Journal of Computing and Security

January 2014, Volume 1, Number 1 (pp. 15–22)

http://www.jcomsec.org

Unified Byte Permutations for the Block Cipher 3D

Hamid Mala [a,*]

[a] *Department of Information Technology Engineering, University of Isfahan, Isfahan, Iran*

ARTICLE INFO.

Article history:
Received: 15 December 2012
Revised: 17 June 2013
Accepted: 10 July 2013
Published Online: 20 December 2013

Keywords:
Block Cipher 3D, Diffusion, Graph Theory, Order/Degree Problem, Permutation

ABSTRACT

3D is a 512-bit block cipher whose design is inspired from the Advanced Encryption Standard (AES). Like the AES, each round of 3D is composed of 4 transformations including a round-key addition, a byte-wise substitution, a byte-wise shuffle and an MDS matrix multiplication. In 3D, two distinct byte-wise permutations are employed for odd and even rounds. In this paper, using concepts from graph theory, we design a unified byte permutation for both odd and even rounds with the same diffusion property as the original cipher. The main advantage of this new transformation is in hardware implementation of the cipher where with less resources we can speed up the encryption/decryption process.

1 Introduction

The Advanced Encryption Standard (AES)[1] is the most world-widely used block cipher in the current century. Since the selection of this 128-bit block cipher as the standard by NIST in 2001, its design rationale has inspired many cryptographic primitives including the LEX stream cipher [2], the ECHO hash function [3], the ALPHA-MAC message authentication code [4] and the 3D block cipher [5].

3D is an AES-based block cipher introduced by Nakahara at CANS 2008. This 22-round block cipher operates on 512-bit blocks and supports 512-bit keys. While AES, as a 128-bit block cipher, has a 4×4 byte state, 3D regards the 512-bit internal state as a $4 \times 4 \times 4$ cube of bytes. Both ciphers have substitution-permutation structure and are composed of 4 components in each round: XORing the internal state by the round subkey, byte-wise substitution, byte permutation, and finally a column-wise linear diffusion obtained through multiplication of a 4×4 MDS (Maximum Distance Separable) matrix by all of the 4-byte columns of the state. Subkey mixing, S-boxes, and column-wise diffusion are the same in both ciphers. However, to take advantage of the three-dimensional states, 3D applies the byte permutation of AES (ShiftRows) in two directions (θ_1 and θ_2 operations) in every two rounds alternately. Thus, here the odd and even rounds of the cipher are not the same.

As NIST considered in the AES competition, the main criteria to evaluate a block cipher include security, efficiency in software and hardware, and flexibility. 3D seems to be secure enough against known block cipher cryptanalysis techniques. Since its proposal in 2008, the most important cryptanalytic results on this cipher include a 10-round impossible differential attack with a data complexity of about 2^{501} chosen plaintexts and a time complexity of about 2^{401} encryptions [6], and a 13-round truncated differential attack with a data complexity of about 2^{469} chosen plaintexts and a time complexity of about 2^{308} encryptions [7]. These attacks do not threaten the practical security of 3D in any way. So, in the absence of any major achievement in the cryptanalysis of the 3D, software and hardware efficiency becomes a substantial measure for its evaluation. Using different transformations in different rounds of a block cipher reduces the implementation efficiency. With respect to 3D, this cipher uses two different byte permutations in odd

* Corresponding author.
Email address: h.mala@eng.ui.ac.ir (H. Mala)
ISSN: 2322-4460

and even rounds. So, when implementing this cipher in hardware, we have to either implement at least two rounds of the cipher or add an extra multiplexer to choose between these two transformations in odd and even rounds. The former solution requires more hardware area and the latter imposes more latency to the encryption/decryption process.

In this paper, we propose a unified byte permutation for all rounds of 3D. The proposed permutation improves the implementation efficiency of the modified 3D with the same effect on the diffusion of three rounds of the cipher as the original byte permutations. The method that we use to obtain this byte permutation is based on concepts from graph theory. In fact, we model the diffusion of three rounds of 3D as properties of a directed graph, and then present graphs satisfying these properties.

1.1 Related Work

The idea of using a graph-theoretic model for diffusion properties of a block cipher is studied by Massey in [8]. He combines a block shuffle, called Armenian shuffle, with a two-block linear transformation, called pseudo-hadamard transform (PHT), to compose the diffusion layer of the well-known SPN (Substitution Permutation Network) block cipher SAFER+. Later, at FSE 2010, by modeling the internal block shuffle of the type-II generalized Feistel structure as a graph, Suzuki et al. show that the diffusion of this block cipher structure can be improved by modifying the traditional cyclic shift of sub-blocks by some new shuffles [9]. The SHA-3 finalist JH uses a permutation P_d over 2^d words in its compression function [10]. In fact, the compression function of JH has an SPN structure which uses the permutation P_d along with a 2×2 MDS matrix as the diffusion layer. It is shown that this structure has an equivalent SPN structure in which distinct permutations are used alternately instead of the same P_d in all rounds.

1.2 Paper Organization

The rest of this paper is organized as follows. Section 2 provides a brief description of 3D and reviews the required concepts of graph theory. Section 3 is dedicated to the main contribution of this work. In Subsection 3.1, we present a graph-theoretic model for a unified diffusion layer for 3D including an optimum byte shuffle which can be employed as an alternative for the 3D's permutations θ_1 and θ_2. Then, a procedure to find graphs satisfying the required diffusion properties and several concrete examples for 64-byte permutations are introduced in Subsection 3.2. The extension for 256-byte permutations is discussed in Subsection 3.3. The implementation benefits of the unified permutation on the 3D are discussed in Section 4.

2 Preliminaries

In this section, we provide a brief description of the block cipher 3D, and briefly review the required background from graph theory.

2.1 Brief Description of the Block Cipher 3D

The block cipher 3D operates on 512-bit blocks under a 512-bit key, both represented as $4 \times 4 \times 4$ states of bytes [5]. The state corresponding to a 64-byte data block $A = (a_0, a_1, ..., a_{63})$ can be represented by a cube of 64 bytes, as shown in Figure 1, or by a 4×16 matrix A, as shown in Table 1.

Each square set of 16 bytes in Figure 1 is called a slice of the state. Since we can index each byte of the state by $16z + 4y + x$, $0 \leq x, y, z \leq 3$, each slice is described by fixing one of these three variables. For example, the slice $z = 0$ includes $(a_0, a_1, ..., a_{15})$, and the slice $x = 3$ represents the lowest horizontal slice. According to Figure 1, the four bytes $4i, 4i + 1, 4i + 2$ and $4i+3$ constitute the column $col(i)$. Conversely, the byte with index j belongs to column $col(\lfloor j/4 \rfloor)$. Each round of 3D applies the following four transformations to the state cube:

- Key Addition k_i: a bit-wise XOR operation between the state cube and the subkey of the current round.
- Substitution γ: this nonlinear operation consists of the byte-wise application of the AES S-box to the 64 bytes of the state cube.
- Byte shuffles θ_1 and θ_2: these two byte permutations are applied on the state cube in odd and even rounds, alternately. θ_1 transfers the byte located in position (x, y, z) into location $(x, y - x \mod 4, z)$. In other words, it transforms the 4×16 state matrix A into $\theta_1(A)$ shown in Table 1.

  On the other hand, as shown in Table 1, θ_2 transfers the byte located in position (x, y, z) into location $(x, y, z - x \mod 4)$.
- Matrix multiplication π: the 4×4 MDS matrix of the Anubis cipher [11] is applied to each of the 16 columns of the state. Since the branch number of this matrix is 5 and the state is 3-dimensional, complete diffusion is achieved in three rounds [5].

The $i - th$ round of 3D can be represented by

$$\tau_i(state) = \pi \circ \theta_{(i \mod 2)+1} \circ \gamma \circ k_{i-1}(state), \quad 1 \leq i \leq 22$$

In the complete cipher the round function is iterated 22 times, where in the last round the π function is replaced by the round key addition k_{22}. The 64-byte state after the application of transformation $T \in \{k_i, \gamma, \theta_1, \theta_2, \pi\}$ in round $1 \leq r \leq 22$ is denoted by X_r^T. Considering the byte numbering and column numbering represented in Figure 1, the byte with

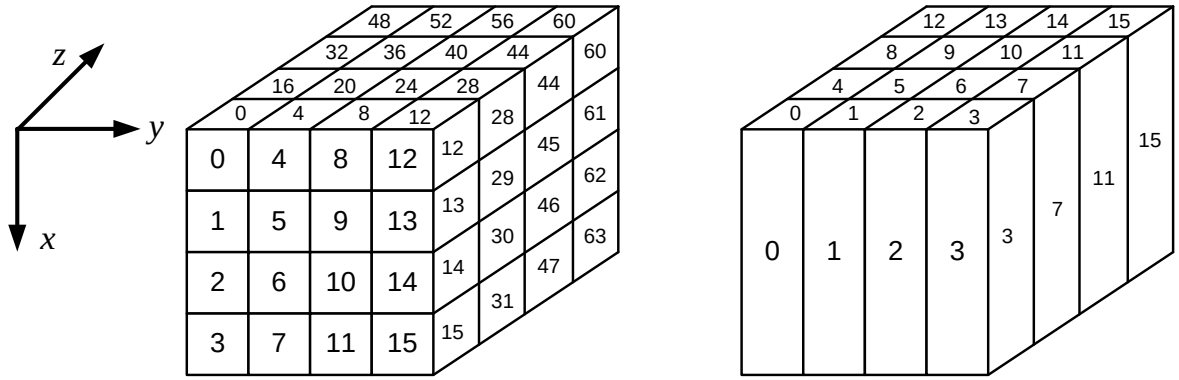

Figure 1. Byte numbering (left) and column numbering (right) for the 64-byte state array of 3D

index i and the column with index c in the state X_r^T are denoted by $X_r^T(i)$ and $X_r^T(col(c))$, respectively.

2.2 Required Terminology of Graph Theory

Here we recall the required terminology of graph theory. A directed graph or digraph G is an ordered pair $G = (V, A)$, where V is a non-empty set of distinct elements called vertices, and A is a set of ordered pairs $(u, v), u, v \in V$ called arcs or edges. The order of a digraph G is the number of vertices in G, i.e., $|V|$, and the size of G is defined as the number of its arcs, i.e., $|A|$. The in-degree of a vertex $v \in G$ is the number of arcs of the form (u_i, v), and its out-degree is defined as the number of arcs of the form (v, u_i). A digraph in which the in-degree equals the out-degree ($= \Delta$) for every vertex in the graph is called a diregular digraph of degree Δ. A sequence $(v_{i_1}, v_{i_2}, ..., v_{i_{l+1}})$ of vertices of the digraph G is called a directed path of length l in G if for $j = 1, 2, ..., l$, $(v_{i_j}, v_{i_{j+1}})$ is an edge in G. A digraph G is called connected if and only if for every ordered pair of vertices (v_i, v_j) in G there exists a directed path from v_i to v_j. The distance between two vertices v_i and v_j in G is denoted by $\delta(v_i, v_j)$ and defined as the length of the shortest directed path from v_i to v_j. The diameter of a digraph G is defined as $\max_{\forall i,j} \delta(v_i, v_j)$.

3 New Byte Permutations for the 3D Cipher

In this section, first we propose a graph-theoretic representation for the diffusion layer of 3D. Next, we try to find a unified byte shuffle to be used instead of θ_1 and θ_2 in every round.

3.1 Graph-Theoretic Model of the Diffusion Layer of 3D

In this section, we introduce a formal notion for the diffusion property of the 3D. What we mean, here, by *diffusion* is the state where a byte in input affects all of the bytes in output. More formally, if $X_{r'}^{T'}(j)$ can be expressed by an equation containing $X_r^T(i)$ for some i and j and $r \leq r'$, we say $X_{r'}^{T'}(j)$ is affected by $X_r^T(i)$.

Proposition 1. *Diffusion properties of the four transformations of 3D are:*

(a) $X_r^k(i)$ *is affected by only* $X_{r-1}^{\pi}(i)$.
(b) $X_r^{\gamma}(i)$ *is affected by only* $X_r^k(i)$.
(c) $X_r^{\theta}(i)$ *is affected by only* $X_r^{\gamma}(\theta^{-1}(i))$, $\theta \in \{\theta_1, \theta_2\}$.
(d) $X_r^{\pi}(i)$ *is affected by all of the four bytes of* $X_r^{\theta}(col(\lfloor i/4 \rfloor))$.

Proof. Since the transformations k_i and γ are bytewise and do not change the place of any bytes, properties (a) and (b) are correct. The transformation θ only changes the position of byte i to the new position $\theta(i)$, so (c) is correct. The fact that the branch number of the MDS matrix of π is five guarantees that each of the four bytes of $X_r^{\theta}(4i, 4i+1, 4i+2, 4i+3)$ affects all of the four bytes of $X_r^{\pi}(col(i))$, so property (d) holds true. □

If all of the output bytes of a cubic state $X_{r_2}^{T_2}$ are affected by $X_{r_1}^{T_1}(i)$, we say $X_{r_1}^{T_1}(i)$ is diffused to all of the bytes of $X_{r_2}^{T_2}$. For instance, as a result of Proposition 1, we can say that $X_1^k(0)$ is diffused to $X_1^{\pi}(col(0))$. Using this concept, we come up with the following definition.

Definition 1. For the 3D, let DR_i be the minimum number of rounds such that the byte with index i of the first round input, $X_0(i)$, is diffused to all bytes of the state cube. Then, the maximum diffusion rounds for 3D, denoted by DR_{max}, is defined as $DR_{max} = max_{0 \leq i \leq 63} DR_i$.

The following proposition is a more formal exposition of the designer's statement saying "complete diffusion is achieved in three round".

Proposition 2. *For the 3D block cipher* $DR_i = 3, 0 \leq i \leq 63$, *consequently* $DR_{max} = 3$.

Proof. Considering Proposition 1, one can easily check that each of the 64 bytes of the $X_0(i)$ is diffused to one complete column after the $\pi \circ \theta_1$ function.

Table 1. The 4×16 state array A and the effect of permutations θ_1 and θ_2 on it

$$A = \begin{pmatrix} a_0 & a_4 & a_8 & a_{12} & a_{16} & a_{20} & a_{24} & a_{28} & a_{32} & a_{36} & a_{40} & a_{44} & a_{48} & a_{52} & a_{56} & a_{60} \\ a_1 & a_5 & a_9 & a_{13} & a_{17} & a_{21} & a_{25} & a_{29} & a_{33} & a_{37} & a_{41} & a_{45} & a_{49} & a_{53} & a_{57} & a_{61} \\ a_2 & a_6 & a_{10} & a_{14} & a_{18} & a_{22} & a_{26} & a_{30} & a_{34} & a_{38} & a_{42} & a_{46} & a_{50} & a_{54} & a_{58} & a_{62} \\ a_3 & a_7 & a_{11} & a_{15} & a_{19} & a_{23} & a_{27} & a_{31} & a_{35} & a_{39} & a_{43} & a_{47} & a_{51} & a_{55} & a_{59} & a_{63} \end{pmatrix}$$

$$\theta_1(A) = \begin{pmatrix} a_0 & a_4 & a_8 & a_{12} & a_{16} & a_{20} & a_{24} & a_{28} & a_{32} & a_{36} & a_{40} & a_{44} & a_{48} & a_{52} & a_{56} & a_{60} \\ a_5 & a_9 & a_{13} & a_1 & a_{21} & a_{25} & a_{29} & a_{17} & a_{37} & a_{41} & a_{45} & a_{33} & a_{53} & a_{57} & a_{61} & a_{49} \\ a_{10} & a_{14} & a_2 & a_6 & a_{26} & a_{30} & a_{18} & a_{22} & a_{42} & a_{46} & a_{34} & a_{38} & a_{58} & a_{62} & a_{50} & a_{54} \\ a_{15} & a_3 & a_7 & a_{11} & a_{31} & a_{19} & a_{23} & a_{27} & a_{47} & a_{35} & a_{39} & a_{33} & a_{63} & a_{51} & a_{55} & a_{59} \end{pmatrix}$$

$$\theta_2(A) = \begin{pmatrix} a_0 & a_4 & a_8 & a_{12} & a_{16} & a_{20} & a_{24} & a_{28} & a_{32} & a_{36} & a_{40} & a_{44} & a_{48} & a_{52} & a_{56} & a_{60} \\ a_{17} & a_{21} & a_{25} & a_{29} & a_{33} & a_{37} & a_{41} & a_{45} & a_{49} & a_{53} & a_{57} & a_{61} & a_1 & a_5 & a_9 & a_{13} \\ a_{34} & a_{38} & a_{42} & a_{46} & a_{50} & a_{54} & a_{58} & a_{62} & a_2 & a_6 & a_{10} & a_{14} & a_{18} & a_{22} & a_{26} & a_{30} \\ a_{51} & a_{55} & a_{59} & a_{63} & a_3 & a_7 & a_{11} & a_{15} & a_{19} & a_{23} & a_{27} & a_{31} & a_{35} & a_{39} & a_{43} & a_{47} \end{pmatrix}$$

The 4 bytes of this column are carried over to four different columns of the same vertical slice after θ_2, and then diffused to a complete vertical slice after the second π. These 16 bytes of this vertical slice are carried over to the 16 columns of the cube by the next θ_1 permutation, and finally, after the third π all the 64 bytes are affected. □

Our objective here is to find a byte shuffle such that by replacing it for θ_1 and θ_2 in 3D we get the same $DR_{max} = 3$. To find such a shuffle, in general form, we have to search for a byte permutation of order 64, $P^* : \{0, 1, ..., 63\} \rightarrow \{0, 1, ..., 63\}$. Obviously, the cost of exhaustive search for this problem is too high; therefore, we present a graph-theoretic interpretation for the diffusion properties of a round of the cipher and then solve this problem.

Proposition 3. *Suppose we replace the byte permutations of 3D, θ_1 and θ_2, by an arbitrary byte permutation $P : \{0, 1, ..., 63\} \rightarrow \{0, 1, ..., 63\}$. Since the π transformation is a 4×4 MDS matrix, which propagates one byte in input to 4 bytes in output, the minimum possible value for DR_{max} is 3. Every byte permutation that satisfies $DR_{max} = 3$ is called an optimum shuffle for 3D.*

The following property for any optimum byte shuffle for 3D guides us to model the problem as a graph.

Proposition 4. *Every optimum byte shuffle for the 3D structure permutes the 4 bytes of each column to four distinct columns.*

Proof. Suppose this is not the case, i.e., there exists an optimum permutation P that permutes 4 bytes of a column to at most 3 distinct columns. Starting with a byte that diffuses to this specific column in the output of the first round, it will diffuse to at most 3 columns in the output of the second round; consequently, it will diffuse to at most 12 columns in the output of the 3rd round. This would contradict the fact that P is an optimum shuffle for 3D. □

Consider a modified version of the 3D cipher where θ_1 and θ_2 are replaced by an optimum shuffle P. Here, every byte in $X_0(i)$ diffuses to a 4-byte column in output of the first round. Now, based on Proposition 4, we can introduce an equivalent graphical representation for the diffusion of this modified cipher.

Definition 2. Corresponding to an optimum byte shuffle P of 3D, we define a directed graph, denoted by $G(P)$ with order 16. Each vertex of $G(P)$ corresponds to one of the 16 columns of the state cube and is labeled with $\{0, 1, ..., 15\}$, compatible with the column numbering of Figure 1. If the shuffle P carries one of the bytes of $col(i)$ to one of the bytes of $col(j)$, a directed arc is formed from node i to node j.

Based on Definition 2 and Proposition 4, the following property is deduced for the digraph $G(P)$.

Corollary 1. *For every vertex of $G(P)$ the in-degree and out-degree is 4.*

The following theorem correlates the optimality of P and the diameter of $G(P)$.

Theorem 1. *For every optimum shuffle of 3D, the corresponding digraph $G(P)$, defined in Definition 2, has diameter 2.*

Proof. Since P is optimum, each column in the output of the first round must diffuse to all 64 bytes in the output of the next two rounds. This is equivalent to the statement that each node in $G(P)$ must have a path of length 2 to each of the 16 vertices of this digraph. □

So, to find optimum shuffles, we have to find diregular digraphs of order 16, degree 4, and diameter 2. The following subsection gives an answer to this query.

3.2 A Procedure to Construct Optimum Permutations

In the context of graph theory, our problem is: "given order $n = 16$ and degree $\Delta = 4$, find a diregular digraph with the minimum diameter". Such a problem, in its general form, is of great interest due to its theoretical appeal and its possible applications in communication networks design. We, first, have to answer the question of existence of such a digraph. This is associated to the well-known *order/degree* problem in graph theory: Given natural numbers n and Δ find the smallest possible diameter $d_{n,\Delta}$ in a digraph of order n and maximum out-degree Δ [12]. The following lower bound for $d_{n,\Delta}$ has been proved in [13].

$$d_{n,\Delta} \geq \lceil log_{\Delta}(n(\Delta-1)+\Delta)\rceil - 1 \qquad (1)$$

In our problem $n = 16$ and $\Delta = 4$, thus based on Equation 1 we have $d_{16,4} \geq \lceil log_4 52\rceil - 1 = 2$. This confirms Proposition 2, i.e., the best possible value for DR_{max} is 3. In [14] Imase and Itoh proposed an algorithm to construct a nearly optimal diregular digraph of order n and degree Δ and diameter $\lceil log_{\Delta} n\rceil$. In our problem, this diameter is $\lceil log_{\Delta} n\rceil = \lceil log_4 16\rceil = 2$, which is of our interest. Customizing the algorithm of [14], we present a procedure to construct a diregular digraph with 16 nodes, diameter 2, and degree 4 as follows.

Algorithm 1 The procedure to construct a 4-diregular digraph of order 16 and diameter 2

Input: A set of 16 nodes $V = \{v_0, v_1, \ldots, v_{15}\}$
Output: A set of 64 directed arcs $A = \{(v_i, v_j)\}$
1: $\Delta \leftarrow 4$ % Δ is the degree of the diregular digraph
2: $n \leftarrow 16$ % n is the order of the digraph
3: $A \leftarrow \varnothing$
4: **for** $i = 0$ to $n-1$ **do**
5: **for** $\alpha = 0$ to $\Delta - 1$ **do**
6: $j \leftarrow \Delta \times i + \alpha \bmod n$
7: $A \leftarrow A \cup (v_i, v_j)$
8: **end for**
9: **end for**
10: **return** A

Table 2. Destination columns computed based on Algorithm 1 for the 16 columns of the state cube

Source column	Destination columns	Source column	Destination columns
0	{0, 1, 2, 3}	8	{0, 1, 2, 3}
1	{4, 5, 6, 7}	9	{4, 5, 6, 7}
2	{8, 9, 10, 11}	10	{8, 9, 10, 11}
3	{12, 13, 14, 15}	11	{12, 13, 14, 15}
4	{0, 1, 2, 3}	12	{0, 1, 2, 3}
5	{4, 5, 6, 7}	13	{4, 5, 6, 7}
6	{8, 9, 10, 11}	14	{8, 9, 10, 11}
7	{12, 13, 14, 15}	15	{12, 13, 14, 15}

As presented in Table 2, the result of this procedure on the state cube is that an optimum permutation P permutes the four bytes of column 0 to the four columns $\{0, 1, 2, 3\}$, the four bytes of column 1 to the four columns $\{4, 5, 6, 7\}$, ... and the four bytes of column 15 to the four columns $\{12, 13, 14, 15\}$. Note that four specific columns are assigned as the destination of four bytes of a specific column but the exact locations in the destination columns are not forced by the procedure. On the other hand, the 16 bytes of slice $y = i, i = 0, 1, 2, 3$ are transposed to the 16 bytes of slice $z = i$, such that four bytes of each source column go to four bytes in four distinct columns. Hence, the first column of the source slice has $16 \times 12 \times 8 \times 4 = 4^4 \times 4!$ possible choices in the destination slice, the second column has $12 \times 9 \times 6 \times 3 = 3^4 \times 4!$ options, the third column has $8 \times 6 \times 4 \times 2 = 2^4 \times 4!$, and, finally, the fourth column has 4! options. Thus, the total number of one slice to one slice transpositions is $(4!)^8$. Since we have four one-slice-to-one-slice transpositions, the total number of optimum permutations defined in the Algorithm 1 is $((4!)^8)^4 = 24^{32} \approx 10^{44}$. Recall that the number of all permutations over 64 bytes is $64! \approx 10^{250}$.

One can easily check that neither θ_1 nor θ_2 are optimum byte permutations. Based on the content of Table 2, for example, 3 bytes out of the 4 bytes of column 0 are moved to three positions in columns 1, 2 and 3, while these columns do not have any path of length 1 back to column 0. This would suffice to conclude that this procedure does not introduce any involutory permutation. However, among the 24^{32} optimum permutations produced through the above procedure, here, we provide several concrete more easily imaginable samples:

(1) P_1: first transposes the four horizontal slices and then applies θ_1. In this manner, the byte located in position (x, y, z) is carried over to the

Table 3. Instances of the optimum permutations on the 4×16 state array A

Permutation	Effect on the array A
$P_1(A)$	$\begin{pmatrix} a_0 & a_{16} & a_{32} & a_{48} & a_4 & a_{20} & a_{36} & a_{52} & a_8 & a_{24} & a_{40} & a_{56} & a_{12} & a_{28} & a_{44} & a_{60} \\ a_5 & a_{21} & a_{37} & a_{53} & a_9 & a_{25} & a_{41} & a_{57} & a_{13} & a_{29} & a_{45} & a_{61} & a_1 & a_{17} & a_{33} & a_{49} \\ a_{10} & a_{26} & a_{42} & a_{58} & a_{14} & a_{30} & a_{46} & a_{62} & a_2 & a_{18} & a_{34} & a_{50} & a_6 & a_{22} & a_{38} & a_{54} \\ a_{15} & a_{31} & a_{47} & a_{63} & a_3 & a_{19} & a_{35} & a_{51} & a_7 & a_{23} & a_{39} & a_{55} & a_{11} & a_{27} & a_{43} & a_{59} \end{pmatrix}$
$P_2(A)$	$\begin{pmatrix} a_0 & a_1 & a_2 & a_3 & a_{16} & a_{17} & a_{18} & a_{19} & a_{32} & a_{33} & a_{34} & a_{35} & a_{48} & a_{49} & a_{50} & a_{51} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{36} & a_{37} & a_{38} & a_{39} & a_{52} & a_{53} & a_{54} & a_{55} & a_4 & a_5 & a_6 & a_7 \\ a_{40} & a_{41} & a_{42} & a_{43} & a_{56} & a_{57} & a_{58} & a_{59} & a_8 & a_9 & a_{10} & a_{11} & a_{24} & a_{25} & a_{26} & a_{27} \\ a_{60} & a_{61} & a_{62} & a_{63} & a_{12} & a_{13} & a_{14} & a_{15} & a_{28} & a_{29} & a_{30} & a_{31} & a_{44} & a_{45} & a_{46} & a_{47} \end{pmatrix}$
$P_3(A)$	$\begin{pmatrix} a_0 & a_{16} & a_{32} & a_{48} & a_1 & a_{17} & a_{33} & a_{49} & a_2 & a_{18} & a_{34} & a_{50} & a_3 & a_{19} & a_{35} & a_{51} \\ a_4 & a_{20} & a_{36} & a_{52} & a_5 & a_{21} & a_{37} & a_{53} & a_6 & a_{22} & a_{38} & a_{54} & a_7 & a_{23} & a_{39} & a_{55} \\ a_8 & a_{24} & a_{40} & a_{56} & a_9 & a_{25} & a_{41} & a_{57} & a_{10} & a_{26} & a_{42} & a_{58} & a_{11} & a_{27} & a_{43} & a_{59} \\ a_{12} & a_{28} & a_{44} & a_{60} & a_{13} & a_{29} & a_{45} & a_{61} & a_{14} & a_{30} & a_{46} & a_{62} & a_{15} & a_{31} & a_{47} & a_{63} \end{pmatrix}$
$P_4(A)$	$\begin{pmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{16} & a_{17} & a_{18} & a_{19} & a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & a_{29} & a_{30} & a_{31} \\ a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} & a_{40} & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} \\ a_{48} & a_{49} & a_{50} & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} & a_{59} & a_{60} & a_{61} & a_{62} & a_{63} \end{pmatrix}$
$P_5(A)$	$\begin{pmatrix} a_{48} & a_{49} & a_{50} & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} & a_{59} & a_{60} & a_{61} & a_{62} & a_{63} \\ a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{16} & a_{17} & a_{18} & a_{19} & a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & a_{29} & a_{30} & a_{31} \\ a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} & a_{40} & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} \end{pmatrix}$

position $(x, z-x \mod 4, y)$. The effect of P_1 on the 4×16 state array A has been demonstrated by $P_1(A)$ in Table 3.

(2) P_2: first applies transformation θ_2 and then transposes the vertical slices $y = 0, 1, 2, 3$. In this manner, the byte located in position (x, y, z) is carried over to the position $(z - x \mod 4, y, x)$. The effect of this permutation on the 4×16 state array A has been shown by $P_2(A)$ in Table 3.

(3) P_3: first transposes the horizontal slices and then transposes the vertical slices $z = 0, 1, 2, 3$, i.e., the byte located in position (x, y, z) is carried over to the position (z, x, y). The effect of this permutation on the 4×16 state array A has been demonstrated by $P_3(A)$ in Table 3.

(4) P_4: first transposes the horizontal slices and then transposes the vertical slices $y = 0, 1, 2, 3$, i.e., the byte located in position (x, y, z) is carried over to the position (y, z, x). The effect of this permutation on the 4×16 state array A has been shown by $P_4(A)$ in Table 3.

The above mentioned permutations have fixed points, that is, for each of them there is at least one byte not permuted by the mentioned P_1, P_2, P_3 and P_4. For instance, $(0, 0, 0)$ is a fixed point for all the four permutations given above. However this is not the case for all possible optimum permutations. An optimum permutation, P_5, without any fixed point is proposed below. P_5 moves the byte in position (x, y, z) to the position $(z, x, y + 1 \mod 4)$. The effect of this permutation on the 4×16 state array A has been demonstrated by $P_5(A)$ in Table 3.

3.3 Extending for a Higher Dimension

3D with its $4 \times 4 \times 4$ cubic state is a natural extension of the AES whose state is a 4×4 matrix. The next

step in extending the AES may be for example a 2048-bit block cipher whose state is a $4 \times 4 \times 4 \times 4$ array of bytes. This 4-dimensional array can be represented as a hypercube, or equivalently four $4 \times 4 \times 4$ cubes. To give a concrete permutation, let us indicate four cubes of a state by $w = 0, 1, 2, 3$ and follow the same indexing based on (x, y, z) in each cube as for the state of 3D in Figure 1. The cube with indicator $w = 0$ includes columns indexing from 0 to 15, the cube with index $w = 1$ includes columns from 16 to 31, the cube with index $w = 2$ includes columns from 32 to 47, and the cube with index $w = 3$ includes columns from 48 to 63. In this manner, the byte located in position (x, y, z, w), $0 \leq x, y, z, w \leq 3$ in the hypercube can be indexed by $64w + 16z + 4y + x$. Let us consider a 2048-bit block cipher that uses four transformations in each round. First, as the round key addition, a 2048-bit round key is XORed to the state. Next, the same 8-bit S-box is applied to all 256 bytes of the state. Then a 256-byte permutation is applied, and finally, the same 4×4 MDS matrix is multiplied to each one of the 64 columns of the state.

Using the proposed method to model the diffusion property of an optimum permutation and the MDS matrix, we can show that, here each byte of the state affects all the 256 bytes of the state after 4 rounds. In our graphical model, we have to look for a diregular digraph of order $n = 64$, diameter 3, and still degree 4. Based on the order/degree problem, the lower bound for the diameter of a diregular graph of order $n = 64$ and $\Delta = 4$ is $\lceil log_{\Delta}(n(\Delta-1)+\Delta)\rceil - 1 = 3$. Moreover, the procedure of Subsection 3.2, when customized for $n = 64$ and $\Delta = 4$, would determine such an optimum permutation over 64 columns. The outcome of this procedure is that, in an optimum 256-byte permutation, the 4 bytes of column $i, i = 0, 1, ..., 63$ are displaced to 4 distinct bytes of columns $4i \mod 64$, $4i + 1 \mod 64$, $4i + 2 \mod 64$ and $4i + 3 \mod 64$. The permutation carrying over the byte located in position (x, y, z, w) into the position (w, x, y, z) follows the procedure of Subsection 3.2, hence, an optimum permutation over 256 bytes. This transformation can be considered as three consecutive matrix transpositions: first transposing in the $w - z$ planes, then in $y - z$ planes, and finally in $x - y$ planes.

4 Concluding Remarks

In this paper, we proposed a graph-theoretic model for the diffusion property of the AES-based block cipher 3D. Like AES, the diffusion layer of 3D is composed of two transformations: a byte shuffle that permutes the 64 bytes of the state cube, and a 4×4 MDS matrix that is multiplied to all 16 columns of the state cube. Each byte of the state affects all the 64 bytes after three rounds. However, this property of 3D is achieved at the cost of using two distinct byte shuffles in odd and even rounds which in turn increases hardware implementation cost of the cipher. In this paper, assuming a unified byte shuffle for all rounds of the cipher, we modeled the diffusion property of 3 rounds as a diregular directed graph of order 16, diameter 2 and degree 4. Then we presented a procedure to obtain such a digraph, which proposes 24^{32} optimum byte shuffles in the $4 \times 4 \times 4$ cubic state. We believe that replacing the same byte permutation for every round of 3D, instead of θ_1 and θ_2, will improve hardware performance of the cipher.

The typical methods used in hardware implementation of a block cipher include *basic iterated*, *partial loop unrolling*, and *full loop unrolling* architectures, each with the potential of several levels of pipelining [15]. The round iterated architecture is the best choice when there exist restrictions on the maximum area of a cryptographic module. For a block cipher like 3D with a big block size, the choice of round iterated is preferable. In this architecture, one round of the cipher is implemented as a combinational logic and supplemented with a single register and a multiplexer before the round. In the first clock cycle, input block of data is fed to the circuit through the multiplexer and stored in the register. In each of the subsequent clock cycles, by proceeding the data block through the circuit, one round of the cipher is evaluated, and then, the result is fed back to the circuit through the multiplexer, and stored in the register.

However, since odd and even rounds of 3D use two distinct shuffles, some adjustments in the basic iterated architecture are inevitable. We have two possibilities:

- we can iterate at least two rounds, or
- implement one round with both θ_1 and θ_2, and an additional multiplexer to select between their outputs. This multiplexer would have two 64-byte inputs in order to select one of them.

Compared with the area required to implement one single round of 3D, the former solution increases the area by a factor of two, and the latter needs an additional area of a big multiplexer as well as one additional permutation. Moreover, the latter solution imposes the latency of the multiplexer and consequently causes a decrease in efficiency of the encryption/decryption process. So, modifying the 3D by replacing one of our proposed shuffles in all rounds instead of θ_1 and θ_2 eliminates these costs. In fact, by unifying the permutation of the cipher, the imposed additional area in the former iterated architecture, and both the imposed area and latency of the additional multiplexer in the latter iterated architecture are suppressed.

References

[1] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer, 2002. ISBN 978-3-642-07646-6. doi: 10.1007/978-3-662-04722-4.

[2] Alex Biryukov. The Design of a Stream Cipher LEX. In Eli Biham and Amr M. Youssef, editors, *Selected Areas in Cryptography*, volume 4356 of *Lecture Notes in Computer Science*, pages 67–75. Springer, 2006. ISBN 978-3-540-74461-0.

[3] Ryad Benadjila, Olivier Billet, Henri Gilbert, Gilles Macario-Rat, Thomas Peyrin, Matt Robshaw, and Yannick Seurin. SHA-3 Proposal: ECHO. Submission to NIST (updated), 2009. URL http://crypto.rd.francetelecom.com/echo/doc/echo_description_1-5.pdf.

[4] Joan Daemen and Vincent Rijmen. A New MAC Construction ALRED and a Specific Instance ALPHA-MAC. In Henri Gilbert and Helena Handschuh, editors, *FSE*, volume 3557 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2005. ISBN 3-540-26541-4.

[5] Jorge Nakahara Jr. 3D: A Three-Dimensional Block Cipher. In Matthew K. Franklin, Lucas Chi Kwong Hui, and Duncan S. Wong, editors, *CANS*, volume 5339 of *Lecture Notes in Computer Science*, pages 252–267. Springer, 2008. ISBN 978-3-540-89640-1.

[6] Jorge Nakahara Jr. New Impossible Differential and Known-Key Distinguishers for the 3D Cipher. In Feng Bao and Jian Weng, editors, *ISPEC*, volume 6672 of *Lecture Notes in Computer Science*, pages 208–221. Springer, 2011. ISBN 978-3-642-21030-3.

[7] Takuma Koyama, Lei Wang, Yu Sasaki, Kazuo Sakiyama, and Kazuo Ohta. New Truncated Differential Cryptanalysis on 3D Block Cipher. In Mark Dermot Ryan, Ben Smyth, and Guilin Wang, editors, *ISPEC*, volume 7232 of *Lecture Notes in Computer Science*, pages 109–125. Springer, 2012. ISBN 978-3-642-29100-5.

[8] James Massey. On the Optimality of SAFER+ Diffusion. In *Proceedings of the Second AES Candidate Conference, National Institute of Standards and Technology*, 1999.

[9] Tomoyasu Suzaki and Kazuhiko Minematsu. Improving the Generalized Feistel. In Seokhie Hong and Tetsu Iwata, editors, *FSE*, volume 6147 of *Lecture Notes in Computer Science*, pages 19–39. Springer, 2010. ISBN 978-3-642-13857-7.

[10] Hongjun Wu. The Hash Function JH. Submission to NIST (round 3), January 2011. URL http://www3.ntu.edu.sg/home/wuhj/research/jh/jh_round3.pdf.

[11] Paulo Barreto and Vincent Rijmen. The Anubis block cipher. *Submission to the NESSIE Project*, 2000.

[12] Mirka Miller and Jozef Siráň. Moore graphs and beyond: a survey of the degree/diameter problem. *The Electronic Journal of Combinatorics*, DS14: 1–61, 2005.

[13] W.G. Bridges and Sam Toueg. On the impossibility of Directed Moore Graphs. *Journal of Combinatorial theory, series B*, 29(3):339–341, 1980.

[14] Makoto Imase and Masaki Itoh. Design to Minimize Diameter on Building-Block Network. *IEEE Trans. Computers*, 30(6):439–442, 1981.

[15] Kris Gaj and Pawel Chodowiec. FPGA and ASIC Implementations of AES. In Cetin Kaya Koc, editor, *Cryptographic Engineering*, pages 235–294. Springer US, 2009. ISBN 978-0-387-71816-3. doi: 10.1007/978-0-387-71817-0_10.

Hamid Mala received his B.S., M.S. and Ph.D. degrees in Electrical Engineering from Isfahan University of Technology (IUT) in 2003, 2006 and 2011, respectively. He joined University of Isfahan (UI) in September 2011 as an Assistant Professor in the Department of Information Technology Engineering. His Research interests include the design and cryptanalysis of block ciphers, digital signatures and cryptographic protocols.