INFORMATICA, 2022, Vol. 33, No. 4, 833–856

DOI: https://doi.org/10.15388/22-INFOR499

CBC Mode of MPF Based Shannon Cipher Defined Over a Non-Commuting Platform Group

Aleksejus MIHALKOVICH*, Matas LEVINSKAS, Lina DINDIENE, Eligijus SAKALAUSKAS

Kaunas University of Technology, Studentu str. 50-324, Lithuania
e-mail: aleksejus.michalkovic@ktu.lt, matas.levinskas@ktu.edu, lina.dindiene@ktu.lt, eligijus.sakalauskas@ktu.lt

Received: May 2022; accepted: November 2022

Abstract. Commonly modern symmetric encryption schemes (e.g. AES) use rather simple actions repeated many times by defining several rounds to calculate the ciphertext. An idea we previously offered was to trade these multiple repeats for one non-linear operation. Recently we proposed a perfectly secure symmetric encryption scheme based on the matrix power function (MPF). However, the platform group we used was commuting. In this paper, we use a non-commuting group whose cardinality is a power of 2 as a platform for MPF. Due to the convenient cardinality value, our scheme is more suitable for practical implementation. Moreover, due to the non-commuting nature of the platform group, some "natural" constraints on the power matrices arise. We think that this fact complicates the cryptanalysis of our proposal. We demonstrate that the newly defined symmetric cipher possesses are perfectly secure as they were previously done for the commuting platform group. Furthermore, we show that the same secret key can be used multiple times to encrypt several plaintexts without loss of security. Relying on the proven properties we construct the cipher block chaining mode of the initial cipher and show that it can withstand an adaptive chosen plaintext attack.

Key words: symmetric cryptography, perfect secrecy, non-commuting cryptography, matrix power function.

1. Introduction

1.1. *Motivation*

Symmetric cryptography came a long way from ancient times. One of the fundamental works in this area was presented in Shannon (1949). There the author introduced a concept nowadays known as the Shannon cipher given by a triplet (Gen(), Enc(), Dec()), where Gen() is a key generation function, Enc() and Dec() are encryption and decryption functions, respectively, as defined in Katz and Lindell (2007). Assuming μ is the plaintext to be encrypted, the major requirement of a symmetric encryption scheme is the following:

$$\mathrm{Dec}(k, \mathrm{Enc}(k, \mu)) = \mu, \tag{1}$$

*Corresponding author.

i.e. decryption function correctly restores the message μ using the same key k. Any properly working symmetric cipher must satisfy this requirement. Proving the correctness of any symmetric cipher relies on verifying identity (1).

In the realm of modern symmetric ciphers, the most secure ones possess an essential property of perfect secrecy – a concept initially defined by Shannon himself. One of the most intuitive definitions can be found in various textbooks like Katz and Lindell (2007) or Boneh and Shoup (2020) and states that a symmetric cipher is perfectly secure if the ciphertext c is statistically independent of the encrypted plaintext μ, i.e.

$$\Pr(c = c_0 \mid \mu = \mu_0) = \Pr(c = c_0), \tag{2}$$

where Pr() denotes the probability of a random event and c_0 and μ_0 are fixed ciphertext and plaintext respectively. We use this definition in Section 4 to show that our cipher satisfies condition (2).

The perfect secrecy property of the one-time pad (OTP) technique was proven by Shannon. To our knowledge, up to our previous works, OTP together with its various modifications remained the only technique with this property. This comes from the fact that perfectly secure ciphers require keys of the same size as the plaintext to be encrypted. Hence, despite achieving this highly desirable property, OTP is mainly viewed as a theoretical concept and is rarely used in practice. Moreover, the OTP falls flat due to its inability to reuse the secret key and becomes an easy prey for active attackers. Interestingly enough, the latter flaw is also the main issue for constructing various encryption modes based on this technique.

Therefore, widely popular symmetric ciphers (e.g. AES) are usually constructed by repeating several rather simple operations multiple times. The more rounds are used, the higher security is achieved. These ciphers can also be adapted for practical implementation via various encryption modes.

Our goal is to show that the perfectly secure cipher can be adaptable for practical implementation. In other words, by using a highly non-linear matrix mapping as opposed to multiple rounds of encryption we can achieve a high-security level while also avoiding the main issue of the OTP.

1.2. *Related Work*

Recently our research group published a paper (Sakalauskas *et al.*, 2020b), where we introduced a symmetric encryption scheme based on a special case of the so-called MPF mapping. In Sakalauskas and Luksys (2012), authors formally defined MPF as a mapping $Mat_n(\mathbb{R}) \times Mat_n(\mathbb{S}) \times Mat_n(\mathbb{R}) \mapsto Mat_n(\mathbb{S})$, where $Mat_n(\cdot)$ denotes a set of square $n \times n$ matrices with entries taken from the specified algebraic structure: a platform semigroup $\mathbb{S}$ or a finite ring of integers $\mathbb{R}$ with cardinality determined by the properties of $\mathbb{S}$. Let us assume, that matrices $\mathbf{X}, \mathbf{Y} \in Mat_n(\mathbb{R})$ and $\mathbf{W}, \mathbf{E} \in Mat_n(\mathbb{S})$. Then we denote

$$^{\mathbf{X}}\mathbf{W}^{\mathbf{Y}} = \mathbf{E}, \tag{3}$$

where each entry of the result matrix $\mathbf{E}$ is computed as follows:

$$e_{ij} = \prod_{k=1}^{n} \prod_{l=1}^{n} w_{kl}^{x_{ik} y_{lj}}. \tag{4}$$

In the paper (Sakalauskas *et al.*, 2020b) we focused on a Sylow group $\mathbb{G}_3$ found in a multiplicative group $\mathbb{Z}_7$. Let us recall that the semigroup $\mathbb{S}$ contains a Sylow group of cardinality p^k if k is the largest power of p dividing the multiplicative order of $\mathbb{S}$, i.e. a number denoted as ord($\mathbb{S}$) such that for every element $s \in \mathbb{S}$ we have $s^{1+\text{ord}(\mathbb{S})} = s$ (Sylow, 1872). We proved in Sakalauskas *et al.* (2020b) that the proposed scheme is perfectly secure.

Our recent research continues the study of MPF applications for symmetric cipher construction but uses a non-commuting platform group. Previously, we published several papers where we proposed new protocols based on MPF defined over non-commuting platform groups (Sakalauskas *et al.*, 2020a; Mihalkovich *et al.*, 2020). In those papers, we proved that the proposed asymmetric cryptographic primitives rely on NP-complete problems (Sakalauskas and Mihalkovich, 2018; Mihalkovich *et al.*, 2020). We used singular matrices to our advantage and showed that non-commuting platform groups and singular matrices contribute to the overall security of the proposed protocols.

In our previous paper (Mihalkovich *et al.*, 2022), we considered the performance of the cipher block chaining (CBC) mode based on MPF mapping. Moreover, we evaluated the computational costs of AES and TDES protocols operating in the CBC mode based on the notion of clock cycles. To achieve a balance between the memory requirements, performance, and statistical properties of our scheme, discussed previously in Levinskas and Mihalkovich (2021), we fixed the main parameters of our cipher at $m = 4$, $p = 4079$ and $q = 2039$. Our results have shown that MPF-based CBC mode outperforms AES-128 by 1.5 times and TDES by roughly 47 times.

Notably, our cipher has another interesting property that was not considered previously in Mihalkovich *et al.* (2022). Since our cipher is based on matrix operations we can achieve a significant boost of performance speed by implementing parallelization of calculations up to m^2 processors during an encryption process of each block. We think that this fact benefits our proposal since other algorithms considered in our previous paper do not have this property.

In this paper, we introduce the CBC mode for our MPF-based cipher and prove its security. We leave the performance evaluation and comparison to other ciphers for our future work. Based on the findings presented in Mihalkovich *et al.* (2022), we expect to achieve similar results for the to-be-presented CBC mode built on the non-commuting group.

1.3. *Our Contributions*

Obviously, singular matrices cannot be used as symmetric keys since the initial message must be restored by applying the same key which is impossible if the inverse matrix does not exist. Hence, to implement the non-commuting platform groups in symmetric encryption we have to define different templates in such a way that power matrices in (3) would be

invertible. Furthermore, as opposed to asymmetric encryption presented in Mihalkovich *et al.* (2020), we also have to make the base matrix **W** as flexible as possible. In other words, we cannot simply fix a template for the base matrix **W** since we want to be able to work with any kind of message without having to adapt them to fit a certain requirement. Therefore, we limit ourselves to defining a template for the power matrix **Y**, thus keeping the restrictions to a minimum.

As mentioned above, we also consider our cipher from the point of view of practical implementation. One obvious drawback of any perfectly secure block cipher is the fact that the encryption key has to be at least as long as the encrypted plaintext. To overcome this obstacle we define the cipher block chaining mode on the basis of our proposal. To prove its resistance against adaptive chosen plaintext attack we define a security game and show that the probability of a win is negligible.

In this paper, we consider a general form of one of the previously explored non-commuting groups, namely the group $\mathtt{M}_{16}$ (Mihalkovich, 2018; Mihalkovich *et al.*, 2020). We define this general form in the next section and present some important facts useful for our goals. These involve the explicit formulas of basic operations and the properties of MPF. In Section 3, we present our main idea – a Shannon cipher based on MPF defined over a non-commuting group. Later in Section 4 we prove the perfect secrecy of our proposal. Moreover, in Section 6 we define the CBC mode of our cipher and consider the security of this scheme in Section 7. As usual, in Section 8 we present our conclusions.

2. Mathematical Background

Let us define two generators a and b which do not commute, i.e. $ab \neq ba$. Furthermore, we define the following relations:

$$\begin{aligned} &R_1 : a^{2^{t-1}} = e; \\ &R_2 : b^2 = e; \\ &R_3 : bab^{-1} = a^{2^{t-2}+1}, \end{aligned} \tag{5}$$

where e is the identity element. Using these relations we can form words of the types $a^\alpha b^\beta$ or $b^\beta a^\alpha$, where $\alpha \in \{0, 1, \ldots, 2^{t-1} - 1\}$ and $\beta \in \{0, 1\}$. Moreover, the set of these words defines the following group:

$$\mathtt{M}_{2^t} = \langle a, b \mid R_1, R_2, R_3 \rangle . \tag{6}$$

Remark 1. We use the notation $\mathtt{M}_{2^t}$ to better distinguish this group from the plaintext matrix **M** and the plaintext space $\mathbb{M}$. Furthermore, we denote the plaintext bit string by μ and entries of the matrix **M** by m_{ij}.

Evidently, the identity element can be written as $e = a^0 b^0 = b^0 a^0$. Furthermore, based on the defined relations R_1 and R_2, we can see that all the powers of the generators

can be reduced modulo 2^{t-1} for generator a and modulo 2 for generator b. Using relations R_1, R_2, R_3, it is possible to derive that each element of the group $\mathtt{M}_{2^t}$ can be represented in the form $b^\beta a^\alpha$. Onwards we call this representation a normal form of the element and use it throughout this paper. Obviously, if $\beta = 0$, we have:

$$a^\alpha b^0 = b^0 a^\alpha.$$

The general representation if $\beta = 1$ is as follows:

$$a^\alpha b = \begin{cases} ba^\alpha, & \text{if } \alpha \text{ is even;} \\ ba^{\alpha+2^{t-2}}, & \text{if } \alpha = 0 \text{ is odd.} \end{cases} \tag{7}$$

The proof of this fact in the special case of $\mathtt{M}_{16}$ was presented in Mihalkovich (2018). Since the idea of the proof remains the same, we omit it to shorten the paper. For this reason, the cardinality of the group $\mathtt{M}_{2^t}$ is 2^t, i.e. the parameter t defines the size of the considered group.

Here we defined the group $\mathtt{M}_{2^t}$ in its most general form. However, special cases of such groups were previously explored by researchers in group theory. For example, $\mathtt{M}_{16}$ is mentioned in Grundman and Smith (1996), where the authors were discussing the groups of cardinality 16, which are not isomorphic to any other group. A total of seven such groups of size 16 were found. In 2010, authors presented a continuation of their research in Grundman and Smith (2010b). There they considered non-abelian groups of size 32 and one of the mentioned groups was $\mathtt{M}_{32}$. Similar non-abelian groups were also explored in Michailov (2007) and Grundman and Smith (2010a).

Expanding the idea to greater sizes grants us opportunities to construct symmetric encryption using $\mathtt{M}_{2^t}$ as a platform group more flexibly. Conveniently, we can now manipulate two parameters, i.e. square matrix size $\mathbf{M}$ and platform group size determined by t. Special cases discussed above are obtained when $t = 4$ or $t = 5$. As mentioned previously, none of these groups are isomorphic to any other groups of the appropriate cardinality.

Let us now present formulas for the basic operations in $\mathtt{M}_{2^t}$. All of the formulas given below are verified using relations R_1, R_2, R_3:

- Multiplication of two elements $w_1, w_2 \in \mathtt{M}_{2^t}$

$$w_1 \cdot w_2 = \begin{cases} b^{\beta_1+\beta_2} a^{\alpha_1+\alpha_2}, & \text{if } \alpha_1 \text{ is even;} \\ b^{\beta_1} a^{\alpha_1+\alpha_2}, & \text{if } \alpha_1 \text{ is odd and } \beta_2 = 0; \\ b^{\beta_1+1} a^{\alpha_1+\alpha_2+2^{t-2}}, & \text{if } \alpha_1 \text{ is odd and } \beta_2 = 1; \end{cases} \tag{8}$$

- Raising of an element $w \in \mathtt{M}_{2^t}$ to a power $n \in \mathbb{Z}_{2^{t-1}}$:

$$w^n = \begin{cases} a^{\alpha n}, & \text{if } \beta = 0; \\ b^n a^{\alpha n}, & \text{if } \beta = 1 \text{ and } \alpha \text{ is even;} \\ b^n a^{\alpha n + 2^{t-2}\left[\frac{n}{2}\right]}, & \text{if } \beta = 1 \text{ and } \alpha \text{ is odd,} \end{cases} \tag{9}$$

where notation $[\frac{n}{2}]$ stands for the integer part of $\frac{n}{2}$.

- Calculating the inverse of the element $w \in \mathtt{M}_{2^t}$:

$$w^{-1} = \begin{cases} a^{-\alpha}, & \text{if } \beta = 0; \\ ba^{-\alpha}, & \text{if } \beta = 1 \text{ and } \alpha \text{ is even}; \\ ba^{2^{t-2}-\alpha}, & \text{if } \beta = 1 \text{ and } \alpha \text{ is odd}. \end{cases} \tag{10}$$

Explicit proofs of these formulas for a special case of $\mathtt{M}_{16}$ can be found in Mihalkovich (2018). Since the idea of these proofs stays the same, we omit them.

We also introduce an extra notation:

$$\mathbf{W} = b^{\mathbf{B}} a^{\mathbf{A}}. \tag{11}$$

This means that each entry w_{ij} of the matrix $\mathbf{W}$ is represented in the normal form

$$w_{ij} = b^{\beta_{ij}} a^{\alpha_{ij}}, \tag{12}$$

where β_{ij} and α_{ij} are entries of matrices $\mathbf{B}$ and $\mathbf{A}$, respectively.

Interestingly enough, by using the group $\mathtt{M}_{2^t}$ as a platform for MPF we also inflict some "natural" restrictions on the set of symmetric keys. This means that any tuple of matrices, which is outside of the specified domain, cannot be used, since the decryption of the ciphertext results in a scrambled mess. Specifically, if $\mathtt{M}_{2^t}$ is used as a platform group, then in general we have:

$$\begin{aligned} &(\mathbf{W}^{\mathbf{Y}})^{\mathbf{Y}^{-1}} \neq \mathbf{W}; \\ &{}^{\mathbf{Y}^{-1}}({}^{\mathbf{Y}}\mathbf{W}) \neq \mathbf{W}; \\ &({}^{\mathbf{Y}}\mathbf{W})^{\mathbf{Y}} \neq {}^{\mathbf{Y}}(\mathbf{W}^{\mathbf{Y}}). \end{aligned} \tag{13}$$

Despite these additional complexities, it is possible to construct a working symmetric encryption protocol. However, we think that these extra complexities may be beneficial for the overall security of our proposal. Similar to the previously published key exchange in Mihalkovich *et al.* (2020), we define a template for power matrix $\mathbf{Y}$, which can be used to achieve correct decryption. Then, due to inequalities (13), anything which disobeys the chosen template makes the decryption incorrect.

Keeping in mind the essence of symmetric encryption, we have chosen to pick power matrices from a subset of permutation matrices modulo 2, i.e. every square power matrix of size n contains exactly n odd entries whereas the rest of the entries are even. In this special case inequalities, (13) turn to equalities regardless of the choice of $\mathbf{W}$. In the next section, we propose Shannon symmetric encryption protocol with this restriction on the power matrices.

Considering security of our protocol we often refer to the following two mappings $\phi : \mathtt{M}_{2^t} \mapsto \mathbb{Z}_2$ and $\psi : \mathtt{M}_{2^t} \mapsto \mathbb{Z}_{2^{t-1}}$ defined below:

$$\phi(b^{\beta}a^{\alpha}) = \beta; \tag{14}$$

$$\psi(b^{\beta}a^{\alpha}) = \alpha. \tag{15}$$

Moreover, we define the matrix analogs of these mappings by applying them to each entry of the matrix $\mathbf{W}$ of the form (11) entry-wise, i.e. we have:

$$\Phi(\mathbf{W}) = \mathbf{B}; \tag{16}$$

$$\Psi(\mathbf{W}) = \mathbf{A}. \tag{17}$$

These mappings will prove helpful to us when showing the validity of the proposed protocol and establishing perfect secrecy property since they allow us to work with the powers of the specific generator.

3. The Proposed Shannon Symmetric Encryption Protocol

Before executing the proposed scheme the size of the group $\mathtt{M}_{2^t}$, defined by t, the size of square matrices n and the shifting parameter κ, defined below in (18), are published online.

3.1. *Key Generation Procedure*

The key generation procedure consists of the following steps:

1. Generate a binary matrix $\mathbf{\Delta}$;
2. Generate matrix $\mathbf{X}$ with random uniformly selected entries from $\mathbb{Z}_{2^{t-1}}$;
3. Generate a temporary matrix $\mathbf{Y}'$ with random uniformly selected entries from $\mathbb{Z}_{2^{t-2}}$;
4. Choose a permutation matrix $\mathbf{P}$ uniformly from the set of permutation matrices $\mathbb{P}_n \subset Mat_n(\mathbb{Z}_2)$ of size $n!$;
5. Define $\mathbf{Y} = 2\mathbf{Y}' + \mathbf{P}$. Calculate $\mathbf{Y}^{-1}$ using the Gauss-Jordan algorithm.

The result of this procedure is a symmetric key $(\mathbf{X}, \mathbf{Y}, \mathbf{\Delta})$. Note that each time the matrix is generated at Steps 1–3 of the presented process no additional restrictions are applied. Also, since $\mathbf{P} = \mathbf{Y} \bmod 2$ is a permutation matrix, the last step of the presented algorithm is always successful, i.e. $\mathbf{Y}$ is invertible. Hence, all the steps of this procedure are executed exactly once since none of them can fail. We also see that due to the definition of matrix $\mathbf{Y}$ both even and odd entries of $\mathbf{Y}$ are distributed uniformly in the subsets of even and odd elements of $\mathbb{Z}_{2^{t-1}}$ respectively.

3.2. *Encryption Function*

Let us assume that a message needs to be encrypted using the generated symmetric key $\vec{\mathbf{K}} = (\mathbf{X}, \mathbf{Y}, \mathbf{\Delta})$. The encryption procedure is as follows:

1. The message is converted to a string of bits of size $t \cdot n^2$. If the message is shorter, then extra symbols are added at the end to achieve the appropriate length. Otherwise, the message is too long.
2. The obtained string of bits is transformed to the matrix format by splitting it into n^2 separate parts of length t each. The outcome of this step is a matrix which we denote by $\mathbf{M}$.
3. The obtained matrix is split into separate matrices $\mathbf{M}_a$ and $\mathbf{M}_b$, where the first bit of each entry of $\mathbf{M}$ gets transported to matrix $\mathbf{M}_b$, whereas the rest of bits are written to matrix $\mathbf{M}_a$, hence obtaining powers of generators b and a respectively.
4. The encryption algorithm is as follows:

$$\begin{aligned} \mathbf{C}_1 &= b^{\mathbf{M}_b+\mathbf{\Delta}} \odot a^{\mathbf{M}_a+\mathbf{X}}; \\ \mathbf{C}_2 &= {}^{\mathbf{Y}}\mathbf{C}_1^{\mathbf{Y}}; \\ \mathbf{C} &= \mathrm{Shift}_\kappa\big(\Phi(\mathbf{C}_2) \parallel \Psi(\mathbf{C}_2)\big) + (\mathbf{\Delta} \parallel \mathbf{X}), \end{aligned} \tag{18}$$

   where $\parallel$ denotes the concatenation of two matrices, Shift_κ is the entry-wise shifting by κ bits (e.g. to the right) operator and the addition is performed appropriate modulo, i.e. matrices $\mathbf{M}_b$ and $\mathbf{\Delta}$ are summed modulo 2, $\mathbf{M}_a$ and $\mathbf{X}$ – modulo 2^{t-1}, and at the last step addition is performed modulo 2^t. In all cases, we omit moduli of addition as the appropriate values are usually clear from the context.
5. The matrix $\mathbf{C}$ is converted into a string of bits by concatenating its entries in the following way:

$$c = c_{11} \parallel c_{12} \parallel \ldots \parallel c_{1n} \parallel c_{21} \parallel c_{22} \parallel \ldots \parallel c_{2n} \parallel c_{nn},$$

   where the first bit of each entry c_{ij} is reserved for the power of generator b and the rest of the bits denote the power of generator a. The string of bits c is the ciphertext of the initial message.

Due to the discussed steps, the encryption function is given by:

$$\mathrm{Enc}(\vec{\mathbf{K}}, \mathbf{M}) = \mathrm{Shift}_\kappa\big(\Phi\big({}^{\mathbf{Y}}(\mathbf{C}_1)^{\mathbf{Y}}\big)\big) \parallel \Psi\big({}^{\mathbf{Y}}\big((\mathbf{C}_1)^{\mathbf{Y}}\big)\big) + (\mathbf{\Delta} \parallel \mathbf{X}), \tag{19}$$

where $\mathbf{M} = \mathbf{M}_b \parallel \mathbf{M}_a$ is the original message represented in matrix form and $\mathbf{C}_1$ is defined as in (18).

3.3. *Decryption Function*

Upon receiving the ciphertext c the following procedure is performed to decrypt the encrypted message using symmetric key $\vec{\mathbf{K}} = (\mathbf{X}, \mathbf{Y}, \mathbf{\Delta})$.

1. The ciphertext c is transformed into matrix form $\mathbf{C}$ by splitting it into n^2 parts of length t.

2. The decryption algorithm is as follows:

$$\begin{aligned}
\mathbf{D}_1 &= \mathrm{Shift}_{t-\kappa}(\mathbf{C} - \mathbf{\Delta} \parallel \mathbf{X}),\\
\mathbf{D}_2 &= b^{\mathbf{D}_{1b}} a^{\mathbf{D}_{1a}},\\
\mathbf{D}_3 &= {}^{\mathbf{Y}^{-1}}\mathbf{D}_2^{\mathbf{Y}^{-1}},\\
\mathbf{D}_a &= \Psi(\mathbf{D}_3) - \mathbf{X},\\
\mathbf{D}_b &= \Phi(\mathbf{D}_3) - \mathbf{\Delta},
\end{aligned} \tag{20}$$

   where $\mathbf{D}_{1b}$ is a binary matrix obtained by splitting the first bits of $\mathbf{D}_1$ and $\mathbf{D}_{1a}$ consists of the leftover bits. Subtraction is to be treated as an inverse of addition in the encryption algorithm (18).
3. Matrices $\mathbf{D}_a$ and $\mathbf{D}_b$ are concatenated together entry-wise, thus producing matrix $\mathbf{D} = \mathbf{D}_b \parallel \mathbf{D}_a$.
4. The obtained matrix $\mathbf{D}$ undergoes the procedure of transformation to a string of bits by concatenating entries of the matrix in a specific way determined by one of the permutation vectors.
5. Junk symbols are removed, if any. The output of this step is the initial message.

Hence, we can define the decryption function as follows:

$$\mathrm{Dec}(\vec{\mathbf{K}}, \mathbf{C}) = \left(\Phi\left({}^{\mathbf{Y}^{-1}}(\mathbf{D}_2)^{\mathbf{Y}^{-1}}\right) - \mathbf{\Delta}\right) \parallel \left(\Psi\left({}^{\mathbf{Y}^{-1}}(\mathbf{D}_2)^{\mathbf{Y}^{-1}}\right) - \mathbf{X}\right), \tag{21}$$

where $\mathbf{C}$ is the received ciphertext represented in matrix form and $\mathbf{D}_2$ is defined as in (20).

3.4. *Proof of Correctness*

Looking at the presented encryption and decryption algorithms we see that $\mathbf{D}_2 = \mathbf{C}_2$ due to definitions of these matrices.

Let us consider an intermediate result $\mathbf{H} = {}^{\mathbf{Y}}\mathbf{C}_1$. Note that entries of matrix T are given by

$$h_{ij} = \prod_{k=1}^{n} c_{1kj}^{y_{ik}}. \tag{22}$$

An important restriction, which helps us to prove the validity of our protocol is the structure of the key matrix $\mathbf{Y}$. Obviously, due to $\mathbf{Y}$ being a permutation matrix modulo 2, it is invertible over $\mathbb{Z}_{2^{t-1}}$, since its determinant is always odd and hence relatively prime with 2^{t-1} for any value of t. Furthermore, since exactly one entry is odd in each row and each column of $\mathbf{Y}$, exactly one of the multipliers in the product (22) can contain generator b and hence this generator can never be cancelled unless raised to an even power. For the same reason, matrix $\mathbf{Y}^{-1}$, which has the same structure as $\mathbf{Y}$, successfully restores the initial matrix $\mathbf{C}_1$ when applied to $\mathbf{H}$, i.e. we have $\mathbf{C}_1 = \mathbf{H}^{\mathbf{Y}^{-1}}$.

We now consider the matrix $\mathbf{C}_2 = \mathbf{H}^{\mathbf{Y}} = {}^{\mathbf{Y}}\mathbf{C}_1^{\mathbf{Y}}$. As claimed in the latter paragraph, the generator b can never be cancelled unless raised to an even power. Hence, as previously, the matrix $\mathbf{Y}^{-1}$ successfully restores matrix $\mathbf{H}$, i.e. $\mathbf{H} = \mathbf{C}_2^{\mathbf{Y}^{-1}}$.

Combining these two observations we gain the following result:

$$\mathbf{D}_2 = {}^{\mathbf{Y}^{-1}}\mathbf{D}_1^{\mathbf{Y}^{-1}} = {}^{\mathbf{Y}^{-1}}\mathbf{C}_2^{\mathbf{Y}^{-1}} = {}^{\mathbf{Y}^{-1}}\left({}^{\mathbf{Y}}\mathbf{C}_1^{\mathbf{Y}}\right)^{\mathbf{Y}^{-1}} = \mathbf{C}_1.$$

Moreover, applying the mappings Φ and Ψ and subtracting appropriate matrices yields the matrix form $\mathbf{M}$ of the initial message, i.e. $\mathbf{D} = \mathbf{M}$.

The matrix $\mathbf{D}$ is now transformed to obtain a string of bits d by concatenating its entries as follows:

$$d = d_{11} \parallel d_{12} \parallel \ldots \parallel d_{1n} \parallel d_{21} \parallel d_{22} \parallel \ldots \parallel d_{2n} \parallel d_{nn}.$$

Relying on the discussed observations, we conclude that d is the bit string representing the initial message with junk symbols at the end. These can now be dropped to leave us with the initial message.

4. Proof of Perfect Secrecy

In this section, we consider the security of the proposed symmetric encryption. Our main goal is to show that our scheme possesses the property of perfect secrecy (2). To achieve this, we start by formulating and proving an important result involving the distribution of the MPF value entries.

Lemma 1. *Let us assume that the entries of the matrix* $\mathbf{W}$ *are random variables distributed uniformly in* $\mathtt{M}_{2^t}$ *and* $\mathbf{Y}$ *is a permutation matrix modulo* 2 *with entries uniformly distributed in the subsets of even and odd elements of* $\mathbb{Z}_{2^{t-1}}$*, respectively. Under these conditions the entries of the MPF exponent value* $\mathbf{E} = {}^{\mathbf{Y}}\mathbf{W}^{\mathbf{Y}}$ *are uniformly distributed in* $\mathtt{M}_{2^t}$.

Proof. Let us apply previously defined mappings $\Phi(\cdot)$ and $\Psi(\cdot)$ to the matrix $\mathbf{W}$ of the form (11). Recall that due to the statement of the lemma, entries of $\Phi(\mathbf{W}) = \mathbf{A}$ and $\Psi(\mathbf{W}) = \mathbf{A}$ are uniformly distributed in $\mathbb{Z}_2$ and $\mathbb{Z}_{2^{t-1}}$, respectively.

Since $\mathbf{Y}$ is a permutation matrix modulo 2, it mixes up the entries of $\mathbf{A}$ without changing them. For this reason, the entries of $\Phi(\mathbf{E})$ are uniformly distributed in $\mathbb{Z}_2$. Hence powers of generator b in matrix $\mathbf{E}$ are uniformly distributed in $\mathbb{Z}_2$.

We now consider the distribution of the powers of generator a in matrix $\mathbf{E}$. Keeping in mind the properties of permutation matrices, without loss of generality we onwards consider a special case of identity permutation, i.e. we assume that odd entries of the matrix $\mathbf{Y}$ are located on its main diagonal. We make a remark regarding the general case of the permutation matrix later in this proof.

Let us focus on the intermediate result $\mathbf{V} = {}^{\mathbf{Y}}\mathbf{W}$ and apply mapping $\Psi(\cdot)$ to this matrix. We can express every entry $\psi(v_{ij})$ as follows:

$$\psi(v_{ij}) = \sum_{k=1}^{n} \lambda_{kj} y_{ik} + \gamma_{ij}, \tag{23}$$

where $\gamma_{ij} \in \{0, 2^{t-2}\}$ can be one of two possible values depending on the number of times extra summand 2^{t-2} was added. We split the sum (23) into two parts based on the parity of entries of the matrix $\mathbf{Y}$. Then, for even values of $\mathbf{Y}$ we have:

$$s_{ij} = \sum_{k=1, k\neq i}^{n} \lambda_{kj} y_{ik} + \gamma_{ij}. \tag{24}$$

Due to the special structure of matrix $\mathbf{Y}$, we have a single summand of the sum (23) containing an odd entry y_{ii}. Hence, we denote

$$u_{ij} = \lambda_{ij} y_{ii}. \tag{25}$$

Note that if $\mathbf{Y}$ is a permutation matrix other than identity modulo 2, then the column index changes in the extracted summand. The omitted index in sum (24) changes as well. These are the only two differences in the general case.

Due to construction, all possible values of the sum (24) lie in the subset of even elements of $\mathbb{Z}_{2^{t-1}}$ and hence we claim that:

$$\sum_{r=0}^{2^{t-2}-1} \Pr(s_{ij} = 2r) = 1, \tag{26}$$

which is true, since these probabilities form a total probability. The exact values of these probabilities are irrelevant.

Considering the only odd summand, we can calculate the following probability:

$$\Pr(u_{ij} = u_0) = \Pr(\lambda_{ij} y_{ii} = u_0) = \Pr\left(\lambda_{ij} = u_0 y_{ii}^{-1}\right) = \frac{1}{2^{t-1}}, \tag{27}$$

where $u_0 \in \mathbb{Z}_{2^{t-1}}$ is fixed. This comes from the fact that $\gcd(y_{ii}, 2^{t-1}) = 1$ and hence y_{ii}^{-1} exists. Moreover, λ_{ij} is uniformly distributed due to the statement of the lemma.

Meshing facts (26) and (27) together we obtain the following result:

$$\begin{aligned} \Pr(\psi(v_{ij}) = z_0) &= \Pr(s_{ij} + u_{ij} = z_0) \\ &= \Pr(u_{ij} = z_0 - 2r) \Pr(s_{ij} = 2r) = \frac{1}{2^{t-1}} \sum_{r=0}^{2^{t-2}-1} \Pr(s_{ij} = 2r) = \frac{1}{2^{t-1}}. \end{aligned} \tag{28}$$

This result means that powers of generator a in an intermediate matrix $\mathbf{V}$ are distributed uniformly in $\mathbb{Z}_{2^{t-1}}$. Note also that since the term γ_{ij} does not play a major part in this

calculation, distributions of power of both generators are independent of each other, i.e. powers of generator b do not in any way affect the distribution of powers of generator a.

Similar calculations of probabilities can be performed for the powers of generator a in the matrix $\mathbf{V}^{\mathbf{Y}} = {}^{\mathbf{Y}}\mathbf{W}^{\mathbf{Y}} = \mathbf{E}$. Relying on the uniform distribution of entries of the matrix $\mathbf{V}$ and properties of the matrix $\mathbf{Y}$ we conclude that powers of generator a in matrix $\mathbf{E}$ are distributed uniformly.

Lastly, since the powers of both generators in matrix $\mathbf{E}$ are distributed uniformly and are independent of each other, the lemma is valid. □

Corollary 1. *The probability* $\Pr(\mathbf{E} = \mathbf{E}_0)$*, where* $\mathbf{E}_0$ *is a fixed matrix defined over* $\mathtt{M}_{2^t}$*, equals*:

$$\Pr(\mathbf{E} = \mathbf{E}_0) = \frac{1}{2^{n^2 t}}. \tag{29}$$

The proved lemma shows that we have obtained evidence of perfect secrecy property for our protocol. We establish this fact by proving the following theorem:

Theorem 1. *Let* $\vec{\mathbf{K}} = (\mathbf{X}, \mathbf{Y}, \mathbf{\Delta})$ *be a random key uniformly chosen from the set of keys* $\mathbb{K}$ *and let* $\mathbf{M}$ *be a random matrix chosen from the set of messages* $\mathbb{M}$ *in an arbitrary way. Assume also that probability distributions of* $\vec{\mathbf{K}}$ *and* $\mathbf{M}$ *are independent and fully determine the distribution of the matrix* $\mathbf{C}$ *in the set of cipher value matrices* $\mathbb{C}$ *together with the encryption algorithm* $\mathrm{Enc}(\cdot)$*. Under these assumptions, the proposed Shannon cipher in* (18) *based on MPF is perfectly secure.*

Proof. Let us consider encryption algorithm (18). Firstly, we turn our attention to matrix $\mathbf{C}_1$ and focus on the powers of generator a. Denoting $\mathbf{M}_a + \mathbf{X} = \mathbf{U}$ we rewrite each entry of matrix U in the following form:

$$u_{ij} = x_{ij} + m_{aij}, \quad i, j \in \{1, \ldots, m\}. \tag{30}$$

Due to the statement of the theorem, entries x_{ij} are chosen at random and are uniformly distributed in $\mathbb{Z}_{2^{t-1}}$, whereas entries m_{aij} are random arbitrary distributed values in $\mathbb{Z}_{2^{t-1}}$. For any fixed matrix $\mathbf{U}_0$ with entries $u_{0ij} \in \mathbb{Z}_{2^{t-1}}$, we have

$$\begin{aligned}\Pr(u_{ij} = u_{0ij}) &= \Pr(x_{ij} = u_{0ij} - m_{aij}) = \\ &= \frac{1}{2^{t-1}} \sum_{m_{0ij} \in \mathbb{Z}_{2^{t-1}}} \Pr(m_{aij} = m_{0ij}) = \frac{1}{2^{t-1}},\end{aligned} \tag{31}$$

where m_{0ij} are fixed elements of $\mathbb{Z}_{2^{t-1}}$.

We now calculate the conditional probabilities of the entries of matrix $\mathbf{U}$:

$$\Pr(u_{ij} = u_{0ij} \mid m_{aij} = m_{0ij}) = \Pr(x_{ij} = u_{0ij} - m_{0ij}) = \frac{1}{2^{t-1}}, \tag{32}$$

since the entries x_{ij} and m_{aij} are independent, and the difference $u_{0ij} - m_{0ij} \in \mathbb{Z}_{2^{t-1}}$.

Another important property of matrix $\mathbf{U}$ is the independence of its entries. Since all $x_{ij}, i, j = 1, \ldots, m$, are independent, for all $u_{0ij} \in \mathbb{Z}_{2^{t-1}}$ we have:

$$\begin{aligned}
&\Pr\Big(\bigcap_{i,j=1}^{n}\{u_{ij} = u_{0ij}\}\Big) = \Pr\Big(\bigcap_{i,j=1}^{n}\{x_{ij} + m_{aij} = u_{0ij}\}\Big)\\
&= \sum_{m\in\mathbb{Z}_{2^{t-1}}} \Pr\Big(\bigcap_{i,j=1}^{n}\{x_{ij} = u_{0ij} - m_{0ij}\}, \bigcap_{i,j=1}^{n}\{m_{aij} = m_{0ij}\}\Big)\\
&= \frac{1}{2^{n^2(t-1)}} \sum_{m_{0ij}\in\mathbb{Z}_{2^{t-1}}} \Pr\Big(\bigcap_{i,j=1}^{n}\{m_{aij} = m_{0ij}\}\Big) = \frac{1}{2^{n^2(t-1)}}.
\end{aligned} \tag{33}$$

In the last step we used the fact that the sum $\sum_{m_{0ij}\in\mathbb{Z}_{2^{t-1}}} \Pr(\bigcap_{i,j=1}^{n}\{m_{aij} = m_{0ij}\})$ is the total probability and hence is equal to 1.

Relying on the obtained equalities (31), (32) and (33) we claim that:

$$\Pr(\mathbf{U} = \mathbf{U}_0) = \Pr(\mathbf{U} = \mathbf{U}_0 \mid \mathbf{M}_a = \mathbf{M}_{a0}) = \frac{1}{2^{n^2(t-1)}}, \tag{34}$$

where $\mathbf{M}_{a0}$ is a fixed matrix defined over $\mathbb{Z}_{2^{t-1}}$.

Similarly, matrix $\mathbf{\Delta}$ is chosen uniformly from $\mathbb{Z}_2$. For this reason, analogous observation holds for the matrix sum $\mathbf{M}_b + \mathbf{\Delta}$, with probability 2^{-n^2}. However, both sums in the expression of $\mathbf{C}_1$ are independent of each other and hence we have:

$$\Pr(\mathbf{C}_1 = \mathbf{C}_{10}) = \Pr(\mathbf{C}_1 = \mathbf{C}_{10} \mid \mathbf{M} = \mathbf{M}_0) = \frac{1}{2^{n^2}} \cdot \frac{1}{2^{n^2(t-1)}} = \frac{1}{2^{tn^2}}, \tag{35}$$

where $\mathbf{C}_{10}$ is a fixed matrix defined over $\mathtt{M}_{2^t}$ and $\mathbf{M}_0$ is a fixed matrix defined over $\mathbb{Z}_{2^t}$. Hence we have shown that the entries of matrix $\mathbf{C}_1$ are uniformly distributed in $\mathtt{M}_{2^t}$.

Let us denote the set of all possible values of the key matrix $\mathbf{Y}$ by $\mathbb{K}_{\mathbf{Y}}$. Note that each matrix from this set reduced modulo 2 is a permutation matrix and hence the cardinality of this set is $|\mathbb{K}_{\mathbf{Y}}| = n! \cdot 2^{n^2(t-2)}$.

We now consider the second step of the encryption algorithm (18), i.e. matrix $\mathbf{C}_2$. Due to Lemma 1, entries of MPF value are uniformly distributed in $\mathtt{M}_{2^t}$. All that is left is to explore the conditional probabilities of its entries which are expressed as follows:

$$\Pr(\mathbf{C}_2 = \mathbf{C}_{20} \mid \mathbf{M} = \mathbf{M}_0) = \frac{\Pr(\mathbf{C}_2 = \mathbf{C}_{20}, \mathbf{M} = \mathbf{M}_0)}{\Pr(M = \mathbf{M}_0)}. \tag{36}$$

Explicit calculations of probability $\Pr(\mathbf{C}_2 = \mathbf{C}_{20}, \mathbf{M} = \mathbf{M}_0)$ are presented below in matrix form for simplicity:

$$\begin{aligned}\Pr(\mathbf{C}_2 = \mathbf{C}_{20}, \mathbf{M} = \mathbf{M}_0) &= \Pr\left({}^{\mathbf{Y}}(\mathbf{C}_1)^{\mathbf{Y}} = \mathbf{C}_{20}, \mathbf{M} = \mathbf{M}_0\right) \\ &= \left(\sum_{\mathbf{Y}_0 \in \mathbb{K}_{\mathbf{Y}}} \Pr\left(\mathbf{C}_1 = {}^{\mathbf{Y}_0^{-1}}(\mathbf{C}_{20})^{\mathbf{Y}_0^{-1}}\right) \cdot \Pr(\mathbf{Y} = \mathbf{Y}_0)\right) \Pr(\mathbf{M} = \mathbf{M}_0) \\ &= \frac{1}{2^{tn^2}} \cdot \left(\sum_{\mathbf{Y}_0 \in \mathbb{K}_{\mathbf{Y}}} \Pr(\mathbf{Y} = \mathbf{Y}_0)\right) \cdot \Pr(\mathbf{M} = \mathbf{M}_0) = \frac{1}{2^{tn^2}} \cdot \Pr(\mathbf{M} = \mathbf{M}_0), \end{aligned} \tag{37}$$

where $\mathbf{Y}_0 \in \mathbb{K}_{\mathbf{Y}}$ is a fixed matrix. Here we used the fact that the entries of $\mathbf{C}_1$ are identically uniformly distributed and are independent of the matrix $\mathbf{M}$. Also, keeping with our notation, the sum $\sum_{\mathbf{Y}_0 \in \mathbb{K}_{\mathbf{Y}}} \Pr(\mathbf{Y} = \mathbf{Y}_0)$ represents a total probability and hence is equal to 1. Note that we use the notation $\Pr(\mathbf{M} = \mathbf{M}_0)$ to indicate the probability of a certain fixed message, which is then split into two parts $\mathbf{M}_a$ and $\mathbf{M}_b$.

We limit ourselves to the matrix form of these calculations since the expression of probability for a single entry of $\mathbf{C}_2$ is much more complicated due to restriction on matrix $\mathbf{Y}$.

Since expression (37) is a numerator of conditional probability (36), we obtain the following result:

$$\Pr(\mathbf{C}_2 = \mathbf{C}_{20} \mid \mathbf{M} = \mathbf{M}_0) = \frac{\frac{1}{2^{tn^2}} \cdot \Pr(\mathbf{M} = \mathbf{M}_0)}{\Pr(\mathbf{M} = \mathbf{M}_0)} = \frac{1}{2^{tn^2}}. \tag{38}$$

Comparing this result to the expression (29), we can see that the distributions match and hence draw a conclusion that entries of the matrix $\mathbf{C}_2$ are independent of plaintext matrix $\mathbf{M}$.

The proof for the last step of the encryption algorithm is analogous to the proof for the first step since the matrix $\mathbf{\Delta} \parallel \mathbf{X}$ consists of uniformly distributed in $\mathbb{Z}_{2^t}$ entries whereas the shifting function does not have an impact on the distribution of the entries of the other matrix summand. □

Due to the proven result, we can see that no information about the plaintext is leaked by the encryption algorithm. This is the essential property any good symmetric cipher should possess.

5. Comparison With One-Time Pad

A classic example of a perfectly secure cipher is the one-time pad scheme proposed by G. Vernam in the early XX century. It uses a key k the size of the message mu and a simple XOR operation $\oplus$ to obtain a ciphertext $c = \mu \oplus k$. Decryption works similarly, i.e. $\mu = c \oplus k$.

However, despite being an ideal cipher, its practical implementation is highly limited. Firstly, the size of the secret key is a big problem, e.g. encrypting a 1GB file requires a key of the same size. Obviously, no user wants to waste his memory space storing such a key. So far in this sense, our cipher seems even worse since the size of the key is about

twice the size of the message. Moreover, regardless of any actions we make, the size of the secret key has to be at least the size of the message for our cipher to remain perfectly secure. This fact is called the Shannon theorem.

The logical question now is if we can gain any benefits by using such a key to encrypt a message. To answer this question we consider another flaw in the one-time pad scheme. It is widely known that reusing the same key k to encrypt messages μ_1, μ_2 results in a catastrophe, i.e. any adversary possessing $c_1 = \mu_1 \oplus k$ and $c_2 = \mu_2 \oplus k$ is able to restore μ_2 given that the plaintext μ_1 is known to him since he can perform the following calculation:

$$c_2 \oplus c_1 \oplus \mu_1 = (\mu_2 \oplus k) \oplus (\mu_1 \oplus k) \oplus \mu_1 = \mu_2. \tag{39}$$

This fact can be viewed as gaining an advantage of 1 in winning the following Attack Game aimed at the recovery of data encrypted by a fixed key k:

Attack Game 1. *For a given symmetric cipher* $\varepsilon = (\mathrm{Enc}(k, \mu), \mathrm{Dec}(k, c))$ *defined over* $(\mathbb{K}, \mathbb{M}, \mathbb{C})$ *define the following attack game*:

1. *The challenger picks at random a secret key* $k \in \mathbb{K}$;
2. *The adversary sends a sequence of queries* $\mu_1, \mu_2, \ldots, \mu_Q$ *of equal size to the challenger*;
3. *The challenger calculates the ciphertexts* $c_1, c_2, \ldots, c_Q$, *where* $c_i = \mathrm{Enc}(k, \mu_i)$, *and sends them to the adversary*;
4. *The adversary outputs a pair* (μ, c), *where* $\mu \in \mathbb{M} \setminus \{\mu_1, \mu_2, \ldots, \mu_Q\}$ *and* $c \in \mathbb{C} \setminus \{c_1, c_2, \ldots, c_Q\}$. *He wins if* $c = \mathrm{Enc}(k, \mu)$.

We let the adversary be adaptive, i.e. he can choose his queries based on the ciphertexts obtained from the challenger.

Obviously, for the case of a one-time pad scheme, an adversary requires a single query, i.e. $Q = 1$. In fact, the secret key k is trivially recoverable in this case.

Let us denote the event of winning the Attack Game 1 by W.

Definition 1. The advantage of the adversary $\mathcal{A}$ in winning the Attack Game 1 is given by

$$KRadv[\mathcal{A}, \varepsilon] = \left| \Pr(W) - \frac{1}{|\mathbb{K}|} \right|, \tag{40}$$

where $|\mathbb{K}|$ denotes the cardinality of the keyspace.

Note that due to expression (39) the adversary may not necessarily obtain the secret key k to win the game as long as he can output a working pair (μ, c). Hence, he has two alternatives to winning: determining the secret key or using the obtained replies to gain a way to output a working pair. The advantage $KRadv[\mathcal{A}, \varepsilon]$ shows how much better than randomly guessing the key can the adversary $\mathcal{A}$ do.

Definition 2. The symmetric cipher is secure under key reuse if for any poly-bounded number of queries Q the advantage $KRadv[\mathcal{A}, \varepsilon]$ is negligible.

As we have seen one-time pad is not secure under key reuse. We prove the following proposition:

Theorem 2. *The Shannon block cipher defined by the encryption algorithm* (18) *and decryption algorithm* (20) *is secure under key reuse.*

Proof. Let us consider both alternatives for winning the Attack Game 1.

Firstly, we consider determining the key strategy. Assume that the adversary $\mathcal{A}$ received ciphertext matrices $\mathbf{C}^{(1)}, \mathbf{C}^{(2)}, \ldots, \mathbf{C}^{(Q)}$ matching the known message matrices $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_Q$. Here we use the upper indexes for $\mathbf{C}$'s to distinguish challenger responses from intermediate results of the encryption algorithm (18). Hence, an adversary can analyse the following system of equations:

$$\begin{cases} \mathbf{C}^{(1)} = \mathrm{Shift}_\kappa\big(2^{t-1}\Phi\big(\mathbf{C}_2^{(1)}\big) + \Psi\big(\mathbf{C}_2^{(1)}\big)\big) + \big(2^{t-1}\boldsymbol{\Delta} + \mathbf{X}\big), \\ \mathbf{C}^{(2)} = \mathrm{Shift}_\kappa\big(2^{t-1}\Phi\big(\mathbf{C}_2^{(2)}\big) + \Psi\big(\mathbf{C}_2^{(2)}\big)\big) + \big(2^{t-1}\boldsymbol{\Delta} + \mathbf{X}\big), \\ \qquad\qquad \ldots \\ \mathbf{C}^{(Q)} = \mathrm{Shift}_\kappa\big(2^{t-1}\Phi\big(\mathbf{C}_2^{(Q)}\big) + \Psi\big(\mathbf{C}_2^{(Q)}\big)\big) + \big(2^{t-1}\boldsymbol{\Delta} + \mathbf{X}\big), \end{cases} \tag{41}$$

where matrices $\mathbf{X}$, $\mathbf{Y}$, $\boldsymbol{\Delta}$ are unknown, $\mathbf{C}_2^{(1)}, \mathbf{C}_2^{(2)}, \ldots, \mathbf{C}_2^{(Q)}$ are intermediate matrices at the second step of the encryption function (19), and $\mathbf{C}^{(1)}, \mathbf{C}^{(2)}, \ldots, \mathbf{C}^{(Q)}$ are its output values, i.e. responses the adversary $\mathcal{A}$ sees. However, simplifying this system is not an easy task, since at the very least we have to take the non-commuting nature of $\mathtt{M}_{2^t}$ into account. In other words, reducing all the equations modulo 2^{t-1} which would remove the non-commuting aspect of $\mathtt{M}_{2^t}$ is not helpful since in expression (19) the matrix $\boldsymbol{\Delta}$ immediately vanishes along with leading bits of the first term. Furthermore, the shifting operator is not action preserving thus any calculations analogous to (39) are inefficient. For example, computing $\mathbf{C}^{(1)} - \mathbf{C}^{(2)}$ we get:

$$\mathbf{C}^{(1)} - \mathbf{C}^{(2)} = \mathrm{Shift}_\kappa\big(2^{t-1}\Phi_1 + \Psi_1\big) - \mathrm{Shift}_\kappa\big(2^{t-1}\Phi_2 + \Psi_2\big),$$

where notations Φ_1, Φ_2, Ψ_1, Ψ_2 are used to shorten the appropriate expressions in (41). We have

$$\mathrm{Shift}_{t-\kappa}\big(\mathbf{C}^{(1)} - \mathbf{C}^{(2)}\big) \neq 2^{t-1}\Phi_1 + \Psi_1 - 2^{t-1}\Phi_2 - \Psi_2$$

and thus we cannot make a new equation based on the obtained responses.

Hence, even knowing the parameter k the adversary $\mathcal{A}$ cannot use this information to formulate an advantageous system of equations to extract the secret key. As such we conclude that the key determination strategy is not applicable.

Hence we consider the other option, i.e. using the responses to find a way of outputting a working pair. In this scenario, we assume that an adversary obtained n^2 matrices $\mathbf{C}^{(1)}, \mathbf{C}^{(2)}, \ldots, \mathbf{C}^{(n^2)}$. Moreover, we can also assume that the correspondent message matrices $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_{n^2}$ are linearly independent and hence form a basis of the linear space $\mathbb{M} = Mat(\mathbb{Z}_{2^t})$. Then he can express each subsequent query $\mathbf{M}_j$, $j > n^2$ as a linear combination of $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_{n^2}$. Also, since $\mathbb{C} = \mathbb{M}$ on the same basis can also be used to express the ciphertext matrices $\mathbf{C}^{(1)}, \mathbf{C}^{(2)}, \ldots, \mathbf{C}^{(n^2)}$, as well as the corresponding response $\mathbf{C}^{(j)}$ as a linear combination of $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_{n^2}$. Hence, the adversary can get the following results:

$$\begin{aligned} \mathbf{M}_j &= \sum_{i=1}^{n^2} \alpha_{ij} \mathbf{M}_i; \\ \mathbf{C}^{(j)} &= \sum_{i=1}^{n^2} \beta_{ij} \mathbf{M}_i. \end{aligned} \tag{42}$$

However, the coefficients α_{ij} and β_{ij} change independently of each other due to the perfect secrecy property of our cipher, thus establishing a non-linear link between these coefficients. In other words, the obtained coefficients β_{ij} seem completely random to $\mathcal{A}$.

For this reason a relation between coefficients α_{ij} and β_{ij} can be viewed as a random permutation mapping $P(\alpha) = \beta$, where $\alpha, \beta \in \mathbb{Z}_{2^{t-1}}^{n^2}$. Define an adversary $\mathcal{B}$ who plays the role of challenger to $\mathcal{A}$ and plays the Attack Game 4.1 (see Boneh and Shoup, 2020) with his challenger. Recall that Attack Game 4.1 is aimed at distinguishing an encryption function from a random permutation. To be self-contained, let us revise this game:

Attack Game 2. *For the block cipher* $\varepsilon = \{\mathrm{Enc}(\vec{\mathbf{K}}, \mathbf{M}), \mathrm{Dec}(\vec{\mathbf{K}}, \mathbf{C})\}$ *we define two experiments. Then for a value* $\beta \in \{0, 1\}$ *we have an Experiment* β:

1. *The challenger selects a function* E_β *as follows*:

$$E_\beta = \begin{cases} \mathrm{Enc}(\vec{\mathbf{K}}, \mathbf{M}), & \text{if } \beta = 0; \\ \mathit{Rand}(\mathbf{M}), & \text{otherwise}. \end{cases}$$

2. *The adversary* $\mathcal{B}$ *submits a sequence of queries, i.e. plaintexts in their matrix form* $\mathbf{M}_i$, *where* $i = 1, 2, \ldots, Q$;
3. *For the* i*-th query the challenger computes* $\mathbf{C}^{(i)} = E_\beta(\mathbf{M}_i)$ *and sends all the* $\mathbf{C}_i$*'s to an adversary;*
4. $\mathcal{B}$ *outputs* $\hat{\beta} \in \{0, 1\}$.

Denote by W_β *the random event that in Experiment* β $\mathcal{B}$ *outputs* 1. *Then* $\mathcal{B}$*'s advantage is defined as*

$$BCadv[\mathcal{A}, \varepsilon] = \left|\Pr(W_1) - \Pr(W_0)\right|.$$

Whenever $\mathcal{B}$ receives a query $\mathbf{M}_j$ from $\mathcal{A}$, he sends it to his challenger and afterward forwards the obtained response $\mathbf{C}_j$ back to $\mathcal{A}$. Steps 1 and 3 of the Attack Game 1 are performed by $\mathcal{B}$'s challenger. Due to the perfect secrecy of our cipher, $\mathcal{B}$'s advantage in winning the Attack Game 2 on his own is negligible, i.e. he cannot tell apart the encryption function from a random permutation. On the other hand, if $\mathcal{A}$ can output a working pair $(\mathbf{M}, \mathbf{C})$ with a non-negligible probability p, then $\mathcal{B}$ can send $\mathbf{M}$ as his $(Q+1)$-st query to his own challenger and achieve an advantage of $p - \epsilon$ in Attack Game 2 if $\mathbf{C} = \mathbf{C}_{Q+1}$. However, to achieve an advantage p, the adversary $\mathcal{A}$ has to distinguish a specific mapping $P(\alpha) = \beta$ among other possible permutations with that particular probability. This would imply that not all choices are equally possible and hence it contradicts the perfect secrecy of our cipher.

As such, we see that the only chance the adversary $\mathcal{A}$ has is to randomly guess a pair $(\mathbf{M}, \mathbf{C})$ and hope for it to work. However, due to the design of Attack Game 1, there are $|\mathbb{M}| - Q$ leftover working pairs out of $(|\mathbb{M}| - Q)^2$ possible pairs. Furthermore, due to restrictions applied to the key matrices $(\mathbf{X}, \mathbf{Y}, \mathbf{\Delta})$, the size of the keyspace is

$$|\mathbb{K}| = 2^{n^2(t-1)} \cdot 2^{n^2(t-2)} n! \cdot 2^{n^2} = 2^{n^2(2t-2)} n!,$$

where each multiplier describes the total choices of $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{\Delta}$, respectively.

Then we can evaluate $\mathcal{A}$'s advantage in Attack Game 1 as follows

$$\begin{aligned} KRadv[\mathcal{A}, \varepsilon] &= \left| \frac{|\mathbb{M}| - Q}{(|\mathbb{M}| - Q)^2} - \frac{1}{2^{n^2(2t-2)} n!} \right| \\ &= \left| \frac{1}{2^{n^2 t} - Q} - \frac{1}{2^{n^2(2t-2)} n!} \right| = \frac{n! 2^{n^2(t-2)} - 1 + Q \cdot 2^{-n^2 t}}{(2^{n^2 t} - Q) n! 2^{n^2(t-2)}}. \end{aligned} \tag{43}$$

Throwing away a negligible term $Q \cdot 2^{-n^2 t}$ and approximating the ratio $\frac{n! 2^{n^2(t-2)} - 1}{n! 2^{n^2(t-2)}} \approx 1$ we obtain the following result:

$$KRadv[\mathcal{A}, \varepsilon] < \frac{1}{2^{n^2 t} - Q},$$

which is the probability of randomly guessing a working pair $(\mathbf{M}, \mathbf{C})$. The obtained advantage is negligible which ends the proof. □

This result is a significant advantage of our cipher over the one-time pad technique. Specifically, as opposed to a one-time pad we do not need to use a different key whenever we use our scheme to encrypt a message. Furthermore, this beneficial property of our cipher greatly outshines the drawback of using a longer key, since it unlocks the implementation of different types of modes, e.g. CBC. This ability follows from the fact that we have to use the same key to encrypt a large number of blocks. As the one-time pad is insecure under key reuse, no encryption mode can ever be constructed on its basis.

6. CBC Mode of the Proposed Block Cipher

The general idea of the CBC mode is to unite encrypted chunks of the message into a chain. To withstand the chosen plaintext attack, our cipher has to be probabilistic. The commonly used solution is to use a randomly generated initialization vector **IV**. In our case, we interpret it as a matrix $\mathbf{C}^{(0)} \in Mat(\mathbb{Z}_{2^t})$. We use this matrix together with the secret key $\vec{\mathbf{K}}$ to create a chain in the following way:

$$\mathbf{C}^{(i)} = \mathrm{Enc}\big(\vec{\mathbf{K}}, \mathbf{M}_i + \mathbf{C}^{(i-1)}\big), \tag{44}$$

where $\mathrm{Enc}(\vec{\mathbf{K}}, \mathbf{M})$ is the encryption function defined by (19). The result of this procedure is the ciphertext

$$c = c_{11}^{(0)} \parallel c_{12}^{(0)} \parallel \ldots \parallel c_{nn}^{(0)} \parallel c_{11}^{(1)} \parallel c_{12}^{(1)} \parallel \ldots \parallel c_{nn}^{(1)} \parallel \ldots \parallel c_{nn}^{(l)},$$

where l denotes the number of blocks. The decryption of a ciphertext is performed as follows:

$$\mathbf{M}_i = \mathrm{Dec}\big(\vec{\mathbf{K}}, \mathbf{C}^{(i)}\big) - \mathbf{C}^{(i-1)}, \tag{45}$$

where $\mathrm{Dec}(\vec{\mathbf{K}}, \mathbf{C})$ is a decryption function defined by (21). The proof of the correctness of CBC mode follows from the result proven in Section 3.4.

We see that the ciphertext is longer than the plaintext which is a common practice when implementing a CBC mode. As the number of blocks gets larger, the CBC mode of our cipher becomes more efficient as compared to the one-time pad technique. Furthermore, the proof of a perfect secrecy property still holds for a single block $\mathbf{M}_i$. However, we emphasize that when referring to the perfect secrecy property we only consider the initial block cipher. Obviously, as the number of blocks increases, the size of the message surpasses the size of the key and hence the CBC mode is not perfectly secure, which is consistent with the Shannon theorem.

7. Resistance Against Chosen Plaintext Attack

In this section, we consider the security of our scheme. More precisely, we turn our attention to the chosen plaintext attack which is aimed at the newly defined CBC mode. Any efficient adversary capable of successfully executing this attack can distinguish a plaintext corresponding to the received ciphertext based on the obtained responses to his queries. Moreover, the adversary is adaptable, which means that he can base his queries on the received information. The formal description of this attack is presented here as the following game:

Attack Game 3. *For a given symmetric cipher* $\varepsilon = (\mathrm{Enc}(k, \mu), \mathrm{Dec}(k, c))$ *defined over* $(\mathbb{K}, \mathbb{M}, \mathbb{C})$ *define the CBC mode* $\varepsilon' = (\mathrm{Enc}'(\vec{\mathbf{K}}, \mu), \mathrm{Dec}'(\vec{\mathbf{K}}, c))$ *using encryption and decryption functions* (44) *and* (45) *respectively. Consider the following attack game*:

1. *The challenger selects a random key* $\vec{\mathbf{K}}$;
2. *The adversary* $\mathcal{A}$ *submits a sequence of queries i.e. plaintext pairs* (μ_{i0}, μ_{i1}) *of equal lengths, where* $i = 1, 2, \ldots, Q$;
3. *For the* i*-th query the challenger computes* $c_i = \mathrm{Enc}'(\vec{\mathbf{K}}, \mu_{i\beta})$*, where* $\beta \in \{0, 1\}$ *is the Experiment indicator, and sends all the* $\mathbf{C}_i$*'s to an adversary*;
4. $\mathcal{A}$ *outputs* $\hat{\beta} \in \{0, 1\}$.

Denote by W_β *the random event that in Experiment* β $\mathcal{A}$ *outputs* 1. *Then* $\mathcal{A}$*'s advantage is defined as*

$$CPAadv\big[\mathcal{A}, \varepsilon'\big] = \big|\Pr(W_1) - \Pr(W_0)\big|.$$

Note that the challenger of the Attack Game 3 always encrypts either the first or second messages of each query. The essence of the presented Attack Game is that an adversary can win it with a non-negligible probability if he can somehow relate the received ciphertext c_i to the correct message in the pair (μ_{i0}, μ_{i1}).

Let us make two important observations. Firstly, the message space of the CBC mode is super-poly. In fact, its size is $|\mathbb{M}| = 2^{n^2 t}$. Secondly, the number of blocks l is poly-bounded and determined by the length of the plaintext as follows:

$$l = \left\lceil \frac{|\mu|}{n^2 t} \right\rceil.$$

For these reasons we rely on a strategy presented in (Boneh and Shoup, 2020) to prove the following claim:

Theorem 3. *Consider probabilistic cipher* $\varepsilon' = \{\mathrm{Enc}'(\vec{\mathbf{K}}, \mu), \mathrm{Dec}'(\vec{\mathbf{K}}, c)\}$. *For all efficient adversaries* $\mathcal{A}$ *their advantage in Attack Game 3 is expressed as follows*:

$$CPAadv\big[\mathcal{A}, \varepsilon'\big] = \frac{Q^2 l^2}{(l+1)2^{n^2 t - 1}} + 2BCadv[\mathcal{B}, \varepsilon], \tag{46}$$

where Q *is the number of queries in Attack Game 3,* l *is the total number of blocks needed to encrypt a plaintext* μ_{ib} *and* $BCadv[\mathcal{B}, \varepsilon]$ *is the advantage of the adversary* $\mathcal{B}$ *in winning the Attack Game 2.*

Before presenting the proof for this theorem, we emphasize that the main adversary in the Attack Game 3 is $\mathcal{A}$. However, he also communicates with adversary $\mathcal{B}$, who attacks the block cipher ε as in Attack Game 2 and forwards $\mathcal{A}$'s queries to his challenger.

Proof. Note that before encrypting the first block of the plaintext $\mu_{i\beta}$ a challenger randomly selects an initialization vector $\mathbf{C}^{(0)}$ and hence the intermediate block $\mathbf{C}_1^{(i)}$ consists of random uniformly distributed entries. Hence, by the construction of our scheme the advantage $CPAadv^*[\mathcal{A}, \varepsilon']$ of adversary $\mathcal{A}$ to win a bit-guessing version (i.e. an adversary

wins the game if $\hat{\beta} = \beta$) of the Attack Game 3 is given by:

$$CPAadv^*[\mathcal{A}, \varepsilon'] = \left|\Pr(W_0) - \frac{1}{2}\right|,$$

i.e. he can do no better than randomly guessing the Experiment indicator β.

To improve his chances $\mathcal{A}$ collaborates with another adversary $\mathcal{B}$ whose purpose is to analyse the original block cipher by playing the Attack Game 2. Adversary $\mathcal{B}$ wins if he can distinguish between the encrypted block and a random permutation. This is where the perfect secrecy property of our cipher plays a significant role. Due to this property, the entries of the ciphertext matrix $\mathbf{C}^{(j)}$ are statistically independent of the entries of the original message block $\mathbf{M}_j$. Hence, this behaviour is indistinguishable from a random permutation and thus the adversary $\mathcal{B}$ cannot gain any significant advantage.

Moreover, since the initialization matrix $\mathbf{C}^{(0)}$ is selected randomly from a significantly large space of possible values (in fact, the size of this space is super-poly), the responses to multiple queries of the same plaintext are almost always distinct. This claim is based on two facts: choosing the same initialization matrix is practically an impossible random event and the encryption function is a one-to-one mapping. As such, $BCadv[\mathcal{B}, \varepsilon]$ can be estimated in the following way:

$$BCadv[\mathcal{B}, \varepsilon] \leqslant \frac{1}{|\mathbb{K}|} = \frac{1}{2^{n^2(2t-2)}n!}.$$

Obviously, this advantage is negligible for all blocks, including the first one. Moreover, it is negligible even compared to the first term of $CPAadv[\mathcal{A}, \varepsilon']$ as can be seen from (46).

The strategy now is to introduce Games 2 and 3 as in the proof of Theorem 5.4 of (Boneh and Shoup, 2020) and evaluate the appropriate results. These games explore the changes influenced by switching from a permutation to a one-to-one mapping and then to many-to-one mapping. These changes are unnoticeable to the adversary under the assumptions that ε is a secure block cipher and the message space is super-poly. Both these assumptions are satisfied for our scheme. We limit ourselves to the essence of these games and leave their detailed description outside of this paper since they are technical and universal for all encryption algorithms. The changes are minor and involve the algebraic structures and actions used in the initial block cipher. In our case – matrix space $Mat_n(\mathbb{Z}_{2^t})$ and entry-wise addition modulo 2^t. □

Note that we used the same Attack Game 2 in this proof as in Section 5. This comes from the fact that a good block cipher should be indistinguishable from a random permutation and unpredictable. As it was shown, Attack Game 2 plays an important role in establishing both of these properties.

Let us end this section by presenting an example of computing the $CPAadv[\mathcal{A}, \varepsilon']$. Inspired by the fact that AES encrypts a 128-bit block, we pick the non-commuting group $\mathtt{M}_{256}$ and consider 4×4 matrices, i.e. public parameters $t = 8$, $n = 4$ and the size of the block is $n^2 t = 4^2 \cdot 8 = 128$ bits. Furthermore, we limit the message size by 2^{32} blocks

and hence can encrypt 2^{39} bits (64 GB) of information. Then we get the following result:

$$CPAadv[\mathcal{A}, \varepsilon'] \leqslant \frac{Q^2 \cdot 2^{64}}{(2^{32}+1)2^{127}} + \frac{1}{2^{223} \cdot 24} \leqslant Q^2 \cdot 2^{-159}.$$

Then by sending $Q = 2^{40}$ queries, the adversary gains an advantage $CPAadv[\mathcal{A}, \varepsilon'] \leqslant 2^{-79}$. Relying on the obtained advantage, a tolerable value may be fixed, thus determining how often must the session key be replaced.

8. Discussion and Conclusions

In this paper, we proposed a new Shannon cipher based on a special case of MPF. Instead of several rounds, our symmetric encryption scheme uses only one round. However, the operations we use are more complex. Moreover, we use a non-commuting platform group in our construction which contributes to the overall security of our cipher.

In our scheme we can manipulate two parameters: the size of square matrices n and the size of the platform group determined by t. This feature makes our scheme flexible and easy to adapt to messages of any length. However, more investigations are needed to make reasonable recommendations for the values of parameters n and t depending on the message length. This is one of the possibilities for future work in this research.

We have proven that our cipher has the property of perfect secrecy and hence the encryption algorithm itself does not leak any information about the plaintext. This is one of the essential properties of a good symmetric encryption scheme.

The perfect secrecy of our block cipher also favourably distinguishes it from a widely used AES scheme, whose perfect secrecy property for a single block to our knowledge has not been established. We also think that a significant boost in the performance of our cipher is because matrix operations can be parallelized and hence the encryption of a single block can be executed on multiple processors. Relying on our findings presented in Mihalkovich *et al.* (2022), we expect that the non-commuting platform group used in our paper also contributes to the performance of our scheme. Since all powers of the elements of $\mathtt{M}_{2^t}$ are reduced modulo 2^t, the reduction process is much simpler than reducing modulo a prime. For this reason, we think that our proposal can produce better results than those presented in Mihalkovich *et al.* (2022). However, verifying this claim requires additional research thus far.

Relying on the fact that our cipher is secure under key reuse, we defined a CBC mode for our cipher. As the message becomes longer, its length surpasses the size of the secret key, and hence due to the Shannon theorem, the perfect secrecy property is lost. However, since perfect secrecy also implies semantic security of a block cipher, we claim that the CBC mode can be considered safe in this weaker sense, i.e. an efficient adversary cannot gain a significant advantage in linking a ciphertext c to the correct plaintext.

Moreover, in Section 7 we have shown that the probabilistic cipher ε' can resist an adaptive chosen plaintext attack, i.e. the previously obtained responses to the sent queries in no way help the efficient adversary to gain a significant advantage in Attack Game 3.

References

Boneh, D., Shoup, V. (2020). A Graduate Course in Applied Cryptography.

Grundman, H.G., Smith, T.L. (1996). Automatic realizability of Galois groups of order 16. In: *Proceedings of the American Mathematical Society, AMS '96*. AMS, Rhode Island, USA, pp. 2631–2640.

Grundman, H.G., Smith, T.L. (2010a). Galois realizability of groups of order 64. *Central European Journal of Mathematics*, 8(5), 846–854.

Grundman, H.G., Smith, T.L. (2010b). Realizability and automatic realizability of Galois groups of order 32. *Central European Journal of Mathematics*, 8(2), 244–260.

Katz, J., Lindell, Y. (2007). *Introduction to Modern Cryptography*. CRC Press, New York.

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. https://doi.org/10.21595/mme.2021.22234.

Michailov, I. (2007). Groups of order 32 as Galois groups. *Serdica Mathematical Journal*, 33(1), 1–34.

Mihalkovich, A. (2018). On the associativity property of MPF over M16. *Lietuvos matematikos rinkinys: Lietuvos matematiku draugijos darbai, Serija A*, 59, 7–12.

Mihalkovich, A., Levinskas, M., Makauskas, P. (2022). MPF based symmetric cipher performance comparison to AES and TDES. *Mathematical Models in Engineering*, 8(2), 15–25. https://doi.org/10.21595/mme.2022.22517.

Mihalkovich, A., Sakalauskas, E., Luksys, K. (2020). Key exchange protocol defined over a non-commuting group based on an NP-complete decisional problem. *Symmetry*, 12, 1389. https://doi.org/10.3390/sym12091389.

Sakalauskas, E., Luksys, K. (2012). Matrix power function and its application to block cipher s-box construction. *International Journal of Innovative Computing, Information and Control*, 8(4), 2655–2664.

Sakalauskas, E., Mihalkovich, A. (2018). MPF problem over modified medial semigroup is NP-complete. *Symmetry*, 10(11), 571. https://doi.org/10.3390/sym10110571.

Sakalauskas, E., Mihalkovich, A., Uselis, A. (2020a). Security analysis of KAP based on enhanced MPF. *IET Information Security*, 14(4), 410–418.

Sakalauskas, E., Dindiene, L., Kilciauskas, A., Luksys, K. (2020b). Perfectly secure Shannon Cipher construction based on the matrix power function. *Symmetry*, 12, 860. https://doi.org/10.3390/sym12050860.

Shannon, C.E. (1949). Communication theory of secrecy systems. *The Bell System Technical Journal*, 28(4), 656–715.

Sylow, M.L. (1872). Théorèmes sur les groupes de substitutions. *Mathematische Annalen*, 5, 584–594. https://doi.org/10.1007/BF01442913.

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

M. Levinskas is currently pursuing a master's degree at Kaunas University of Technology. He is a member of Identification and Cryptography Research Group and performs investigations in symmetric cryptography.

L. Dindiene obtained her PhD in 2016 and is currently a lecturer at Kaunas University of Technology. She is a member of Identification and Cryptography Research Group and investigates statistical and probabilistic properties of cryptographic primitives.

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