

# THE UPGRADED PETRI NET MODELS OF ONE STEGANOGRAPHY ALGORITHM

Perica Štrbac<sup>1</sup>, Pavle Štrbac<sup>2</sup>, Nemanja Maček<sup>3</sup>

**Abstract:** The main goal of this paper is to present Upgraded Petri Net models for an image steganography algorithm, to transform these models into programming code, and to check out the models. The first model includes as follows: importing two images (one cover and one secret image we want to encrypt and hide), making a histogram of an image, making a crypt map, hiding an encrypted image into the carrier image, and making a steganography image. The second model extracts a hidden encrypted image from a steganography image and decrypts the extracted image. Input parameters are: the start point of encrypt mode, the start point of the map for encrypt, the start point of hiding bytes, a carrier image, a secret image, and an output image path. After several modeling, simulation, and analysis cycles, the models were translated into programming code. We further use this code to check our models. These histograms have been created for both the carrier and the steganography image. The practical results confirm the correctness of the two Upgraded Petri Net models referring to our steganography hiding image algorithm.

**Key words:** steganography, Upgraded Petri Nets, encryption, histogram, modeling, simulation, analysis

## 1. INTRODUCTION

The basic idea is to make two Upgraded Petri Net (UPN) models of a steganography algorithm, convert them into the software program, and check the algorithm. The models refer to a map of information bytes, twice encode some information, and then hide the information in a cover image. This information is saved in the least significant bit (LSB) or in two right bits of the cover image bytes [1].

The UPNs are a specific class of Petri-nets for modeling, simulating, and analyzing complex processes [2]. We used one original Petri Manager (PeM) software package to create our models. The two graph models related to our steganography algorithm were created using the PeM software. The first model maps our main image by using its histogram and doing two encryptions. For mapping, we choose one of the color histograms of the main image. First encryption employs pairs of bytes. The second encryption uses even bytes. The second model refers to extracting the encrypted image from the steganographic image and decrypting this encrypted image.

Finally, these models were transformed into a Python program [3]. The main goal is to check the UPN models by executing this Python code.

## 2. THE UPN STEGANOGRAPHY ALGORITHM MODELS

This chapter shows the basic theory of the UPN and shows the two UPN models of our steganography algorithm.

### 2.1. Basic theory of the UPNs

The UPN is defined as a nine-tuple  $C = (P, T, F, B, \mu, \theta, TF, TFL, PAF)$ , where  $P$  and  $T$  are sets of places and transitions, respectively, while  $F, B, \mu, \theta, TF, TFL$  and  $PAF$  are functions (front, back, marking, timing, transition, firing level and place attributes, respectively) [2].

## 2.2. The UPN steganography algorithm models

[illegible]

<sup>1</sup>Perica Štrbac, Academy of Technical and Art Applied Studies, School of Electrical and Computer Engineering, Belgrade, 283 Vojvode Stepe st., e-mail: pericas@viser.edu.rs

<sup>2</sup>Pavle Štrbac, Osterus d.o.o, Novi Sad, 24 Vase Stajića St. e-mail: strbacpavlejob@gmail.com

<sup>3</sup>Nemanja Maček, Academy of Technical and Art Applied Studies, School of Electrical and Computer Engineering, Belgrade, 283 Vojvode Stepe st., e-mail: nmacek@viser.edu.rs

of the set of transitions  $\{t-5, t-6, t-7\}$  gives a new marking  $\mu = (0,1,1,0,0,0,0,9,0,0, 0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0, 0,0,0,0)$ . These transitions refer to the load cover (carrier) image, loading the secret image, and setting the steganography mode of our algorithm, respectively. The place p-2 models the cover image, place t-3 models in the secret image, and place t-8 models in our steganography mode. The marking of the place p-8 is set to value 9 because of multiplicity  $B(t-7, p-8)=9$ . Now the set of transitions  $\{t-17, t-18, t-8, t-22, t-26, t-27, t-31, t-32, t-34, t-35\}$  is enabled. The transitions t-17 and t-18 conflict and only one of them will be fired. Firing of the transition t-17 refers to the correct parameters of the cover image and secret image, whereas firing of the transition t-18 refers to incorrect parameters of the cover image and secret image. These parameters are the size and format of the cover image and secret image.

The first possible firing set of transitions is  $\{t-17, t-8, t-22, t-26, t-27, t-31, t-32, t-34, t-35, t-43\}$ . This modeling corrects the parameters of the cover image and secret image, and the parameters of the secret image; set histogram selector; set mapping mode, set first encrypt mode, set second encrypt mode, set start of histogram map, set histogram map save mode, set start of encrypted bytes, set encrypted bytes save mode. The new marking is  $\mu = (0,0,0,0,0,0,0,0,1,0, 6,1,0,0,0,1,0,0,0,1, 1,0,0,0,2,2,0,1,1,0, 0,0,0,1,0,0,0,0,0, 0,0,1,0)$ . Parallel firing of the set of transitions  $\{t-1, t-2, t-3, t-4, t-20, t-21\}$  gives a new marking is  $\mu = (0,0,0,1,1,1,1,0,1,0, 0,0,0,3,1,1,0,0,0,1, 1,0,0,0,2,2,0,1,1,0, 0,0,0,0,0,0,0,0,0, 0,0,1,0)$ . This state refers to the created four histograms (RGB, R, G, B, respectively) of the secret image, refers to the created cover image vector, and refers to the created secret image vector. By firing of the set of transition  $\{t-9, t-10, t-11, t-12, t-44\}$  one of the possible new marking  $\mu = (0,0,0,1,1,1,0,0,0,1,0,0,0,2,1,1,0,0,0,1,1,0,0,0,2,2,0,1,1,0, 0,0,0,0,0,0,0, 0,0,0, 0,0,0,1)$ . This marking refers to choosing the B histogram (placed in the place p-10), and to saving steganography mode into the cover image. Now, the set of transition  $\{t-13, t-14, t-15, t-23\}$  is enabled. The parallel firing of this set of transitions gives a new marking  $\mu = (0,0,0,0,0,0,0,0,0,0, 0,0,0,2,1,0,1,0,0,1, 1,0,0,0,2,2,0,1,1,0, 0,0,0,0,0,0,0, 0,0,0,1)$ . The firing  $\{t-13, t-14, t-15\}$  saves UPN model marking by removing markers from the places p-4, p-5 and p-6 because the front functions with inhibitor arcs are  $F^l(t-13, p-9) = F^l(t-14, p-9) = F^l(t-15, p-9) = 1$ . The firing of  $\{t-23\}$  makes the list of 2-tuple (place p-17) where the first member of this tuple is a byte value, while the second one is the frequency of that byte value. Firing of the transition sequence  $\{t-24\}, \{t-25\}$  sorts the list by frequency in the first level and sorts the list by byte value in the second level. Now, the new marking is  $\mu = (0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0,0, 0,2,1,1,0,0,0,2,2,0,1,1,0, 0,0,0,0,0,0,0,0,0,0,0,0,0,1)$ , and the final list created. This list is used for mapping the secret image vector (the transition t-28 is enabled) and for saving this list into the cover image vector (the transition t-33 is enabled). For saving the list into the cover image we use the final list (the place p-19), map start value (the place p-25), map save mode (the place p-26), and cover image vector (the place p-14). After firing of the transition set  $\{t-28, t-33\}$  a new marking is available as  $\mu = (0,0,0,0,0,0,0,0,0,0, 0,0,0,1,0,0,0,0,0,1, 1,1,0,0,1,1,1,1,0, 0,0,0,0,0,0,0,0,0, 0,0,0,1)$ . This tells that the map list is saved into the cover image (p-27), and secret image bytes are mapped (p-22). By firing the transition set again  $\{t-29, t-37\}$  the new marking is  $\mu = (0,0,0,0,0,0,0,0,0,0, 0,0,0,1,0,0,0,0,0,0, 1,0,1,0,1,1,1,1,0,0, 0,0,1,0,0,0,0,0,0, 0,0,0,1)$ . This state represents the operation of encryption 1 on mapped bytes, and choosing one of two possible options. The first option refers to the use of one LSB of the cover image, while the second one refers to two LSBs of the cover image. We use conflict to model this state ( $\mu(p-33)=1, F(t-38, p-33)=1, F(t-39, p-33)=1$ ), so the only one of transitions t-38, t-39 can be fired. The set of transitions  $\{t-30, t-38, t-39\}$  is enabled. The transition t-30 will be fired, and this refers to the operation of encryption 2 on previously encrypted bytes. If the transition t-38 was fired, then we will use as follows: 1LSB, 0xFE as an AND-bitwise value for the cover byte, 0x01 as an AND-bitwise value for the encrypted byte, SH1 for shifting by 1 bit, and LOOP8 for 8 loops per encrypted byte. If the transition t-39 was fired, then we will use 2LSB, 0xFC as an AND-bitwise value for the cover byte, 0x03 as an AND-bitwise value for the encrypted byte, SH2 for shifting by 2 bits, and LOOP4 for 4 loops per encrypted byte. If the transition t-38 was fired then the transition t-41 will be enabled. On the other side, if the transition t-39 was fired then the transition t-42 will be enabled. In both cases, by firing the transition t-41 or t-42 the new marking is  $\mu = (0,0,0,0,0,0,0,0,0,0, 0,0,0,1,0,0,0,0,0,0, 0,0,0,1,1,0,1,0,0, 0,0,0,0,0,0,0,0,0, 0,1,0,1)$ . Now, the transition t-36 is enabled. This means that

encrypted bytes can be saved into the cover image according to map start, map save mode, encrypted bytes start, LSB parameters, cover image vector, and encrypted bytes.

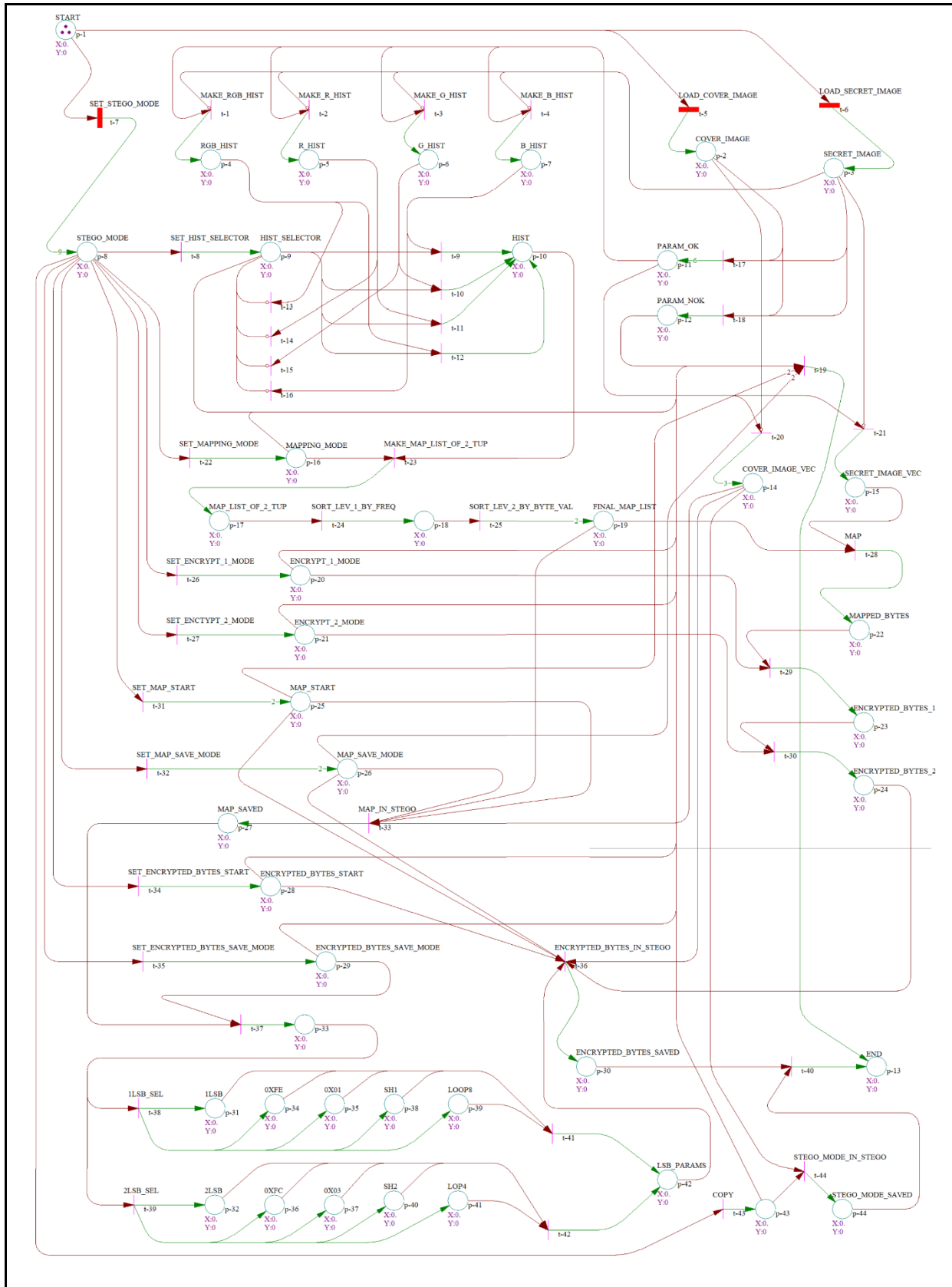


Figure 1 – The UPN model: put a secret image in a cover image

The new marking now is  $\mu = (0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,1, 0,0,0,0,0,0,0,0,0,0, 0,0,0,1)$ . This represents that encrypted bytes are saved into the cover image. By firing the transition t-40 the UPN model will finish its execution and reach the death node (the place p-13). The final marking is  $\mu = (0,0,0,0,0,0,0,0,0,0, 0,0,1,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0, 0,0,0,0)$ .

The second possible firing set of transition is {t-18, t-8, t-22, t-26, t-27, t-31, t-32, t-34, t-35, t-43}. The new marking is  $\mu = (0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,1,1,0,0,0,2,2,0,1,1,0, 0,0,0,1,0, 0,0,0,0,0,0,0,1,0)$ . This model shows the incorrect parameters of the cover and secret image (place p-12). By firing the transition t-19 the model goes into the death node (the place p-13). The final marking is  $\mu = (0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0)$ .

Figure 2 depicts the UPN model of our steganography algorithm for getting a secret image from a cover image. A place p-1 models the start of the algorithm. The  $\mu_0$  of the model is equal to  $(1,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0)$ . Firing of the transition {t-1} gives a new marking  $\mu = (0,3,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0)$ . This transition refers to loading our steganography image. The firing of the transition t-2 models gets steganography mode from the steganography image and makes new marking of the model  $\mu = (0,2,9,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0)$ . Now, the set of transition {t-3, t-4, t-5, t-6, t-7, t-8, t-9, t-14, t-15} is enabled. Firing of this set refers to getting parameters as follows: start histogram, map histogram, start secret image, map secret image, second encryption, first encryption, LSB mode, SH mode, and LOOP mode. A new marking of the model is  $\mu = (0,2,0,1,1,1,1,2,2,2, 0,0,0,0,0,0,0,1,1,0)$ . The firing of the set of transitions {t-10, t-11, t-12} represents getting the histogram and setting AND value for our encrypted image and AND value for our cover image. A new marking is  $\mu = (0,1,0,0,0,1,1,1,1,1, 0,1,1,1,0,0,0,1,1,0)$ . These parameters were used to get the encrypted image (firing of the transition t-13). Now, the marking is  $\mu = (0,0,0,0,0,0,0,0,0,0, 0,0,1,1,0,0,0,1,1,0)$ . Further, the model goes through a sequence of firing transitions {t-16}, {t-17}. The sequence refers to the second decryption and first decryption, respectively. Now, a new marking is  $\mu = (0,0,0,0,0,0,0,0,0,0, 0,0,1,0,0,1,0,0,0,0)$ . The firing of the transition t-18 in the model represents the state of the completed secret image. Finally, by firing the transition t-19 the model goes into the death node  $\mu = (0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,1)$ .

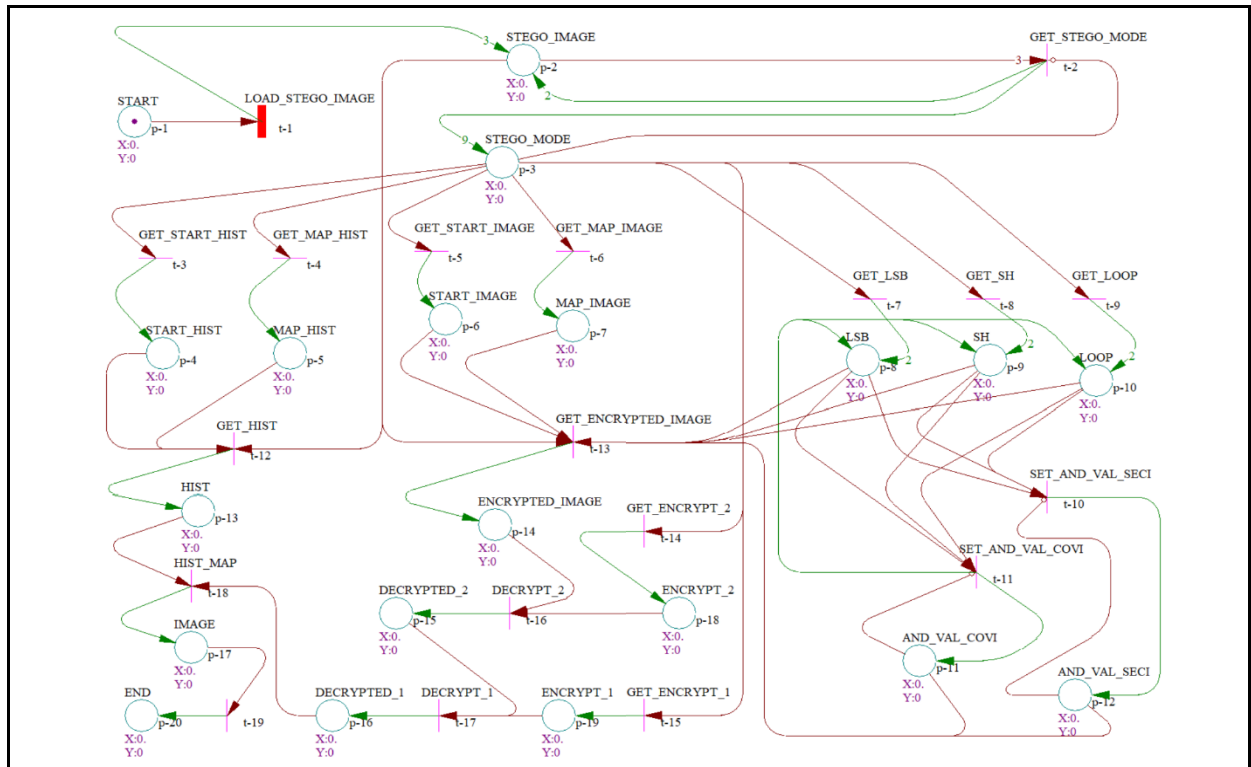


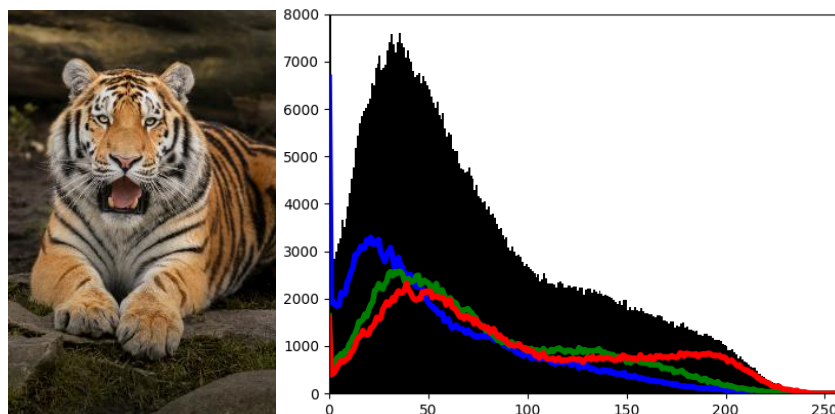
Figure 2 – The UPN model: get a secret image from a steganography image

### 3. EXPERIMENTAL RESULTS

Inputs in our program for putting a secret image in the cover image are a cover image, a secret image, a start point of the steganography mode bytes, a start point of the histogram bytes, a start point of the encrypted bytes, encrypt mode 1, encrypt mode 2, the LSB parameters, a mode of saving histogram bytes, a mode for saving encrypted bytes. Outputs of the program are as follows: a steganography image, a histogram of the cover image, a histogram of the steganography image, and all the histograms of the secret image.



*Figure 3 – The cover image [4]*



*Figure 4 – The secret image [5] with appropriate histograms*

The cover image resolution is 4368x2448 pixels. This image is scaled down from 4000x6000 pixels to a new resolution of 400x600 pixels. Both images are saved as PNG images. Figure 3 depicts both the cover and the secret image with appropriate histograms (RGB, R, G, B). In our experiments, we randomly chose one of our secret image histograms to map bytes of its image.

The input of our program for getting the secret from the stego-image is only the path for this secret image. All necessary parameters will be obtained from the steganography image. Figure 5 shows the cover image histogram and steganography image histogram for two LSB options (1-bit LSB and 2-bit LSB). The difference between the cover image histogram and the steganography 1-bit LSB histogram is shown (barely perceptible difference) [6],[7]. The difference between the cover and stego-image 2-bit LSB histograms is shown (thus presenting a visibly noticeable difference).

The shown steganography algorithm does lossless image hiding and extracting. So, we can use this algorithm to hide text (cf. [8],[9]).

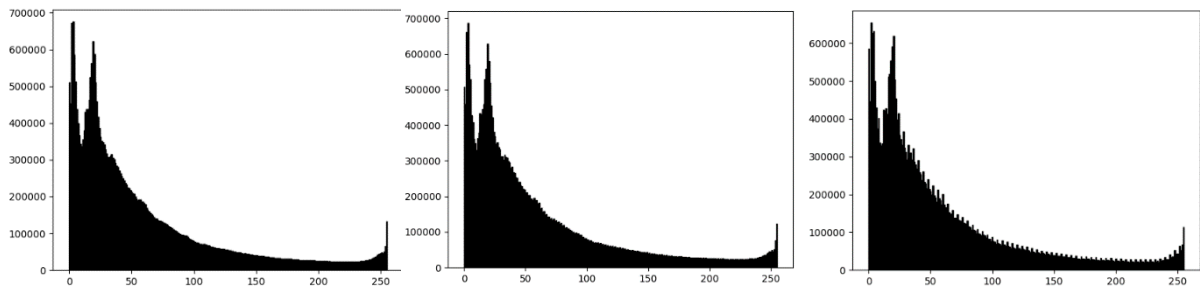


Figure 5 – The cover, steganography for the 1-bit LSB option, and the steganography image histogram for the 2-bit LSB option. Presented left to right.

## 4. CONCLUSION

The paper shows the Upgraded Petri Nets models of the steganography algorithm. Two UPN models are presented. The first model refers to putting a secret image in a cover image. This model uses various input parameters as follows: a histogram selector; a mapping mode, an encrypted mode for the first encryption, an encrypted mode for the second encryption, a start of histogram map, a histogram map save mode, a start of encrypted bytes, and a encrypted bytes save mode. This model refers to loading a cover and a secret image, making histograms of a secret image, mapping the secret image bytes, encrypting the secret image bytes, to saving encrypted bytes in a cover image. The second model refers to getting a secret image from a cover image according to the mentioned parameters. These UPN models were transformed into appropriate Python programs. By executing this program, the correctness of these UPN models was checked. The experimental results are shown, especially with the impact of changing the histogram of a steganography image concerning the appropriate cover image. Further research involves arithmetic compression of the secret image blocks.

## 5. REFERENCES

- [1] Singh, A. K., Singh, J., Singh, H. V.: *Steganography in Images Using LSB Technique*, International Journal of Latest Trends in Engineering and Technology, Vol. 5 Issue 1 January 2015, pp. 426-430.
- [2] Strbac, P., Milovanovic, G.V.: *Upgraded Petri net model and analysis of adaptive and static arithmetic coding*, Elsevier: Mathematical and Computer Modelling Vol. 58, 2013, pp. 1548-1562.
- [3] Sengupta, R., Umarani, C.: *Steganography Using Python*, IITM Journal of Management and IT, Volume 12, Issue 1, January-June 2021, pp. 40-43.
- [4] Downloaded from: <https://www.vecteezy.com/photo/48445334-lush-tropical-jungle-paradise-with-exotic-plants-and-waterfall-nature-s-bounty-in-paradise> December 2024.
- [5] Downloaded from: <https://www.vecteezy.com/photo/21164258-photo-of-a-siberian-tiger> December 2024.
- [6] Choudhury, B., Das R., Baruahhistr A.: *A Novel Steganalysis Method Based on Histogram Analysis*, Lecture Notes in Electrical Engineering, Vol. 315, November 2015, pp. 779-789.
- [7] Ki-Hyun, J.: *Comparative Histogram Analysis of LSB-based Image Steganography*, WSEAS Transactions on Systems and Control, Vol. 13, January 2018, pp. 103-112.
- [8] Rani, N., Chaudhary, J.: *Text Steganography Techniques: A Review*, Neha International Journal of Engineering Trends and Technology (IJETT) – Volume 4 Issue 7- July 201, pp. 3013-3015.
- [9] Majeed, M. A., Rossilawati, S., Shukur, Z.: *New Text Steganography Technique based on multilayer Encoding with Format-Preserving Encryption and Huffman Coding*, International Journal of Advanced Computer Science and Applications, Vol. 13, No. 12, 2022, pp. 163-172.