ISSN 2395-8618

Real-Time Simulation of the Occlusion-Noise Attack and a Mitigation Proposal

Manuel Alejandro Cardona-López, Juan Carlos Chimal-Eguía,
Víctor Manuel Silva-García, Rolando Flores-Carapia, Julieta Mazzetti-Riveros

***Abstract*—The performance of image transmission can be affected by noises. These can be produced by natural conditions or attacks induced by third parties. The main inconvenience of noise in images is the loss of visual quality. The higher the percentage of noise in the image, the lower the clarity of information obtained. For this reason, in this paper, occlusion noise is simulated in real-time during the download of an image. The simulation includes the malicious interference of a third agent with the data transfer between two mobile devices. In this way, the receiver realizes the damage only after the image reproduction is complete. The purpose is to demonstrate the effects of this type of noise and the difficulties in understanding damaged information. Additionally, the present work shows the utility of mitigation in occlusion attacks, despite sometimes not being completely avoided. The proposed solution involves a permutation that spreads the noise across all rows. Consequently, each row has both initial values and damaged pixels.**

***Index Terms*—Occlusion attack, noise, image encryption, permutation.**

I. Introduction

Noise in images is a frequent feature in image transmission due to natural communication conditions [1-3]. It can also be produced as part of a cyber-attack [4-6]. Consequently, the inconvenience for human vision lies in the difficulty of recognizing a damaged image caused by the noise. For this reason, various noises have been examined [7-9]. One such noise is the occlusion noise attack, which involves replacing the original pixels from an image with other values. Due to the vulnerability of information during sending and receiving processes, it is essential to address these attacks. They often cannot be avoided because their execution depends on third parties.

In that sense, different proposals have emerged to keep information secure [10-12]. In summary, occlusion noise exposes information integrity, particularly in channels with vulnerability problems. An occlusion attack can occur when hackers are unable to cut off a communication channel. Therefore, a possible option is to obstruct it with noise for a few seconds. The result is the loss of a certain number of pixels within the image. All these changes occur during the time in which the interference is present. This damaged area is called noise and generates values in the pixels that are different from the original ones.

For this reason, the present work demonstrates a computational simulation of the occlusion noise attack. It occurs in real-time while the sender and receiver establish communication through two different mobile devices. We explore the scenario when the receiver downloads an image from the sender's server, and then the occlusion noise is executed by a third party (attacker). However, the mobile device does not perceive the noise interference during the download. It is only realized by the user once the image has been completely displayed on the screen. To mitigate the damage caused by the attack, a pixel permutation is proposed. The main goals of this paper are to show the damage in images caused by occlusion noise and how it can be mitigated with a uniform permutation.

The paper is divided as follows. It begins with Section 2, which illustrates the theoretical bases for the development of the application. In this case, it discusses the permutation applied to the arrangement of image pixels. Also, the tools for measuring uniformity in the permutation are included. Section 3 contains the simulation development, while Section 4 presents the obtained results. Finally, Section 5 includes the conclusions.

II. Knowledge Bases of the Proposal

In this section, we briefly describe each lexical resource, and show the type of information that the semantic parser extracts from these knowledge bases.

A. Linear Congruential Generator Algorithm

The Random number generation was made through the modular linear congruence method with a variable time seed. A linear congruence is a linear equation in Zm.

Congruence of the form:

$$ax \equiv b(mod m), \tag{1}$$

where m is a positive integer, a and b are integers, and x is an integer variable, is called linear congruence or linear congruence equation.

Solving this equation consists of finding all the integers x that satisfy the equivalent Diophantine equation:

$$ax + my = b\,. \tag{2}$$

In the case of Zm, the equation will have a solution if and only if $gcf(a,m)|b$, and in this case it will have exactly $d = gcf(a,m)$ different solutions in Zm of the shape:

$$x = x_0 + (m - y)/d, t = 0,1,2,3,\ldots,d - 1\,, \tag{3}$$

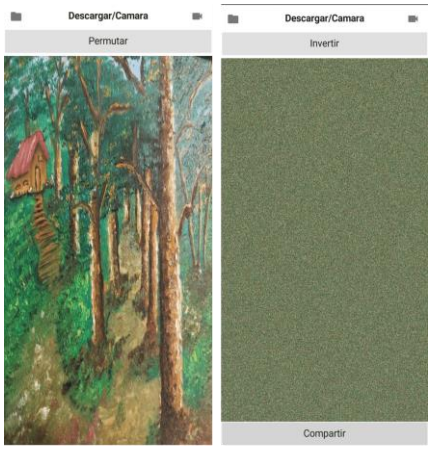
where x_0 is a particular solution of the Diophantine equation $ax + my = b$.

Manuscript received on 17/04/2023, accepted for publication on 27/06/2023.

Manuel Alejandro Cardona-López, Juan Carlos Chimal-Eguía are with the Instituto Politécnico Nacional (IPN), Centro de Investigación en Computación (CIC), México ({mcardonal2022, chimal}@cic.ipn.mx).

Víctor Manuel Silva-García, Rolando Flores-Carapia, Julieta Mazzetti-Riveros are with the Instituto Politécnico Nacional (IPN), Centro de Innovación y Desarrollo Tecnológico en Cómputo (CIDETEC), Mexico City, Mexico ({vsilvag, rfloresca}@ipn.mx, jmazzettir2200@alumno.ipn.mx).

ISSN 2395-8618



(a) (b)

Fig. 1. Main screen of mobile application (a) Original image, (b) Permuted image

The method of modular linear congruences consists of the following equation:

$$Z_n = (aZ_{n-1} + C) mod m\,, \tag{4}$$

where a, c, m, Z_0 are known positive integers, n ∈ N and Z_0 is the seed that the method uses.

This sequence generates a set of pseudo-random numbers that has a complete period if the following conditions are met:

1. The Greatest Common Factor of (m, a) is equal to 1.
2. If p is a prime such that p/m then $p/(a - 1)$.
3. If $4/m$ then $4/(a - 1)$.

B. Permutation

The pixel permutation procedure consists of multiplying the image width by the length of the image. Therefore, the number of elements to be scrambled corresponds to the total number of pixels in the image. The goal is to vary the position of the pixels. Given a positive integer n, the following set is defined in Equation (5):

$$Z\_n = \{m \in N \mid 0 \leq m \leq n! - 1\}\,. \tag{5}$$

Any element of Z_m can be expressed according to Equation (6). That is, expressed on a factorial basis:

$$m = A_0(n-1)! + A_1(n-2)! + \ldots A_n - 2(1)! + A_{n-1}(0)!\,. \tag{6}$$

On the other hand, according to the division algorithm, the A_i are unique. Moreover, A_n-1 = 0. Thus, the values Ai satisfy Equation (7):

$$0 \leq A\_i < (n - i)\ with\ 0 \leq i \leq (n - 2). \tag{6}$$

In this paper, the set to scramble is $\{0, 1, \cdots, n - 1\}$.

Furthermore, it is shown that the algorithm defines a bijective function [13].

The latter is highlighted because it is convenient that there are two different permutations for two positive integers $m_1, m_2 \in Z_n$. This allows us to build dynamic permutations and boxes for a cryptosystem.

C. Goodness-of-Fit Test

This tool is a statistical hypothesis test, where the null and alternative hypotheses are as follows:

– Null hypothesis. The string of bits is random.
– Alternative hypothesis. The string of bits is not random.

In addition, it is necessary to define a statistic and a level of significance, which in this investigation is $\alpha = 0.01$ [14], that determines a region of acceptance or rejection and, subsequently, the decision rule.

The statistic is shown in Equation (8), which has a χ^2 distribution with $n - 1$ degrees of freedom. Regarding the variables involved in Equation (7), it is noted that o_i, exp correspond to the observed and expected values. Furthermore, considering that each primary color is described with 256 levels (a byte), we conclude that the degrees of freedom are $n - 1 = 255$. With this argument, and in accordance with the central limit theorem, in this paper we assume that the variable χ^2 approximates a normal distribution $N(\mu, \sigma)$; where $\mu = 255$ and $\sigma = 22.58$ [15]:

$$x^2 = \sum_{i}^{k} \frac{(o_i - exp)^2}{exp}\,. \tag{8}$$

On the other hand, according to the significance level value defined above, it follows that when $\chi^2 < 308$, the null hypothesis is accepted. On the contrary, it is rejected if $\chi^2 \geq 308$.

III. Simulation Operation

We proceed to detail the procedures that the sender and the receiver must follow in the process of exchanging information through the application, along with a exposition of the nature of the noise that may affect such transmission.

A. Permutation Application

First, an image has to be selected to permute. The simulation application allows taking a photo from the camera device. Afterward, the selected photo will appear in the background, ready to be swapped. The procedure for permuting the image pixels considers preloading a permuted sequence of numbers. The user has two options: using the home screen background image from the camera or an image previously captured by themselves. The permutation algorithm works once the button “Permute” is pushed. Subsequently, the permuted image will appear on the mobile device screen. For the image-sending process, the user shares by means of selecting the receiver to whom they want to send the image. A representation of this stage is indicated in Fig. 1.

B. Noise Application

This application is considered a complement to the Android operating system. It contains information about whether an image download is in progress and, if so, provides a real-time graphical representation of the downloaded percentage. Additionally, the application offers the ability to activate a button that simulates interference in the communication path between the sender and the recipient.

ISSN 2395-8618

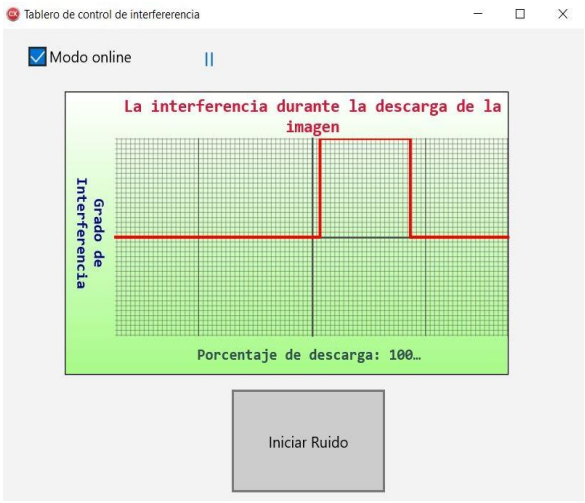


Fig. 2. Noise application interface

Fig. 3. Download in mobile application (a) Downloading progress, (b) Damaged image

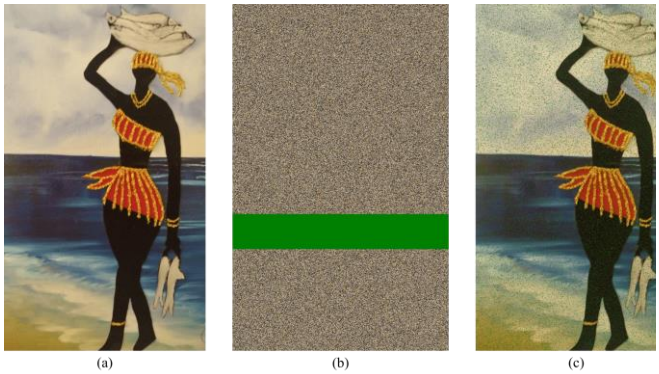


Fig. 4. *Colores del océano* simulation results (a) Original image, (b) Permuted image with occlusion noise at 10%, (c) Image after mitigation

This simulation is reflected in the mentioned graph, where the value will be equal to 1 if interference is active and 0 if it is not. In this situation, the x-axis represents the download percentage. At the beginning of the axis, 0% download is indicated, signifying that no download process is ongoing. At the point of intersection with the y-axis, 50% download is represented, while at the end of the axis, 100% download is indicated, signifying the download is complete.

On the other hand, the y-axis is used to determine the presence of interference. If the graph shows a value of 0 on the x-axis, it means there is no interference present, implying no damage occurs. However, if the graph is not in this position, it indicates the presence of interference in communication.

The mobile application can monitor the interference state and consider this in the download process. When this condition occurs, the application suspends communication with the server to acquire rows of the image and instead proceeds to fill those rows with a green color. This action is known as damage. Consequently, it involves the loss and subsequent replacement of the pixels that were supposed to be downloaded during the interference period. The percentage of damage caused to the image is also perceptible through this application. The user can observe the duration of the interference, expressed as a percentage, and determine to what extent the image was affected, even before the download is complete. The interface is shown in Fig. 2.

C. Noise Interference during Download

In this stage, the receiver to whom the image was sent. From there, the user will select the image they wish to download using the corresponding user credentials. After choosing the image to download, a screen will display a message stating “Downloading...”; at the bottom, a progress bar will reflect the download’s progress percentage. Meanwhile, when the receiver begins the image download, in the desktop program’s interface, the third user can observe the image download progress. In this graphic, the left half represents the bottom half of the image, while the right half corresponds to the upper half of the image.

The “Start Noise” introduces interference during the image download. While the “Stop Noise” button interrupts the image’s damage. Once the download is complete, the graph will resemble the resultant image. An illustrative representation is presented in Fig. 3.

IV. Results

The following simulation demonstrates the step-by-step evolution of an image from its original state to the final permuted version. This article segment provides a comprehensive look at the stages involved in image permutation, presenting a trio of visual representations: *Colores del océano* in Fig. 4, *Princesa* in Fig. 5, and *Bosque escondido* in Fig. 6.

These pictures explore the image permutation capabilities of this simulator.

In Table 1, the random distribution of pixels is tested for the permutation algorithm proposed. Since the linear congruential generator is applied, the seed is included. The results indicate that all images have a random pixel distribution.

V. Conclusions

In this paper, a real-time simulation of occlusion noise is presented. The attack occurs while an image is being

ISSN 2395-8618

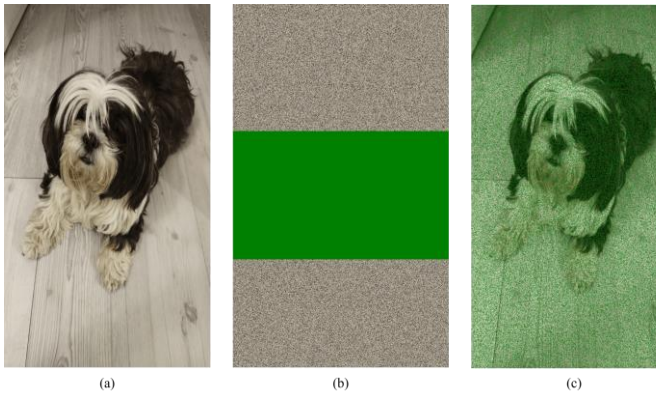


Fig. 5. *Princesa* simulation results (a) Original image, (b) Permuted image with occlusion noise at 35%, (c) Image after mitigation

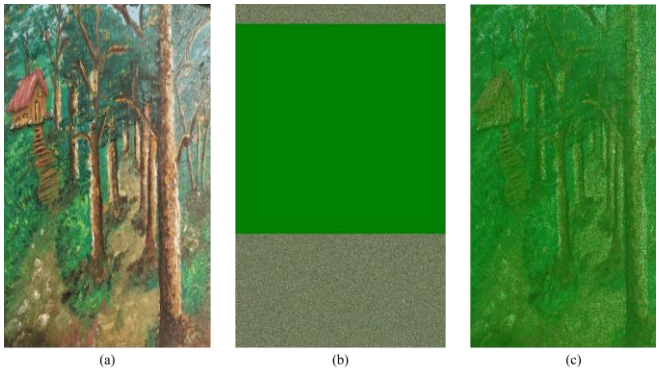


Fig. 6. *Bosque escondido* simulation results (a) Original image, (b) Permuted image with occlusion noise at 60%, (c) Image after mitigation

TABLE I
PIXELS RANDOM DISTRIBUTION TEST

Image	Width	Height	Seed	χ^2	Statistic	Accepted
Colores	560	1000	2132545692	1025.71	1073.65	A
Princesa	748	1000	1280689831	1048.79	1073.65	A
Bosque	1000	750	2079249579	765.5	813.784	A

downloaded by a third party. Under these circumstances, we examine the damage to it.

Additionally, a permutation is proposed to mitigate the effects and spread out the noise. Consequently, the image information is recognizable even at a 60% loss of information.

On the other hand, this kind of intrusion does not aim to obtain information but replaces the original data with noise. In this sense, the mitigation can work when communication cannot wait to be transferred and is interrupted by an attack.

Finally, since the information is received with noise, an image filter should be included to improve the visual quality.

ACKNOWLEDGMENTS

The authors would like to thank the Instituto Politécnico Nacional of Mexico (Secretaría Académica, Comisión de Operación y Fomento de Actividades Académicas COFAA, SIP, CIDETEC, and CIC), and the CONAHCyT for their support in the development of this work.

REFERENCES

[1] W. Du and S. Tian, "Transformer and GAN based super-resolution reconstruction network for medical images," *Tsinghua Science and Technology*, vol. 29, pp. 197–206, 2024. DOI: 10.26599/TST.2022.9010071.

[2] J. Alatalo, T. Sipola, and M. Rantonen, "Improved difference images for change detection classifiers in SAR imagery using deep learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 6, pp. 1–14, 2023. DOI: 10.1109/TGRS.2023.3324994.

[3] L. He, J. Shan, and D. Aliaga, "Generative building feature estimation from satellite images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61 pp. 1–13, 2023. DOI: 10.1109/TGRS.2023.3242284.

[4] M.W. Hafiz, W.K. Lee, S.O. Hwang, M. Khan, and A. Latif, "Discrete logarithmic factorial problem and Einstein crystal model based public-key cryptosystem for digital content confidentiality," *IEEE Access*, vol. 10, pp. 102119–102134, 2022. DOI: 10.1109/ACCESS.2022.3207781.

[5] H. Nazir, I.S. Bajwa, S. Abdullah, R. Kazmi, and M. Samiullah, "A color image encryption scheme combining hyperchaos and genetic codes," *IEEE Access*, vol. 10, pp. 14480–14495, 2022. DOI: 10.1109/ACCESS.2022.3143096.

[6] M. Jin, L. Yu, K. Zhou, and Q. Yi, "Occlusion tolerant object recognition using visual memory selection model," *Applied Intelligence*, vol. 52, pp. 15575–15599, 2022.

[7] Z. He, X. Lan, J. Yuan, and W. Cao, "Multi-layer noise reshaping and perceptual optimization for effective adversarial attack of images," *Applied Intelligence*, vol. 53, pp. 7408–7422, 2023.

[8] P. Aberna and L. Agilandeeswari, "Digital image and video watermarking: Methodologies, attacks, applications, and future directions," *Multimedia Tools and Applications*, pp. 1–61, 2023.

[9] S. Jung, M. Chung, and Y.G. Shin, "Adversarial example detection by predicting adversarial noise in the frequency domain," *Multimedia Tools and Applications*, vol. 82, pp. 25235–25251, 2023.

[10] M. K. Rusia and D. K. Singh, "A comprehensive survey on techniques to handle face identity threats: Challenges and opportunities," *Multimedia Tools and Applications*, vol. 82, pp. 1669–1748, 2023.

[11] X. Wang, J. Liu, S. Chen, and G. Wei, "Effective light field de-occlusion network based on swin transformer," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, pp. 2590–2599, 2023.

[12] G. Verma, W. He, and X. Peng, "A novel four image encryption approach in sparse domain based on biometric keys," *Multimedia Tools and Applications*, vol. 82, pp. 22889–22904, 2023.

[13] J. A. Gallian, *Contemporary abstract algebra*, CRC Press, Boca Raton, 2021.

[14] Z. Liu, J. Shen, R. Barfield, J. Schwartz, A. A. Baccarelli, and X. Lin, "Large-scale hypothesis testing for causal mediation effects with applications in genome-wide epigenetic studies," *Journal of the American Statistical Association* vol. 117, pp. 67–81, 2022.

[15] P. Bourgade, K. Mody, and M. Pain, "Optimal local law and central limit theorem for β-ensembles," *Communications in Mathematical Physics*, vol. 390, pp. 1017–1079, 2022.