

Efficient binary diffusion matrix structures for dynamic key-dependent cryptographic algorithms

Hassan N. Noura^{a,*}, Ali Chehab^b

^a Univ. Bourgogne Franche-Comté (UBFC), FEMTO-ST Institute, CNRS, Belfort, France

^b Electrical and Computer Engineering, American University of Beirut (AUB), Beirut, Lebanon

ARTICLE INFO

Keywords:

Dynamic cryptography
Binary diffusion matrix forms
Invertible or non-invertible diffusion primitives
Cryptographic analysis

ABSTRACT

In this paper, we propose a new mechanism for the generation of dynamic binary diffusion matrices, with a flexible dimension ($n \times n$), to be used in the design of new symmetric cryptographic algorithms. The proposed framework defines four primary invertible and non-invertible binary diffusion matrix forms. The advantages of these forms stem from the dynamic key approach, where each primary matrix requires the construction of two pseudo-random sub-matrices (Mu and Mv) with a size that depends on two variables (m and l). A cryptographic analysis was carried out to detect the optimal size of m and l for the construction of the matrices to achieve the best possible cryptographic performance and thus, to provide better immunity against different types of attacks. The results showed that the optimal size for m is $\in \{\frac{n}{2} - 1, \frac{n}{2}, \frac{n}{2} + 1\}$, and for $l = n - m$. Accordingly, the proposed scheme was designed with an optimal size of m and l , which resulted an acceptable linear branch number and low fixed numbers. In comparison to the existing static diffusion techniques, the proposed solution offers a higher security level since the diffusion matrices are constantly changing and they depend of the dynamic key, which is unknown to attackers.

1. Introduction

The current acceleration of digital transformation led to a sharp increase in security threats and attacks associated with drastic impacts in numerous domains. As such, it is critical to enhance the security posture of online services and data; there is a dire need to deploy security services to protect data, either in transit or at rest, and to secure the confidentiality, integrity, authentication and availability of online systems. Such security services are typically based on cryptographic algorithms and protocols [1], and they typically classified as symmetric, asymmetric or un-keyed algorithms. In Symmetric Cryptographic Algorithms (SCA), the same secret key is shared between the communicating entities. In modern systems and applications, such a key is referred to as the session key. While in the case of Asymmetric Cryptographic Algorithms (ACA), each entity has its own public and private key pair. ACAs provide different security services, whereby the public key is typically used for encryption and signature verification, and the private key for decryption and signature generation, and both keys are used for key exchange of symmetric keys.)

SCAs ensure data confidentiality by encrypting data either in block mode or stream mode. On the other hand, keyed hash functions are used to achieve data integrity and source authentication. SCAs are

typically used for data confidentiality since they are less computationally expensive when compared to ACAs. The security of any SCA is related to the performance of its round function that consists of two main cryptographic primitives, diffusion (linear branch number) and confusion (linear and differential probability approximation). This round function should be iterated for r rounds to be considered strong against analytic attacks. In this work, we focus on the diffusion process since it exhibits much higher computational complexity as compared to the confusion process.

The diffusion process is the basis of any SCA cipher, hash function, key derivation function, and Pseudo Random-Number Generator (PRNG). Hence, designing a robust diffusion process has attracted the attention of many researchers. In general, block ciphers and coding schemes utilize an invertible diffusion matrix, while a non-invertible one can be used for stream ciphers, hash functions, key derivation functions, and PRNGs [2,3].

The main objective of the confusion and diffusion processes is to ensure the avalanche effect at the local (byte or word) and global (block) levels. The output bits should be very sensitive to a slight change of the input bits [4]. In principle, the diffusion process can be applied by using either static or dynamic diffusion primitives. The majority of existing SCAs adopt the static structure due to the fact that the

* Corresponding author.

E-mail address: hassan.noura@univ-fcomte.fr (H.N. Noura).

proper selection of static substitution and diffusion primitives ensures the maximum cryptographic performance. This is typically done by using substitution tables or non-linear transformations with minimum linear and differential probability, in addition to the strict avalanche criteria [5]. The diffusion matrix, on the other hand, should have a maximum linear branch number. A number of static construction techniques of binary diffusion matrices were presented in [6–10].

A dynamic SCA has a key-dependent structure where the substitution and diffusion primitives depend on a secret key, such as the Blowfish algorithm [11], which uses a variable substitution table.

The security level of existing symmetric ciphers depends on the number of rounds r ; a higher r increases the security at the expense of additional computational complexity and delay, and thus, there is always a trade-off between the security level and the required performance. Static encryption algorithms have proven their resistance against analytic cryptanalysis but with an additional overhead in terms of latency and required resources [12–15]. Moreover, the fixed structures lend themselves to future potential attacks that would compromise the secret keys being used; the static primitives are made public and known to attackers and thus, they are vulnerable to advanced statistical analysis and new types of cryptanalysis [16].

On the other hand, there is a current trend to rely on dynamic cryptographic algorithms such as the ones presented in [12–15]. In such a case, the cryptographic primitives used for confusion and diffusion vary with each new input message, and the mode of operation could be as well dynamic as described in [17].

Typically, the dynamic approach requires an optimization process to reduce the initialization overhead associated with the generation of the dynamic cryptographic primitives, for each input block or small message. One possible mechanism is to generate offline a set of cryptographic primitives such as the binary diffusion matrices proposed in this paper, and then, whenever there is a need to update the primitives, new cryptographic primitives will simply be selected in a pseudo-random manner. This will tremendously decrease the complexity of the dynamic cryptographic algorithm.

Furthermore, When designing a dynamic structure, several requirements must be met: (a) simplicity of the operations (e.g., logical operations) to reduce the computational complexity, (b) flexibility to facilitate hardware and software implementations, and most importantly, (c) achieving a high cryptographic performance to protect the system against various types of attacks.

The motivation for this work stems from the fact that there are no existing approaches that satisfy the above requirements, and hence, there is a need for a scheme that provides a good cryptographic performance in a pseudo-random manner, through a new flexible, dynamic, robust diffusion technique for a modern SCAs. This diffusion primitive consists of a linear transformation with a matrix representation that could be either binary over Galois Field $GF(2)$, or integer over $GF(2^m)$, where m represents the precision (number of bits).

It should be noted that the binary diffusion matrices have suitable implementation properties for 8-bit, 32-bit and 64-bit processors. The simplicity of operations inherent within the proposed dynamic diffusion primitive, along with the appropriate security level, make of it an adequate choice to be used in modern cryptographic algorithms.

1.1. Problem and motivation

A binary diffusion matrix exhibits low computational complexity and simple implementation since it depends on the logical “Exclusive OR” operation. The challenge is to design a binary diffusion matrix that exhibits as well a strong cryptographic performance. The number of possible vectors that form an $(n \times n)$ invertible matrix over field q is [18]:

$$\prod_{k=1}^n (q^n - q^{k-1}). \tag{1}$$

So the probability of this matrix being invertible is:

$$\frac{\prod_{k=1}^n (q^n - q^{k-1})}{q^{n^2}} = \prod_{k=1}^n (1 - q^{k-1-n}) < 1 - \frac{1}{q}. \tag{2}$$

This probability decreases as n grows, which limits the flexibility property. Also, when $q = 2$, which corresponds to the binary case, the invertibility probability converges to ≈ 0.288788 as $n \rightarrow \infty$, as indicated in [18]. Therefore, there are much fewer invertible matrices in the binary field as compared to the integer field. Hence, a good bound cannot be obtained in the binary Galois field. Also, a large q (e.g., 2^8 , 2^{16}) ensures linear independence among the elements of the diffusion vector. Their probability is high, close to 0.9961 for $q = 2^8$, but the corresponding arithmetic operations require more than one clock cycle, which reduces the efficiency of such integer diffusion matrix.

To address this limitation, the authors in [16] proposed four dynamic integer invertible forms of diffusion matrices where one sub-matrix is required for each form. However, these forms have one limitation due to the fact that the existing sub-matrices do not have equal priorities; there is always a sub-matrix that has higher values compared to the other sub-matrices. This limitation was then solved in [19] by applying a shuffling algorithm to the lines, followed by shuffling the columns, for each diffusion matrix. This solution was adapted for secret encoding of packets. However, the solution is not highly efficient because of the required complexity since it is based on integer matrix multiplications.

1.2. Contribution

This work targets the design of a construction technique for a new type of invertible and non-invertible binary diffusion matrices that are dynamic and flexible. For this purpose, the algebraic integer diffusion matrix form of [16,19] is adapted to the Galois binary field matrix form. Each of these primary forms requires two sub-matrices to be constructed (Mu and Mv). However, these primary matrix forms, on their own, do not ensure the best possible cryptographic performance. To that end, we propose a scheme with a binary matrix multiplication between two primary matrix forms to produce the final $(n \times n)$ either invertible or non-invertible diffusion matrix. The resulting binary matrices exhibit a higher cryptographic performance when compared to a single primary form. From a performance perspective, the computational complexity for the generation of the final diffusion matrix is relatively low; it is simply based on the iteration of a stream cipher to construct two different matrix forms, in addition to the binary matrix multiplication in the Galois binary field. This does not require any optimization, and it provides a high security level at a low overhead cost.

The obtained results show that the optimal dimensions of sub-matrices, which yield the best possible cryptographic performance, is for $l = m = \frac{n}{2}$. Such invertible diffusion matrices could be used in block ciphers. The non-invertible dynamic primitive, with the one-way property, could be used in stream ciphers, hash functions and pseudo-random number generators.

The design of these new types of diffusion matrices has the following advantages:

- A dynamic structure based on a dynamic key.
- A designer has the choice of an invertible or a non-invertible matrix depending on the required cryptographic task.
- The binary form simplifies the hardware and software implementations.
- Flexibility in terms of dimensions and precision.
- Good cryptographic properties: high branch number and low fixed points.

Finally, it is important to note that the construction of binary diffusion matrices advances the field of symmetric dynamic cryptographic algorithms, which strikes a good balance between the impact on performance and the security level. This was confirmed by the obtained results for $l = m = \frac{n}{2}$, especially in terms of linear branch number and fixed points.

1.3. Organization

The rest of this paper is organized as follows. Section 2 presents the related work, and Section 3 describes the required background and preliminaries for this work, in addition to a review of the related dynamic diffusion primitives, and a discussion of the challenges to achieve a good cryptographic performance. The invertible dynamic and flexible integer diffusion forms are defined in Section 4, in addition to the proposed scheme for the construction of the final binary diffusion matrices that exhibit a high cryptographic performance. The non-invertible forms are described in Section 5, while Section 6 analyzes the cryptographic properties of the proposed approach to identify the optimal number of the primary matrix forms, and the size of the corresponding sub-matrices. Finally, Section 7 concludes this work and discusses its perspectives.

2. Related work

As mentioned previously, the main issue of traditional SCAs is the high cost in terms of computations due to the repeated iterations of a round function [20]. For example, the minimum number of rounds, for existing block ciphers, is 4 for the Hummingbird2 cipher, which is not suitable for emerging systems [12,20]. Recently, researchers shifted their attention towards the design of new cryptographic structures that exhibit a high security level while requiring a low computational overhead. The researchers in [21,22] indicated clearly the need for novel lightweight SCAs that are suitable for real-time applications and for devices that are constrained in terms of computational power, memory capacity and battery life.

Several lightweight ciphers have been presented, in the ongoing effort to address the computational complexity issue, such as LED (PHOTON family) [23,24], ITUbee [25], RECTANGLE (Substitution-Permutation Network (SPN) based) [26], AKF (Feistel based) [27], Simon and Speck [28]. Then, SIMECK, a combination of the Speck and Simon ciphers was proposed in [29]. However, in [30], SIMECK proved to be susceptible to random byte and bit-flip attacks. Other lightweight block ciphers were proposed recently such as LiCi (SPN based) [31], BORON [32], PRESENT [33], GIFT [34], and CHAM [35].

In recent research, such as QTL [36], Substitution and Permutation Networks (SPN) were combined with Feistel Networks (FN) [37] to leverage the benefits of both concepts (SPN and FN). However, this did not eliminate the need for a high number of rounds. Thus, these lightweight SCAs still use the multi-round structure but with a simple round function, which does not cater for constrained devices and real-time applications. On the other hand, chaotic cryptographic algorithms have been proposed to address this issue. However, these solutions suffer from different security and performance challenges such as their vulnerability to a variety of attacks, in addition to performance difficulties such as the need for conversion operations, floating-point computations, as well as a complex hardware implementation [38,39].

Note that elliptic curve cryptography has been also used by researchers to design lightweight cipher schemes [40]. TWINE is a primary work in this direction, as described in [41,42].

In order to design a lightweight round function, and to reduce the number of rounds, several lightweight cryptographic algorithms adopted the dynamic cryptographic concept [5,20,43–45]. These solutions minimize the number of rounds while maintaining a high security level. The cipher schemes in [13,43,44] use diffusion matrices and they

require two rounds when compared to [20], which requires only one round and processes two blocks at a time.

In this work, we aim to enhance the previous works by designing a new technique to generate simple key-dependent flexible binary diffusion matrices, instead of integer diffusion matrices. In addition, we propose an update process for the binary diffusion matrices by simply permuting lines or columns to preserve the invertibility or the non-invertibility property, and to reduce the cost of re-generation of the binary diffusion primitives.

There are different types of diffusion primitives that are presented in the literature such as the Maximum Distance Separable (MDS) code approach used in the Advanced Encryption Standard (AES) [46]. There is also the Binary Matrix (BM) approach, which has an advantage over MDS since its implementation requires only XOR operations, while the MDS process consumes more than 75% of the AES execution time, in addition to other XOR operations [47].

Recent block ciphers are relying on binary diffusion matrices such as Camellia [48] and ARIA [49], which require a maximum branch number and thus, they are referred to as Maximum Distance Binary Linear (MDBL) codes [50]. The maximum branch number for (8×8) and (16×16) binary matrices is 5 and 8, respectively. This means that the input difference and the corresponding output difference across these matrices have a total weight of 5 and 8, respectively. Except for our previous work [51], all other BM candidates have a static nature, as described by Koo et al. [52]; the authors presented a static method to build a (16×16) binary matrix with a branch number of 8. The authors of [53] presented the construction of (32×32) BMs with a branch number of 10. Recently, in [54], the authors presented another static algebraic binary method to build (8×8) and (16×16) matrices with a maximum branch number. Additionally, in [55] another static BMs for lightweight block ciphers and hash functions were investigated. However, invertible BMs cannot be used in the design of hash functions due to the required one-way property, which necessitates the use of non-invertible BMs.

These approaches have a static structure that is invertible and with a fixed size, which makes them limited in terms of flexibility. A configurable size of the diffusion matrix offers the flexibility required for enhanced robustness of a cryptographic algorithm, and paves the way for the generation of new invertible and non-invertible matrices that are needed in most SCAs.

3. Background & preliminaries

In this section, we describe the concepts behind the dynamic key-dependent cryptographic algorithms, and the dynamic diffusion primitives (matrix forms), and we illustrate the existing invertible and non-invertible forms.

3.1. Dynamic key-dependent cryptographic algorithms

The classical symmetric cryptographic algorithms, such as AES, are based on a static structure: the substitution box has fixed values, and the column multiplication is always performed with the same matrix. These two primitives remain unchanged throughout the encryption and decryption processes.

In a dynamic encryption scheme, the secret key is used to generate a dynamic key, which changes for every new message or for a set of messages. The cryptographic primitives, which are used for substitution and/or permutation in the encryption process, are based on the dynamic key and hence, they also change with each new message.

As such, in classical encryption, the primitives are made public, and what remains as a secret is just the key. However, in dynamic encryption, the primitives are variable and no longer known, which renders attacks much harder to perform.

Table 1

Table of abbreviation.

Abbreviation	Explanation
SCA	Symmetric Cryptographic Algorithm
ACA	Asymmetric Cryptographic Algorithms
PRNG	Pseudo Random-Number Generators
SPN	Substitution-Permutation Network
FN	Feistel Networks
AES	Advanced Encryption Standard
BM	Binary Matrix
NBM	Non-invertible Binary Matrix
MDS	Maximum Distance Separable
MDBL	Maximum Distance Binary Linear
NM	Non-invertible diffusion Matrix
BN	Branch Number
FP	Fixed Points
SAC	Strict Avalanche Criterion
BIC	Output Bit Independence Criterion

3.2. Dynamic binary diffusion matrix structures

The proposed diffusion matrix has binary values as opposed to integer ones, and thus, the arithmetic operations of addition and multiplication are simply performed using logical operations (XOR). This decreases tremendously the computational complexity of the diffusion process. Also, the construction technique of these binary diffusion matrices is simple. The contribution is mainly based on the construction of such simple, efficient and flexible binary diffusion matrices, both in invertible and non-invertible forms, while preserving the security strength of the cryptographic primitives.

Next, we describe the mathematical procedure of the solution presented in [51], along with the matrix construction technique used to produce an invertible or non-invertible dynamic binary diffusion matrix. Then, we analyze the solution to illustrate its limited cryptographic performance. The list of used abbreviations is shown in Table 1, and the list of used notations is presented in Table 2, to make it easier for the reader to follow the proposed approach.

3.3. Invertible binary diffusion matrix form

The dynamic diffusion invertible matrix used in [51] is based on a particular matrix structure, the invertible Bijective 2D matrix, that is simple with an acceptable computational overhead. The 2D matrix is presented in Eq. (3):

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}; \det(M) = ad - bc \tag{3}$$

To ensure invertibility, $\det(M) = 1 \Rightarrow ad = 1 + bc$. We consider that d is equal to a , which leads to $a^2 = 1 + bc \Rightarrow bc = a^2 - 1 = (a - 1)(a + 1)$. As a result, b and c become equal to $(a + 1)$ and $(a - 1)$, respectively. Then, the form of the secret matrix can be re-written in Eq. (4) as:

$$M = \begin{bmatrix} a & a + 1 \\ a - 1 & a \end{bmatrix} \tag{4}$$

As such, the invertible diffusion matrix is based on a single parameter, a , which results in a poor cryptographic performance, as it will be shown later on. The parameter a is replaced by the dynamic pseudo-random sub matrix A to form a flexible secret matrix, having a dimension equal to n , as shown in Eq. (5).

$$M = \begin{bmatrix} A & A + I_m \\ A - I_m & A \end{bmatrix} \tag{5}$$

where I_m and A are the identity matrix and the non-zero matrix of size $\frac{n}{2}$, respectively. The elements of A can be freely chosen from any Galois field such that M is full rank. The invertibility of a matrix M , which is

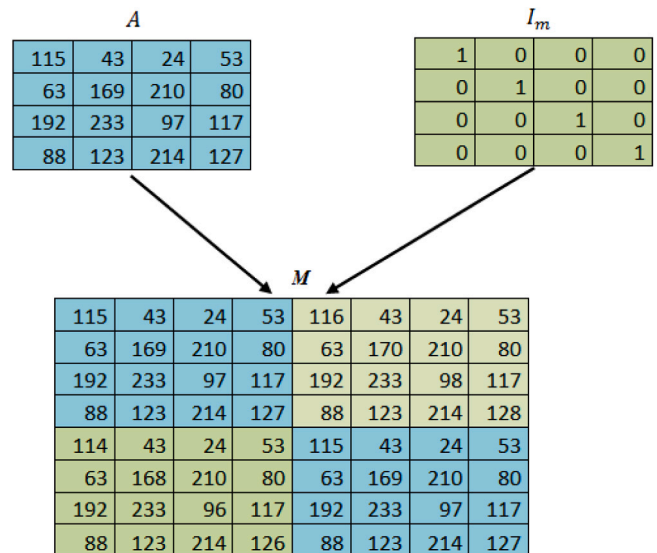


Fig. 1. An example of the integer diffusion matrix construction technique of [51] (Eq. (5)) for the construction of an invertible dynamic, flexible, integer diffusion matrix M for $n = 8$.

based on four sub-matrices (A , B , C , and D), can be proven as follows:

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{6}$$

The determinant of M is given by Eq. (7):

$$\begin{aligned} \det(M) &= \det(A) \times \det(D - CA^{-1}B) \\ &= \det(A) \times \det(D - CBA^{-1}) \\ &= \det(A) \times \det(A - A^2A^{-1} + I_m^2 \times A^{-1}) \\ &= \det(A) \times \det(A - A + A^{-1}) \\ &= \det(A) \times \det(A^{-1}) \\ &= \det(A \times A^{-1}) \\ &= \det(I_m) \\ &= 1 \end{aligned} \tag{7}$$

where $C \times B = (A^2 - I_m^2)$; $A^2 \times A^{-1} = A$; and $A^{-1}I_m^2 = A^{-1}$. Therefore, the necessary condition to have an inverse matrix is satisfied. Fig. 1 illustrates an example of the different elements that are used to build an invertible dynamic integer matrix D , for $n = 8$. Note that this diffusion matrix is limited to one sub-matrix (A).

The corresponding inverse integer matrix form, M^{-1} , is obtained based on the inverse rule and is defined in Eq. (8):

$$M^{-1} = \begin{bmatrix} D & -B \\ -C & A \end{bmatrix} \tag{8}$$

Therefore, M^{-1} can be obtained according to Eq. (9):

$$M^{-1} = \begin{bmatrix} A & -(A + I_m) \\ -(A - I_m) & A \end{bmatrix} \tag{9}$$

We adopt the integer form (M) to produce a binary diffusion matrix (BM), that can be implemented in the binary Galois field. The binary form inherits the same properties of the integer form in terms of flexibility, invertibility, and dynamicity. This approach yields a better efficiency due to the reduced computational complexity. Also, the lower number of the required clock cycles increases the throughput and reduces the energy consumption.

The binary diffusion matrix (BM) can be obtained by:

- Replacing the addition and subtraction arithmetic operations with the logical operation Exclusive-OR (XOR);

Table 2
Table of notation.

Symbol	Definition
n	Dimension of the diffusion matrix
$x \cdot y$	Integer matrix multiplication between two matrices x and y
$x \odot y$	The proposed adapted binary matrix multiplication between two matrices x and y
\oplus	The bitwise addition, modulo 2, of two bit strings of equal length
BN	Branch number criterion of a $n \times n$ diffusion matrix
FP	Fixed points
BM	Invertible binary diffusion matrix
NBM	Non-invertible binary diffusion matrix
M_u	Matrix of size $m \times l$, where $l = n - m$. M_u is a sub-matrix that replaces the parameter u of the 2×2 diffusion matrix.
M_v	Matrix of size $l \times m$. M_v is a sub-matrix that replaces the parameter v of the 2×2 diffusion matrix.
I_m	Identity matrix of size $m \times m$
I_l	Identity matrix of size $l \times l$

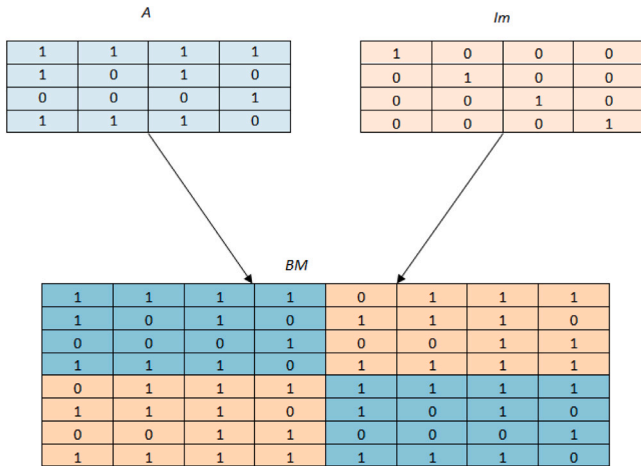


Fig. 2. An example of the binary diffusion matrix construction technique of [51] for a flexible invertible dynamic binary diffusion matrix M for $n = 8$.

- Replacing the integer sub-matrix A with a binary one.

Then, the binary matrix form becomes as in Eq. (10):

$$BM = BM^{-1} = \begin{bmatrix} A & A \oplus I_m \\ A \oplus I_m & A \end{bmatrix} \quad (10)$$

The determinant of the binary matrix form is expressed in Eq. (11):

$$\begin{aligned} \det(BM) &= \det(A) \wedge \det(D \oplus C \wedge A^{-1} \wedge B) \\ &= \det(A) \wedge \det(D \oplus C \wedge B \wedge A^{-1}) \\ &= \det(A) \wedge \det(A \oplus A^2 \wedge A^{-1} \oplus I_m^2 \wedge A^{-1}) \\ &= \det(A) \wedge \det(A \oplus A \oplus A^{-1}) \\ &= \det(A) \wedge \det(A^{-1}) \\ &= \det(A \wedge A^{-1}) \\ &= \det(I_m) \\ &= 1 \end{aligned} \quad (11)$$

where $C \wedge B = (A^2 \oplus I_m^2)$; $A^2 \wedge A^{-1} = A$; $A \wedge A^{-1} = I_m$; and $A^{-1} \wedge I_m^2 = A^{-1}$.

The calculation of the inverse binary matrix, BM^{-1} , is described in [56]. BM^{-1} is equal to BM based on the inverse rule of Eq. (8). Note that the determinant in the binary field is equal to the determinant in the integer field modulo 2. In Fig. 2, an example of different binary elements (A, I_m) is used to build the invertible dynamic binary matrix BM for $n = 8$.

The binary mixing process is described in Algorithm 1, using a binary diffusion matrix BM , and an input data vector $Z = \{z_1, z_2, \dots, z_n\}$ with length n .

Algorithm 1 Binary diffusion algorithm

```

1: procedure BINARY_MIXING( $Z, BM$ )
2:    $n \leftarrow \text{length}(Z)$ 
3:    $x \leftarrow 0$ 
4:   for  $j \leftarrow 1$  to  $n$  do
5:     if  $BM_{i,j} \neq 0$  then
6:        $x \leftarrow x \oplus z_i$ 
7:     end if
8:   end for
9:   return  $x$ 
10: end procedure

```

3.4. Related dynamic non-invertible diffusion matrices

For the non-invertible form, $\det(M) = 0 \Rightarrow a \times d = b \times c$. We consider that c is equal to a which leads to $d = b$ as presented in [57]. Then, the form of the non-invertible diffusion Matrix (NM) becomes as expressed in Eq. (12):

$$NM = \begin{bmatrix} a & b \\ a & b \end{bmatrix} \quad (12)$$

Hence, the non-invertible diffusion matrix requires two parameters (a, b) compared to the invertible form. Next, we replace a and b by the sub matrices A and B , to form a non-invertible flexible diffusion matrix, with a dimension equal to n , as expressed in Eq. (13):

$$NM = NBM = \begin{bmatrix} A & B \\ A & B \end{bmatrix} \quad (13)$$

This form can be either integer or binary, without requiring any additional modifications compared to the invertible form. We either use values for A and B from an integer Galois field to generate (NM), or we generate a binary non-invertible diffusion matrix (NBM) using binary values $\{0, 1\}$. The determinant of NM or NBM is always equals to 0, which ensures the non-invertibility property. Fig. 3 illustrates an example of different binary elements (A, B) that are used to build a non-invertible dynamic binary matrix NBM for $n = 8$.

4. Construction technique of dynamic, flexible, integer & binary diffusion matrices

In this section, and to overcome the limitations of the solution presented in the previous section, we propose and describe a new diffusion matrix form. Different diffusion forms are obtained by converting the integer diffusion matrix forms in [16,19] to the binary field. First, we start by describing the integer diffusion form and then, we present the conversion technique, and we explain the process for enhancing the cryptographic performance.

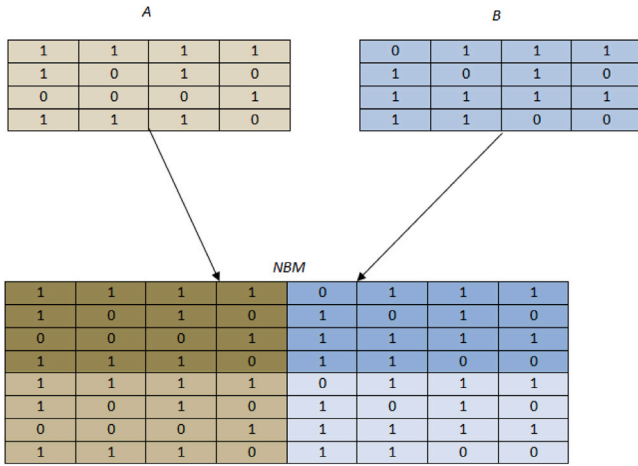


Fig. 3. The binary adaptation of a non-invertible matrix as constructed in [57], for a flexible diffusion matrix M with $n = 8$.

4.1. Primary dynamic invertible integer diffusion matrices forms

The proposed scheme, for the generation of integer diffusion forms, is presented in [16,19]. As described previously, the dynamic invertible diffusion matrix has a determinant equals to ± 1 , whereas, the non-invertible diffusion matrix has a determinant of 0.

The invertible diffusion matrices are generated with a determinant value of 1 (non-singular matrix), in order to avoid the floating-point operation that results from the division operation. These forms also take into consideration the requirements needed for simple hardware and software implementations. The goal is to leverage the forms' characteristics in the integer field, and to map them into the binary field. Next, we describe the scheme for generating the four flexible, invertible and key-dependent diffusion matrices M of [16,19]. To this end, we adopt the determinant rule, $ad = 1 + bc$, and we fix one parameter to ± 1 . For example, if $a = 1 \rightarrow d = \pm 1 + bc$, which yields into the following matrix form having dimensions of (2×2) , as presented in Eq. (14):

$$A_1 = \begin{bmatrix} 1 & b \\ c & 1 + b \times c \end{bmatrix} \quad (14)$$

To simplify the representation, two variables (u, v) are used to express the matrix forms, as illustrated in Eqs. (15)–(18):

$$A_1 = \begin{bmatrix} 1 & u \\ v & 1 + u \times v \end{bmatrix} \quad (15)$$

$$A_2 = \begin{bmatrix} u & 1 \\ 1 + u \times v & v \end{bmatrix} \quad (16)$$

$$A_3 = \begin{bmatrix} 1 + u \times v & v \\ u & 1 \end{bmatrix} \quad (17)$$

$$A_4 = \begin{bmatrix} v & 1 + u \times v \\ 1 & u \end{bmatrix} \quad (18)$$

Moreover, the diffusion matrix should have a flexible dimension n , to enable its extension according to the size of the input block. Also, the diffusion matrix should satisfy the variable structure (dynamicity), which we consider as a principal condition for modern diffusion primitives.

The four diffusion integer matrix forms M_1, M_2, M_3 , and M_4 with size $(n \times n)$ are built from the four structures, A_1, A_2, A_3 , and A_4 , respectively. These new forms are presented in Eqs. (19)–(22), respectively:

$$M_1 = \begin{bmatrix} I_m & Mu \\ Mv & I_l + Mu \cdot Mv \end{bmatrix} \quad (19)$$

$$M_2 = \begin{bmatrix} Mu & I_m \\ I_l + Mu \cdot Mv & Mv \end{bmatrix} \quad (20)$$

$$M_3 = \begin{bmatrix} I_l + Mu \cdot Mv & Mv \\ Mu & I_m \end{bmatrix} \quad (21)$$

$$M_4 = \begin{bmatrix} Mv & I_l + Mu \cdot Mv \\ I_m & Mu \end{bmatrix} \quad (22)$$

In these formulations, the elements u and v are replaced by the sub matrices Mu and Mv , respectively, to form these flexible and dynamic primary integer matrices, each having a dimension of n . The “ \cdot ” represents the integer matrix multiplication as described in [58,59]. Two sub-matrices Mu and Mv are required to form each diffusion matrix form.

The determinant of the various matrices $M_w (w = 1, 2, 3, 4)$ is equal to ± 1 , hence, these matrices are invertible. I_m and I_l are two identity matrices of size m and l , respectively. Mu and Mv are two non-zero matrices of size $(m \times l)$ and $(l \times m)$, respectively, with $l = n - m$. The elements of Mu and Mv can be freely chosen from any Galois field such that M_w is full rank.

For enhanced cryptographic performance, different Mu and Mv are preferred for the various structures, which increases the key space of the diffusion matrix. However, using different M_u and M_v , for the various structures, requires the generation of more key-streams, which results into additional computational complexity. The invertibility of each matrix can be proven by calculating its determinant (Eq. (7)). For example, when $M = M_1$, the determinant is obtained according to Eq. (23):

$$\begin{aligned} \det(M_1) &= \det(I_m) \times \det(I_l + M_{v1}M_{u1} - M_{v1}I_l^{-1}M_{u1}) \\ &= \det(I_m) \times \det(I_l + M_{v1}M_{u1} - M_{v1}M_{u1}) \\ &= \det(I_m) \times \det(I_l) = 1 \end{aligned} \quad (23)$$

As a result, each integer diffusion matrix has an inverse matrix, M_w^{-1} , that can be obtained by using Eq. (8), and as expressed in Eqs. (24)–(27):

$$M_1^{-1} = \begin{bmatrix} I_l + Mv \cdot Mu & -Mu \\ -Mv & I_m \end{bmatrix} \quad (24)$$

$$M_2^{-1} = \begin{bmatrix} Mv & -I_m \\ -(I_l + Mv \cdot Mu) & Mu \end{bmatrix} \quad (25)$$

$$M_3^{-1} = \begin{bmatrix} I_m & -Mv \\ -Mu & I_l + Mv \cdot Mu \end{bmatrix} \quad (26)$$

$$M_4^{-1} = \begin{bmatrix} Mu & -(I_l + Mv \cdot Mu) \\ -I_m & Mv \end{bmatrix} \quad (27)$$

4.2. Adaptation of the invertible binary diffusion matrices forms to the binary galois field

The technique proposed to build a final binary diffusion matrix is based on several steps, where the first step is mapping the integer diffusion matrices of Section 4 into binary. Also, we employ the minimum and necessary number of different diffusion forms to provide the maximum possible cryptographic performance.

From previous section, the binary diffusion matrix (BM_w) is adapted from the integer form (M_w), and inherits its properties: flexibility, invertibility, and dynamicity. This approach ensures a better cryptographic performance when compared to the previous binary solution (described in Section 3.3) and it exhibits a better efficiency when compared to the integer proposition of [19].

The binary diffusion form ($BM_w, w = 1, 2, 3, 4$) is obtained by applying four modifications:

- (1) Replacing the addition and subtraction operations by the logical Exclusive-OR (\oplus) operation;
- (2) Replacing the multiplication operation by the logical “AND” operation (\wedge);
- (3) Performing the matrix multiplication in the binary Galois field, using the first and second modifications, and as expressed in Algorithm 2;

- (4) Employing binary sub-matrices (Mu and Mv) instead of integer ones.

Algorithm 2 Proposed Binary Matrix Multiplication Algorithm

```

1: procedure  $Z = \text{MOD\_MAT\_BINARY\_MULT}(X, Y)$ 
2:    $[m, n] = \text{size}(X)$ 
3:    $q = \text{NumberofColumn}(Y)$ 
4:   for  $i \leftarrow 1$  to  $m$  do
5:     for  $j \leftarrow 1$  to  $q$  do
6:        $tmp \leftarrow 0$ 
7:       for  $w \leftarrow 1$  to  $n$  do
8:          $tmp \leftarrow tmp \oplus (X(i, k) \wedge Y(k, j))$ 
9:       end for
10:       $Z(i, j) \leftarrow tmp$ 
11:    end for
12:  end for
13:  return  $Z$ 
14: end procedure
  
```

The obtained binary diffusion forms are illustrated in Eqs. (28)–(31):

$$BM_1 = \begin{bmatrix} I_m & Mu \\ Mv & I_l \oplus (Mu \odot Mv) \end{bmatrix} \quad (28)$$

$$BM_2 = \begin{bmatrix} Mu & I_m \\ I_l \oplus (Mu \odot Mv) & Mv \end{bmatrix} \quad (29)$$

$$BM_3 = \begin{bmatrix} I_l \oplus (Mu \odot Mv) & Mv \\ Mu & I_m \end{bmatrix} \quad (30)$$

$$BM_4 = \begin{bmatrix} Mv & I_l \oplus (Mu \odot Mv) \\ I_m & Mu \end{bmatrix} \quad (31)$$

where \odot represents the proposed adapted binary matrix multiplication over the Galois binary field as described in Algorithm 2. The calculation of the inverse binary matrix BM_w^{-1} is described in [56], where the determinant of the binary diffusion matrix is 1. Note that, in order to calculate the determinant in binary field, the previously mentioned modifications should also be applied. The corresponding inverse diffusion matrix forms are given in Eqs. (32)–(35):

$$BM_1^{-1} = \begin{bmatrix} I_l \oplus (Mv \odot Mu) & Mu \\ Mv & I_m \end{bmatrix} \quad (32)$$

$$BM_2^{-1} = \begin{bmatrix} Mv & I_m \\ I_l \oplus (Mv \odot Mu) & Mu \end{bmatrix} \quad (33)$$

$$BM_3^{-1} = \begin{bmatrix} I_m & Mv \\ Mu & I_l \oplus (Mv \odot Mu) \end{bmatrix} \quad (34)$$

$$BM_4^{-1} = \begin{bmatrix} Mu & I_l \oplus (Mv \odot Mu) \\ I_m & Mv \end{bmatrix} \quad (35)$$

BM_i and BM_i^{-1} , $i = 1, 2, 3, 4$ always have determinants equal to one.

In the following, we describe the proposed technique to construct diffusion matrices based on two primary diffusion matrices. The final binary diffusion matrix is invertible if the both primary binary diffusion matrices are invertible. If only one of the primary diffusion matrices is invertible, the final diffusion matrix is non-invertible.

4.3. Constructing technique of the final binary invertible diffusion matrices

A diffusion matrix that is based on a single diffusion form cannot provide maximum performance as we will prove in Section 6. For this reason, an extended analysis is done to quantify the minimum number of different primary diffusion forms necessary to provide the maximum cryptographic performance. In fact, the construction technique of the final binary diffusion matrix form necessitates the detection of the required number of different primary diffusion matrices BM_w that should be mixed by using the proposed binary matrix multiplication operation (presented in Algorithm 2) to produce the final diffusion

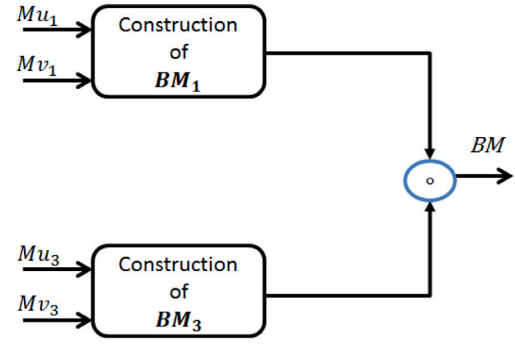


Fig. 4. Architecture of the proposed invertible diffusion matrix construction using two different invertible primary diffusion forms ($BM = BM_1 \odot BM_3$).

matrix $BM = \odot_{w=1}^j BM_w$, where j can be 2, 3, or 4. The required number of matrix multiplication is $j - 1$.

Moreover, to produce a primary diffusion matrix, we apply the following two steps:

- (1) Construction of two different binary sub-matrices (Mu_w and Mv_w), in addition to two identity matrices I_l and I_m to form one primary diffusion matrix BM_w , $w = 1, 2, 3, 4$.
- (2) A part of BM_w requires the binary matrix multiplication between two sub-matrices ($Mv_w \odot Mu_w$), and addition of an identity matrix IL to the output using the logical operation (\oplus). Then, these components are used to construct the corresponding primary diffusion matrix BM_w .

In Fig. 4, we show the process for the construction of two primary diffusion matrix forms, with different sub-matrices, that are used to produce a final binary diffusion matrix based on the adopted binary matrix multiplication.

According to the obtained results (Figs. 9 and 10), we need to select two pairs of matrix forms. We recommend the use of (BM_1, BM_3) or (BM_2, BM_4) for the generation of the final binary diffusion matrix, which yields better results than other pairs.

In summary, the expression of the final diffusion matrix BM is given by Eq. (36), when the selected pair is (BM_1, BM_3) , and by Eq. (37), when the selected pair is (BM_2, BM_4) :

$$BM = \odot_{w=0}^1 BM_{2 \times w+1} = BM_1 \odot BM_3 \quad (36)$$

$$BM = \odot_{w=1}^2 BM_{2 \times w} = BM_2 \odot BM_4 \quad (37)$$

Fig. 4 illustrates the architecture of the proposed approach for building the final invertible diffusion matrices. It uses two different primary forms to build the final diffusion matrix, which are multiplied using the proposed binary matrix multiplication \odot . An example of such construction of BM is shown in Fig. 5, for $n = 8$, where $l = m = \frac{n}{2} = 4$.

Moreover, the inverse of the final diffusion matrix BM^{-1} is obtained using the binary matrix multiplication of the inverse matrices of the corresponding primary matrix forms. For $BM = BM_1 \odot BM_3$, BM^{-1} is obtained according to Eq. (38):

$$BM^{-1} = \odot_{w=0}^1 BM_{2 \times w+1}^{-1} = BM_1^{-1} \odot BM_3^{-1} \quad (38)$$

For $BM = BM_2 \odot BM_4$, BM^{-1} is obtained according to Eq. (39):

$$BM^{-1} = \odot_{w=1}^2 BM_{2 \times w}^{-1} = BM_2^{-1} \odot BM_4^{-1} \quad (39)$$

Another advantage of the proposed construction technique is if the fact that the inverse BM^{-1} is obtained in a similar manner to BM . Moreover, the inverse matrix operation is not required according to Eq. (32). We use the same sub-matrices and operations to form the primary diffusion inverse matrix form, which in turn reduces the latency of diffusion primitives construction at the decryption side.

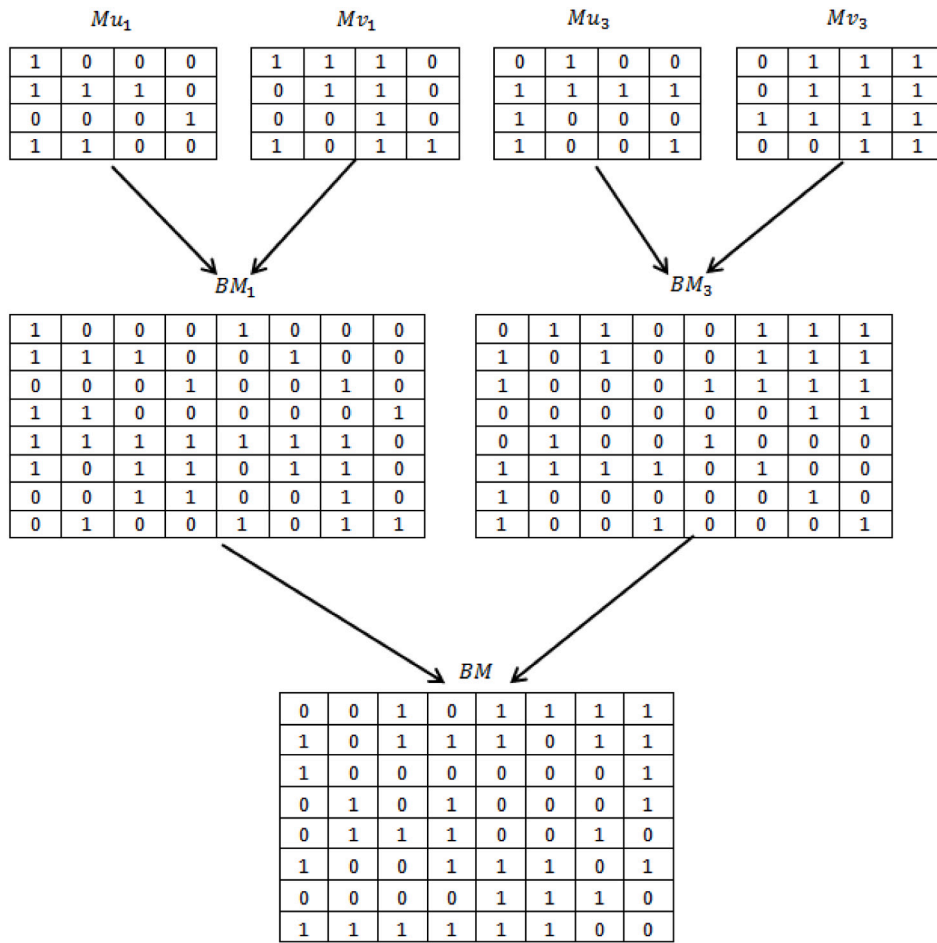


Fig. 5. A construction example of the invertible binary diffusion matrix for $n = 8$.

In Fig. 7, we show two numerical examples of binary diffusion matrices with maximum possible branch number, which is 5 for $n = 12$, and 6 for $n = 16$. This confirms the ability to generate a variable binary diffusion matrix with the maximum desirable diffusion properties.

5. Proposed non-invertible integer and binary diffusion matrices

A non-invertible diffusion matrix guarantees the one-way property, which is a principal condition for several cryptographic algorithms such as hash functions, stream ciphers and key derivation functions. Such a matrix, NBM , will be generated after defining the non-invertible primary integer matrix forms. These will be adapted in a similar manner to the invertible case, as described in Section 4.2.

5.1. Primary dynamic non-invertible integer diffusion matrices forms

First, the construction of non-invertible integer diffusion matrices will be carried out for a size (2×2) , which should have a determinant equals to 0 (singular matrix). Thus, $\det(A) = 0$, then $ad = bc$. To obtain such a form, we fix one parameter to ± 1 . For example, if $a = 1 \rightarrow d = \pm bc$, this results in one of 4 forms, as presented in Eq. (40):

$$A_1 = \begin{bmatrix} 1 & b \\ c & b \times c \end{bmatrix} \tag{40}$$

To simplify the representation, each non-invertible diffusion primary matrix form also has two variable parameters (u, v) that can

be different for different forms. This leads into 4 new forms (see Eqs. (41)–(44)):

$$NA_1 = \begin{bmatrix} 1 & u \\ v & uv \end{bmatrix} \tag{41}$$

$$NA_2 = \begin{bmatrix} u & 1 \\ uv & v \end{bmatrix} \tag{42}$$

$$NA_3 = \begin{bmatrix} uv & v \\ u & 1 \end{bmatrix} \tag{43}$$

$$NA_4 = \begin{bmatrix} v & uv \\ 1 & u \end{bmatrix} \tag{44}$$

This formulation is illustrated again below where u and v are replaced with the sub matrices Mu and Mv , respectively. Then, we obtain the flexible non-invertible primary diffusion matrix with dimension n .

NM_1, NM_2, NM_3 , and NM_4 are four primary flexible, non-invertible diffusion matrices, which are built using the previous four structures NA_1, NA_2, NA_3 , and NA_4 , respectively, and they are presented in Eqs. (45)–(48):

$$NM_1 = \begin{bmatrix} I_m & Mu \\ Mv & Mu \cdot Mv \end{bmatrix} \tag{45}$$

$$NM_2 = \begin{bmatrix} Mu & I_m \\ Mu \cdot Mv & Mv \end{bmatrix} \tag{46}$$

$$NM_3 = \begin{bmatrix} Mu \cdot Mv & Mv \\ Mu & I_m \end{bmatrix} \tag{47}$$

$$NM_4 = \begin{bmatrix} Mv & Mu \cdot Mv \\ I_m & Mu \end{bmatrix} \tag{48}$$

For example, if $M = NM_0$, the determinant of M_0 is given by Eq. (49):

$$\begin{aligned} \det(NM_0) &= \det(I_m) \times \det(M_{v0}M_{u0} - M_{v0}I_l^{-1}M_{u0}) \\ &= \det(I_m) \times \det(M_{v0}M_{u0} - M_{v0}M_{u0}) \\ &= \det(I_m) \times \det(Z) = 0 \\ &= 1 \times 0 = 0 \end{aligned} \quad (49)$$

where Z is a zero matrix and its determinant is equal to zero (since all elements are equal to zero). This means that the necessary condition to obtain the inverse matrix is not possible, and hence, retrieving the original input is not also possible.

5.2. Adaptation of the non-invertible binary diffusion matrices forms to the binary galois field

The proposed form for the non-invertible Binary Matrices NBM_i , $i = 1, 2, 3, 4$ can be obtained using the same modification that was described previously for the invertible ones. Moreover, the proposed dynamic, non-invertible, flexible, primary binary diffusion forms are presented below in Eqs. (50)–(53):

$$BNM_1 = \begin{bmatrix} I_m & Mu \\ Mv & (Mu \odot Mv) \end{bmatrix} \quad (50)$$

$$BNM_2 = \begin{bmatrix} Mu & I_m \\ (Mu \odot Mv) & Mv \end{bmatrix} \quad (51)$$

$$BNM_3 = \begin{bmatrix} (Mu \odot Mv) & Mv \\ Mu & I_m \end{bmatrix} \quad (52)$$

$$BNM_4 = \begin{bmatrix} Mv & (Mu \odot Mv) \\ I_m & Mu \end{bmatrix} \quad (53)$$

5.3. Constructing technique of the final binary non-invertible diffusion matrices

The experimental results are similar to the results of Figs. 9 and 10. Hence, the number of fixed points is very low, according to Fig. 10, when multiplying two primary forms as compared to a single primary diffusion matrix form, or when multiplying more than two primary forms. Similarly, a higher branch number is reached in this case.

As in the invertible case, we select the pairs (NBM_1, NBM_3) or (NBM_2, NBM_4) as they provide the best choice for the final non-invertible diffusion matrix. NBM is expressed by Eq. (54), if the selected pair is (NBM_1, NBM_3) , and by Eq. (55), if the selected pair is (NBM_2, NBM_4) :

$$NBM = \odot_{w=0}^1 NBM_{2 \times w+1} = NBM_1 \odot NBM_3 \quad (54)$$

$$NBM = \odot_{w=1}^2 NBM_{2 \times w} = NBM_2 \odot NBM_4 \quad (55)$$

Note that different binary sub-matrices $(Mu$ and $Mv)$ and $(Mu'$ and $Mv')$ are used to build both primary forms. Then, the same approach is used for the construction of the invertible diffusion matrix, except for the fact that the primary matrix forms are replaced with the non-invertible ones (Eq. (53)), as seen in Fig. 6.

6. Security analysis

In this section, we perform several security tests and we analyze their results, to assess the feasibility and strength of the proposed diffusion algorithm. The simulation experiments include BN , FP , in addition to SAC , and BIC .

Note that we used MATLAB (MATrix LABoratory) since it offers iterative analysis with a programming language that expresses matrix and array mathematics directly; this makes it the simulation environment of choice for our work. Furthermore, The experimental setup includes MATLAB version R2018a, and a DELL laptop machine Intel(R)

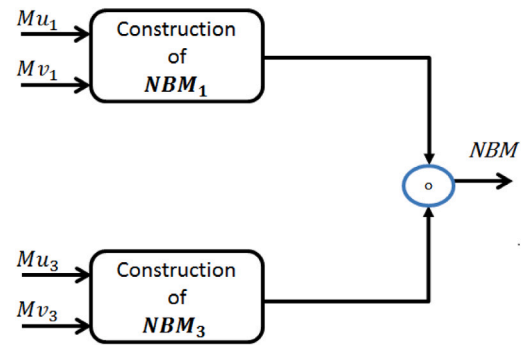


Fig. 6. Architecture of the construction technique of a non-invertible diffusion matrix using two different non-invertible primary diffusion forms ($NBM = NBM_1 \odot NBM_3$), using the adapted binary multiplication.

Core(TM) i7-7600U CPU @ 2.80 GHz (4 CPUs), running Windows 10, with OS memory of 16,266 MB RAM.

Next, we describe the used cryptographic primitives, and we analyze the cryptographic performance of the proposed scheme to identify the optimal parameters for the construction of diffusion matrices with a solid cryptographic level.

6.1. Cryptographic metrics

We quantify the following cryptographic metrics to assess the robustness of the generated binary diffusion matrices:

- (1) The branch number [60]
- (2) The number of the fixed points [61]
- (3) Strict Avalanche Criterion (SAC) and the Output Bit Independence Criterion (BIC) [62]

6.1.1. Branch number

The branch number of a diffusion matrix represents the diffusion rate, and it quantifies its security against traditional attacks such as linear and differential cryptanalysis. Also, it denotes the minimum number of active elements (different) for any two consecutive rounds. The diffusion matrix is a linear transformation that is represented as a matrix. Let n be the number of elements in a diffusion matrix D , where the size of each input and output element is m bits. Therefore, the diffusion matrix can be defined as $D : (\{0, 1\}^m)^n \rightarrow (\{0, 1\}^m)^n$.

The branch number $\beta(D)$ of an $(n \times n)$ diffusion matrix D is calculated according to Eq. (56):

$$\beta(D) = \min\{wt(x) + wt(D \cdot x^t)\} \quad | \quad x \neq 0 \quad (56)$$

where $(.)^t$ is the transposed matrix, $x = \{x_1, x_2, \dots, x_n\} \in (\{0, 1\}^m)^n$, $x_i \in \{0, 1\}^m$, $i = 1, 2, \dots, n$, and $wt(c)$ denotes the Hamming weight of a code word c (the number of nonzero bits in c).

6.1.2. Number of fixed points

The effect of the number of Fixed Points (FP) of the diffusion operation is described in [61]. When the number of fixed points in a linear transformation greatly exceeds the expected number of a random linear transformation, this indicates a poor diffusion of the transformation since the bits in these blocks are left unchanged when producing the output blocks. Note that the expected number of fixed points in a random permutation is one [61].

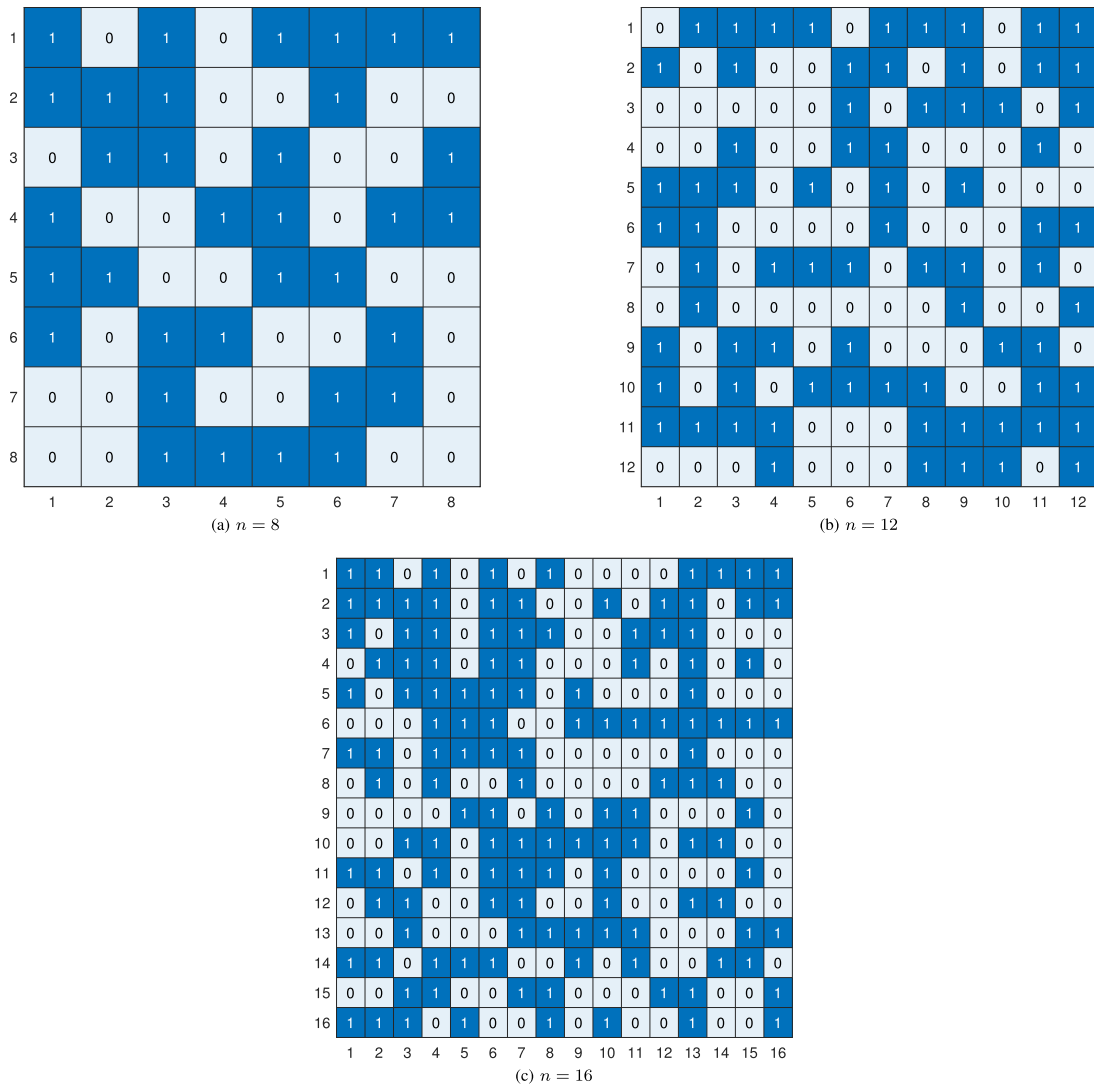


Fig. 7. A numerical representation of three binary final invertible matrices, for (a) $n = 8$, (b) $n = 12$, and (c) $n = 16$. The matrices have the maximum possible branch number, which is 4 for (a), 5 for (b) and 6 for (c).

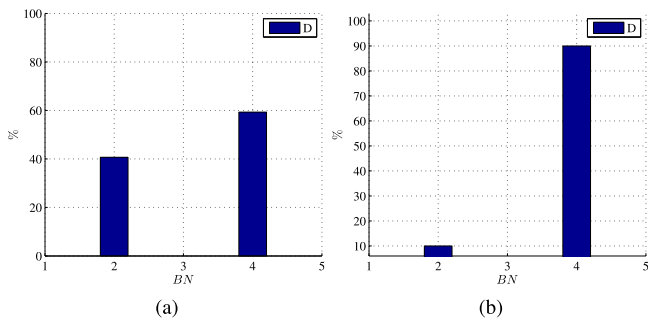


Fig. 8. Distribution of branch numbers for the diffusion form of [51]: (a) for $n = 8$ and (b) for $n = 16$, for 1000 randomly generated diffusion matrices.

6.1.3. Strict avalanche criterion (SAC)

The effect of the confusion and diffusion properties is known as the avalanche effect [63]. Webster and Tavares have defined the Strict Avalanche Criterion (SAC) [62] in order to quantify the avalanche effect of the substitution tables. In contrast, we have used it to quantify the robustness of the proposed diffusion matrix construction technique.

We perform the mapping between the input and diffused output elements. Each input is a vector of n bits, with values varying between 0 and $2^n - 1$; these will be diffused when generating a binary diffusion matrix. SAC is an essential characteristic, and a strong cryptographic primitive (confusion or diffusion) should satisfy this criterion, which states that, if one bit in the input block is changed, half of the output bits should change [5].

6.1.4. Output bit independence criterion (BIC)

BIC is another essential property that has also been described by Webster and Tavares [62]. BIC is used to validate the safe use of a substitution table or a non-linear transformation. The importance of the BIC criterion lies in the fact that it specifies the following condition: two output bits j, k should change independently [5], when a single input bit i is changed, for all i, j and k .

6.2. Cryptographic performance of the related binary diffusion matrix

In this section, we assess the cryptographic performance of the previous diffusion matrix forms presented in [51], and we highlight their limitations to justify the need for the new scheme. In this experiment, we generate 1000 random binary diffusion matrices, for ($n = 8$) and for

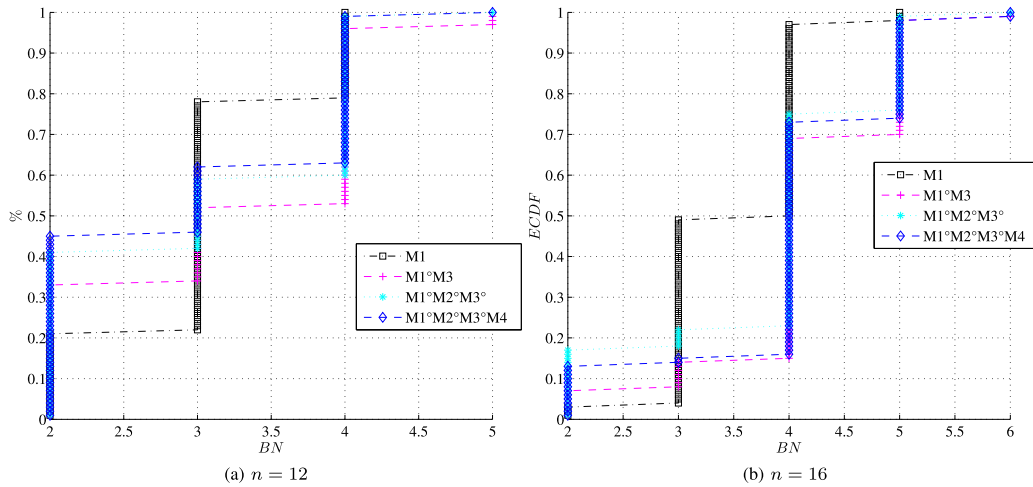


Fig. 9. Variation of the ECDF of the branch number for the produced diffusion using a single or several primary diffusion forms for (a) $n = 12$ and (b) $n = 16$.

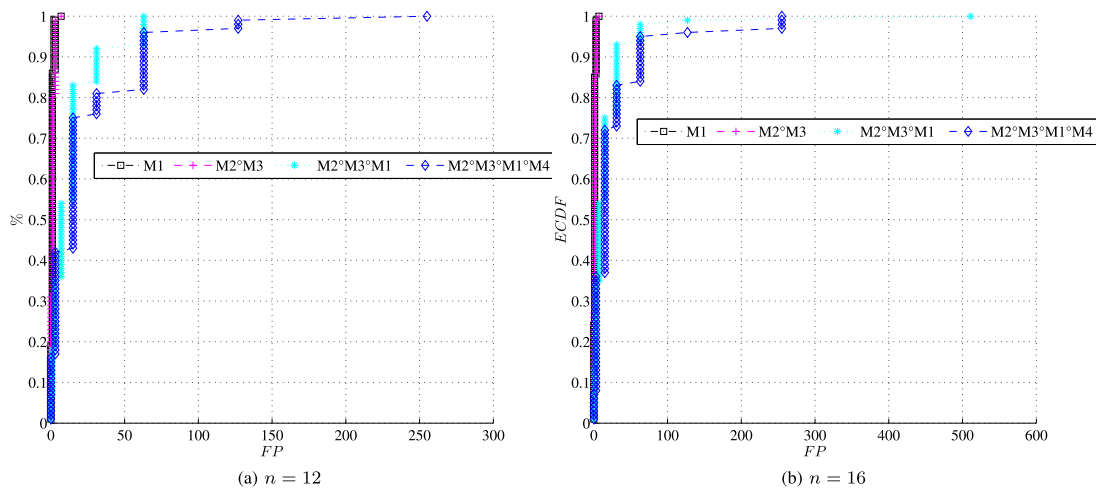


Fig. 10. Variation of the ECDF of the fixed points for the produced diffusion matrices using single or several primary diffusion forms for (a) $n = 12$ and (b) $n = 16$.

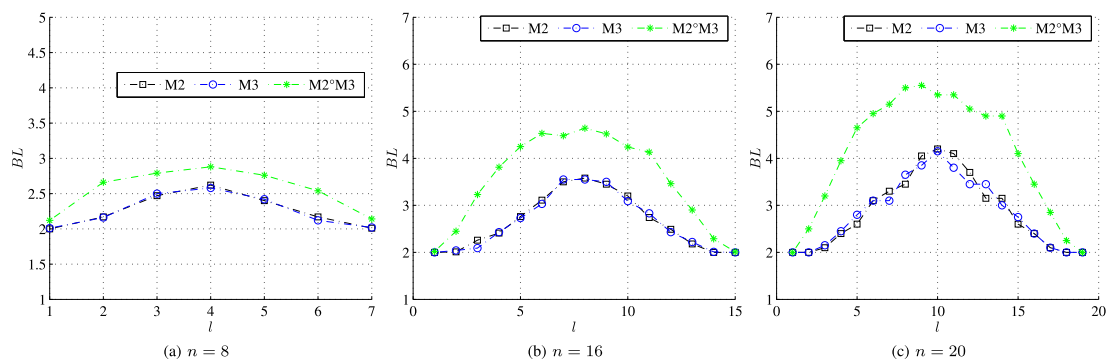


Fig. 11. The variation of the branch number for (a) $n = 8$, (b) $n = 12$, and (c) $n = 16$.

($n = 16$). We plot the distribution of the branch numbers in Fig. 8(a)–(b), respectively. The results indicate that with the increase of n , the percentage of the BN value of 4 becomes more significant, while the percentage of the lower branch number of value 2, decreases. The maximum value of BN remains 4 with higher values of n , which is considered as a hard limitation with this form.

Next, we show the percentage distribution of the FP for the same diffusion matrices, with $n = 8$ and $n = 16$, in Tables 3 and 4, respectively. The results indicate that the number of FP is not lowered, and the diffusion matrices have $BN \leq 4$. These results confirm the limited cryptographic performance of the solution. However, to the best of our knowledge, it is the first approach that provides the two important

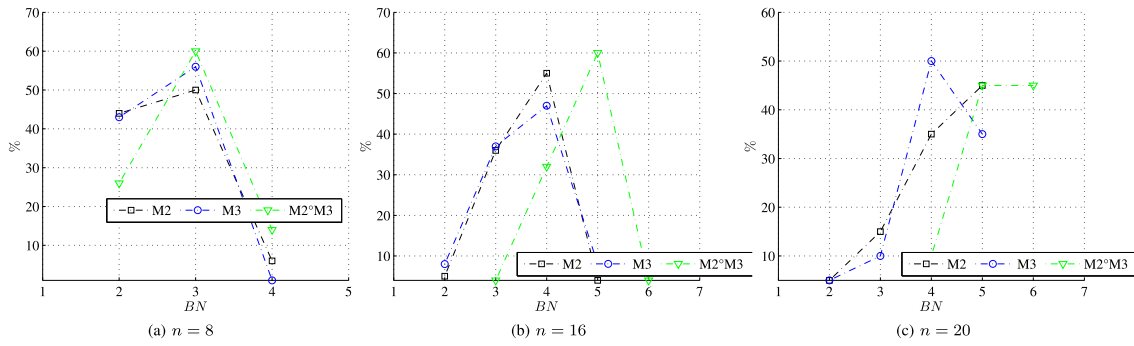


Fig. 12. Variation of the corresponding distribution of the branch number for different values of n , for $m = l = \frac{n}{2}$.

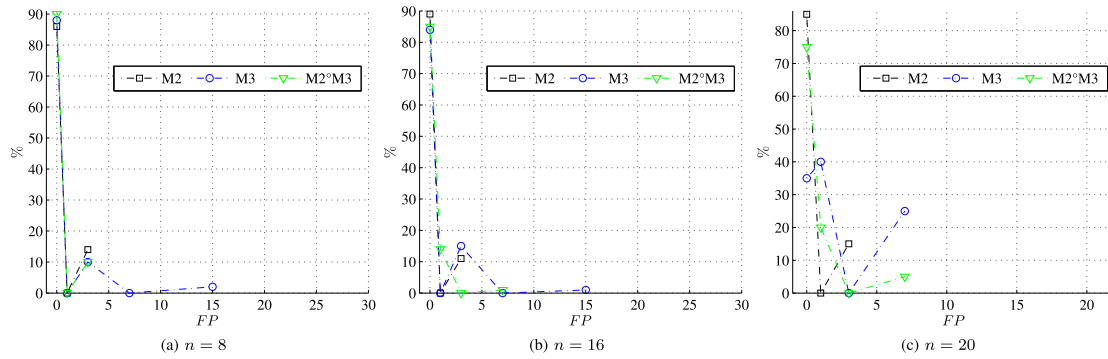


Fig. 13. Variation of the percentage of the number of fixed points for (a) $n = 8$, (b) $n = 12$, and (c) $n = 16$, with $m = l = \frac{n}{2}$.

Table 3

Percent distribution of the fixed points for 1000 random produced invertible binary diffusion matrices using the previous solution for $n = 8$.

FP	15	31	63	127
%	88.3	11.3	0.01	0.39

Table 4

Percent distribution of the fixed points for 1000 random produced invertible binary diffusion matrices using the previous solution for $n = 16$.

FP	255	511	1023	2047
%	88	11	0	1

properties of flexibility and dynamicity. Note that the results of BN or FP are variable, and they depend on the produced pseudo-random sub-matrices (M_u, M_v).

6.3. Identifying the required number of primary forms to construct the final diffusion matrix

In this experiment, we generate 1000 primary diffusion matrices for each form, for $n = 12$ and for $n = 16$. Then, we produce the final diffusion matrices using either a single form, 2, 3, or 4 forms via the proposed matrix multiplication ($BM = \odot_{w=1}^j BM_w, w = 1, 2, 3, 4$). We plot the Empirical Cumulative Distribution Function (ECDF) of BN and FP in Figs. 9 and 10.

The results show that using a single primary diffusion matrix form does not achieve a high branch number. The maximum BN is 5 for $n = 12$, and 6 for $n = 16$, and it is obtained when using more than one primary diffusion form. Moreover, close to 50% of the produced diffusion matrices have a $BN \geq 4$ for $n = 12$, and close to 84% of produced diffusion matrices have a $BN \geq 4$ for $n = 16$. Therefore, we propose to use two primary diffusion matrix forms to achieve the required cryptographic performance, as compared to a single primary matrix.

Also, this is efficient since the multiplication is applied just twice, which strikes a good balance between low computational complexity and good cryptographic performance.

6.4. Detecting the optimal size of primary matrix forms

In this experiment, we track the variation of BN versus l , for different sizes of n (8, 16 and 20). The results, shown in Fig. 11, indicate a symmetric variation; the maximum values of BN are reached for $m \in \{\frac{n}{2} - 1, \frac{n}{2}, \frac{n}{2} + 1\}$. The distribution of BN is shown in Fig. 12, for $m = l = \frac{n}{2}$. We can see that BN increases with the increase of n , and mixing two different primary diffusion matrices achieve a better BN compared to a single primary diffusion matrix form. Note that the maximum values of BN (4, 5, 6, and 7) are reached for $n = 8, 12, 16, 20$, respectively. Compared to Fig. 8, the branch number increases from 4 to 6, for $n = 16$, which indicates an enhanced diffusion property, when n is increased.

Similar result are obtained for the FP criterion whereby the variation of FP is symmetric, and it reaches its low bound, for $m \in \{\frac{n}{2} - 1, \frac{n}{2}, \frac{n}{2} + 1\}$, for different values of n . The distributions of FP for different sizes of n , and for $m = l = \frac{n}{2}$, is shown in Fig. 13. The results indicate clearly that the maximum number of FP is always low as n increases, and a numerical result is presented in Table 5. These results indicate that the optimal configuration to produce binary diffusion matrices, invertible and non-invertible, is for $m = l = \frac{n}{2}$.

Hence, this solution achieves the main objective of modern symmetric cryptography, and outperforms the solution presented in [51]. In summary, the obtained results range from acceptable to high values, and the latter are achievable.

Unlike traditional tests, SAC is applied on the output, the produced diffusion matrix, which is invertible and satisfies the bijectivity property. The output matrix could be considered as a substitution box, however, it does not have a nonlinear degree. In Fig. 14(a)–(b), we show the variation of the average SAC values (mean of $n \times n$ values of the dependence matrix) versus m , for $n = 8$ and 16, respectively. We

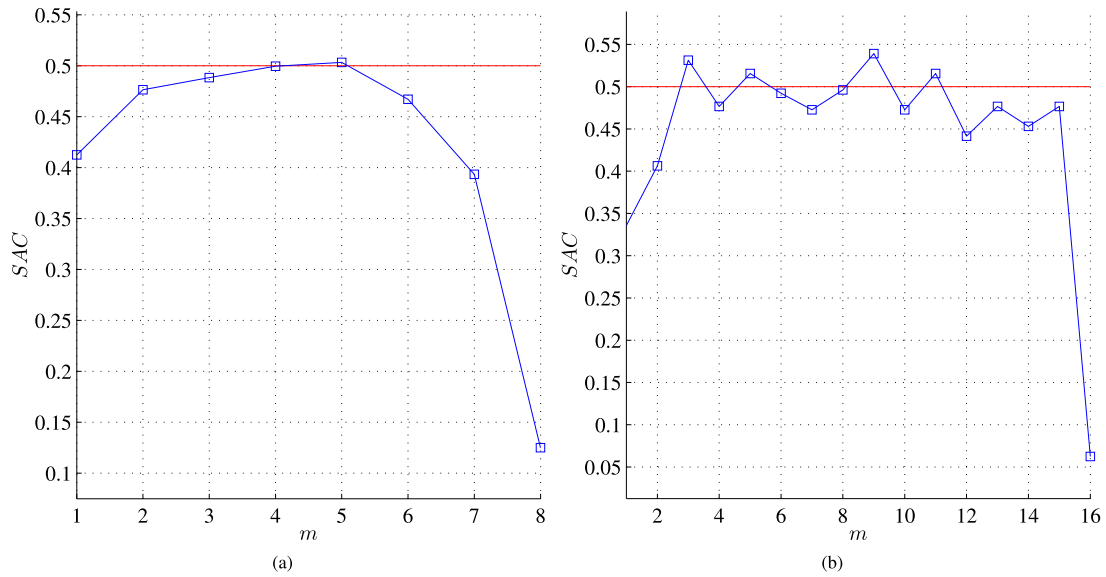


Fig. 14. Variation of the average of SAC versus m for (a) $n = 8$ and (b) $n = 16$.

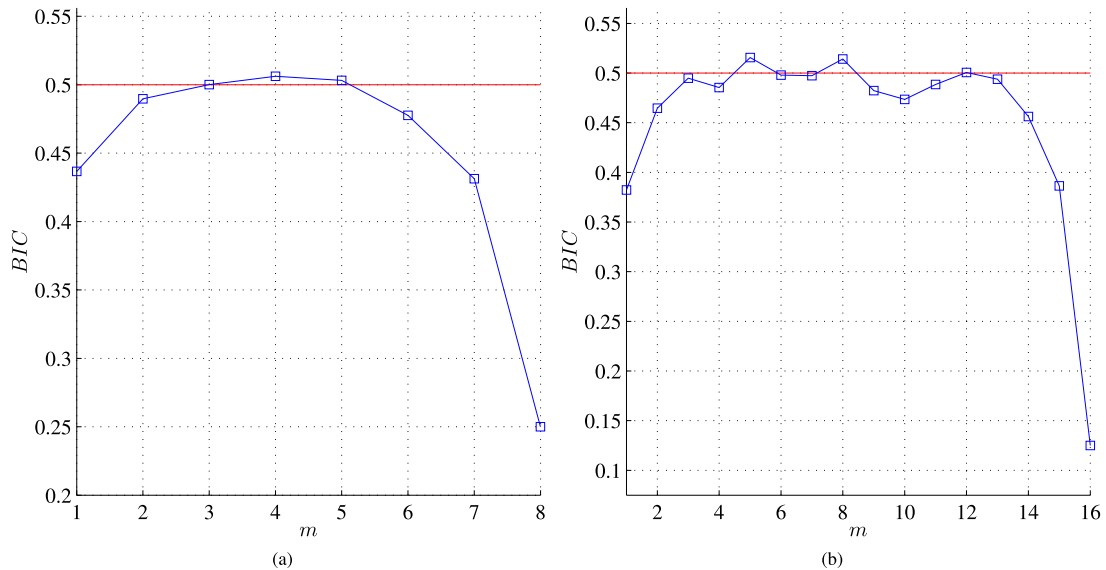


Fig. 15. Variation of the average of BIC versus m for (a) $n = 8$ and (b) $n = 16$.

Table 5

The percentage distribution of the fixed points for 1000 random produced invertible binary diffusion matrices by using the proposed construction technique for $n = 16$.

FP	0	1	3	7
%	87.9000	11.499	0.001	0.6

can observe that the SAC values converge closely to the ideal value of 0.5, for $m \in \{\frac{n}{2} - 1, \frac{n}{2}, \frac{n}{2} + 1\}$. These results are similar to, and thus, they confirm the results obtained for BN and FP.

Also, in this paper, BIC is used to find the optimal size of m to produce binary diffusion matrices with the desired cryptographic properties. The BIC test analyzes the relationship between the input vector and the output diffusion vector of an $(n \times n)$ binary diffusion matrix. We track the variation of the average values of BIC (the mean of $n \times n$ values of the BIC matrix without the diagonal elements) versus m , and we show the results in Fig. 15(a)–(b), for $n = 8$ and 16, respectively. It can be clearly shown that the BIC values are very close to the ideal value of $\frac{1}{2}$, for $m \in \{\frac{n}{2} - 1, \frac{n}{2}, \frac{n}{2} + 1\}$, which is similar to the

results of SAC. Accordingly, m can be set to $\frac{n}{2}$ as it provides acceptable cryptographic properties.

7. Conclusion

In this paper, we followed the dynamic key-dependent approach, and we presented several dynamic, key-dependent primary diffusion matrix forms. These forms are derived using specific algebraic operations that are simple, and yet they ensure the diffusion property. These matrix forms are adapted from the integer field into the binary Galois field, by modifying the matrix multiplication operation. The advantages of the proposed matrix forms are related to their flexibility, dynamicity, and the support of both invertible and non-invertible matrices. To the best of our knowledge, this work presents the first technique to construct key-dependent binary diffusion matrices that exhibit low computational complexity and good cryptographic performance. The obtained results proved that the maximum cryptographic performance is achieved when $m \in \{\frac{n}{2} - 1, \frac{n}{2}, \frac{n}{2} + 1\}$. Then, we proposed a scheme to

enhance the cryptographic performance of the produced binary diffusion matrices by multiplying together different primary matrix forms. Also, the results indicated clearly that using only two different primary binary diffusion forms is sufficient to strike a good balance between cryptographic performance and efficiency; this makes the proposed solutions appropriate for a dynamic SCA.

CRedit authorship contribution statement

Hassan N. Noura: Software, Writing – original draft, Writing – reviewing and editing, Conceptualization, Methodology. **Ali Chehab:** Validation, Writing – original draft, Reviewing and editing, Conceptualization, Methodology.

Data availability

No data was used for the research described in the article.

Funding

The EIPHI Graduate School (contract “ANR-17-EURE-0002”) provided funding for this work.

References

- [1] Eom Sungwook, Huh Jun-Ho. Group signature with restrictive linkability: minimizing privacy exposure in ubiquitous environment. *J Ambient Intell Humaniz Comput* 2018;1–11.
- [2] Menezes AJ, Van Oorschot PC, Vanstone SA. Handbook of applied cryptography. 1997, cited By (since 1996)5122.
- [3] Stallings William. Cryptography and network security: principles and practice. NJ: Pearson Upper Saddle River; 2017.
- [4] Fiestel H. Cryptography and computer privacy, Vol. 228. *Scientific American*; 1973, p. 15–23.
- [5] Noura Hassan N, Salman Ola, Kaaniche Nesrine, Sklavos Nicolas, Chehab Ali, Couturier Raphaël. Tresc: Towards redesigning existing symmetric ciphers. *Microprocess Microsyst* 2020;103478.
- [6] Li Xiaodan, Wu Wenling. Constructing binary matrices with good implementation properties for low-latency block ciphers based on lai-massey structure. *Comput J* 2021.
- [7] Rishakani Akbar Mahmoodi, Mirzaee Shamsabad Mohammad Reza, Dehnavi Seyed Mojtaba, Amiri Mohammad Amin, Maimani Hamidreza, Bagheri Nasour. Lightweight 4x4 mds matrices for hardware-oriented cryptographic primitives. *The ISC Int J Inf Secur* 2019;11(1):35–46.
- [8] Yang Yumeng, Zeng Xiangyong, Wang Shi. Construction of lightweight involutory mds matrices. *Des Codes Cryptogr* 2021;89(7):1453–83.
- [9] Pehlivanoğlu Meltem Kurt, Büyüksaraçoğlu Sakallı Fatma, Sakallı Muharerem Tolga. On the construction of low-latency 32 x32 binary mds matrices from ghadamard matrices. *Int J Inf Secur Sci* 2021;10(4):111–8.
- [10] Aslan Bora, Sakallı Muharerem Tolga. Algebraic construction of cryptographically good binary linear transformations. *Secur Commun Netw* 2014;7(1):53–63.
- [11] Schneier Bruce. Description of a new variable-length key, 64-bit block cipher (blowfish). In: Anderson Ross, editor. *Fast software encryption. Lecture notes in computer science*, Vol. 809, Springer Berlin Heidelberg; 1994, p. 191–204.
- [12] Noura Hassan N, Salman Ola, Couturier Raphaël, Chehab Ali. Lorca: Lightweight round block and stream cipher algorithms for iov systems. *Veh Commun* 2021;100416.
- [13] Noura Hassan N, Noura Mohamad, Chehab Ali, Mohammad M Mansour, Raphaël Couturier. Efficient and secure cipher scheme for multimedia contents. *Multimedia Tools Appl* 2019;78:14837–66.
- [14] Noura Hassan, Chehab Ali, Noura Mohamad, Couturier Raphaël, Mansour Mohammad M. Lightweight, dynamic and efficient image encryption scheme. *Multimedia Tools Appl* 2018;1–35.
- [15] Noura Hassan, Sleem Lama, Noura Mohamad, Mansour Mohammad M, Chehab Ali, Couturier Raphaël. A new efficient lightweight and secure image cipher scheme. *Multimedia Tools Appl* 2017;1–28.
- [16] Noura Hassan. Design and simulation of efficient chaos based generators, crypto-systems and hash functions (Ph.D. thesis), 2012.
- [17] Noura Hassan N, Chehab Ali, Couturier Raphael. Efficient & secure cipher scheme with dynamic key-dependent mode of operation. *Signal Process, Image Commun* 2019;78:448–64.
- [18] Dai A, Kim C, Kim J. Invertibility probability of binary matrices.
- [19] Noura Hassan, Steven Martin, Agha Khaldoun Al. E3SN - efficient security scheme for sensor networks. In: *SECRYPT - 10th international conference on security and cryptography*. Reykjavik, Iceland; 2013.
- [20] Noura Hassan, Chehab Ali, Sleem Lama, Noura Mohamad, Couturier Raphaël, Mansour Mohammad M. One round cipher algorithm for multimedia iot devices. *Multimedia Tools Appl* 2018;77(14):18383–413.
- [21] McKay Kerry A, Bassham Larry, Turan Meltem Sönmez, Mouha Nicky. Report on lightweight cryptography. Nist draft nistir, 8114, 2016.
- [22] Poschmann Axel York. Lightweight cryptography: cryptographic engineering for a pervasive world (Ph.D. thesis), Citeseer; 2009.
- [23] Guo Jian, Peyrin Thomas, Poschmann Axel. The photon family of lightweight hash functions. In: *Annual cryptography conference*. Springer; 2011, p. 222–39.
- [24] Guo Jian, Peyrin Thomas, Poschmann Axel, Robshaw Matt. The led block cipher. In: *International workshop on cryptographic hardware and embedded systems*. Springer; 2011, p. 326–41.
- [25] Karakoç Ferhat, Demirci Hüseyin, Emre Harmancı A. Itubee: a software oriented lightweight block cipher. In: *International workshop on lightweight cryptography for security and privacy*. Springer; 2013, p. 16–27.
- [26] Zhang Wentao, Bao Zhenzhen, Lin Dongdai, Rijmen Vincent, Yang Bohan, Verbauwhe Ingrid. Rectangle: a bit-slice lightweight block cipher suitable for multiple platforms. *Sci China Inf Sci* 2015;58(12):1–15.
- [27] Karakoç Ferhat, Demirci Hüseyin, Harmancı AE. Akf: A key alternating feistel scheme for lightweight cipher designs. *Inform Process Lett* 2015;115(2):359–67.
- [28] Beaulieu Ray, Shors Douglas, Smith Jason, Treatman-Clark Stefan, Weeks Bryan, Wingers Louis. Simon and speck: Block ciphers for the internet of things. *IACR Cryptol ePrint Arch* 2015;2015:585.
- [29] Yang Gangqiang, Zhu Bo, Suder Valentin, Aagaard Mark D, Gong Guang. The simeck family of lightweight block ciphers. In: *International workshop on cryptographic hardware and embedded systems*. Springer; 2015, p. 307–29.
- [30] Nalla Venu, Sahu Rajeev Anand, Saraswat Vishal. Differential fault attack on simeck. In: *Proceedings of the third workshop on cryptography and security in computing systems*, 2016 p. 45–8.
- [31] Patil Jagdish, Bansod Gaurav, Kant Kumar Shashi. Lici: A new ultra-lightweight block cipher. In: 2017 international conference on emerging trends & innovation in ICT (ICEI). IEEE; 2017, p. 40–5.
- [32] Bansod Gaurav, Pisharoty Narayan, Patil Abhijit. Boron: an ultra-lightweight and low power encryption design for pervasive computing. *Front Inf Technol Electron Eng* 2017;18(3):317–31.
- [33] Bogdanov Andrey, Knudsen Lars R, Leander Gregor, Paar Christof, Poschmann Axel, Robshaw Matthew JB, Seurin Yannick, Vikkelseo Charlotte. Present: An ultra-lightweight block cipher. In: *International workshop on cryptographic hardware and embedded systems*. Springer; 2007, p. 450–66.
- [34] Banik Subhadeep, Pandey Sumit Kumar, Peyrin Thomas, Sasaki Yu, Sim Siang Meng, Todo Yosuke. Gift: a small present. In: *International conference on cryptographic hardware and embedded systems*. Springer; 2017, p. 321–45.
- [35] Koo Bonwook, Roh Dongyoung, Kim Hyeonjin, Jung Younghoon, Lee Dong-Geon, Kwon Daesung. Cham: a family of lightweight block ciphers for resource-constrained devices. In: *International conference on information security and cryptography*. Springer; 2017, p. 3–25.
- [36] Li Lang, Liu Botao, Wang Hui. Qtl: a new ultra-lightweight block cipher. *Microprocess Microsyst* 2016;45:45–55.
- [37] Li Lang, Liu Botao, Zhou Yimeng, Zou Yi. Sfn: A new lightweight block cipher. *Microprocess Microsyst* 2018;60:138–50.
- [38] Noura Hassan, Guyeux Christophe, Chehab Ali, Mansour Mohammad, Couturier Raphaël. Efficient chaotic encryption scheme with OFB mode. *Int J Bifurcation Chaos* 2019;29(05).
- [39] Hassan Noura. Conception et simulation des générateurs, crypto-systèmes et fonctions de hachage basés chaos performants (Ph.D. thesis), Université de Nantes; 2012.
- [40] Kim SuHyun, Lee ImYeong. Iot device security based on proxy re-encryption. *J Ambient Intell Humaniz Comput* 2018;9(4):1267–73.
- [41] Suzaki Tomoyasu, Minematsu Kazuhiko, Morioka Sumio, Kobayashi Eita. Twine: A lightweight, versatile block cipher. In: *ECRYPT workshop on lightweight cryptography*, Vol. 2011. 2011.
- [42] Wei Yuechuan, Xu Peng, Rong Yisheng. Related-key impossible differential cryptanalysis on lightweight cipher twine. *J Ambient Intell Humaniz Comput* 2019;10(2):509–17.
- [43] Noura Hassan, Sleem Lama, Noura Mohamad, Mansour Mohammad M, Chehab Ali, Couturier Raphaël. A new efficient lightweight and secure image cipher scheme. *Multimedia Tools Appl* 2017.
- [44] Noura H, Courousse D. Method of encryption with dynamic diffusion and confusion layers. 2016, WO Patent App. PCT/EP2015/078, 372.
- [45] Melki Reem, Noura Hassan N, Mansour Mohammad M, Chehab Ali. An efficient ofdm-based encryption scheme using a dynamic key approach. *IEEE Internet Things J* 2018.
- [46] Miller Frederic P, Vandome Agnes F, McBrewhster John. *Advanced encryption standard*. Alpha Press; 2009.
- [47] Nakahara Jr J, Abrahão E. A new involutory mds matrix for the aes. *Int J Netw Secur* 2009;9(2):109–16.

- [48] Aoki K, Ichikawa T, Kanda M, Matsui M, Moriai S, Nakajima J, Tokita T. Camellia: A 128-bit block cipher suitable for multiple platforms - design and analysis. LNCS 2001;2012(2012):39–56, cited By (since 1996)104.
- [49] Kwon D, Kim J, Park S, Sung SH, Sohn Y, Song JH, Yeom Y, Yoon E-J, Lee S, Lee J, Chee S, Han D, Hong J. New block cipher: aria. Lect Notes Comput Sci 2004;2971:432–45, cited By (since 1996)22.
- [50] Kwon D, Sung SH, Song JH, Park S. Design of block ciphers and coding theory. Trends Math 2005;8(1):13–20.
- [51] Noura Hassan, Martin Steven, Agha Khaldoun Al, Chahine Khaled. ERSS-RLNC: Efficient and robust secure scheme for random linear network coding. Comput Netw 2014;75(Part A(0)):99–112.
- [52] Koo BonWook, Jang HwanSeok, Song JungHwan. Constructing and cryptanalysis of a 16×16 binary matrix as a diffusion layer. In: Chae Ki-Joon, Yung Moti, editors. Information security applications. Lecture notes in computer science, Vol. 2908, Springer Berlin Heidelberg; 2004, p. 489–503.
- [53] Koo Bon Wook, Jang Hwan Seok, Song Jung Hwan. On constructing of a 32×32 binary matrix as a diffusion layer for a 256-bit block cipher. In: International conference on information security and cryptology. Springer; 2006, p. 51–64.
- [54] Aslan B, Sakalli MT. Algebraic construction of cryptographically good binary linear transformations. Secur Commun Netw 2014;7(1):53–63.
- [55] Sakallı Muharrem Tolga, Akleylek Sedat, Aslan Bora, Bulus Ercan, Sakallı Fatma Büyüksaraçoğlu. On the construction of 20×20 and 24×24 binary matrices with good implementation properties for lightweight block ciphers and hash functions. Math Probl Eng 2015.
- [56] Ledley RS, NATIONAL BIOMEDICAL RESEARCH FOUNDATION SILVER SPRING MD . The inverse of a boolean matrix. Defense Technical Information Center; 1965.
- [57] Noura Hassan, Chehab Ali. Efficient and robust keyed hash function based on artificial neural networks. In: 2021 international symposium on networks, computers and communications (ISNCC). 2021, p. 1–7.
- [58] Birmes J, Asanovic K, Chin C, Demmel J. Optimizing matrix multiply using PHiPAC: a portable, high-performance, ansi C coding methodology. In: International conference on supercomputing. 1997.
- [59] Brent RP. Algorithms for matrix multiplication. Technical report TR-CS-70-157, Stanford University; 1970.
- [60] Daemen Joan, Rijmen Vincent. The design of rijndael. Secaucus, NJ, USA: SpringerVerlag New York, Inc; 2002.
- [61] Z'aba Muhammad Reza. Analysis of linear relationships in block ciphers (Ph.D. thesis), Queensland University of Technology; 2010.
- [62] Webster AF, Tavares Stafford E. On the design of s-boxes. In: Advances in cryptology—CRYPTO'85 proceedings. Springer; 1986, p. 523–34.
- [63] Fiestel Horst. Cryptography and computer privacy. Sci Am 1973;228(5):15–23.