INFORMATICA, 2025, Vol. 36, No. 4, 1013–1035

DOI: https://doi.org/10.15388/25-INFOR608

Implementation of an Enhanced MPF-Based Block Cipher to Encrypt Digital Images

Jokubas ZITKEVICIUS[1], Aleksejus MIHALKOVICH[2,3,*], Eligijus SAKALAUSKAS[2]

[1] *Vilnius University, Naugarduko str. 24, Lithuania*
[2] *Kaunas University of Technology, Studentu str. 50, Lithuania*
[3] *Vilnius University, Kaunas Faculty, Muitines str. 8, Lithuania*
e-mail: jokubas.zitkevicius@mif.stud.vu.lt, aleksejus.michalkovic@ktu.lt, eligijus.sakalauskas@ktu.lt

Received: June 2025; accepted: October 2025

Abstract. In this paper, we present an enhanced version of a previously published symmetric block cipher implemented for the encryption of digital images. We introduce an additional step of using Arnold's cat map prior to encryption to improve its quality. After inspecting the statistical characteristics of the ciphertexts for the electronic codebook (ECB) and cipher block chaining (CBC) modes, we found that with this additional step, our schemes produce high entropy ciphertexts for both regular and monochromatic images. Therefore, the results obtained in this paper show that our schemes are indifferent to the Advanced Encryption Standard (AES) cipher. Moreover, due to an effective parallelization of matrix operations, we think that our proposal can be executed reasonably fast.

Key words: non-commutative cryptography, symmetric encryption, statistical analysis.

1. Introduction

The need to hide sensitive information has existed since ancient times. Nowadays, it has become even more important due to the rapid development of technology and multimedia. Transmissions of images and videos play an important role in medical systems, military image databases, and other areas (Li, 2006). Furthermore, our everyday electronic devices are used to transmit images, video clips, or music over wireless networks. Therefore, various algorithms based on mathematical calculations were proposed to securely hide this information, which gave birth to modern cryptography.

One of the simplest ideas was introduced in 1919 by Vernam. In modern terms, his proposal can be formalized as follows: given the message bit string μ, the sender can use the key string k of the same length and add it to the original message bit-wise to obtain the ciphertext $c = \mu \oplus k$. Here, we used $\oplus$ to denote the XOR operation. The receiver uses the ciphertext c to restore the original message by adding k to it bit-wise, i.e. we have $\mu = c \oplus k$. The presented idea is now known as the Vernam cipher.

*Corresponding author.

Modern cryptography has gone a long way since then. In this work, we focus on the symmetric cryptography branch, which covers block and stream ciphers to encrypt messages. For implementation purposes, block ciphers are usually converted into stream ciphers by applying them in various modes of encryption to obtain the ciphertext.

One of the most popular block ciphers was standardized in Dworkin *et al.* (2001) and is now known as the Advanced Encryption Standard (AES). It uses a 128, 192, or 256-bit secret key to encrypt a 4×4 block of 16-bit entries. Depending on the size of the secret key, either 10, 12 or 14 rounds of four steps are performed.

Symmetric encryption is also a tool for hiding visual information. Over the years, many image encryption techniques have been proposed. One of the most popular ways to encrypt an image is by applying some chaotic mapping, e.g. the logistic map (Zia *et al.*, 2022; Zhang and Liu, 2023). We use one of such maps, namely Arnold's cat mapping, in our paper. These mappings can produce satisfactory results when used with statistically independent sampling and binary key derivation (Dinu and Frunzete, 2025). Many researchers recommend chaos-based encryption techniques for images due to their computational efficiency (Zia *et al.*, 2022). However, generating large chaotic sequences is time-consuming, and, therefore, the real-time application of such schemes is problematic. For this reason, parallelization of computations is used in Daoui *et al.* (2023) to speed up the running time. This is where the matrix computations can be helpful due to the natural parallelization of matrix operations. We view it as one of the advantages of our scheme over other schemes.

Other researchers incorporate elliptic curves into their work (Hernández-Díaz *et al.*, 2021; Chillali and Oughdir, 2023; Nagaraj *et al.*, 2015; Singh *et al.*, 2024). For example, in Hernández-Díaz *et al.* (2021), elliptic curves are used to generate encryption keys, while in Singh *et al.* (2024), elliptic curve points combined with a hash function are utilized to create dynamic S-boxes. However, calculations in elliptic curves are more complex than the structures discussed in this paper. Additionally, our approach relies solely on a highly non-linear mapping called the matrix power function (MPF), rather than a combination of two different methods.

In our paper (Dindiene *et al.*, 2022), we presented a block cipher based on MPF. We have shown how to encrypt a single block of data using MPF as opposed to the multiple rounds approach used in AES. Later, in our papers (Levinskas and Mihalkovich, 2021; Mihalkovich *et al.*, 2022a), we analysed the statistical properties and performance of our proposal implemented in various modes of encryption. The results obtained in Mihalkovich *et al.* (2022a) have shown that by carefully choosing the parameters of the system, our proposal can be executed in time comparable to the AES cipher. Moreover, in Mihalkovich *et al.* (2022c) and Mihalkovich *et al.* (2022b), we have extended our research to non-commuting algebraic structures.

While the obtained results have shown some promise, in this paper, we make several changes as compared to the original idea. The most significant change comes from applying an extra mapping to the original data prior to encrypting it with our cipher. We demonstrate how this additional step affects the quality of encryption and compare our results to AES. The proposed changes can be summarized as follows:

- We apply Arnold's cat mapping to the initial image, aiming to make it more chaotic prior to encrypting.
- We implement the Galois fields in our scheme, aiming to simplify the implementation of our cipher.
- We now use a different key matrix for each of the colour channels.

The rest of the paper is organized as follows: in Section 2 we present the key definitions of the algebraic structures and functions we use throughout the paper; in Section 3 we introduce the changes to the original idea; in Section 4 we investigate the quality of encryption in two modes, where the finite field of integers $\mathbb{Z}_p$ is used as a platform group; in Section 5 we implement Galois Fields in our proposal and compare the quality of encryption to the results of the previous section. As mentioned previously, we also compare our results to analogous modes of AES in both sections. Conclusions are presented at the end of the paper.

2. Mathematical Background

First, we consider the finite field of integers $\mathbb{Z}_p$, where p is a prime.

Definition 1 (Detomi *et al.*, 2025). If q^k is the highest power of a prime q dividing $p-1$, then a multiplicative subgroup $\mathbb{G}_q$ of order q^k in $\mathbb{Z}_p^*$ is called a *Sylow q-subgroup* of $\mathbb{Z}_p^*$.

In this paper, we use a special case of Sylow q-subgroup, where $k = 1$ and two primes p and q are linked via a relation $p = 2q + 1$. Therefore, the Sylow subgroup has the following representation:

$$\mathbb{G}_q = \left\langle g^{2i} \,\middle|\, i = 0, 1, \ldots, q-1 \right\rangle, \tag{1}$$

where g is a generator of the multiplicative group $\mathbb{Z}_p^*$. Notably, each non-identity element of $\mathbb{G}$ generates the whole group. We also define a mapping $\gamma : \mathbb{Z}_q \mapsto \mathbb{G}_q$ as follows:

$$\gamma(x) = g^{2x}.$$

Clearly, γ is an invertible mapping. In this paper, we use this mapping to interchange between the elements of $\mathbb{Z}_q$ and $\mathbb{G}_q$ to fit our needs.

Note that the Sylow subgroup can be defined in a more general case. However, our research does not require considering it.

Using the multiplicative group $\mathbb{G}_q$ and a ring of integers $\mathbb{Z}_q$, we can define the following mapping:

Definition 2 (Sakalauskas *et al.*, 2008). Assume that $\mathbf{W} \in \mathbb{G}_q^{m\times m}$ is an $m \times m$ matrix with entries in $\mathbb{G}_q$, and $\mathbf{X}, \mathbf{Y} \in \mathbb{Z}_q^{m\times m}$ are two matrices with entries in $\mathbb{Z}_q$. We define *the two-sided matrix power function* (MPF) as a mapping $\mathbb{Z}_q^{m\times m} \times \mathbb{G}_q^{m\times m} \times \mathbb{Z}_q^{m\times m} \mapsto \mathbb{G}_q^{m\times m}$

denoted by $\mathbf{E} =^{\mathbf{X}} \mathbf{W}^{\mathbf{Y}}$, where the entries of the MPF value matrix $\mathbf{E}$ are calculated as follows:

$$e_{ij} = \prod_{k=1}^{m} \prod_{l=1}^{m} w_{kl}^{x_{ik} \cdot y_{lj}}. \tag{2}$$

Definition 3 (Sakalauskas and Mihalkovich, 2014). We refer to the group $\mathbb{G}_q$ in Definition 2 as a *platform group*.

Definition 4 (Sakalauskas and Mihalkovich, 2014). We refer to the group $\mathbb{Z}_q$ in Definition 2 as a *power ring*.

In this paper, we compare two platform groups: the Sylow q-subgroup $\mathbb{G}_q$ and the Galois Field $\mathbb{GF}(q^k)$.

In our previous paper (Dindiene *et al.*, 2022), we considered a symmetric block cipher based on the MPF, where we used the Sylow q-subgroup $\mathbb{G}_q$ as a platform group along with an additional one-to-one mapping Γ (an entry-wise application of γ), which was used to map the initial values of the pixels to the elements of $\mathbb{G}_q$. However, as we have shown in Levinskas and Mihalkovich (2021), the function Γ does not help to achieve good statistical properties of our cipher. Therefore, here we use a so-called Arnold's cat map defined below:

Definition 5 (Pardede *et al.*, 2018). The generalized form of *Arnold's cat map* for digital $N \times N$ image applications is a chaotic discrete map:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & a \\ b & ab+1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \bmod N, \tag{3}$$

where each element of the set $\{x, x', y, y'\}$ comes from $\mathbb{Z}_N$, whereas a and b are two integers.

In this paper, we use the parameter values $a = 1$ and $b = 1$. Note that Arnold's cat map is invertible since the determinant of the transformation matrix is equal to 1.

Furthermore, in our research, we use the Hadamard product, otherwise known as the entry-wise multiplication of two matrices, and its inverse, denoted by an upper index H. In other words, we have:

Definition 6. The matrix $\mathbf{V} = \mathbf{W}^H$ is a Hadamard inverse of matrix $\mathbf{W}$, if its entries are the multiplicative inverses of the corresponding entries of $\mathbf{W}$, i.e.

$$v_{ij} = w_{ij}^{-1}. \tag{4}$$

Also, we define the 3×3 correlation matrix Σ_{avg}, where each row corresponds to a distinct channel, and each column represents a different direction of a neighbouring pair:

horizontal, vertical, or diagonal. Each entry in the matrix corresponds to the correlation coefficient between neighbouring pixel intensities for the respective pair type.

DEFINITION 7 (Broumandnia, 2020). For an $m \times m$ matrix, the types of neighbouring pairs are defined as follows:

- For each row i, the horizontal neighbouring pair consists of pixels at positions $\{(i, j), (i, j+1)\}$, where $1 \leqslant j < m$.
- For each column j, the vertical neighbouring pair consists of pixels at positions $\{(i, j), (i+1, j)\}$, where $1 \leqslant i < m$.
- For each pixel at position (i, j), the diagonal neighbouring pair consists of pixels at positions $\{(i, j), (i+1, j+1)\}$, where $1 \leqslant i, j < m$.

We use this correlation matrix as one of the tools to investigate relations between the entries of the encrypted blocks.

3. MPF-Based Block Cipher with Arnold's Cat Map

In this section, we revise the block cipher considered in Dindiene *et al.* (2022). We present here a modified version of our cipher, since we implement Arnold's cat map together with the previously used function Γ. Therefore, we introduce an additional step applied to the whole image to improve the quality of encryption.

Let μ be a string of bits representing the initial message. We cut this string into m^2 8-bit chunks and therefore obtain its representation in the following form:

$$\mu = \mu_{11} \| \mu_{12} \| \dots \| \mu_{1m} \| \mu_{21} \| \mu_{22} \| \dots \| \mu_{2m} \| \dots \mu_{mm},$$

where $\|$ is the concatenation operation. Furthermore, if the message is too short, we append the appropriate amount of random bits at the end of the message. We can now construct the matrix representation $\mathbf{M}$ of the initial message, where the entries of $\mathbf{M}$ are integers $\mu_{ij} \in [0, 255]$. We apply Arnold's cat map to the matrix $\mathbf{M}$ a fixed number of times and denote the obtained result by $\mathbf{M}' = \mathrm{ACM}(\tau, \mathbf{M})$, where τ is the number of iterations. We now encrypt $\mathbf{M}'$ using a secret key – a triplet of matrices $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$, where $\mathbf{X}, \mathbf{Y} \in \mathbb{Z}_q^{m \times m}$, $\mathbf{Z} \in \mathbb{G}_q^{m \times m}$, $q > 255$, matrix $\mathbf{X}$ does not contain any zero entries and $\mathbf{Y}$ is invertible. For application purposes, we want to keep the values of q and p as small as possible. Hence, in this paper, we set $q = 281$ and $p = 563$.

The encryption algorithm consists of the following steps:

$$\mathbf{S}_1 \equiv \left(\mathbf{X} + \mathbf{M}'\right) \bmod q;$$
$$\mathbf{S}_2 \equiv \left(\mathbf{Z} \odot {}^{\mathbf{Y}}\Gamma(\mathbf{S}_1)^{\mathbf{Y}}\right) \bmod p;$$
$$\mathbf{S} \equiv \left(\Gamma^{-1}(\mathbf{S_2}) + \mathbf{X}\right) \bmod q,$$

where $\Gamma(\mathbf{X}) : \mathbb{Z}_q^{m\times m} \mapsto \mathbb{G}_q^{m\times m}$ is a publicly known one-to-one mapping which replaces entries of matrix $\mathbf{X}$ with elements from $\mathbb{G}_q$ – a Sylow subgroup of $\mathbb{Z}_p$. Clearly, $\Gamma^{-1}(\mathbf{S}_2)$ is the inverse transformation. We use $\odot$ to denote the Hadamard product of two matrices.

Note that we use both q and p to perform modular operations. Therefore, we specify the modulus for each step of the encryption algorithm. Clearly, the modular arithmetic is applied entry-wise.

The decryption algorithm works similarly in reverse, i.e.:

$$\begin{aligned}
&\mathbf{D}_1 \equiv (\mathbf{S} - \mathbf{X}) \bmod q;\\
&\mathbf{D}_2 \equiv {}^{\mathbf{Y}^{-1}}\left(\Gamma(\mathbf{D}_1) \odot \mathbf{Z}^H\right)^{\mathbf{Y}^{-1}} \bmod p;\\
&\mathbf{M}' \equiv \left(\Gamma^{-1}(\mathbf{D}_2) - \mathbf{X}\right) \bmod q.
\end{aligned}$$

Now all that remains is to apply Arnold's cat map to $\mathbf{M}'$ in reverse to obtain the initial message in the matrix form $\mathbf{M}$ and then concatenate its entries as presented above to restore the message μ.

Previously, in Dindiene *et al.* (2022), we have shown that the considered block cipher is perfectly secure, i.e. it satisfies the following property:

$$\Pr[c = c_0 | \mu = \mu_0] = \Pr[c = c_0], \tag{5}$$

where c_0 and μ_0 are some fixed ciphertext and plaintext values. In other words, the ciphertext is statistically independent of the original message.

Perfect secrecy property was previously proven for the Vernam cipher. However, due to one-time secret keys and several other disadvantages, the Vernam cipher remains a purely theoretical algorithm: while easy to understand, it is never used in the real world.

One disadvantage of any perfectly secure cipher is the size of the secret key, since it has to be at least as long as the plaintext. This, together with one-time use of the key, prevents applying any modes of encryption to long messages. On the other hand, we have shown in Dindiene *et al.* (2022) that in our case, the secret key can be reused. Therefore, the perfectly secure block cipher we consider can be transformed into a stream cipher by applying modes of symmetric encryption. We consider two of these modes in this paper.

4. Modes of Symmetric Encryption

Like AES, our cipher operates on blocks. Therefore, to convert our idea to a stream cipher, we have to apply one of the encryption modes. Two of such modes considered in this paper are the electronic codebook (ECB) mode and the cipher block chaining (CBC) mode. A well-known fact is that the ECB mode is considered insecure even for such ciphers as AES. In this paper, we explore how Arnold's cat mapping affects the quality of the ECB mode of our block cipher. Moreover, we perform a comparison of CBC modes for AES and our proposal.

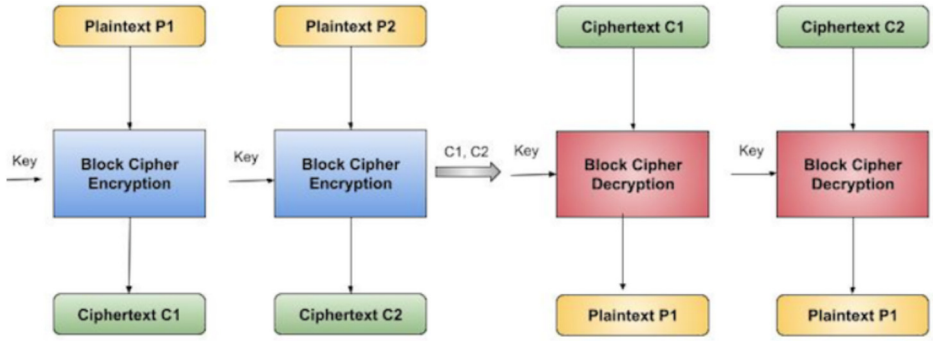


Fig. 1. ECB Mode.

4.1. *ECB Mode*

ECB mode of symmetric encryption is the simplest of modes, since it is applied block-wise (Elashry *et al.*, 2012). The encrypted blocks are restored using the same key by performing the actions in reverse as shown in the previous section. The general idea of ECB mode is presented in Fig. 1.

Since the digital images are in RGB format, we perform the algebraic process for each of the three colours separately. Therefore, the whole image's information is stored in a three-dimensional array. Block size is set by default to 4, i.e. the size of the single-coloured sub-image is 4×4. The same size is inherited for the ciphers and other inter-components of the encryption. We encrypt each of the colour matrices using separate keys rather than the same one as we did in our previous papers.

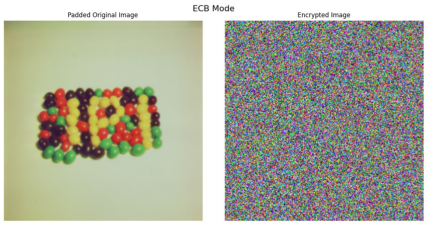
Note that since all intensities of the image pixels are represented as 8-bit integers, we can interpret them as elements of $\mathbb{Z}_q$ and map them to the multiplicative Sylow q-subgroup $\mathbb{G}_q$ using a mapping γ.

We generate matrices $\mathbf{X} \in \mathbb{Z}_q^{4\times4}$, $\mathbf{Y} \in \mathbb{Z}_q^{4\times4}$, $\mathbf{Z} \in \mathbb{G}_q$ with random entries uniformly distributed in the appropriate sets. We also remove 0 from the set of possible values when generating $\mathbf{Y}$. Therefore, we obtain a key triplet $\vec{\mathbf{K}} = (\mathbf{X}, \mathbf{Y}, \mathbf{Z})$. We generate a separate key for each colour, thus obtaining a 3-part key $\mathbf{K}_{rgb} = (\vec{\mathbf{K}}_r, \vec{\mathbf{K}}_g, \vec{\mathbf{K}}_b)$. Then we perform the ECB mode of our block cipher for each of the colour matrices using an appropriate key. Example results for various images[1] are shown in Figs. 2a–2d.

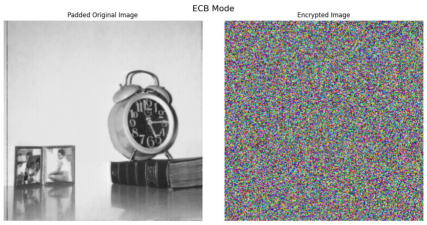
Let us compare this result to the ECB mode of the AES cipher. Using the same images as the original plaintexts, we get the results presented in Figs. 3a–3d.

We can see that the ciphertexts produced using both ciphers in ECB mode look chaotic. However, this is not always the case for other images. A well-known downside of the ECB is noticeable patterns in the encrypted image. For this reason, this mode of encryption is not used even for such secure ciphers as AES. These patterns can also be seen if our

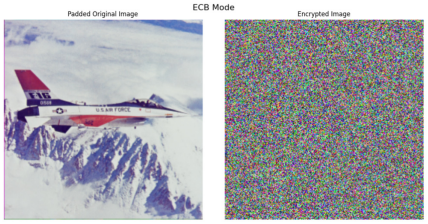
[1] All the images used in this paper are open access and can be found at https://sipi.usc.edu/database/database.php?volume=misc



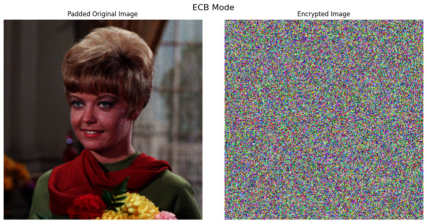
(a) 'Skittles.png'



(b) 'Clock.png'

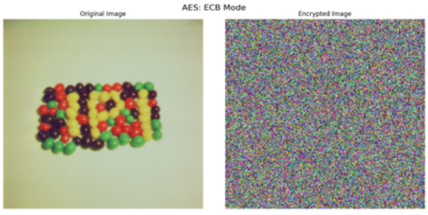


(c) 'Plane.png'

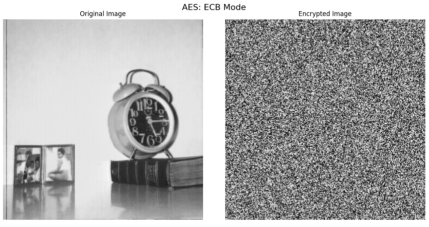


(d) 'Woman.png'

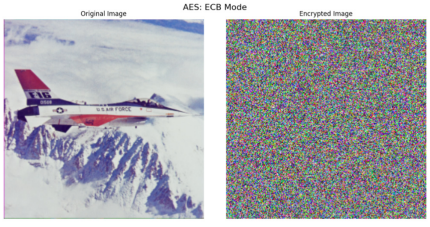
Fig. 2. ECB Mode of MPF-based cipher applied to various images, Arnold's cat map not used.



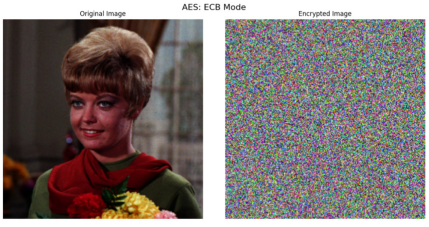
(a) 'Skittles.png'



(b) 'Clock.png'



(c) 'Plane.png'



(d) 'Woman.png'

Fig. 3. ECB Mode of AES cipher applied to various images.

previous version of the MPF-based block cipher is applied in ECB mode, where only the mapping Γ was used. We present an example of this below in Fig. 4.

We can observe that the generated cipher has obvious patterns in the form of noticeable strips. This is the consequence of the ECB mode: repeatable low-intensity blocks produce

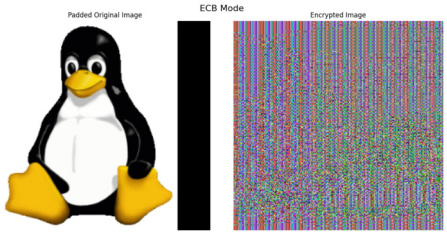
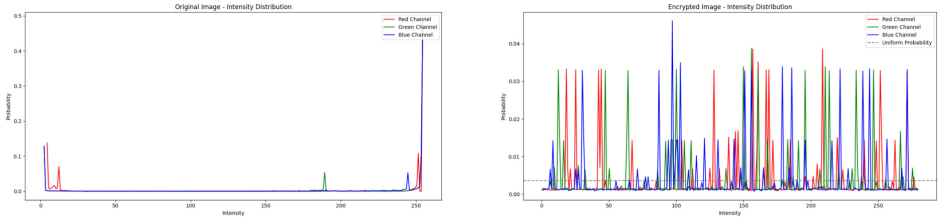


Fig. 4. ECB Mode of MPF-based cipher applied to Linux penguin, Arnold's cat map is not used.



(a) Plaintext (b) Ciphertext

Fig. 5. Pixel intensity distribution for the Linux penguin image and its ciphertext.

a pattern in the encrypted image, which reduces the quality of encryption. Interestingly enough, even using three separate keys instead of one does not improve the result by much.

Let us now plot the distribution of the colour intensities for the original Linux penguin image and its ciphertext in Fig. 5a and 5b, respectively. We can see from Fig. 5a that intensities greater than 250 dominate in the initial picture. After applying the ECB mode (Fig. 5b), roughly one third of intensity probabilities are higher than 3 times the uniform probability, which strongly suggests a non-uniform distribution.

Let us observe the average correlation between the pixels. In reference to Definition 7, we output the Σ_{avg} matrix as:

$$\Sigma_{\text{avg}} = \begin{bmatrix} 0.158 & 0.227 & 0.084 \\ -0.041 & 0.259 & -0.051 \\ -0.181 & 0.222 & -0.11 \end{bmatrix}. \tag{6}$$

We can see that the maximum absolute correlation was 0.259, which shows a weak, yet statistically significant, positive correlation for the green channel vertically. Also, for each row, the value in the second column is the largest one, which suggests that vertical correlation dominates in the initial picture. This is correct because the low-intensity block pattern repeats mostly vertically.

(a) Linux penguin (b) Blank image

Fig. 6. ECB Mode of AES cipher applied to monochromatic images.

Furthermore, we perform the analysis of the entropy for each channel. Asif *et al.* (2022) We output the vector that consists of entropies for each channel.

$$\mathbf{H}_{RGB} = \begin{bmatrix} 6.904 \\ 6.862 \\ 6.865 \end{bmatrix}. \tag{7}$$

Since the maximum theoretical entropy is

$$h(x) = -\sum_{i=0}^{280} p(x_i)\log_2 p(x_i) = -281 \cdot 281^{-1} \cdot \log_2\left(281^{-1}\right) \approx 8.134, \tag{8}$$

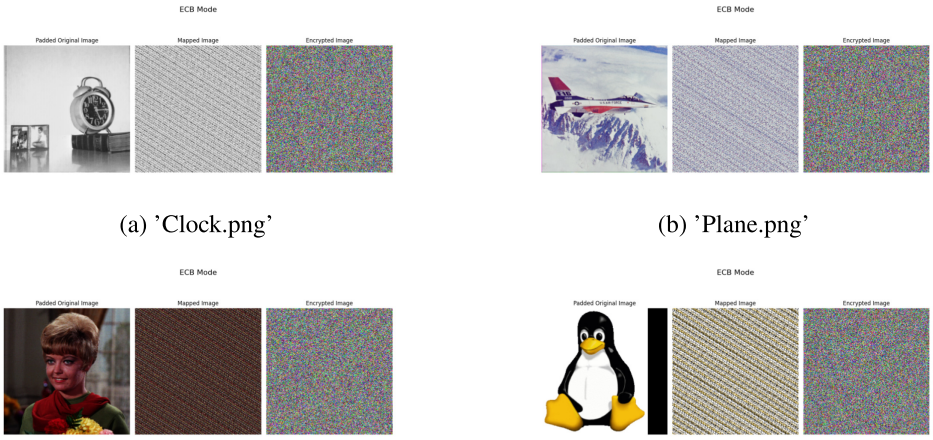
so the result is far from a statistically sufficient one.

The main difference between the Linux penguin and the presented images is that the latter image can be viewed as somewhat monochromatic since there is a dominating white colour. We can also see a similar problem of encryption quality with the AES cipher in ECB mode presented in Fig. 6a. Wherever there are more colours in the original image, the ciphertext looks more chaotic than the areas of dominating white colour. This can be clearly seen if we consider a blank (totally white) image and apply the AES cipher in the ECB mode to it. The result is presented in Fig. 6b.

Now we apply the ECB mode of encryption with the same parameters on an image transformed by Arnold's cat map, where $\tau = 5$. In other words, for each colour, we first view the whole image as a large matrix and apply Arnold's cat map to it. Then we split the obtained output matrix into blocks for encryption using ECB mode. The results for various images are presented below in Figs. 7a–7d.

The implementation of Arnold's cat map to the Linux penguin image reduced the maximum absolute correlation to 0.009, and the average entropy for a channel is 8.131. Therefore, we can see that Arnold's cat map enhances the statistical results of the ECB mode. This is due to the removal of dominance of the white colour (see the middle image in Fig. 7d).

However, if we apply Arnold's cat mapping to a blank image, no changes happen since all the values in the image matrix are the same. Therefore, the encryption quality of the ECB mode of our cipher is unaffected in this case. Hence, while we managed to cope



(a) 'Clock.png' (b) 'Plane.png'

(c) 'Woman.png' (d) Linux penguin

Fig. 7. ECB Mode of MPF-based cipher applied to various images, Arnold's cat map is used.

with some weaknesses of the ECB mode using an additional mapping, the ciphertext of a monochromatic image still has low quality.

4.2. *CBC Mode*

The CBC encryption mode is more advanced than the ECB mode. It requires an initialization vector, which is sampled randomly from a uniform distribution (Li *et al.*, 2023). This initialization vector is added to the sub-image using XOR or standard addition. The resulting matrix is then encrypted using a block cipher. This encrypted block is passed on as a new initialization vector for a new iteration. Then we repeat this process for each of the other given sub-images until all blocks are encrypted. Since the process of encrypting each block is dependent on previous iterations, it forms a chain of blocks.

The decryption of CBC starts from the encrypted block that was calculated in the latest iteration. We decrypt the blocks until the step with the initialization vector. We repeat the steps of decryption for each cipher block until we reach the very first deciphered block, where we subtract the public initialization vector.

The general idea of the CBC mode is presented below:

To implement this mode for the MPF-based block cipher, we use the initialization vector in the form of a 4×4 matrix with entries randomly sampled from $\mathbb{Z}_{256}$. We denote this matrix by $\mathbf{C}_0$. Then the encryption is performed as follows:

$$\begin{aligned}
\mathbf{C}_{i1} &\equiv \left(\mathbf{X} + \mathbf{C}_{i-1} + \mathbf{M}_i'\right) \bmod q \\
\mathbf{C}_{i2} &\equiv \left(\mathbf{Z} \odot {}^{\mathbf{Y}}\Gamma(\mathbf{C}_{i1})^{\mathbf{Y}}\right) \bmod p; \\
\mathbf{C}_i &\equiv \left(\Gamma^{-1}(\mathbf{C_{i2}}) + \mathbf{X}\right) \bmod q,
\end{aligned}$$

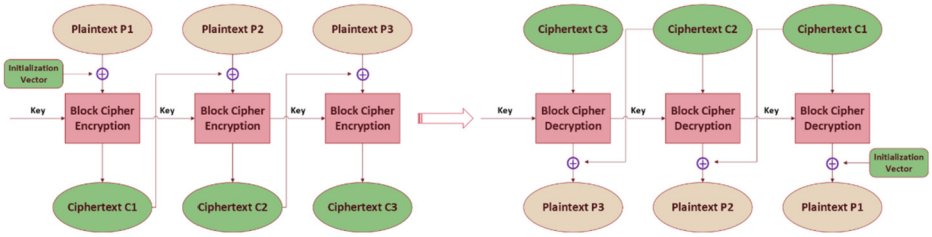
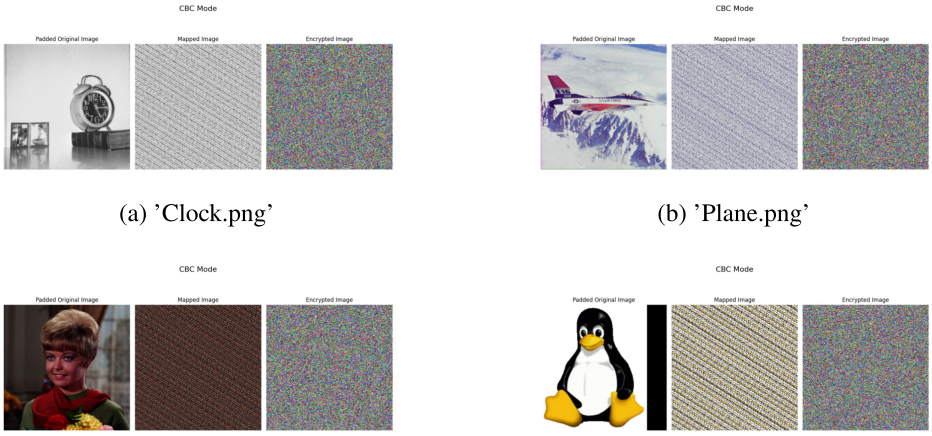


Fig. 8. CBC Mode.



(a) 'Clock.png' (b) 'Plane.png'

(c) 'Woman.png' (d) Linux penguin

Fig. 9. CBC Mode of MPF-based cipher applied to various images, Arnold's cat map is used.

where $\mathbf{C}_i$ is the i-th encrypted block and matrices $\mathbf{C}_{i1}$ and $\mathbf{C}_{i2}$ are intermediate results corresponding to that block. The obtained matrix $\mathbf{C}_1$ and all of the upcoming encrypted blocks are used as initialization vectors for the next block encryption.

The decryption works similarly, but all encryption steps are reversed, just as in the previous section.

We apply the CBC mode for each colour individually, generating a new initialization vector for each colour matrix. Example results are displayed below in Figs. 9a–9d. We use ACM$(5, \mathbf{M})$ mapping and apply a CBC mode to the obtained result.

Let us consider the quality of encryption of two images, namely the 'Woman.png' and the Linux penguin image presented in Fig. 9c and Fig. 9d. First, we present the correlation matrices Σ_{avg} for the obtained encrypted images:

$$\Sigma_{\text{avg}}(\text{'Woman.png'}) = \begin{bmatrix} 0.001 & -0.002 & -0.004 \\ -0.004 & -0.004 & 0.005 \\ -0.005 & 0.000 & 0.003 \end{bmatrix},$$

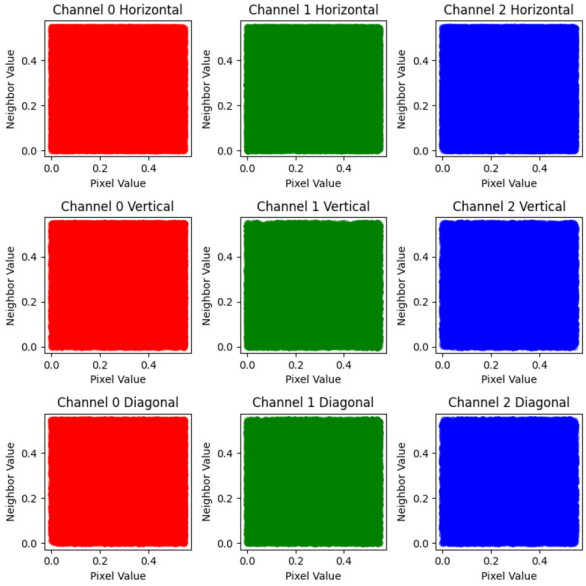


Fig. 10. Scatter plot for Linux penguin image using MPF-based block cipher in CBC, ACM(5, **M**) was used.

$$\Sigma_{\text{avg}}(\text{Linux penguin}) = \begin{bmatrix} -0.007 & 0.002 & 0.002 \\ -0.001 & -0.001 & 0.007 \\ -0.007 & 0.002 & -0.009 \end{bmatrix}.$$

The correlations above indicate significantly low or a lack of correlation. Furthermore, the results are similar for both images. Therefore, the adversary cannot distinguish between two images based on the encrypted image, which is evidence of the semantic security property.

Let us also present the scatter plot for the correlations of the encrypted Linux penguin in Fig. 10.

We can see that there are no apparent correlations between the observations.

Now we consider the entropy for each channel of the encrypted images:

$$\mathbf{H}_{RGB}(\text{'Woman.png'}) = \begin{bmatrix} 8.131 \\ 8.131 \\ 8.131 \end{bmatrix}, \qquad \mathbf{H}_{RGB}(\text{Linux penguin}) = \begin{bmatrix} 8.131 \\ 8.131 \\ 8.131 \end{bmatrix}, \quad (9)$$

where the target entropy is 8.134, which means the observed values for each channel are very close to the target entropy. Furthermore, we have obtained the same entropy values for both encrypted images. Therefore, the encrypted images are statistically indistinguishable from the white noise.

Now we repeat the previously described process 5 times with the Linux penguin image, each time generating the keys and initialization vectors at random. We observe how the calculated entropy for each encrypted image relates to the theoretical maximum in Fig. 11.

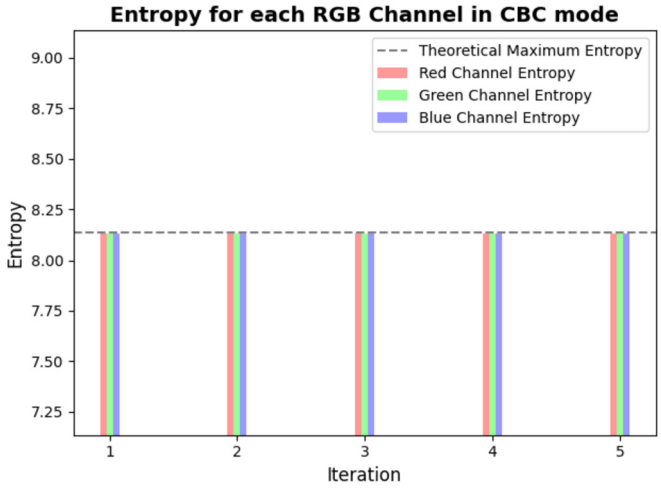


Fig. 11. Entropy plot for the encrypted Linux penguin image.

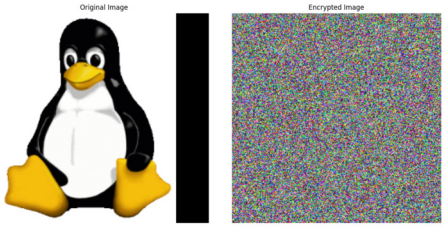


Fig. 12. CBC Mode of AES-128 cipher applied to Linux penguin image.

We can see that the entropies remained relatively close to the theoretical maximum, which is represented by a dashed line.

Let us compare the obtained results to the CBC mode of the AES-128 block cipher. We use the same Linux penguin image. The result of AES encryption is given below in Fig. 12.

The correlation matrix Σ_{avg} for the obtained encrypted image is:

$$\Sigma_{\text{avg}} = \begin{bmatrix} 0.002 & 0.003 & 0.004 \\ 0.003 & -0.001 & -0.002 \\ -0.001 & -0.001 & -0.005 \end{bmatrix}. \tag{10}$$

AES cipher maximum absolute correlation is 0.005, and the entropies for each channel are

$$\mathbf{H}_{RGB} = \begin{bmatrix} 7.997 \\ 7.997 \\ 7.997 \end{bmatrix}, \tag{11}$$

whereas the target entropy is 8.0.

We can conclude that for the considered platform group $\mathbb{G}_q$, the CBC mode of the MPF-based block cipher with Arnold's cat map is statistically indifferent to the AES cipher.

5. Galois Field

We can also use the Galois fields in our approach for better comparison with AES. In this paper, we consider the Galois field $\mathbb{GF}(2^9)$ as the platform group. We use the irreducible polynomial $\phi(x) = x^9 + x^4 + 1$ due to the adjusted default parameter in the Python library for the Galois field. As in the case of AES, the private key matrices $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ have size 4×4 with their entries $x_{ij} \in [1, 256]$, $y_{ij} \in [1, 510]$, $z_{ij} \in \mathbb{GF}(2^9) \setminus \{0\}$. Moreover, $\mathbf{Y}$ is an invertible matrix. The entries of the message block matrix $\mathbf{M}$ are integers in the range $[0, 255]$. As previously, we apply Arnold's cat map to the matrix $\mathbf{M}$ τ times and denote the obtained result by $\mathbf{M}' = \mathrm{ACM}(\tau, \mathbf{M})$. We now use the secret key $\vec{\mathbf{K}} = (\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ to encrypt a single block $\mathbf{M}'$ as follows:

$$\begin{aligned}
&\mathbf{S}_1 = \mathbf{X} + \mathbf{M}';\\
&\mathbf{S}_2 \equiv \left(\mathbf{Z} \odot {}^{\mathbf{Y}}\Gamma(\mathbf{S}_1)^{\mathbf{Y}}\right) \bmod \phi(x);\\
&\mathbf{S} \equiv \left(\Gamma^{-1}(\mathbf{S_2}) + \mathbf{X}\right) \bmod 2^9,
\end{aligned}$$

where $\Gamma(\mathbf{S}_1) : (1 + \mathbb{Z}_{511})^{4\times 4} \mapsto \mathbb{GF}^{4\times 4}(2^9)$ is a publicly known one-to-one mapping which replaces entries of matrix $\mathbf{S}_1$ with elements from $\mathbb{GF}(2^9)$. Clearly, $\Gamma^{-1}(\mathbf{S}_2)$ is the inverse transformation. Of course, the simplest case of Γ is to interpret entries of $\mathbf{S}_1$ as polynomials in $\mathbb{GF}(2^9)$. However, Γ can also permute elements of $\mathbb{GF}(2^9)$.

Note that the first step of encryption does not use modular arithmetic, i.e. since the entries of key matrix $\mathbf{X}$ are in the range $[1, 256]$, whereas the entries of $\mathbf{M}'$ are in the range $[0, 255]$ their sum $\mathbf{S}_1$ contains entries in the range $[1, 511]$ and therefore can be interpreted as an element of $(1 + \mathbb{Z}_{511})^{4\times 4}$.

The decryption algorithm is as follows:

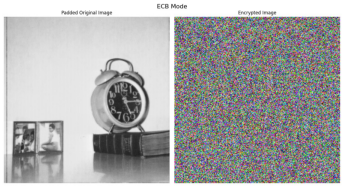
$$\begin{aligned}
&\mathbf{D}_1 \equiv (\mathbf{S} - \mathbf{X}) \bmod 2^9;\\
&\mathbf{D}_2 \equiv {}^{\mathbf{Y}^{-1}}\left(\Gamma(\mathbf{D}_1) \odot \mathbf{Z}^H\right)^{\mathbf{Y}^{-1}} \bmod \phi(x);\\
&\mathbf{M}' = \Gamma^{-1}(\mathbf{D}_2) - \mathbf{X},
\end{aligned}$$

where we see that at the last step, no modular arithmetic is used.

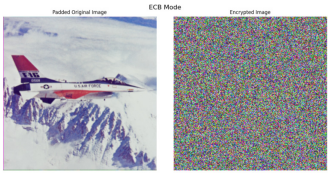
5.1. *ECB Mode*

Let us consider digital images from the previous section and encrypt them using $\mathbb{GF}(2^9)$ as a platform group. We implement the ECB mode.

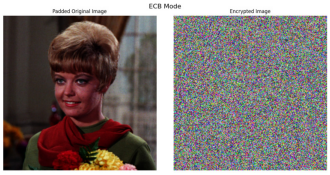
We can see from the encryption algorithm that the entries of the encrypted block $\mathbf{S}$ are integers in the range $[0, 511]$. Therefore, the maximum pixel intensity is 511. These



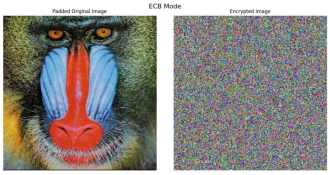
(a) 'Clock.png'



(b) 'Plane.png'

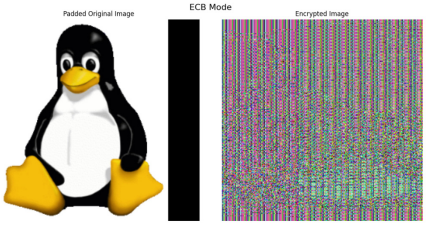


(c) 'Woman.png'

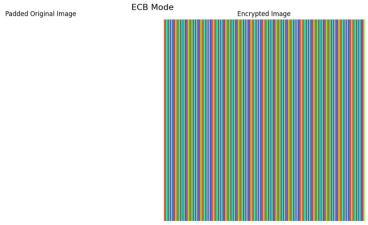


(d) 'Mandrill.png'

Fig. 13. ECB Mode of our cipher with $\mathbb{GF}(2^9)$ platform group applied to various image.



(a) Linux penguin



(b) Blank image

Fig. 14. ECB Mode of our cipher with $\mathbb{GF}(2^9)$ platform group applied to monochromatic image.

intensities have to be normalized for plotting and visualization by taking the integer part of s_{ij} divided by 2, i.e. we have:

$$s'_{ij} = \left\lfloor \frac{s_{ij}}{2} \right\rfloor.$$

We can then visualize the normalized block $\mathbf{S}'$.

Let us present several images encrypted using the ECB mode (see Fig. 13).

Now, let us consider the two monochromatic images from the previous section. The results are presented below in Figs. 14a–14b.

We can see that the low quality of the ECB mode for monochromatic images is still an issue regardless of the algebraic structure used. Therefore, we implement Arnold's cat mapping as previously for the Linux penguin image. The result is presented below in Fig. 15.

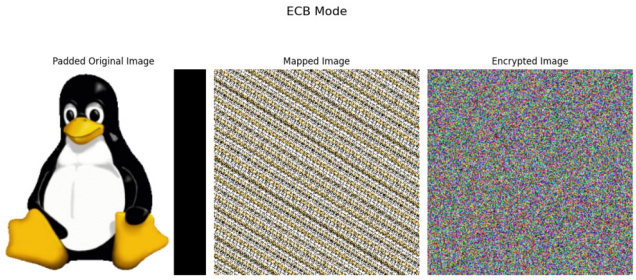


Fig. 15. Linux penguin.

Let us present the correlation matrix Σ_{avg} for the encrypted image:

$$\Sigma_{\text{avg}} = \begin{bmatrix} 0.009 & -0.005 & 0.000 \\ -0.004 & 0.001 & 0.005 \\ 0.005 & -0.002 & 0.005 \end{bmatrix}. \tag{12}$$

We can see from the entries of the correlation matrix Σ_{avg} that $|\sigma_{ij}| < 0.010$, which means there is slim to no correlation in the cipher.

The cipher entropy is calculated as:

$$\mathbf{H}_{RGB} = \begin{bmatrix} 8.992 \\ 8.994 \\ 8.995 \end{bmatrix}, \tag{13}$$

Since the theoretical maximum entropy for $\mathbb{GF}(2^9)$ is 9, the presented results suggest very high entropy, i.e. the encrypted image looks chaotic and surpasses the quality of the ECB mode of AES cipher for the same image.

5.2. *CBC Mode*

In the CBC mode, the initialization vector $\mathbf{C}_0$ is a block-size matrix whose entries are generated from a uniform distribution $\mathcal{U}(0, 255)$. Since each cipher consists of 9-bit elements, we append the cipher list by truncating the most significant bit, transforming each newly obtained $\mathbf{C}_{i-1}$ to an 8-bit entries matrix. Therefore, the encryption algorithm is as follows:

$$\begin{aligned} \mathbf{C}_{i1} &= \mathbf{X} + \left(\mathbf{M}'_i \oplus \left(\mathbf{C}_{i-1} \bmod 2^8\right)\right) \\ \mathbf{C}_{i2} &\equiv \left(\mathbf{Z} \odot {}^{\mathbf{Y}}\Gamma(\mathbf{C}_{i1})^{\mathbf{Y}}\right) \bmod \phi(x); \\ \mathbf{C}_i &\equiv \left(\Gamma^{-1}(\mathbf{C_{i2}}) + \mathbf{X}\right) \bmod 2^9, \end{aligned}$$

where $\oplus$ denotes the bit-wise XOR operation. Note that at step $\mathbf{C}_{i1}$, only the matrix $\mathbf{C}_{i-1}$ is reduced modulo 2^8. No modular arithmetic is used when adding the private key matrix $\mathbf{X}$.

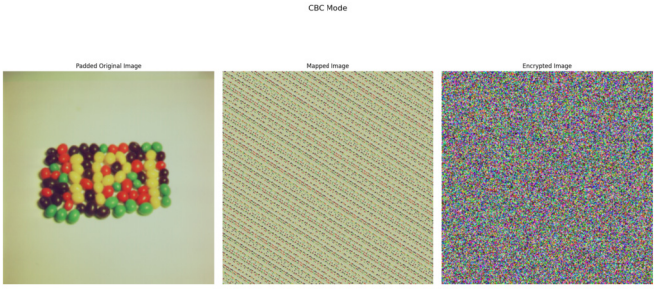


Fig. 16. CBC Mode of our cipher with $\mathbb{GF}(2^9)$ platform group applied to Skitlles.png image.

Decryption algorithm reverses these steps as follows:

$$\mathbf{D}_{i1} \equiv (\mathbf{C}_i - \mathbf{X}) \bmod 2^9;$$
$$\mathbf{D}_{i2} \equiv {}^{\mathbf{Y}^{-1}}\left(\Gamma(\mathbf{D}_{i1}) \odot \mathbf{Z}^H\right)^{\mathbf{Y}^{-1}} \bmod \phi(x);$$
$$\mathbf{D}_{i3} = \left(\Gamma^{-1}(\mathbf{D}_2) - \mathbf{X}\right);$$
$$\mathbf{M}' = \mathbf{D}_{i3} \oplus \left(\mathbf{C}_{i-1} \bmod 2^8\right).$$

Similarly to the encryption algorithm, we reduce only the matrix $\mathbf{C}_{i-1}$ modulo 2^8 when calculating $\mathbf{M}'$. The reduced matrix is then entry-wise XOR'ed with $\mathbf{D}_{i3}$.

Let us now consider the image 'Skittles.png'. We investigate the statistical results of the CBC mode of encryption for this image. The output of the CBC mode of our cipher is presented in Fig. 16.

As previously, let us present the values of the correlation matrix and the entropy vector:

$$\Sigma_{\text{avg}} = \begin{bmatrix} 0.008 & -0.005 & 0.000 \\ 0.006 & -0.003 & 0.001 \\ -0.004 & -0.006 & -0.001 \end{bmatrix}, \qquad \mathbf{H}_{RGB} = \begin{bmatrix} 8.995 \\ 8.994 \\ 8.995 \end{bmatrix}. \tag{14}$$

These results indicate very low correlations and high chaos among the pixel intensities in the cipher.

Moreover, let us plot the distribution of the unnormalized pixel intensities of the encrypted image in Fig. 17. We can see that the intensity probabilities for all colours fluctuate throughout the intensity interval, but the probability difference from the uniform distribution is less than 0.001 for each intensity. To further analyse the uniformity of the cipher, we perform the Chi-Square Test on each colour intensity set that is divided into 32 bins. We plot the frequency histograms for each colour in Fig. 18. In this case, p-values (up to 3 decimal places) of the Chi-Square test for red, green, and blue channels are: 0.063, 0.835, and 0.600 respectively. Therefore, with 95% probability, we do not reject the null hypothesis that the values are uniformly distributed. However, the results of the Chi-Square test might change drastically if given another private key, so the numerical results of the Chi-Square test only suggest that for this specifically randomly generated key, we cannot reject the null hypothesis with 95% probability.

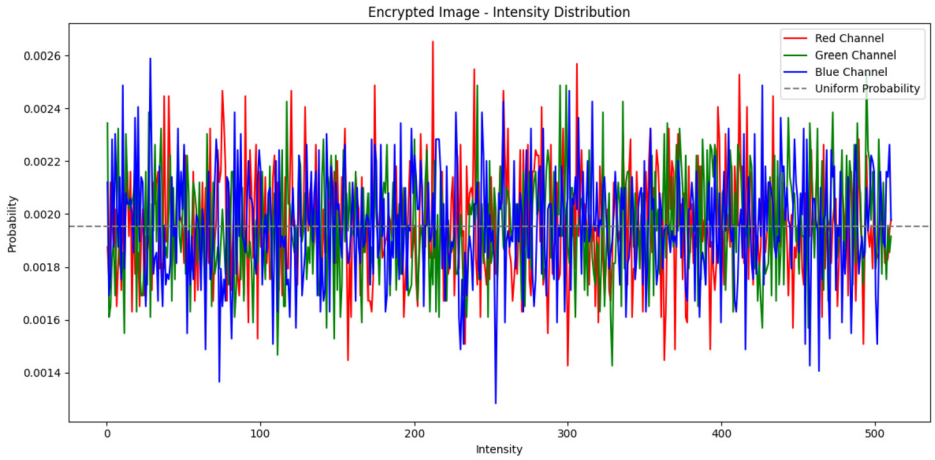


Fig. 17. Cipher Interpolated line chart of pixel intensities for digital image Skittles.png encrypted in CBC mode.

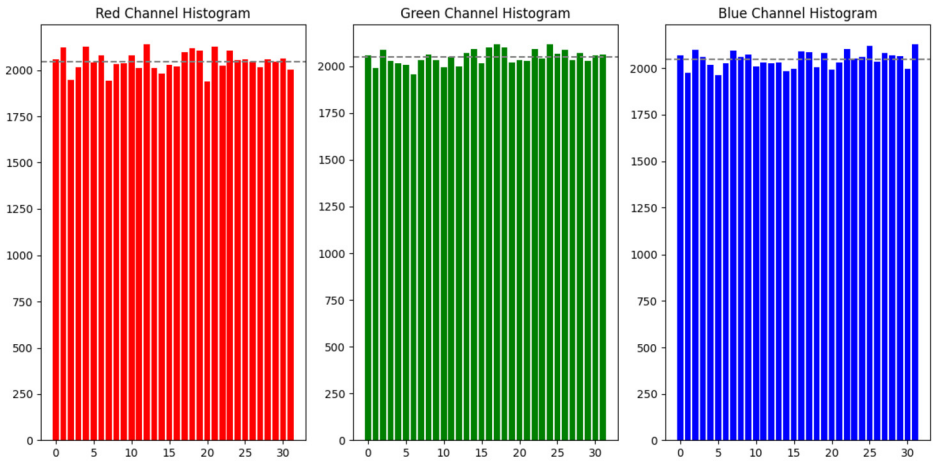


Fig. 18. Binned RGB intensity charts.

6. Statistical Characteristics of the MPF-Based Block Cipher

The number of changing pixel rates (NPCR) and the unified average changed intensity (UACI) are the two most common statistical characteristics used to evaluate the strength of image encryption ciphers with respect to differential attacks (Wu *et al.*, 2011). The results of the numerical analysis using these metrics need to be compared to the critical values that these statistics are theoretically expected to reach.

Strict avalanche criterion resides in different fields of cryptography, and it is generalized by the avalanche effect (Castro *et al.*, 2005). It measures the amount of non-linearity in substitution boxes (also known as s-boxes), and it is a key component in AES (Castro *et al.*, 2005).

Table 1
Average NPCR and UACI values.

Image	Resized image size	Average NPCR, %	Average UACI, %
Linux penguin	16×16	99.630	33.359
Linux penguin	32×32	99.609	33.379
Linux penguin	64×64	99.597	33.504
Skittles.png	16×16	99.576	33.495
Skittles.png	32×32	99.574	33.458
Skittles.png	64×64	99.592	33.471

Let us explore these characteristics for our cipher defined over the Galois field. We consider two images: 'Skittles.png' and the Linux penguin. The experiment is based on comparing two ciphertexts of 4×4 block size and is performed as follows:

- We settle on the initial block plaintext $\mathbf{M}_0$ and encrypt it as the ciphertext $\mathbf{C}_0$.
- For each bit in each of the plaintext $\mathbf{M}_0$ coordinates we change one bit to get $\mathbf{M}_i$, where i denotes the i-th changed bit. Hence, $\mathbf{M}_0$ and $\mathbf{M}_i$ differ by exactly one bit.
- We encrypt $\mathbf{M}_i$ to calculate ciphertext $\mathbf{C}_i$.
- Finally, we compare $\mathbf{C}_0$ and $\mathbf{C}_i$.
- After i runs through all possible values, we consider the next block.
- After collecting data from all of the blocks, we calculate the average statistics.

The first comparison is whether the specific coordinate pixel value changed or not – this difference is captured by the NPCR. The expected value for this statistic is given by the following expression:

$$\mathbb{E}(NPCR) = \frac{L-1}{L} \cdot 100\%, \tag{15}$$

where $L-1$ is the upper bound of the uniform distribution $\mathcal{U}(0, L-1)$. In our case, we have $L = 512$ and therefore $\mathbb{E}(NPCR) \approx 99.8\%$.

Also, we keep track of the absolute difference in ciphertexts' pixel intensities and use the UACI metric to measure it. The expected value is:

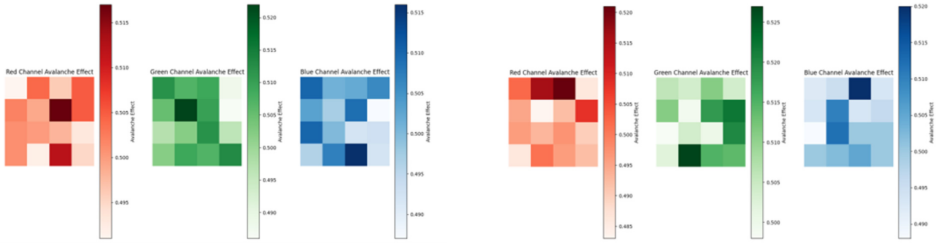
$$\mathbb{E}(UACI) = \frac{L+1}{3L} \cdot 100\%, \tag{16}$$

and therefore in our setup we have $\mathbb{E}(UACI) = 33.4\%$.

We resize both considered images to a 4×4 block size and perform 50 iterations, and calculate the average NPCR and UACI values for all iterations. During the experiment, we do not use Arnold's cat mapping, since otherwise we would not evaluate the encryption scheme appropriately. We output the values in Table 1.

We can observe that the obtained values of both characteristics are close to the expected ones. Therefore, we do not reject the hypothesis that the ciphertexts are generated independently of each other.

Now we calculate the avalanche effect of our cipher. Since we consider 4×4 blocks, each entry has 8 bits that can be changed in the initial image, and the block consists of



(a) 'Skittles.png' (b) Linux penguin

Fig. 19. Heatmaps for the avalanche criterion of our cipher with $\mathbb{GF}(2^9)$ platform group.

three channels, we perform exactly $4 \cdot 4 \cdot 8 \cdot 3 = 384$ iterations. For both images, we output the results as heatmaps for each channel and for each coordinate separately.

We can observe that for both images, the results are similar (see Fig. 19). The variance from the expected value of 0.5 is approximately 0.02, which shows an insignificant difference in value variation during the experiment, despite the coordinate or the channel. Therefore, from the avalanche effect perspective, the theoretically expected value coincided with the experimental result.

We can also observe that there is no clear pattern of higher or lower values. The results are relatively random, although the confidence interval, which includes the value of 0.5, is narrow.

7. Conclusion

In this paper, we have introduced improvements to our original block cipher, presented in Dindiene *et al.* (2022), e.g. we now use individual keys for each of the RGB colours, and we implemented Arnold's cat mapping to the initial digital image to improve the quality of encryption. Also, we implemented Galois Fields in our proposal, making it more convenient for practical applications.

We considered both versions of our block cipher in ECB and CBC modes of encryption and analysed the quality of encrypted images using statistical characteristics. The obtained results have shown that the implementation of Arnold's cat mapping plays an important part in the overall quality of encryption. For example, we have demonstrated in Fig. 9c and 9d that the considered plaintexts are statistically indistinguishable given the ciphertext, which is evidence for the semantic security of our proposal. We calculated the correlation matrices for both images. The maximum absolute correlations of 0.005 and 0.009, respectively, have shown the lack of statistically significant linear relations between the pixels, which is a desirable result for a secure cipher. Also, we saw that the entropy for both images is 8.131, which is close to the theoretical maximum. Therefore, the statistical results obtained for the CBC mode of our block cipher are indistinguishable from the AES.

Implementing Arnold's cat mapping proved especially useful for monochromatic images. We have shown in Fig. 7 that our proposal, together with Arnold's cat mapping,

surpasses the AES cipher in ECB mode. Moreover, the maximum absolute correlation of the ciphertext was 0.009, which is similar to the result obtained for the CBC mode of encryption. However, as can be seen from Fig. 14b, the ECB mode is still not recommended for our cipher, since Arnold's cat mapping cannot remove the dominance of one colour in this case. On the other hand, while ciphertexts in Figs. 4 and 14a are still low quality, they seem more chaotic than the ciphertext obtained from the ECB mode of AES (see Fig. 6a), even without additional scrambling.

We also considered the avalanche effect of our proposal. The obtained results have shown that we can expect an average of 99.6% value of NPCR and a 33.5% value of UACI statistical characteristics. These results are fairly close to the desired values of these characteristics. Furthermore, we can expect the avalanche effect in the range [0.48; 0.52], which is close to the ideal value of 0.5.

Note, however, that in our paper we have fixed the parameters of Arnold's cat mapping, as well as the number of its iterations. In the future, it may be interesting to explore the effect of scrambling with other parameter values.

Also, to shorten this paper, we have left out the performance analysis for the enhanced version of our cipher. Though from our previous work, there are indications that our proposal can be executed reasonably fast, it may be necessary to consider the influence of Arnold's cat mapping and parallelization on the performance of our cipher.

References

Asif, M., Asamoah, J.K.K., Hazzazi, M.M., Alharbi, A.R., Ashraf, M.U., Alghamdi, A.M. (2022). A novel image encryption technique based on cyclic codes over galois field. *Computational Intelligence and Neuroscience*, 2022, 1912603. https://doi.org/10.1155/2022/1912603.

Broumandnia, A. (2020). Image encryption algorithm based on the finite fields in chaotic maps. *Journal of Information Security and Applications*, 54(4), 102553.

Castro, J.C.H., Sierra, J.M., Seznec, A., Izquierdo, A., Ribagorda, A. (2005). The strict avalanche criterion randomness test. *Mathematics and Computers in Simulation*, 68(1), 1–7.

Chillali, S., Oughdir, L. (2023). Image encryption algorithm based on elliptic curves. *WSEAS Transactions on Signal Processing*, 19, 184–191.

Daoui, A., Yamni, M., Chelloug, S.A., Wani, M.A., El-Latif, A.A.A. (2023). Efficient image encryption scheme using novel 1D multiparametric dynamical tent map and parallel computing. *Mathematics*, 11(7), 1589.

Detomi, E., Morigi, M., Shumyatsky, P. (2025). Commuting probability for the Sylow subgroups of a profinite group. *Mathematische Zeitschrift*, 309(3). https://doi.org/10.1007/s00209-025-03686-x.

Dindiene, L., Mihalkovich, A., Luksys, K., Sakalauskas, E. (2022). Matrix power function based block cipher operating in CBC mode. *Mathematics*, 10(12), 2123.

Dinu, A., Frunzete, M. (2025). Image encryption using chaotic maps: development, application, and analysis. *Mathematics*, 13(16), 2588.

Dworkin, M.J., Barker, E.B., Nechvatal, J.R., Foti, J., Bassham, L.E., Roback, E., Jr, J.F.D. (2001). Advanced Encryption Standard (AES). Last Modified: 2021-03-01T01:03-05:00. https://www.nist.gov/publications/advanced-encryption-standard-aes.

Elashry, I.F., Faragallah, O.S., Abbas, A.M., El-Rabaie, S., El-Samie, F.E.A. (2012). A new method for encrypting images with few details using Rijndael and RC6 block ciphers in the electronic code book mode. *Information Security Journal*, 21(4), 193–205. https://doi.org/10.1080/19393555.2011.654319.

Hernández-Díaz, E., Pérez-Meana, H., Silva-García, V., Flores-Carapia, R. (2021). JPEG images encryption scheme using elliptic curves and a new S-Box generated by chaos. *Electronics*, 10(4), 413.

Levinskas, M., Mihalkovich, A. (2021). Avalanche effect and bit independence criterion of perfectly secure Shannon cipher based on matrix power. *Mathematical Models in Engineering*, 7(3), 50–53.

Li, C. (2006). Cryptanalyses of some multimedia encryption schemes. *Cryptology ePrint Archive*.

Li, Y., Song, C., Dong, J., Zheng, H. (2023). An efficient encryption method for smart grid data based on improved CBC mode. *Journal of King Saud University – Computer and information sciences*, 35(9), 101744. https://doi.org/10.1016/j.jksuci.2023.101744.

Mihalkovich, A., Levinskas, M., Makauskas, P. (2022a). MPF based symmetric cipher performance comparison to AES and TDES. *Mathematical Models in Engineering*, 8(2), 15–25.

Mihalkovich, A., Levinskas, M., Sakalauskas, E. (2022b). Counter mode of the Shannon block cipher based on MPF defined over a non-commuting group. *Mathematics*, 10(18), 3363.

Mihalkovich, A., Levinskas, M., Dindiene, L., Sakalauskas, E. (2022c). CBC mode of MPF based Shannon cipher defined over a non-commuting platform group. *Informatica*, 33(4), 833–856. https://doi.org/10.15388/22-INFOR499.

Nagaraj, S., Raju, G., Rao, K.K. (2015). Image encryption using elliptic curve cryptograhy and matrix. *Procedia Computer Science*, 48, 276–281.

Pardede, A.M.H., Abdullah, D., Kusuma, B.S., Kurniawan, C., Suwarni, Iskandar, A., Putri, L.D., Erliana, C.I., Irwansyah, D., Saleh, A.A., Sriadhi, S., Hartono, H. (2018). Retraction: digital image security application with Arnold Cat Map (ACM) *Journal of Physics. Conference Series*, 1114(1), 12152. https://doi.org/10.1088/1742-6596/1114/1/012152.

Sakalauskas, E., Mihalkovich, A. (2014). New asymmetric cipher of non-commuting cryptography class based on matrix power function. *Informatica*, 25(2), 283–298.

Sakalauskas, E., Listopadskis, N., Tvarijonas, P. (2008). Key agreement protocol (KAP) based on matrix power function. *Advanced Studies in Software and Knowledge Engineering*, 2 (4), 92–96.

Singh, L.D., Lahoty, A., Devi, C., Dey, D., Saikai, P., Devi, K.S., Singh, K.M. (2024). Image encryption using dynamic S-boxes generated using elliptic curve points and chaotic system. *Journal of Information Security and Applications*, 83, 103793.

Wu, Y., Noonan, J.P., Agaian, S., (2011). NPCR and UACI randomness tests for image encryption. *Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT)*, 1(2), 31–38.

Zhang, B., Liu, L. (2023). Chaos-based image encryption: review, application, and challenges. *Mathematics*, 11(11), 2585.

Zia, U., McCartney, M., Scotney, B., Martinez, J., AbuTair, M., Memon, J., Sajjad, A. (2022). Survey on image encryption techniques using chaotic maps in spatial, transform and spatiotemporal domains. *International Journal of Information Security*, 21(4), 917–935.

J. Zitkevicius obtained a bachelor's degree in applied mathematics at Kaunas University of Technology in 2025. He is a member of Identification and Cryptography Research Group and performs investigations in symmetric and asymmetric cryptography.

A. Mihalkovich obtained his PhD in 2015 and is currently an associate professor at the Department of Applied Mathematics at Kaunas University of Technology. He is a member of Identification and Cryptography Research Group and performs various investigations in symmetric and asymmetric cryptography.

E. Sakalauskas is currently a full professor at the Department of Applied Mathematics at Kaunas University of Technology. He is the head of Identification and Cryptography Research Group and performs various investigations in symmetric and asymmetric cryptography.