Assimilating Byte Permutations of the 3D Block Cipher

Hamid Mala

Department of Information Technology Engineering,
University of Isfahan, Isfahan, Iran
{h.mala@eng.ui.ac.ir

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. 3D uses two different byte-wise shuffles for odd and even rounds. In this paper, using the concepts of graph theory, we design a unified byte permutation for both odd and even rounds with the same diffusion property for the cipher. The main advantage of this new transformation is in the hardware implementation of the cipher where with less resources we can speed up the encryption/decryption process.

Key Words: 3D block cipher, diffusion, byte permutation, graph theory.

1 Introduction

The Advanced Encryption Standard (AES)[6] has been 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 [3], the ECHO hash function [2], the ALPHA-MAC message authentication code [5] and the 3D block cipher [13].

3D is an AES-based block cipher proposed by Nakahara at CANS 2008. This 22-round block cipher operates on 512-bit blocks and supports a 512-bit key. 3D was designed to have a 4 times larger block size than that of the 128-bit AES. While AES 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 with the round subkey, byte-wise substitution, byte permutation, and finally a column-wise linear diffusion which is obtained through multiplication of a 4×4 MDS (Maximum Distance Separable) matrix to each 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) to two directions (θ_1 and θ_2 operations) in every two rounds alternately. Thus, odd and even rounds of the cipher are not the same.

As NIST has 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 [12], 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 [9]. 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 evaluation of 3D. 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 and even rounds. So, when implementing this cipher in the 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 area of the hardware and the latter imposes more latency to the encryption/decryption process.

In this paper, we propose a unified byte permutation for all rounds. While our proposed permutation has the same effect on the diffusion of the three rounds of the cipher as the original byte permutations, it improves the implementation efficiency of the modified 3D. The method we use to obtain this byte permutation is based on the concepts of graph theory. In fact, we model the diffusion of three rounds of the 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 (part of) a diffusion layer of a block cipher was considered by Massey in [10]. He combined 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 Type-II generalized Feistel structure as a graph, Suzuki et al. showed that the diffusion property of this block cipher structure can be improved by only changing the internal block shuffle rather than the traditional cyclic shift [14].

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. In Section 3, as the main contribution of this work, we present a graph-theoretic model for a assimilated diffusion layer for 3D which includes an optimum byte shuffle which can be replaced as an alternative for the 3D's permutations θ_1 and θ_2. Then, we present a procedure to find graphs satisfying the required diffusion properties. Finally, the implementation benefits achieved by this modification in the 3D are discussed in Section 4.

2 Preliminaries

2.1 Brief Description of the 3D Block Cipher

The 3D block cipher operates on 512-bit blocks under a 512-bit user key, both of which are represented as $4 \times 4 \times 4$ states of bytes [13]. The cubic state for a 64-byte data block $A = (a_0, a_1, ..., a_{63})$ can be represented either by a cube of 64 bytes, as illustrated in Figure 1, or by a 4×16 matrix as below.

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

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 $16y + 4x + z,\ 0 \leq x, y, z \leq 3$, each slice is described by fixing one of these three variables. For example, the slice $y = 0$ includes $(a_0, a_1, ..., a_{15})$, and the slice $z = 3$ represents the most lower horizontal slice. According to Figure 1, the four bytes $4i, 4i+1, 4i+2$ and $4i+3$ constitute the column $col(i)$. Inversely, the byte with index j is located in column $col(\lfloor j/4 \rfloor)$. A 3D round 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 transforms the above state cube A into

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

  while θ_2 transforms the state cube into

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

- Matrix multiplication π: the 4×4 MDS matrix of the Anubis cipher [1] is applied to each of the 16 columns of the state. The branch number of this matrix is 5. Since the state is 3-dimensional, complete diffusion is achieved in three rounds, in combination with θ_1 and θ_2 [13].

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

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

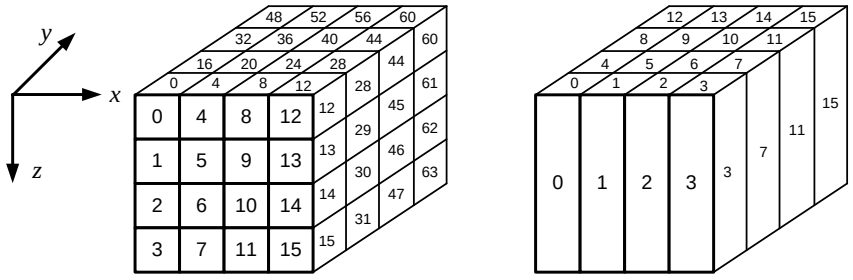

Fig. 1. Byte numbering (left) and column numbering (right) in cubic state of 3D

For 22-round 3D the round function is iterated 21 times, and 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 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 (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 d. A sequence $(v_1, v_2, ..., v_{l+1})$ of vertices of the digraph G where $(v_i, v_{i+1}), i = 1, 2, ..., l$ is an edge in G is called a directed path of length l in G. A digraph G is called connected if and only if for every ordered pair of vertices (v_1, v_2) in G there exists a directed path from v_1 to v_2. 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. Note that $\delta(v_i, v_j)$ may be unequal to $\delta(v_j, v_i)$. The diameter of a digraph G is defined as $\max_{\forall i,j} \delta(v_i, v_j)$.

3 New Byte Shuffle for the 3D cipher

In this section, first we propose a graph-theoretic representation of diffusion layer of the 3D block cipher. Then, 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 of the diffusion property of the 3D. What we mean by diffusion here is the state that a byte in input affects all of the bytes of

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* $X_{r-1}^{\pi}(i)$.
(d) $X_r^{\gamma}(i)$ *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 byte-wise 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 each 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 $X_{r_1}^{T_1}(i)$ has 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 make 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 a 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 rounds".

Proposition 2. *For the 3D block cipher* $DR_i = 3, 0 \leq i \leq 63$*, and 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. The 4 bytes of this column are carried over to four different columns of a 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 goal to find a byte shuffle such that by replacing it instead of 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 heavy to be done. So, 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. Each byte permutation satisfying* $DR_{max} = 3$ *is called an optimum shuffle for 3D.*

The following property for any optimum byte shuffle for 3D conducts 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 different 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 columns. So, 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, and consequently it will diffuse to at most 12 columns in the output of the 3rd round. This contradicts the fact that P is an optimum shuffle for 3D.

Consider a modified 3D cipher in which θ_1 and θ_2 are replaced by an optimum shuffle P. In this cipher, each 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 $G(P)$ is corresponding to one of the 16 columns of state cube and is labeled with $\{0,1,...,15\}$ compatible with the column numbering of Figure 1. There is a directed arc from node i to node j if the shuffle P carries one of the bytes of $col(i)$ to one of the bytes of $col(j)$.*

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 proposition correlates the optimumness 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 two 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 Shuffles

In the context of graph theory, our problem is finding of a diregular digraph with minimum diameter given order $n = 16$ and degree $\Delta = 4$. 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 related to the well-known *order/degree* problem in the 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 [11]. The following lower bound for $d_{n,\Delta}$ has been proven in [4].

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

In our problem $n = 16$ and $\Delta = 4$, thus based on (4) we have $d_{16,4} \geq \lceil log_4 52\rceil - 1 = 3$. This confirms Proposition 2, i.e., the best possible value for DR_{max} is 3. In [8] Imase and Itoh proposed and 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 [8], we present a procedure to construct a diregular digraph with 16 nodes, diameter 2, and degree 4 as below.

```
1: Input : A set of 16 nodes V = {v_0, v_1, ..., v_15}
2: Output : A set A of 64 directed arcs (v_i, v_j)
3: Δ = 4   % Δ is the degree of the diregular digraph
4: n = 16   % n is the order of the digraph
5: A = ∅
6: α = 0, 1, ..., Δ − 1
7: for i = 0 to n − 1 do
8:     for j = 0 to n − 1 do
9:         if j = Δ × i + α mod n  then
10:            A = A ∪ (v_i, v_j)
11:        end if
12:    end for
13: end for
```

Fig. 2. The procedure to construct a 4-diregular digraph of order 16 and diameter 2

As Table 1 shows, the result of this procedure on the state cube is that 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 assigned. On the other hand, each column is assigned for times as a destination column. Thus, the total number of options for the four destination bytes of each of the columns $\{0,1,2,3\}$ is 4^4, while for each of the four columns $\{4,5,6,7\}$ is 3^4, for each of the four columns $\{8,9,10,11\}$ is 2^4, and for the each of the remaining four columns is 1. So the total number of optimum permutations defined in the algorithm of Figure 1, is $(4!)^4 = 331776$.

One can easily check that neither θ_1 nor θ_2 are optimum byte permutations. However, among the 331776 optimum shuffles produced by the above procedure, several shuffles are more easily imaginable: (1) First transposing the four horizontal slices and then applying the θ_1 transformation, (2) first applying the θ_2 transformation and then transposing the vertical slices $y = 1,2,3,4$, and (3) first transposing the horizontal slices and then transposing the vertical (no matter with fixed x or y) slices.

Table 1. Destination columns computed based on Algorithm in Figure 2 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	$\{12,13,14,15\}$

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: First a byte shuffle permutes the 64 bytes of the state cube and then, as a local diffusion, a 4×4 MDS matrix 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 different byte shuffles in odd and even rounds which in turn increases the 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 331776 optimum byte shuffles in the $4 \times 4 \times 4$ cubic state. We believe that using the same byte shuffle for every round of 3D, instead of θ_1 and θ_2, will improve the hardware performance of the cipher.

The typical methods used in the hardware implementation of a block cipher include basic iterated, partial loop unrolling, and full loop unrolling architectures, each of them can have several level of pipelining [7]. Where there exists a limit on the maximum area of a cryptographic unit, the round iterated architecture is the best choice. Regard to 3D, due to the big size of the state, the choice of iterated round is more preferable. One round of the cipher is implemented as a combinational logic and supplemented with a single register and a multiplexer. In the first clock cycle, input block of data is fed to the circuit through the multiplexer and stored in the register. In each subsequent clock cycle, one round of the cipher is evaluated, the result is fed back to the circuit through the multiplexer, and stored in the register. However, since the odd and even rounds of the 3D uses two different shuffles, some reforms in the iterated architecture is inevitable. We can iterate at least two rounds or implement one round with an additional multiplexer to select between the two 64-byte outputs of θ_1 and θ_2. Compared with the area required to implement one round of 3D, the former solution increases the area by a factor of two, and the latter increases the area of a big multiplexer. Moreover, the latter solution imposes the latency of the multiplexer and consequently will cause a decrease in the encryption/decryption process. So, modifying the 3D by our proposed shuffles

will improve the area in the former iterated architecture and both area and latency in the latter iterated architecture.

3D with its cubic state is a natural extension of the AES whose state is a 4×4 matrix. So, the next step in extending the AES may be design of a (for example 2048-bit) block cipher whose state is a hypercube, i.e., four parallel cubes. In this case, we can use the same subkey addition, 8-bit S-boxes, and MDS matrices used in AES or 3D. But we need to design a 256 to 256 byte shuffle. Using our method to model the diffusion property of an optimum shuffle 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 64, diameter 3, and still degree 4. Customizing the procedure of Section 3.2 for $n = 64$ and $\Delta = 4$ determines such an optimum permutation over columns which easily gives us the desired byte shuffles.

References

1. Barreto, P.S.L.M., Rijmen, V.: The ANUBIS Block Cipher. In: 1st NESSIE Workshop, Heverlee, Belgium (2000)
2. Benadjila, R., Billet, O., Gilbert, H., Macario-Rat, G., Peyrin, T., Robshaw, M., Seurin, Y., SHA-3 Proposal: ECHO (version 1.5), SHA-3 submission (2009)
3. Biryukov, A.: The Design of a Stream Cipher LEX, Proceedings of Selected Areas in Cryptography 2006, LNCS, vol. 4356, pp. 67-75. Springer, Heidelberg (2007)
4. Bridges, W.G., Toueg, S.: On the impossibility of directed Moore graphs. Journal of Combinatorial Theory, Series B29, No. 3, pp. 339–341. (1980)
5. Daemen, J., Rijmen, V.: A New MAC Construction ALRED and a Specific Instance, ALPHA-MAC. Fast Software Encryption 2005, LNCS, vol. 3557, pp. 1-17. Springer, Heidelberg (2005)
6. Daemen, J., Rijmen, V.: The design of Rijndael: AES– the Advanced Encryption Standard. Springer (2002)
7. Gaj, K., Chodowiec, P.: FPGA and ASIC Implementations of AES, Chapter 10. In: Koc, C.K. (ed.), Cryptographic Engineering, pp. 235-320, Springer, Dec. 2008. ISBN-10: 0387718168. ISBN-13: 978-0387718163.
8. Imase, M., Itoh, M.: Design to Minimize diameter on Building-Block Network. IEEE Trans. on Computers, vol. C-30, pp. 439–442, June 1981.
9. Koyama, T., Wang, L., Sasaki, Y., Sakiyama, K., Ohta, K.: New Truncated Differential Cryptanalysis on 3D Block Cipher. In: Ryan, M.D., Smyth, B., Wang, G. (eds.), ISPEC 2012. LNCS 7232, pp. 109-125. Springer, Heidelberg (2012)
10. Massey, J.: On the Optimality of SAFER+ Diffusion. In: Second AES Candidate Conference. National Institute of Standards and Technology (1999)
11. Miller, M., Siran, J.: Moore graphs and beyond: A survey of the degree/diameter problem. The Electronic Journal of Combinatorics, Dynamic survey D (2005)
12. Nakahara Jr, J.: New Impossible Differential and Known-Key Distinguishers for the 3D Cipher. In: Bao, F., Weng, J. (eds.), ISPEC 2011. LNCS, vol. 6672, pp. 208-221. Springer, Heidelberg (2011)
13. Nakahara Jr., J.: 3D: a Three-Dimensional Block Cipher. In: Franklin, M.K., Hui, L.C.K., Wong, D.S. (eds.) CANS 2008. LNCS, vol. 5339, pp. 252-267. Springer, Heidelberg (2008)
14. Suzuki, T., Minematsu, K., Improving the Generalized Feistel. In: Hong, S., Iwata, T. (eds.), FSE 2010. LNCS, vol. 6147, pp. 19–39. Springer, Heidelberg (2010)