Embedding a Digital Watermark into an Image Using a Hamming Code and a Solver of the Square Root of a Prime Number

Nataliia Kukharska
Department of Security of Information Technologies
Lviv Polytechnic National University
Lviv, Ukraine
ORCID 0000-0002-0896-8361

Andrii Lagun
Department of Information Systems and Technologies
Lviv Polytechnic National University
Lviv, Ukraine
ORCID 0000-0001-7856-9174

Serhiy Semenyuk
Department of Security of Information Technologies
Lviv Polytechnic National University
Lviv, Ukraine
ORCID 0000-0002-8143-5887

Orest Polotai
Department of Information Security Management
Lviv State University of Life Safety
Lviv, Ukraine
ORCID 0000-0003-4593-8601

Oles Hospodarskyy
Department of Security of Information Technologies
Lviv Polytechnic National University
Lviv, Ukraine
ORCID 0009-0005-9088-3015

Mykhailo Ozhha
Department of Computer Technologies
Ternopil Volodymyr Hnatiuk National Pedagogical University
Ternopil, Ukraine
ORCID 0000-0002-6954-0318

***Abstract*—In today's digital world, images are an essential tool for transmitting information, which is actively used in education, journalism, medicine, military, banking, and other fields. However, with the development of technologies, the problem of protecting digital images from unauthorized use, forgery, and distortion arises. One of the most effective methods of protection is the use of digital watermarks (DW), which allow authenticating content, tracking its distribution, and preventing counterfeiting. This paper considers the use of fragile digital watermarks. An algorithm for embedding and detecting a digital watermark based on the Hamming code and a pseudo-random number generator (PRNG) derived from the calculation of the square root of a prime number is proposed. The process of generating and extracting digital watermarks, which provides high resistance to attacks and accurate detection of modified areas, is described. The effectiveness of the proposed approach is evaluated using the metrics for detecting modified and unmodified pixels (TPR, TNR, FPR, FNR). Experiments with various types of attacks (copying, collaging, adding text, deleting objects) demonstrate the ability of the proposed scheme to reliably detect changes in images. The obtained results confirm the high efficiency of the method for authenticating digital images and protecting their integrity.**

Keywords*—*information security, fragile digital watermarks, image authentication, digital content protection, Hamming code, pseudo-random number generator, solver of the square root of a prime number.

I. INTRODUCTION

In today's digital world, images play a key role in communicating information by providing a visual representation of data. They are widely used in education, journalism, medicine, the military, banking, and other fields. In advertising and social media, photos and illustrations are used to attract attention and build trust. They can convey complex ideas, concepts, and facts faster and more effectively than text. Images do not require translation, making them a universal way of communication. People tend to remember information better if it is presented in a visual form. Images evoke emotions and create associations, enhancing the perception of a message. Given the availability of technologies for processing and copying digital images, the issue of protecting them from unauthorized use, misappropriation, and distortion is becoming increasingly important.

Over the past decades, researchers have proposed various solutions for protecting multimedia data. One of these solutions is to embed invisible labels (digital watermarks) in the source data.

In general, digital watermarks (DW) have many applications, namely [1]:

- *Copyright protection.* Unique identifiers or logos embedded in digital content help to confirm ownership, and prevent unauthorized use or distribution of images. If used in this way, the digital watermarks must be resistant to various attacks.
- *Authentication.* DWs allow you to check whether the original content has been altered. If the watermark is missing or damaged, this may indicate unauthorized editing or forgery.
- *Tracking distribution.* Digital watermarks can contain information about the source of the content or its recipient. For example, the owner can embed various tags in copies distributed to different customers, which will allow them to track the source of illegal copies and detect leaks.
- *Protection against counterfeiting.* DWs make it difficult to counterfeit digital content, as it is a complicated task to remove or replace a watermark without damaging the original.
- *Indexing multimedia data*, such as video and audio files, news, images, etc., is another well-known application of digital watermarks. Comments or any tags/genres embedded in the original content ensure their efficient search, cataloging, and identification in open databases or on the network.

There are three types of DW: robust, fragile, and semi-fragile [1]. Robust ones are resistant to many types of distortion, fragile ones are destroyed with a small modification of the container, and semi-fragile ones are resistant to some distortions and unstable to others.

979-8-3315-9543-2/25/$31.00 ©2025 IEEE

Governments and organizations are already preparing for this possibility by developing post-quantum cryptographic algorithms, which are designed to withstand quantum attacks and are expected to be standardized by NIST in the coming years.

In Fig.2 we can see an estimation of the development of quantum computing, and accordingly, the breakpoint in which a first practical attack can happen on current encryption schemes. Accordingly we can see the estimation of the implementation of PQC in financial systems. The timeline illustrates two key projections: one for when practical quantum attacks on encryption may become feasible, in around a decade of development, and another for when post-quantum cryptography (PQC) is expected to be fully implemented in financial systems which is a decade after, leaving a space of possible attack scenarios, that are dependent on how fast will the defense systems be able to comply with these emerging security needs.

V. Conclusion And Future Work

This paper has explored the necessity of post-quantum cryptography (PQC) integrated as quantum-resistant encryption within financial systems. The estimated timeline suggests a decade-long process, highlighting the urgency of transitioning to PQC solutions. Implementing PQC within financial systems requires a structured and phased approach, combining hybrid cryptographic techniques for seamless migration. Future research should focus on optimizing PQC performance, addressing regulatory compliance, and ensuring widespread adoption across financial institutions. By taking decisive action now, financial systems can fortify their security infrastructure, mitigating the risks posed by quantum adversaries and ensuring the long-term resilience of digital financial transactions.

This study provides a practical framework for integrating PQC into financial infrastructure. Experimental benchmarks indicate that while PQC schemes increase resource usage modestly, they remain viable for real-time applications. Hybrid cryptography, particularly Kyber + X25519 for TLS 1.3, offers a balanced path forward.

Our current work is implementing hybrid practical solutions for all of the schemes that are mentioned above and trying to combine previous encryption schemes ([21], [22], [23]) with PQC schemes.

Future work should explore establishing regulatory clarity and certification for PQC algorithms, automating key lifecycle and post-quantum certificate rotation, ensuring secure coexistence of classical and quantum-resistant systems, and optimizing PQC implementations for performance-critical financial environments.

Acknowledgment

The authors wish to acknowledge Ruppin Academic Center for its support in the different stages of this research. This study was funded by grant number 33144.

References

[1] Shor, P. W. (1994). "Algorithms for quantum computation: Discrete logarithms and factoring." Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS).

[2] National Institute of Standards and Technology (NIST). (2022). "Post-Quantum Cryptography Standardization." Retrieved from https://csrc.nist.gov/Projects/Post-Quantum-Cryptography

[3] Bernstein, D. J., & Lange, T. (2017). "Post-quantum cryptography." Nature, 549(7671), 188-194.

[4] Kwiatkowski, K., & Stebila, D. (2020). "Post-quantum TLS without handshake signatures." Proceedings of the ACM Conference on Computer and Communications Security (CCS).

[5] Alkim, E., Ducas, L., Pöppelmann, T., & Schwabe, P. (2016). "Post-quantum key exchange—A new hope." Proceedings of the 25th USENIX Security Symposium.

[6] Aggarwal, D., Brennen, G. K., Lee, T., Santha, M., & Tomamichel, M. (2017). "Quantum attacks on Bitcoin, and how to protect against them." Ledger, 3, 68-90.

[7] Hülsing, A., Rijneveld, J., Schwabe, P., & Szefer, J. (2018). "Mitigating multi-target attacks in hash-based signatures with key-updating mechanisms." Proceedings of the International Conference on Applied Cryptography and Network Security (ACNS).

[8] National Institute of Standards and Technology (NIST). (2022). "FrodoKEM, BIKE, and NTRUEncrypt submissions." Retrieved from https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions

[9] Bindel, N., Brendel, J., Fischlin, M., Goncalves, B., & Stebila, D. (2019). "Hybrid key exchange in TLS 1.3." Proceedings of the International Conference on Applied Cryptography and Network Security (ACNS).

[10] Bos, J. W., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J. M., & Stehlé, D. (2018). "CRYSTALS-Kyber and CRYSTALS-Dilithium: Strong and efficient post-quantum cryptography." IEEE Transactions on Cryptographic Hardware and Embedded Systems, 2018(1), 237-256.

[11] Saraireh, S. (2013). A secure data communication system using cryptography and steganography. International Journal of Computer Networks & Communications (IJCNC) Vol, 5.

[12] Rivest, R. L. (2005, July). Perspectives on financial cryptography. In Financial Cryptography: First International Conference, FC'97 Anguilla, British West Indies February 24–28, 1997 Proceedings (pp. 145-149). Berlin, Heidelberg: Springer Berlin Heidelberg.

[13] Li, Y., Gai, K., Qiu, L., Qiu, M., & Zhao, H. (2017). Intelligent cryptography approach for secure distributed big data storage in cloud computing. Information Sciences, 387, 103-115.

[14] Breiling, B., Dieber, B., Pinzger, M., & Rass, S. (2021). A cryptography-powered infrastructure to ensure the integrity of robot workflows. Journal of Cybersecurity and Privacy, 1(1), 93-118.

[15] Dworkin, M. J., Barker, E. B., Nechvatal, J. R., Foti, J., Bassham, L. E., Roback, E., & Dray Jr, J. F. (2001). Advanced encryption standard (AES).

[16] Rivest, R. L. (1985). RSA chips (past/present/future). In Advances in Cryptology: Proceedings of EUROCRYPT 84 A Workshop on the Theory and Application of Cryptographic Techniques Paris, France, April 9–11, 1984 3 (pp. 159-165). Springer Berlin Heidelberg.

[17] Das, M. L., & Samdaria, N. (2014). On the security of SSL/TLS-enabled applications. Applied Computing and informatics, 10(1-2), 68-81.

[18] Galbraith, S. D. (2012). Mathematics of public key cryptography. Cambridge University Press.

[19] Buchmann, J. (2004). Introduction to cryptography (Vol. 335). New York: Springer.

[20] Koblitz, N. (2007). The uneasy relationship between mathematics and cryptography. Notices of the AMS, 54(8), 972-979.

[21] Voloch, N. (2017). MSSP for 2-D sets with unknown parameters and a cryptographic application. Contemp. Eng. Sci., 10(19), 921-931.

[22] Voloch, N., & Raicol, N. V. B. (2024, September). A Cryptographic Encryption Scheme based on a Pythagorean Triplets Manufacturing Formula. In 2024 14th International Conference on Advanced Computer Information Technologies (ACIT) (pp. 529-534). IEEE.

[23] Zadok, Y., Voloch, N., Voloch-Bloch, N., & Hajaj, M. M. (2025). Multiple Subset Problem as an encryption scheme for communications. In Computational Problems in Science and Engineering II (pp. 147-156). Cham: Springer Nature Switzerland

In this paper, we will consider fragile DWs, the main purpose of which is to authenticate an image, to determine whether it has been modified or tampered with after creation. Fragile DWs not only indicate the fact of image modification but also provide information about the location of the changes.

II. Related Work

Prasad and Pal [2] proposed an active scheme for detecting fakes in a digital image, which also allows to determine the location of the modified area. To do this, a three-bit authentication code is generated based on the (7,4) Hamming code applied to the first four most significant bits (MSB) of each pixel, which is subsequently hidden in the same pixel.

According to the authors of [2], the fragile watermark they created is secure, since the process of calculating the authentication code is carried out using secret parameters generated from the logistic map. Given the pseudo-random behaviour of the logistic map, as well as its high sensitivity to initial conditions and control parameters, it is almost impossible to calculate or guess the sequence of numbers obtained with its help. At the same time, Benrhouma, having conducted a cryptanalysis of this scheme, noted its security flaws. Two types of attacks (offline and online), which he carried out and described in his paper [3], were successful. He managed to find a way to find out the values of the elements of the lists K and R used in the scheme [2] as secret keys for embedding digital watermarks. With these keys, it is easy to tamper with an image in such a way that it is not detected during the watermark extraction process. In his subsequent work [4], Benrhouma proposed an improved version of the scheme [2] to eliminate security flaws by encrypting the position using the XOR operation with the element from the list R that contributes to the formation of the watermark. A new pseudo-random number sequence is generated using the logistic map algorithm, then converted into zeros and ones based on a secret threshold.

Like a standard logistic map, its binary version with a threshold demonstrates high sensitivity to initial conditions. This means that slight changes in the values of the initial parameters lead to significantly different output sequences, which can be either an advantage or a disadvantage depending on the context of use. For the scheme [4], this nuance is more of an advantage than a disadvantage. In some cases, the generated bit sequences may have a limited period length after which they begin to repeat, which is undesirable in cryptographic applications where long, non-repeating sequences are required. Despite their chaotic behaviour, logistic maps are deterministic systems. This means that, given known initial conditions and parameters, an attacker can reproduce or predict the generated sequence, which jeopardizes the security of the cryptographic system. Also, the use of a threshold function can lead to an uneven distribution of bits in the resulting sequence, which reduces its randomness and can be detected by statistical tests. Another weakness of using a bit logistic map with a threshold is the difficulty of choosing the optimal parameters to ensure the chaotic behaviour of pseudo-random sequences. An incorrect choice of parameters can lead to predictability or low quality of the generated sequences. Taking into account all the above circumstances, in the scheme [4], we propose to use a generator based on calculating the square root of a prime number as an alternative to the logistic map for forming a pseudo-random sequence.

We investigated steganographic approaches for embedding confidential information in digital BMP images [5]. Also, earlier in [6], we considered the use of a pseudo-random number generator (PRNG) based on the calculation of the square root of a prime number, with the aim of increasing the variability of information concealment by the LSB method. The proposed crypto-steganographic scheme demonstrated good performance.

In general, a pseudo-random number generator based on the square root of a prime number has several advantages [7–9], namely:

- The square root of a prime number is an irrational number, which provides an almost infinite non-periodic sequence in its decimal or binary representation. The high entropy of the generator is significant for its cryptographic use.
- Such a generator is resistant to predictive analysis. Due to the complexity of the exact calculation of the square root and its chaotic nature, it is extremely difficult to predict the next value in the sequence without knowing the initial parameter.
- A generator based on the square root calculator of a prime number can be used to generate cryptographically strong keys since the task of recovering the original number from its fragments is mathematically difficult.
- The use of fast algorithms for calculating the square root (for example, the method discussed in [10]) allows obtaining pseudo-random sequences with minimal computational cost.

III. Proposed Method Theory

A. Digital watermark embedding algorithm

The digital watermark will be embedded in an image consisting of $M \times N$ pixels. To do this, we will implement a series of steps described below.

Step 1. To generate the secret list, we will use a pseudo-random number generator based on calculating the square root of a prime number. Such a calculator must be able to calculate a large number of bits of the root, in our case at least $M \times N$ decimal places. Moreover, each digit must be determined with high accuracy. Failure to comply with this requirement can lead to the loss of pseudo-randomness of the resulting sequence. Since the bits of the square root must be calculated accurately, rough approximation methods may be suitable only for obtaining the initial value.

Step 2. Further, we will consider only the fractional part of the resulting irrational number. Let's divide it into separate fragments of a certain length. As a result, we will get a sequence of numbers $A = A_i$ $(i = 1{:}\, M \times N)$, which's length should correspond to the number of pixels in the image.

The resulting sequence A_i we will convert to a sequence of pseudo-random numbers K_i with values in the range from 0 to 7 using (1).

$$K_i = A_i mod 8 \quad (1)$$

For each i-th pixel of the image, we will assign the i-th number of the sequence.

Step 3. Let's calculate the elements of the list:

$$R = \{K_i, (K_i + 1) mod 8, \dots, (K_i + 7) mod 8\}. \quad (2)$$

Step 4. The numerical value of the color intensity of the i-th pixel is represented in binary form. Let's extract the four MSB and calculate the (7,4) Hamming code: $c = (c_7, c_6, c_5, c_4, c_3, c_2, c_1)$, where the three lowest bits $W = (c_3, c_2, c_1)$ are the calculated bits.

Step 5. Let's represent W in decimal form, which will result in an integer T.

Step 6. Let's find T in the list R and memorize its position number as j. Here and further, the indexing of elements in the list starts from zero.

Step 7. Again, we will use a pseudo-random number generator based on the square root of a prime number (not necessarily the one from Step 1), but this time in binary form [6, 10]. To encrypt the position number j, we will consider the fractional part of the resulting irrational number and divide it into fragments of 3 bits. Let's calculate E by applying the XOR operation, the arguments of which are the i-th three-bit fragment S_i of the obtained pseudo-random sequence and the position number j in its binary representation: $E = S_i \oplus j$. Let's represent E in the format of a decimal number.

Step 8. Let's calculate $z = P_i mod 8$, where P_i is the color intensity of the i-th pixel.

Step 9. Let's create a list:

$$PR = \{P_i - z + t,\ t = 0,1, \dots, 7\}. \quad (3)$$

Step 10. Replace the value of the i-th pixel with the value of the element at position E in the PR list.

We implement steps 3-10 for all $M \times N$ pixels to generate a digital image watermark.

Digital watermark extraction algorithm

Consider an image with an embedded digital watermark of size $M \times N$.

Let's describe an algorithm that can be used to detect not only the fact of image tampering, but also to determine the location of modifications.

Step 1. We will generate the same pseudo-random sequence as when building a digital watermark, that is, we will calculate the square root of the same prime number.

Step 2. We will take into account only the fractional part of the calculated square root and divide it into fragments containing the same number of bits as when creating the digital watermark. Using (1), we transform the resulting sequence into a pseudo-random sequence of numbers K_i with values from 0 to 7.

Step 3. For each pixel, using K_i as a basis, we will create a list R using (2).

Step 4. Having represented the color intensity of the i-th pixel in binary form, for the four MSB we will calculate the Hamming code $W = (c_3, c_2, c_1)$ and represent it in decimal format T.

Step 5. Let's find the position of T in the list R and save it as j.

Step 6. Let's generate a pseudo-random sequence based on the square root calculator of a prime number in binary representation [6, 10]. Once we have an infinite binary irrational number, we will split it into 3-bit fragments S_i.

Step 7. Let's calculate for the i-th pixel: $Em = PW_i mod 8$, which is the position in the list PR of the digital watermark encrypted by the XOR operation.

Step 8. To decrypt the item number, we will represent it in binary form and calculate $jm = S_i \oplus Em$. The result is converted to decimal.

Step 9. Let's compare the value of jm with the value of j, which indicates the position of the Hamming code W in the list R. If they are not equal, it means that the pixel is modified.

Let's repeat steps 3-9 for all pixels of the image with the embedded digital watermark. At the same time, we will generate a black-and-white image, with unmodified pixels in black and modified pixels in white, i.e., those for which the equality condition in step 9 is not met. Flowcharts of the embedding, extraction, and tamper detection schemes are given in [4].

IV. An Illustrative Example

A. Steps of embedding

Step 1. Let's calculate the square root, for example, of the number 3, and we will get the infinite irrational number $sqrt(3) = 1.732050807568877 \dots$

Step 2. Let's divide the part after the decimal point into separate fragments, for example, 4 digits long: 7320 3205 2050 0508 … We will match the first pixel of the image with the first element of the pseudo-random sequence 7320 and apply the operation $7320 mod 8 = 0$. However, it is not necessary to start from the first digit after the decimal point. We can start splitting into fragments from any other position. Then the number of this position will play the role of a key.

Step 3. Let's calculate according to the scheme (2):

$R = \{0,1,2,3,4,5,6,7\}$.

Step 4. Let's suppose the first pixel of the image has a value of $P_1 = 178$, whose binary representation is 1011 0010. For its four most significant bits, we will calculate the Hamming code $c = \{1,0,1,1,0,1,0\}$, where $W = \{0,1,0\}$ are the calculated bits responsible for the digital watermark.

Step 5. Let's convert W to decimal and get $T = 2$.

Step 6. Let's search for T with value 2 in the list R and fix its position number $j = 2$.

Step 7. Using the algorithm [6,10], let's calculate, for example, the square root of 2. We will get an infinite binary irrational number:

$$sqrt(2) = 1.01101010000001001111001 \dots \quad (4)$$

Let's use its fractional part to form a sequence of three-bit fragments 011 110 101 010 101 …

We will encrypt the position number j=3 by first representing it as a sequence of three bits. Then we will apply the XOR operation: E={0,1,1}⊕{0,1,0}={0,0,1}. In the decimal format $E = 1$.

Step 8. For the first pixel, we will calculate: $z = 178 mod 8 = 2$.

Step 9. Let's define the elements of the list by (3): $PR = \{176,177,178,179,180,181,182,183\}$.

Step 10. Let's create a digital watermark for the first pixel by selecting the element with position $E = 1$ from the list. Its value is 177.

Let's repeat the procedure for creating a digital watermark described in the example of the first pixel is for the rest of the image pixels.

B. Extraction Steps

Let's say the first pixel of the image has been changed, and now its intensity value is 191 (instead of 177). Let's demonstrate the implementation of the above steps for digital watermark extraction and check whether the algorithm in question will allow us to detect the modification.

Step 1. We will generate the same pseudo-random sequence as when building a digital watermark by calculating the square root of the same prime number: $sqrt(3) = 1,732050807568877$ ...

Step 2. Next, we will consider the sequence of four-digit fragments 7320 3205 2050 ... For the first pixel, we will take the first number 7320, and calculate $7320 mod 8 = 0$.

Step 3. Let's create a list: R={0,1,2,3,4,5,6,7}.

Step 4. We will convert the value of the digital watermark 191 to the bit sequence 10111111 and calculate the Hamming code, taking into account the four most significant bits {1,0,1,1,0,1,0}.

Let's represent {0,1,0} in the decimal code $T = 2$.

Step 5. We will consider the list R and find the T in it that equals 2. Its position number is $j = 2$.

Step 6. Let's calculate the square root of the prime number 2. We will get an infinite binary irrational number (4), from the fractional part of which we will extract a series of three-bit sequences 011 110 101 010 101….

Step 7. Let's calculate for the first pixel: $191 mod 8 = 7$, which is the value of the position E of the digital watermark in the list PR encrypted using the XOR operation.

Step 8. To decrypt the position number, we will represent it in binary form {1,1,1} and calculate it: $jm = \{0,1,1\} \oplus \{1,1,1\} = \{1,0,0\}$. Let's convert the result to a decimal number and get 4.

Step 9. Let's compare the value $j = 2$, which was obtained in step 5, with the value $jm = 4$. They are not equal. This means that the pixel is modified.

V. Study of the Effectiveness of the Scheme

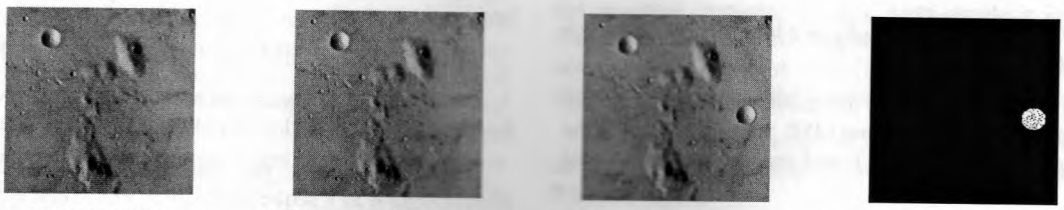
To evaluate the effectiveness of the proposed scheme, we will calculate the coefficients:

- TN (True Negative) – the number of unmodified pixels whose unchangedness is confirmed by the digital watermark.
- FP (False Positive) – the number of unmodified pixels that the watermark displays as modified.
- TP (True Positive) – the number of modified pixels that the watermark recognizes as modified.
- FN (False Negative) – the number of modified pixels that the watermark indicates as unmodified.
- TPR – a rate that shows the percentage of correctly recognized modified pixels: $TPR(\%) = \frac{TP \times 100}{TP+FN}$.
- TNR – a rate that shows the percentage of correctly recognized unmodified pixels by the watermark: $TNR(\%) = \frac{TN \times 100}{TN+FP}$.
- FPR – the false positive rate, which is determined: $FPR(\%) = \frac{FP \times 100}{TN+FP}$.
- FNR – the false negative rate, which is calculated as: $FNR(\%) = \frac{FN \times 100}{TP+FN}$.
- ρ – the intervention coefficient determined by: $\rho(\%) = \frac{(FN+TP) \times 100}{M \times N}$.

Here we will consider different types of attacks on digitally watermarked images. They violate the authenticity and copyright of the image:

- *Copy-paste attack.* This method involves modifying a watermarked image by inserting fragments copied from the same image in other places (Fig. 1).
- *Collage attack.* It involves copying elements from one watermarked image to another image that also contains a watermark (Fig. 2).
- *Text-addition attack.* This attack involves adding text elements to a watermarked image (Fig. 3).
- *Object deletion attack.* In this case, the image is altered by removing certain parts or objects (Fig. 4).

Table 1 shows the quantitative characteristics of these experiments.

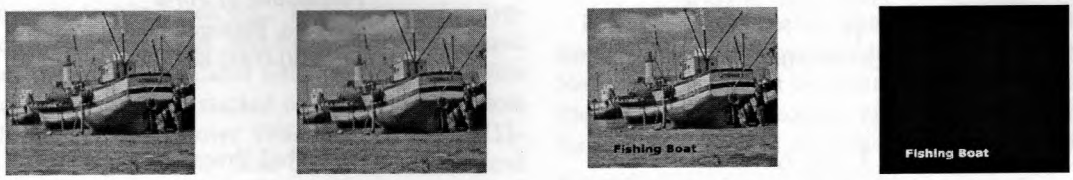


(a) Moon surface (b) Watermarked Moon surface (c) Manipulated Moon surface (d) Tampering map

Fig. 1. Effectiveness against copy-paste attack

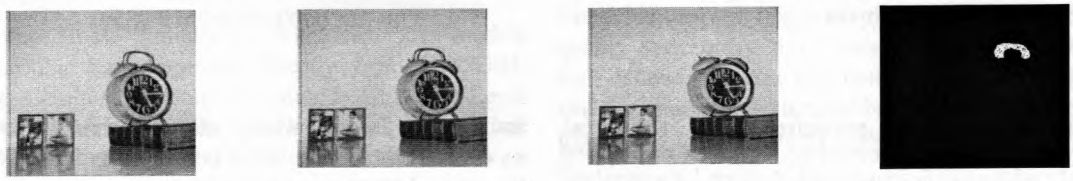
(a) Watermarked Truck (b) Watermarked Car and APCs (d) Manipulated Truck (d) Tampering map

Fig. 2. Effectiveness against collage attack



(a) Fishing Boat (b) Watermarked Fishing Boat (c) Manipulated Fishing Boat (d) Tampering map

Fig. 3. Effectiveness against text-addition attack



(a) Clock (b) Watermarked Clock (c) Manipulated Clock (d) Tampering map

Fig. 4. Effectiveness against object deletion attack

TABLE I. PARAMETERS THAT SHOW THE EFFICIENCY OF THE SCHEME

	Moon surface	**Truck**	**Fishing Boat**	**Clock**
TP	636	2566	2569	510
TN	64807	259258	259201	64970
FP	93	320	374	56
FN	0	0	0	0
FPR %	0.1433	0.1233	0.1441	0.0861
FNR %	0	0	0	0
TPR %	100	100	100	100
TNR %	99.8567	99.8767	99.8559	99.9139
ρ %	0.9705	0.9789	0.9800	0.7782
Tampered	636	2566	2569	510
Non tampered	64900	259578	259575	65026
PSNR	37.725	37.908	37.919	37.825

VI. CONCLUSION

Fig. 1–4 and the parameters listed in Tab. 1 demonstrate that in all experiments, DWs embedded using the described algorithm enable effective forgery detection and localization of modified areas. The method provides 100% accuracy in detecting changed pixels, zero missed modifications, and a very low false positive rate (0.0861%–0.1441%), which confirms its high reliability for image authentication.

The comparative analysis of the proposed approach with similar schemes, particularly those based on logistic maps [4], revealed the same PSNR indicators (37.9 dB and 37.74 dB, respectively), indicating a similar quality of visual image preservation after watermark embedding. However, the method has the advantage of using a PRNG based on the square root of a prime number, offering higher entropy, non-periodicity, better resistance to prediction, and no reliance on complex initial parameters typical for chaotic maps.

REFERENCES

[1] G. Konakhovych, D. Progonov and O. Puzyrenko, Komp'juterna steghanoghrafichna obrobka j analiz muljtymedijnykh danykh. Kyiv: Centr navchaljnoji literatury, 2018.

[2] S. Prasad and AK Pal, "Hamming code and logistic-map based pixel-level active forgery detection scheme using fragile watermarking", Multimedia Tools and Applications, 2020, vol. 79, No. 29, pp. 20897–20928.

[3] O. Benrhouma,"Cryptanalysis of a hamming code and logistic-map based pixel-level active forgery detection scheme," International Journal of Advanced Computer Science and Applications, 2022, vol. 13, No. 2, pp. 663–668.

[4] O. Benrhouma, "Improvement of a Hamming code and logistic-map based pixel-level active forgery detection scheme," International Journal of Advanced and Applied Sciences, 2022, Vol. 9, No. 9, pp. 127–135.

[5] N. Kukharska, A. Lagun and O. Polotai, "The steganographic approach to data protection using Arnold algorithm and the pixel-value differencing method," Proceedings of the 2020 IEEE 3rd International Conference on Data Stream Mining and Processing, DSMP 2020, 2020, pp. 174–177.

[6] N. Kukharska, A. Lagun and O. Yashchyk, "Crypto-steganographic system based on the solver of the square root of a prime number," Proceedings – International Conference on Advanced Computer Information Technologies, ACIT, 2024, pp. 535–538.

[7] V. Dimitrievska Ristovska, " Pseudo random generator based on irrational numbers", ICT-Innovations 2017, Data Driven Innovations, Conference Web Proceedings, 2017, pp. 105–113.

[8] O. Horyachyy, L. Moroz and V. Otenko, "Simple effective fast inverse square root algorithm with two magic constants," International Journal of Computing, 2019, Vol. 18, No. 4, pp. 461–470.

[9] V. Dimitrievska Ristovska and V. Bakeva, "Comprasion of the results obtained by pseudo random number generator based on irrational numbers," Mathematical Modeling, 2017,Vol. 1, No. 4, pp. 167–170.

[10] A. Horpeniuk and N. Luzhetska, " Generator of pseudo-random numbers of the square root of simple numbers," Bulletin of the National university "Lviv Polytechnic" Automation, measurement and control, 2013, No. 753, pp. 45–50