# 3D: A Three-Dimensional Block Cipher

Jorge Nakahara Jr.

École Polytechnique Fédérale de Lausanne
EPFL, 1015 Lausanne, Switzerland
jorge_nakahara@yahoo.com.br

**Abstract.** The main contribution of this paper is a new iterated secret-key block cipher called 3D, inspired by the AES cipher. The 3D cipher has an SPN design, operates on 512-bit blocks, uses 512-bit keys, iterates 22 rounds, and employs a 3-dimensional state, instead of the 2-dimensional matrix of the AES. The main innovation of 3D includes the multi-dimensional state, generalizing the design of Rijndael, and allowing block sizes beyond the 256-bit boundary. This features motivates the use of 3D as a building block for compression functions in hash functions, MAC and stream cipher constructions requiring large internal states. We explain the design decisions and discuss the security of 3D under several attack settings.

**Keywords:** block cipher design, 3-dimensional state.

## 1 Introduction

Secret-key ciphers, such as block and stream ciphers, are designed for fast encryption of large volumes of data. This paper describes a block cipher called 3D, inspired by the design of the AES [16] and with some innovative designs. In the AES, plaintext, ciphertext, subkeys and intermediate data blocks are represented by a 2-dimensional 4× Nb state matrix of bytes, where Nb is the number of 32-bit words in a text block. For example, the state matrix of a $4t$-byte text block, $A = (a_0, a_1, a_2, \ldots, a_{4t-1})$, can be represented

$$\text{State matrix} = \begin{pmatrix} a_0 & a_4 & \ldots & a_{4t-4} \\ a_1 & a_5 & \ldots & a_{4t-3} \\ a_2 & a_6 & \ldots & a_{4t-2} \\ a_3 & a_7 & \ldots & a_{4t-1} \end{pmatrix} , \qquad (1)$$

with bytes inserted columnwise. This state matrix provides not only a compact representation of the plaintext and ciphertext blocks, but was also motivated by two round transformations in Rijndael: ShiftRows and MixColumns [16]. The former explicitly operates on the rows of the state, while the latter operates only on the columns of the state.

M.K. Franklin, L.C.K. Hui, and D.S. Wong (Eds.): CANS 2008, LNCS 5339, pp. 252–267, 2008.

In Rijndael, the block size is variable and ranges from 128 up to 256 bits in steps of 32 bits [16,26]. In the AES, complete text diffusion is achieved in two rounds, due to a combination of ShiftRows and MixColumns over a 4 × 4 state matrix. Key diffusion, though, takes longer depending on the key size. As the block size increases, it takes more rounds to guarantee fast diffusion for both text and key bits. This may be a reason for the upperbound of 256 bits for the block size in AES. This fact motivates our research, leading to 3D, with a larger block size (512 bits) which makes it attractive as a building block in the Miyaguchi-Preneel, Davies-Meyer or Matyas-Meyer-Oseas construction of compression functions (in this setting, it can be compared to SHA-512 [15]) in hash functions [29, p.340], and for stream modes of operation (OFB, CFB) whose security depends on the size of the internal cipher state, and in pseudorandom number generators [29, p.173].

This paper is organized as follows: Sect. 2 describes the new block cipher 3D; Sect. 3 describes the key schedule algorithm of 3D; Sect. 4 shows a security analyses of 3D; Sect. 5 estimates the software performance of 3D; Sect. 6 concludes the paper.

## 2 The 3D Block Cipher

The 3D block cipher operates on 512-bit blocks and uses 512-bit keys, both of which are represented as a 4 × 4 × 4 state of bytes (a 3-dimensional cube). The state for a 64-byte data block, $A = (a_0, a_1, \ldots, a_{63})$, is denoted

$$\text{State} = \left(\begin{array}{cccc|cccc|cccc|cccc} a_0 & a_4 & a_8 & a_{12} & a_{16} & a_{20} & a_{24} & a_{28} & a_{32} & a_{36} & a_{40} & a_{44} & a_{48} & a_{52} & a_{56} & a_{60} \\ a_1 & a_5 & a_9 & a_{13} & a_{17} & a_{21} & a_{25} & a_{29} & a_{33} & a_{37} & a_{41} & a_{45} & a_{49} & a_{53} & a_{57} & a_{61} \\ a_2 & a_6 & a_{10} & a_{14} & a_{18} & a_{22} & a_{26} & a_{30} & a_{34} & a_{38} & a_{42} & a_{46} & a_{50} & a_{54} & a_{58} & a_{62} \\ a_3 & a_7 & a_{11} & a_{15} & a_{19} & a_{23} & a_{27} & a_{31} & a_{35} & a_{39} & a_{43} & a_{47} & a_{51} & a_{55} & a_{59} & a_{63} \end{array}\right), \quad (2)$$

with bytes inserted columnwise. Each square set of 16 bytes is called a slice of the state (Fig. 1). Since all three dimensions of the state are equal, we set an orientation in (2): the set ($a_0$, $a_1$, …, $a_{15}$) represents the front slice or first vertical slice; the set $(a_{16}, a_{17}, \ldots, a_{31})$ represents the second vertical slice, and so on. These slices are relevant for operation $\theta_1$, described later. Other vertical slices exist, such as ($a_0$, $a_1$, $a_2$, $a_3$, $a_{16}$, $a_{17}$, $a_{18}$, $a_{19}$, $a_{32}$, $a_{33}$, $a_{34}$, $a_{35}$, $a_{48}$, $a_{49}$, $a_{50}$, $a_{51}$), which is relevant for operation $\theta_2$, described later.

A reason for the 512-bit user key is that key-recovery attacks applied either on top or at the bottom of a given distinguisher will have to recover 512 subkey bits with a complexity of $2^{512}$, which is about the exhaustive key search effort, and the same size of the codebook. If the user key was larger, say 1024 bits, shortcut attacks would become less expensive.

The round transformations in 3D are denoted:

- $\kappa_i$: a 4 × 4 × 4 state of bytes representing the 512-bit $i$-th round subkey is exclusive-ored bytewise to the $i$-th round state; the exclusive-or operation is an involution, and does not seem susceptible to weak keys/subkeys [11];

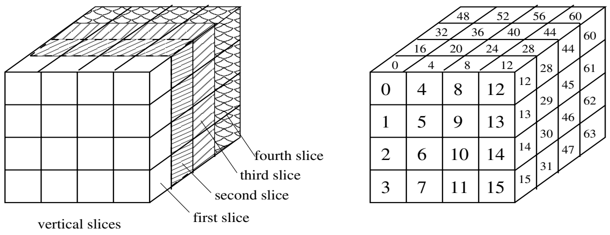

**Fig. 1.** 3D state with vertical slices and byte numbering

- $\gamma$: this nonlinear operation is responsible for the confusion property [35] in 3D, and consists of the bytewise application of the AES S-box to all bytes of the state;
- $\theta_1$, $\theta_2$: these diffusion operations [35] are applied in alternate rounds in 3D. They are identical to ShiftRows in AES, but since the state is 3-dimensional, two different sets of vertical slices of the state (Fig. 1) are affected in turn. $\theta_1$ operates on the vertical slices in Fig. 1, and turn (2) into

$$\left(\begin{array}{cccc|cccc|cccc|cccc} \mathsf{a}_0 & a_4 & a_8 & a_{12} & \mathsf{a}_{16} & a_{20} & a_{24} & a_{28} & \mathsf{a}_{32} & a_{36} & a_{40} & a_{44} & \mathsf{a}_{48} & a_{52} & a_{56} & a_{60} \\ a_5 & a_9 & a_{13} & \mathsf{a}_1 & a_{21} & a_{25} & a_{29} & \mathsf{a}_{17} & a_{37} & a_{41} & a_{45} & \mathsf{a}_{33} & a_{53} & a_{57} & a_{61} & \mathsf{a}_{49} \\ a_{10} & a_{14} & \mathsf{a}_2 & a_6 & a_{26} & a_{30} & \mathsf{a}_{18} & a_{22} & a_{42} & a_{46} & \mathsf{a}_{34} & a_{38} & a_{58} & a_{62} & \mathsf{a}_{50} & a_{54} \\ a_{15} & \mathsf{a}_3 & a_7 & a_{11} & a_{31} & \mathsf{a}_{19} & a_{23} & a_{27} & a_{47} & \mathsf{a}_{35} & a_{39} & a_{43} & a_{63} & \mathsf{a}_{51} & a_{55} & a_{59} \end{array}\right); \qquad (3)$$

$\theta_2$ operates similarly, but transforms (2) into

$$\left(\begin{array}{cccc|cccc|cccc|cccc} \mathsf{a}_0 & a_4 & a_8 & a_{12} & \mathsf{a}_{16} & a_{20} & a_{24} & a_{28} & \mathsf{a}_{32} & a_{36} & a_{40} & a_{44} & \mathsf{a}_{48} & a_{52} & a_{56} & a_{60} \\ \mathsf{a}_{17} & a_{21} & a_{25} & a_{29} & \mathsf{a}_{33} & a_{37} & a_{41} & a_{45} & \mathsf{a}_{49} & a_{53} & a_{57} & a_{61} & \mathsf{a}_1 & a_5 & a_9 & a_{13} \\ \mathsf{a}_{34} & a_{38} & a_{42} & a_{46} & \mathsf{a}_{50} & a_{54} & a_{58} & a_{62} & \mathsf{a}_2 & a_6 & a_{10} & a_{14} & \mathsf{a}_{18} & a_{22} & a_{26} & a_{30} \\ \mathsf{a}_{51} & a_{55} & a_{59} & a_{63} & \mathsf{a}_3 & a_7 & a_{11} & a_{15} & \mathsf{a}_{19} & a_{23} & a_{27} & a_{31} & \mathsf{a}_{35} & a_{39} & a_{43} & a_{47} \end{array}\right); \qquad (4)$$

- $\pi$: the $4 \times 4$ MDS matrix of the Anubis cipher [2] is applied to each column of every vertical slice of the state in (2). The branch number [16] of the Anubis matrix is 5 since it satisfies the MDS (Maximum Distance Separable) property [27]. Since the state is 3-dimensional, complete diffusion is achieved in three rounds, in combination with $\theta_1$ and $\theta_2$. This matrix is an involution, which guarantees the same diffusion power and computational cost for both the encryption and decryption operations. One matrix multiplication by a column of a slice of the state costs 4 xors and 5 xtimes, where xtimes means multiplication by 2 (or the polynomial $\mathsf{x}$) in GF($2^8$). Thus, one matrix multiplication by one slice costs 16 xors and 20 xtimes. For one state matrix the cost is 64 xors and 80 xtimes. Let an input slice to $\pi$ be denoted ($a_0$, $a_1$, ..., $a_{15}$), and the output slice be $(b_0, b_1, \ldots, b_{15})$. An example of the $\pi$ transformation for a single slice is

$$\begin{pmatrix} 01_x & 02_x & 04_x & 06_x \\ 02_x & 01_x & 06_x & 04_x \\ 04_x & 06_x & 01_x & 02_x \\ 06_x & 04_x & 02_x & 01_x \end{pmatrix} \cdot \begin{pmatrix} a_0 & a_4 & a_8 & a_{12} \\ a_1 & a_5 & a_9 & a_{13} \\ a_2 & a_6 & a_{10} & a_{14} \\ a_3 & a_7 & a_{11} & a_{15} \end{pmatrix} = \begin{pmatrix} b_0 & b_4 & b_8 & b_{12} \\ b_1 & b_5 & b_9 & b_{13} \\ b_2 & b_6 & b_{10} & b_{14} \\ b_3 & b_7 & b_{11} & b_{15} \end{pmatrix}, \qquad (5)$$

where the subscript $_\mathtt{x}$ denotes hexadecimal notation.

All round transformations in 3D operate bytewise. Bytes are treated as elements over $\mathrm{GF}(2^8) = \mathrm{GF}(2)[\mathrm{x}]/(m(x))$, where $m(x) = x^8 + x^4 + x^3 + x + 1$ is the same irreducible polynomial of AES. A polynomial $p(x) = \sum_{i=0}^{t} a_i \cdot x^i \in \mathrm{GF}(2)[x]$, with $a_i \in \mathrm{GF}(2)$, for $0 \leq i \leq t$, will be denoted by the numerical value $\sum_{i=0}^{t} a_i \cdot 2^i$, and is shortly represented in hexadecimal notation. For example, $m(x) = \mathtt{11B}_\mathtt{x}$.

The $i$-th full round of 3D, encrypting a text block $X$, is denoted $\tau_i(X) = \pi \circ \theta_{i \bmod 2+1} \circ \gamma \circ \kappa_i(X) = \pi(\theta_{i \bmod 2+1}(\gamma(\kappa_i(X))))$, namely function composition, $\circ$, operates in right-to-left order. The last round does not include $\pi$, and is denoted $\eta_{r-1}(X) = \theta_{(r-1) \bmod 2+1} \circ \gamma \circ \kappa_{r-1}(X)$. The inverse of a full round is $\tau_i^{-1}(X) = \kappa_i^{-1} \circ \gamma^{-1} \circ \theta_{i \bmod 2+1}^{-1} \circ \pi^{-1}(X)$, and the inverse of the last round is $\eta_{r-1}^{-1}(X) = \kappa_{r-1}^{-1} \circ \gamma^{-1} \circ \theta_{(r-1) \bmod 2+1}^{-1}$. Notice that $\kappa_i$ is the only key-dependent round operation, whereas $\gamma$, $\theta_1$, $\theta_2$ and $\pi$ are fixed key-independent transformations. Furthermore, there is an output transformation after $\eta_{r-1}$ consisting of $\kappa_r$, the $r$-th round subkey.

Properties of the round components include:

(a) $\kappa_i = \kappa_i^{-1}$, because the exclusive-or operation is an involution;
(b) $\gamma^{-1} \neq \gamma$ because the AES S-box is not an involution but has order 277182, namely, $\gamma^{277182}[x] = x$, $\forall x \in \mathrm{GF}(2^8)$ (see [34]);
(c) $\theta_i \neq \theta_i^{-1}$, $i \in \{1, 2\}$, because the inverse of $\theta_i$ requires displacing rows in the opposite direction in each slice; the order of $\theta_i$ is 4, that is, $\theta_i^4(X) = X$, for $i = 1, 2$;
(d) $\pi = \pi^{-1}$, because the Anubis matrix is an involution;
(e) $\gamma \circ \theta_i = \theta_i \circ \gamma$, namely, $\gamma$ and $\theta_i$ commute, for $i \in \{1, 2\}$, since both operate bytewise; similarly, $\gamma^{-1} \circ \theta_i^{-1} = \theta_i^{-1} \circ \gamma^{-1}$;
(f) $\kappa_i \circ \pi = \pi \circ \kappa_i'$, where $\kappa_i' = \pi^{-1}(\kappa_i)$;
(g) $\kappa_i \circ \theta_{i \bmod 2+1} = \theta_{i \bmod 2+1} \circ \kappa_i^*$, where $\kappa_i^* = \theta_{i \bmod 2+1}^{-1}(\kappa_i)$;
(h) $\kappa_i \circ \gamma \neq \gamma \circ \kappa_i$ because $\gamma$ is a non-linear operation with respect to exclusive-or;
(i) $\pi \circ \theta_{i \bmod 2+1} \neq \theta_{i \bmod 2+1} \circ \pi$ because $\pi$ operates on columns of the state while $\theta_{i \bmod 2+1}$ operates on rows of the state.

Using these properties, one can prove that the encryption and decryption frameworks of 3D are similar. Consider the $r$-round 3D encryption of a plaintext block $P$, resulting in the ciphertext block

$$C = \kappa_r \circ \eta_{r-1} \circ \bigcirc_{i=0}^{r-2} \tau_i(P) =$$
$$\kappa_r \circ \theta_{(r-1) \bmod 2+1} \circ \gamma \circ \kappa_{r-1} \circ \bigcirc_{i=0}^{r-2} (\pi \circ \theta_{i \bmod 2+1} \circ \gamma \circ \kappa_i)(P)\,. \qquad (6)$$

The decryption scheme is

$$P = \bigcirc_{i=r-2}^{0} \tau_i^{-1} \circ \eta_{r-1}^{-1} \circ \kappa_r^{-1}(C)\,, \tag{7}$$

which can be expressed as $P = \bigcirc_{i=r-2}^{0}(\kappa_i^{-1} \circ \gamma^{-1} \circ \theta_{i \bmod 2+1}^{-1} \circ \pi^{-1}) \circ \eta_{r-1}^{-1} \circ \kappa_r^{-1}(C)$. From (a) and (d): $P = \bigcirc_{i=r-2}^{0}(\kappa_i \circ \gamma^{-1} \circ \theta_{i \bmod 2+1}^{-1} \circ \pi) \circ \eta_{r-1}^{-1} \circ \kappa_r(C)$. From (e) and (f): $P = \bigcirc_{i=r-2}^{0}(\kappa_i \circ \theta_{i \bmod 2+1}^{-1} \circ \gamma^{-1} \circ \pi) \circ \kappa_{r-1} \circ \theta_{(r-1) \bmod 2+1}^{-1} \circ \gamma^{-1} \circ \kappa_r(C) = \kappa_0 \circ \theta_1^{-1} \circ \gamma^{-1} \circ \bigcirc_{i=r-1}^{1}(\pi \circ \kappa_i \circ \theta_{i \bmod 2+1}^{-1} \circ \gamma^{-1}) \circ \kappa_r(C) = \kappa_0 \circ \theta_1^{-1} \circ \gamma^{-1} \circ \bigcirc_{i=r-1}^{1}(\kappa_i^* \circ \pi \circ \theta_{i \bmod 2+1}^{-1} \circ \gamma^{-1}) \circ \kappa_r(C) = \kappa_0 \circ \theta_1^{-1} \circ \gamma^{-1} \circ \kappa_1^* \circ \bigcirc_{i=r}^{1}(\pi \circ \theta_{(i-1) \bmod 2+1}^{-1} \circ \gamma^{-1} \circ \kappa_i^*)(C)$,

where $\kappa_r^* = \kappa_r$ and $\kappa_i^* = \pi(\kappa_i)$, for $i < r$. Thus, the encryption (6) and decryption (7) frameworks are similar, except for the order of some round subkeys, and some inverse transformations. Consequently, both schemes have the same cryptographic strength [21].

Properties (h) and (i) show that the round subkeys cannot be moved around or sorted out from the other round transformations because of the non-commutativity property. Thus, it is not possible to arbitrarily remove key-independent cipher operations, such as $\gamma$, $\theta_1$, $\theta_2$ and $\pi$.

The suggested number of rounds for 3D is 22. This decision is in line with Rijndael, where the block size ranges from 128 up to 256 bits, and roughly one round is added for every additional 32 bits in the block size. Thus, assuming Rijndael with 256-bit block iterates 14 rounds, 3D iterates 22 rounds. This number of rounds is more than enough to counter the attacks described in Sect. 4, and further, there is still a large margin of security. For performance comparison, the AES operates on 128-bit blocks, and iterates (up to) 14 rounds; 3D encrypts four times more texts at a time but iterates $8/14 \approx 57\%$ more rounds.

## 3 Key Schedule of 3D

For $r$-round 3D encryption and decryption operations, $(r+1)$ 512-bit subkeys are needed. The number of rounds is $r = 22$, as explained in Sect. 2.

The key schedule works as follows:

- the 512-bit user key $K = (k_0, k_1, \ldots, k_{63})$ becomes the first round subkey;
- in [36], Wu described an attack on block ciphers with a variable number of rounds. A suggested countermeasure is to combine the number of rounds in the cipher, for instance, in the key schedule, so that cipher instances with different number of rounds are not useful for this attack. This suggestion has been adopted in 3D. Let $\kappa^*$ represent the exclusive-or of the subkey state with a constant $4 \times 4 \times 4$ matrix depending on the number of rounds, $r$, and the Anubis matrix:

$$\left(\begin{array}{cccc|cccc|cccc|cccc} r & 2r & 4r & 6r & 2r & r & 6r & 4r & 4r & 6r & r & 2r & 6r & 4r & 2r & r \\ 2r & r & 6r & 4r & 4r & 6r & r & 2r & 6r & 4r & 2r & r & r & 2r & 4r & 6r \\ 4r & 6r & r & 2r & 6r & 4r & 2r & r & r & 2r & 4r & 6r & 2r & r & 6r & 4r \\ 6r & 4r & 2r & r & r & 2r & 4r & 6r & 2r & r & 6r & 4r & 4r & 6r & r & 2r \end{array}\right), \qquad (8)$$

where multiplication is in $\mathrm{GF}(2^8)$; these constants are used to avoid patterns in the user key to propagate to round subkeys. Without these constants, a user key with all bytes equal could lead to subkeys with all bytes equal, for instance. It could make 3D susceptible to related-key [4], slide or advanced slide attacks [8], independent of the number of rounds.

- the remaining round subkeys are computed as $K_i = \pi \circ \theta_{i \bmod 2+1} \circ \gamma' \circ \kappa^*(K_{i-1})$, $i \geq 1$, and $K_0 = K$. The transformation $\gamma'$ consists of the bytewise application of the AES S-box to alternate columns of the state as follows:

$$\left(\begin{array}{cccc|cccc|cccc|cccc} S[a_0] & a_4 & a_8 & a_{12} & a_{16} & S[a_{20}] & a_{24} & a_{28} & a_{32} & a_{36} & S[a_{40}] & a_{44} & a_{48} & a_{52} & a_{56} & S[a_{60}] \\ S[a_1] & a_5 & a_9 & a_{13} & a_{17} & S[a_{21}] & a_{25} & a_{29} & a_{33} & a_{37} & S[a_{41}] & a_{45} & a_{49} & a_{53} & a_{57} & S[a_{61}] \\ S[a_2] & a_6 & a_{10} & a_{14} & a_{18} & S[a_{22}] & a_{26} & a_{30} & a_{34} & a_{38} & S[a_{42}] & a_{46} & a_{50} & a_{54} & a_{58} & S[a_{62}] \\ S[a_3] & a_7 & a_{11} & a_{15} & a_{19} & S[a_{23}] & a_{27} & a_{31} & a_{35} & a_{39} & S[a_{43}] & a_{47} & a_{51} & a_{55} & a_{59} & S[a_{63}] \end{array}\right). \qquad (9)$$

The encryption subkey generation can be performed on-the-fly. Storing the last round subkey, $K_r$, instead of $K$ allows on-the-fly decryption subkey generation since all key schedule operations are invertible: $K_i = \kappa^* \circ \gamma'^{-1} \circ \theta^{-1}_{i \bmod 2+1} \circ \pi(K_{i+1})$, $0 \leq i < 22$.

Due to the similarity with the encryption framework, it can be shown that complete key diffusion is achieved after three subkeys are generated, that is, every byte of $K_3$ already depends on every byte of $K_0$.

As for performance, notice that the key schedule costs slightly less than a single encryption, although both use similar operations.

## 4 Security Analyses

In the following, we analyse 3D under several attack settings.

### 4.1 Plaintext Leakage

Due to the birthday paradox [29], after about $2^{n/2}$ encryptions, either in ECB or CBC modes, an $n$-bit block cipher starts to leak information about the plaintext [21], in a ciphertext-only (CO) setting. For 3D, this leakage happens after $2^{512/2} = 2^{256}$ block encryptions (or decryptions), which sets an upper-bound on the number of plaintext blocks encrypted before the key has to be changed.

### 4.2 Related-Key Attack

In [3], Biham developed an attack method on arbitrary $n$-bit block ciphers, that depends only on the key size. Thus, his attack is also independent of the number of rounds. This attack is supported by the birthday paradox, and has complexity $2^{k/2}$ encryptions, for a $k$-bit user key. Even though for 3D the key size is equal to the block size, the corresponding attack complexity is $2^{512/2} = 2^{256}$, which matches the attack complexity in Sect. 4.1.

There are many kinds of related-key attacks, such as in Sect. 4.5 and [4,20], and all of them depend on the design of the key schedule algorithm. The key schedule of 3D shares components with its encryption framework (Sect. 3). It implies that (xor) difference propagation works similarly in both schemes, which is relevant for related-key attacks, where the adversary cannot choose the key, but knows or can choose a relationship between keys used for encryption. In particular, it takes three (full) rounds for any single byte difference to spread across the full key state, that is, a single byte difference in the user key(s) will affect the (full) third round subkey and beyond. Consequently, related-key attacks are not expected to be effective against 3D, since any nonzero difference in the key spreads to the entire key state after the third subkey (comparatively, complete diffusion in the key schedule of AES takes six or more rounds depending on the key size).

### 4.3 Non-surjective and Davies' Attacks

Non-surjective attacks on block ciphers have been suggested by Rijmen *et al.* in [33], motivated by non-surjective round functions in Feistel ciphers, such as CAST and Khufu [30]. Similarly, Davies' attack [13] exploit the Feistel structure of DES, and subkey bits shared between neighboring S-boxes. The 3D cipher follows an SPN design, and not only its round function, but also its internal components are bijective mappings. Moreover, no subkey bits are duplicated or shared among S-boxes. Therefore, non-surjective attacks do not apply to 3D.

### 4.4 Interpolation, Higher-Order Differential and $\chi^2$ Attacks

In [17], Jakobsen and Knudsen described attacks on a cipher called PURE and on a variant of the SHARK block cipher [32]. In both cases the attacks were made possible because the ciphers had a compact algebraic (polynomial/rational) expression which could be solved with manageable complexity (up to a certain number of rounds). We have not found any compact (polynomial) representation of round function of 3D (over GF($2^8$)), or of its round components which leads to an effective attack (to the full 22-round 3D). We do not consider expressions such as in [14], which although compact, did not lead to an effective attack on AES. Analogously, because of the non-linear order of the S-box, we do not expect higher-order differential attacks [23,25] to succeed against 3D. Following a similar reasoning, we do not expect $\chi^2$ attacks [24] nor mod-n attacks [19] to apply to 3D.

### 4.5 Slide and Advanced-Slide Attacks

In [7,8], the slide and advanced-slide attacks were described against Feistel ciphers whose key schedules had a periodic behavior. Moreover, in these attacks, symmetries in the cipher framework, suggested that this structure could be twisted and slided in order to partially match another copy of itself. Thus, these attacks depend on a self-similarity in the cipher structure, and a degree of periodicity in the key schedule. In 3D, the key schedule was designed to avoid patterns in the user key to propagate to the subkeys, including the periodicity necessary in [7,8]. Moreover, there is a round asymmetry due to $\theta_1$ and $\theta_2$. We conclude that such attacks do not apply to 3D (Sect. 2).

### 4.6 Truncated Differential Analysis

As a preliminary differential analysis [5], consider truncated differentials [23], such as (10), where '$\varDelta$' stands for an arbitrary nonzero exclusive-or byte difference, while '0' stands for a zero byte difference.

$$\begin{gathered}
\left(\begin{array}{cccc|cccc|cccc|cccc}
\varDelta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\varDelta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\varDelta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\varDelta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)
\overset{\pi\circ\theta_1\circ\gamma\circ\kappa_0}{\longrightarrow}
\left(\begin{array}{cccc|cccc|cccc|cccc}
\varDelta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right) \\
\overset{\pi\circ\theta_2\circ\gamma\circ\kappa_1}{\longrightarrow}
\left(\begin{array}{cccc|cccc|cccc|cccc}
\varDelta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\varDelta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\varDelta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\varDelta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)
\overset{\theta_1\circ\gamma\circ\kappa_2}{\longrightarrow} \\
\left(\begin{array}{cccc|cccc|cccc|cccc}
\varDelta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \varDelta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \varDelta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \varDelta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)
\overset{\pi}{\longrightarrow}
\left(\begin{array}{cccc|cccc|cccc|cccc}
\varDelta & \varDelta & \varDelta & \varDelta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\varDelta & \varDelta & \varDelta & \varDelta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\varDelta & \varDelta & \varDelta & \varDelta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\varDelta & \varDelta & \varDelta & \varDelta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right) \\
\overset{\theta_2\circ\gamma\circ\kappa_3}{\longrightarrow}
\left(\begin{array}{cccc|cccc|cccc|cccc}
\varDelta & \varDelta & \varDelta & \varDelta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \varDelta & \varDelta & \varDelta & \varDelta \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \varDelta & \varDelta & \varDelta & \varDelta & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \varDelta & \varDelta & \varDelta & \varDelta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)
\overset{\pi}{\longrightarrow} \\
\left(\begin{array}{cccc|cccc|cccc|cccc}
\varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta \\
\varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta \\
\varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta \\
\varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta & \varDelta
\end{array}\right)
\end{gathered} \tag{10}$$

The 4-round truncated differential (10) demonstrates that complete text diffusion in 3D is achieved in exactly three rounds (see the last three rounds). There are 25 active S-boxes [9] in (10), and this fact is independent of the position of the single $\varDelta$ byte difference after the first round. Analogously, this behaviour is independent of (10) starting with a round using $\theta_1$ or $\theta_2$. Notice that (10) holds with probability $2^8/2^{32} = 2^{-24}$, due to the difference propagation after the first round, where four byte differences turn into a single byte difference. The remaining difference propagation patterns hold with certainty. Comparatively, for 4-round AES there are also at least 25 active S-boxes. These figures show that 3D has the same expected resistance to differential cryptanalysis (DC) as the AES.

An advantage of truncated differentials (compared to conventional differential characteristics) is that the probability of the former is independent of the S-boxes used in the cipher.

Consider the (hypothetical) 2-round iterative truncated differential (11), that holds with probability $(2^8/2^{32})^4 = 2^{-96}$. This probability accounts for the $\theta_2$ transformation in which four nonzero byte differences in the same column become a single output difference after each slice of the state. Each such event has probability $2^8/2^{23} = 2^{-24}$.

$$\left(\begin{array}{cccc|cccc|cccc|cccc} \varDelta&0&0&0&\varDelta&0&0&0&\varDelta&0&0&0&\varDelta&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{array}\right) \xrightarrow{\pi\circ\theta_1\circ\gamma\circ\kappa_0} \left(\begin{array}{cccc|cccc|cccc|cccc} \varDelta&0&0&0&\varDelta&0&0&0&\varDelta&0&0&0&\varDelta&0&0&0\\ \varDelta&0&0&0&\varDelta&0&0&0&\varDelta&0&0&0&\varDelta&0&0&0\\ \varDelta&0&0&0&\varDelta&0&0&0&\varDelta&0&0&0&\varDelta&0&0&0\\ \varDelta&0&0&0&\varDelta&0&0&0&\varDelta&0&0&0&\varDelta&0&0&0 \end{array}\right)$$

$$\xrightarrow{\pi\circ\theta_2\circ\gamma\circ\kappa_1} \left(\begin{array}{cccc|cccc|cccc|cccc} \varDelta&0&0&0&\varDelta&0&0&0&\varDelta&0&0&0&\varDelta&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{array}\right) \tag{11}$$

For a random permutation the output difference of (11) would appear with probability about $(2^{-8})^{48} = 2^{-384}$ because there are 48 zero byte differences at the output. Thus, (11) is a distinguisher of 3D from a random permutation, for up to six rounds, with probability $(2^{-96})^3 = 2^{-288}$. Repeating (11) four times, namely, for eight rounds, leads to a probability of $(2^{-96})^4 = 2^{-384}$, which is the same as for a random permutation. One can also start the distinguisher with the state after $\pi\circ\theta_1\circ\gamma\circ\kappa_0$. That means that the iterative truncated differential start in an even round (with $\theta_2$). The results are analogous.

Suppose one makes a pool of $2^{32}$ chosen plaintexts (CP) in which the bytes in positions (0,16,32,48) of the state (2) range over all possible 32-bit values, while the remaining bytes are arbitrary constants. This pool leads to about $2^{63}$ text pairs (plaintext and ciphertexts). The output difference contains 60 zero byte differences. Thus, one expects that $2^{63}\cdot(2^{-8})^{60} = 2^{-417} < 1$ pairs satisfy the output difference of (11). This approach does not work.

Suppose one tries to guess the 16 bytes of $\mathrm{AK}_0$, and use pools of $2^{128}$ plaintexts with difference at bytes in positions (0, 5, 10, 15, 16, 21, 26, 31, 32, 37, 42, 47, 48, 53, 58, 63) of the state. Each such pool can lead to $2^{128}(2^{128}-1)/2 \approx 2^{255}$ text pairs. Still $2^{255}\cdot(2^{-8})^{60} = 2^{-225} < 1$ survives filtering by output difference of (11). Notice that due to the structure of 3D, with $\theta_1$ and $\theta_2$ in every other round, any iterative differential needs to have an even number of rounds, otherwise, it could not be concatenated to itself.

### 4.7 Linear Analysis

A linear distinguisher [28] for 3D would be similar to (10) except that $\varDelta$ is replaced by $\varGamma$, denoting a nonzero bitmask, while 0 denotes a zero (empty or trivial) bitmask. Thus, such linear distinguisher would reach three rounds with 21 active S-boxes, and according to [16], the associated bias would be $(2^{-4})^{21} = 2^{-84}$. For four rounds, the bias would be $(2^{-4})^{25} = 2^{-100}$. Comparatively, for the AES the number of active S-boxes across four rounds is at least 25. These figures show that 3D has the same expected resistance to linear cryptanalysis as the AES. The questions of linear hulls [31] and multiple linear relations [18] in 3D are left as open problems.

### 4.8 Multiset Analysis

Consider the first-order multiset [12,6] distinguisher in (12), where 'A' denotes an active byte, 'P' denotes a passive byte, 'B' denotes a balanced byte and '?' denotes an unpredictable byte exclusive-or sum. The distinguisher (12) reaches 4.25 rounds, or more precisely, $\kappa_4 \circ \pi \circ \theta_2 \circ \gamma \circ \kappa_3 \circ \pi \circ \theta_1 \circ \gamma \circ \kappa_2 \circ \pi \circ \theta_2 \circ \gamma \circ \kappa_1 \circ \pi \circ \theta_1 \circ \gamma \circ \kappa_0$.

$$
\begin{gathered}
\left(\begin{array}{cccc|cccc|cccc|cccc}
A&P&P&P&P&P&P&P&P&P&P&P&P&P&P&P\\
P&P&P&P&P&P&P&P&P&P&P&P&P&P&P&P\\
P&P&P&P&P&P&P&P&P&P&P&P&P&P&P&P\\
P&P&P&P&P&P&P&P&P&P&P&P&P&P&P&P
\end{array}\right) \xrightarrow{\pi\circ\theta_1\circ\gamma\circ\kappa_0} \\
\left(\begin{array}{cccc|cccc|cccc|cccc}
A&P&P&P&P&P&P&P&P&P&P&P&P&P&P&P\\
A&P&P&P&P&P&P&P&P&P&P&P&P&P&P&P\\
A&P&P&P&P&P&P&P&P&P&P&P&P&P&P&P\\
A&P&P&P&P&P&P&P&P&P&P&P&P&P&P&P
\end{array}\right) \xrightarrow{\theta_2\circ\gamma\circ\kappa_1} \\
\left(\begin{array}{cccc|cccc|cccc|cccc}
A&P&P&P&P&P&P&P&P&P&P&P&P&P&P&P\\
P&P&P&P&P&P&P&P&P&P&P&P&A&P&P&P\\
P&P&P&P&P&P&P&P&A&P&P&P&P&P&P&P\\
P&P&P&P&A&P&P&P&P&P&P&P&P&P&P&P
\end{array}\right) \xrightarrow{\pi} \\
\left(\begin{array}{cccc|cccc|cccc|cccc}
A&P&P&P&A&P&P&P&A&P&P&P&A&P&P&P\\
A&P&P&P&A&P&P&P&A&P&P&P&A&P&P&P\\
A&P&P&P&A&P&P&P&A&P&P&P&A&P&P&P\\
A&P&P&P&A&P&P&P&A&P&P&P&A&P&P&P
\end{array}\right) \xrightarrow{\theta_1\circ\gamma\circ\kappa_2} \\
\left(\begin{array}{cccc|cccc|cccc|cccc}
A&P&P&P&A&P&P&P&A&P&P&P&A&P&P&P\\
P&P&P&A&P&P&P&A&P&P&P&A&P&P&P&A\\
P&P&A&P&P&P&A&P&P&P&A&P&P&P&A&P\\
P&A&P&P&P&A&P&P&P&A&P&P&P&A&P&P
\end{array}\right) \xrightarrow{\pi} \\
\left(\begin{array}{cccc|cccc|cccc|cccc}
A&A&A&A&A&A&A&A&A&A&A&A&A&A&A&A\\
A&A&A&A&A&A&A&A&A&A&A&A&A&A&A&A\\
A&A&A&A&A&A&A&A&A&A&A&A&A&A&A&A\\
A&A&A&A&A&A&A&A&A&A&A&A&A&A&A&A
\end{array}\right) \xrightarrow{\kappa_4\circ\pi\circ\theta_2\circ\gamma\circ\kappa_3} \\
\left(\begin{array}{cccc|cccc|cccc|cccc}
B&B&B&B&B&B&B&B&B&B&B&B&B&B&B&B\\
B&B&B&B&B&B&B&B&B&B&B&B&B&B&B&B\\
B&B&B&B&B&B&B&B&B&B&B&B&B&B&B&B\\
B&B&B&B&B&B&B&B&B&B&B&B&B&B&B&B
\end{array}\right) \xrightarrow{\gamma}
\left(\begin{array}{cccc|cccc|cccc|cccc}
?&?&?&?&?&?&?&?&?&?&?&?&?&?&?&?\\
?&?&?&?&?&?&?&?&?&?&?&?&?&?&?&?\\
?&?&?&?&?&?&?&?&?&?&?&?&?&?&?&?\\
?&?&?&?&?&?&?&?&?&?&?&?&?&?&?&?
\end{array}\right)
\end{gathered} \tag{12}
$$

An attack on 4.75-round 3D using (12) would partially decrypt $\kappa_5 \circ \theta_1 \circ \gamma$, and recover $\kappa_5$ bytewise. The distinguisher (12) provides an 8-bit condition. After two $\lambda$-sets [12], there remains $2^8 \cdot (2^{-8})^2 < 1$ wrong subkey byte candidates. The cost per subkey byte is therefore, $2 \cdot 2^8 = 2^9$ chosen plaintexts (CP), $2^8 \cdot 2^8 + 2^8 \approx 2^{16}$ computations of $\kappa_5 \circ \theta_1 \circ \gamma$. That means $64 \cdot 2^{16} \cdot 0.75/4.25 \approx 2^{19.5}$ 4.75-round computations.

Consider now the higher-order multiset distinguisher (13) that uses $\lambda$-sets with $2^{128}$ texts. A byte belonging to a 128-bit active word is denoted $A^*$ to indicate that although the bytes are scattered across the state, they jointly form a 128-bit active word; similarly, a byte belonging to a 128-bit balanced word is denoted $B^*$; a byte belonging to a 128-bit passive word is denoted $P^*$; a byte belonging to a 128-bit even word is denoted $E^*$; different subscripts indicate different 128-bit words.

$$\left(\begin{array}{cccc|cccc|cccc|cccc} A^* & P & P & P & P & P & P & P & P & P & P & P & P & P & P & P \\ P & A^* & P & P & P & P & P & P & P & P & P & P & P & P & P & P \\ P & P & A^* & P & P & P & P & P & P & P & P & P & P & P & P & P \\ P & P & P & A^* & P & P & P & P & P & P & P & P & P & P & P & P \end{array}\right) \xrightarrow{\theta_1 \circ \gamma \circ \kappa_0}$$

$$\left(\begin{array}{cccc|cccc|cccc|cccc} A^* & P & P & P & P & P & P & P & P & P & P & P & P & P & P & P \\ A^* & P & P & P & P & P & P & P & P & P & P & P & P & P & P & P \\ A^* & P & P & P & P & P & P & P & P & P & P & P & P & P & P & P \\ A^* & P & P & P & P & P & P & P & P & P & P & P & P & P & P & P \end{array}\right) \xrightarrow{\theta_2 \circ \gamma \circ \kappa_1 \circ \pi}$$

$$\left(\begin{array}{cccc|cccc|cccc|cccc} A^* & P & P & P & P & P & P & P & P & P & P & P & P & P & P & P \\ P & P & P & P & P & P & P & P & P & P & P & P & A^* & P & P & P \\ P & P & P & P & P & P & P & P & A^* & P & P & P & P & P & P & P \\ P & P & P & P & A_* & P & P & P & P & P & P & P & P & P & P & P \end{array}\right) \xrightarrow{\pi}$$

$$\left(\begin{array}{cccc|cccc|cccc|cccc} E_1^* & P & P & P & E_2^* & P & P & P & E_3^* & P & P & P & E_4^* & P & P & P \\ E_1^* & P & P & P & E_2^* & P & P & P & E_3^* & P & P & P & E_4^* & P & P & P \\ E_1^* & P & P & P & E_2^* & P & P & P & E_3^* & P & P & P & E_4^* & P & P & P \\ E_1^* & P & P & P & E_2^* & P & P & P & E_3^* & P & P & P & E_4^* & P & P & P \end{array}\right) \xrightarrow{\theta_1 \circ \gamma \circ \kappa_2}$$

$$\left(\begin{array}{cccc|cccc|cccc|cccc} E_1^* & P & P & P & E_2^* & P & P & P & E_3^* & P & P & P & E_4^* & P & P & P \\ P & P & P & E_1^* & P & P & P & E_2^* & P & P & P & E_3^* & P & P & P & E_4^* \\ P & P & E_1^* & P & P & P & E_2^* & P & P & P & E_3^* & P & P & P & E_4^* & P \\ P & E_1^* & P & P & P & E_2^* & P & P & P & E_3^* & P & P & P & E_4^* & P & P \end{array}\right) \xrightarrow{\pi}$$

$$\left(\begin{array}{cccc|cccc|cccc|cccc} E_1^* & E_1^* & E_1^* & E_1^* & E_2^* & E_2^* & E_2^* & E_2^* & E_3^* & E_3^* & E_3^* & E_3^* & E_4^* & E_4^* & E_4^* & E_4^* \\ E_1^* & E_1^* & E_1^* & E_1^* & E_2^* & E_2^* & E_2^* & E_2^* & E_3^* & E_3^* & E_3^* & E_3^* & E_4^* & E_4^* & E_4^* & E_4^* \\ E_1^* & E_1^* & E_1^* & E_1^* & E_2^* & E_2^* & E_2^* & E_2^* & E_3^* & E_3^* & E_3^* & E_3^* & E_4^* & E_4^* & E_4^* & E_4^* \\ E_1^* & E_1^* & E_1^* & E_1^* & E_2^* & E_2^* & E_2^* & E_2^* & E_3^* & E_3^* & E_3^* & E_3^* & E_4^* & E_4^* & E_4^* & E_4^* \end{array}\right) \xrightarrow{\pi \circ \theta_2 \circ \gamma \circ \kappa_3}$$

$$\left(\begin{array}{cccc|cccc|cccc|cccc} A_1^* & A_1^* & A_1^* & A_1^* & A_2^* & A_2^* & A_2^* & A_2^* & A_3^* & A_3^* & A_3^* & A_3^* & A_4^* & A_4^* & A_4^* & A_4^* \\ A_1^* & A_1^* & A_1^* & A_1^* & A_2^* & A_2^* & A_2^* & A_2^* & A_3^* & A_3^* & A_3^* & A_3^* & A_4^* & A_4^* & A_4^* & A_4^* \\ A_1^* & A_1^* & A_1^* & A_1^* & A_2^* & A_2^* & A_2^* & A_2^* & A_3^* & A_3^* & A_3^* & A_3^* & A_4^* & A_4^* & A_4^* & A_4^* \\ A_1^* & A_1^* & A_1^* & A_1^* & A_2^* & A_2^* & A_2^* & A_2^* & A_3^* & A_3^* & A_3^* & A_3^* & A_4^* & A_4^* & A_4^* & A_4^* \end{array}\right) \xrightarrow{\kappa_5 \pi \circ \theta_1 \circ \gamma \circ \kappa_4}$$

$$\left(\begin{array}{cccc|cccc|cccc|cccc} B & B & B & B & B & B & B & B & B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B & B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B & B & B & B & B & B & B & B & B \\ B & B & B & B & B & B & B & B & B & B & B & B & B & B & B & B \end{array}\right) \xrightarrow{\gamma}$$

$$\left(\begin{array}{cccc|cccc|cccc|cccc} ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \end{array}\right) \tag{13}$$

Thus, (13) covers 5.25 rounds. An attack on 5.75-round 3D using (13) would partially decrypt $\kappa_6 \circ \theta_2 \circ \gamma$, and recover $\kappa_6$ bytewise, by comparing if each byte position before $\gamma$ is balanced. The distinguisher (13) provides an 8-bit condition per byte. After two $\lambda$-sets [12], there remains $2^8 \cdot (2^{-8})^2 < 1$ wrong subkey byte candidates. The cost per subkey byte is therefore, $2 \cdot 2^{128} = 2^{129}$ chosen plaintexts (CP), $2^8 \cdot 2^{128} = 2^{136}$ computations of $\kappa_6 \circ \theta_2 \circ \gamma$. That means $64 \cdot 2^{136} \cdot 0.75/5.75 \approx 2^{139}$ 5.75-round computations.

### 4.9 Impossible Differential Analysis

The impossible differential (ID) technique was formerly described in [22]. A 4.75-round impossible differential distinguisher of 3D is depicted in (14), where '$\Delta$' denotes a nonzero byte difference, '0' denotes a zero byte difference, and '?' denotes an unknown difference (can be zero or not). There are two truncated differentials in (14) that hold with certainty, one in the encryption direction, covering $\pi \circ \theta_2 \circ \gamma \circ \kappa_3 \circ \pi \circ \theta_1 \circ \gamma \circ \kappa_2 \circ \pi \circ \theta_2 \circ \gamma \circ \kappa_1$ and the other in the decryption direction, covering $\kappa_4 \circ \gamma^{-1} \circ \theta_1^{-1} \circ \pi \circ \kappa_5 \circ \gamma^{-1} \circ \theta_2^{-1}$. The contradiction in difference propagation (denoted $\not\rightarrow$ and $\not\leftarrow$) happens after the third $\pi$ layer:

there are four zero byte differences in the decryption direction after $\pi$, while all these bytes are nonzero before $\pi$. There are similar ID distinguishers that cause contradiction in the other slices of the state.

$$
\begin{gathered}
\left(\begin{array}{cccc|cccc|cccc|cccc}
\varDelta&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{array}\right)
\overset{\pi\circ\theta_2\circ\gamma\circ\kappa_1}{\longrightarrow}
\left(\begin{array}{cccc|cccc|cccc|cccc}
\varDelta&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
\varDelta&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
\varDelta&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
\varDelta&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{array}\right)\\
\overset{\theta_1\circ\gamma\circ\kappa_2}{\longrightarrow}
\left(\begin{array}{cccc|cccc|cccc|cccc}
\varDelta&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&\varDelta&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&\varDelta&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&\varDelta&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{array}\right)
\overset{\pi}{\longrightarrow}\\
\left(\begin{array}{cccc|cccc|cccc|cccc}
\varDelta&\varDelta&\varDelta&\varDelta&0&0&0&0&0&0&0&0&0&0&0&0\\
\varDelta&\varDelta&\varDelta&\varDelta&0&0&0&0&0&0&0&0&0&0&0&0\\
\varDelta&\varDelta&\varDelta&\varDelta&0&0&0&0&0&0&0&0&0&0&0&0\\
\varDelta&\varDelta&\varDelta&\varDelta&0&0&0&0&0&0&0&0&0&0&0&0
\end{array}\right)\\
\overset{\theta_2\circ\gamma\circ\kappa_3}{\longrightarrow}
\left(\begin{array}{cccc|cccc|cccc|cccc}
\varDelta&\varDelta&\varDelta&\varDelta&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&\varDelta&\varDelta&\varDelta&\varDelta\\
0&0&0&0&0&0&0&0&\varDelta&\varDelta&\varDelta&\varDelta&0&0&0&0\\
0&0&0&0&\varDelta&\varDelta&\varDelta&\varDelta&0&0&0&0&0&0&0&0
\end{array}\right)
\overset{\pi}{\longrightarrow}\\
\left(\begin{array}{cccc|cccc|cccc|cccc}
\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta\\
\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta\\
\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta\\
\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta
\end{array}\right)
\overset{\kappa_4\circ\gamma^{-1}\circ\theta_1^{-1}}{\not\leftarrow}\\
\left(\begin{array}{cccc|cccc|cccc|cccc}
0&?&?&?&?&?&?&?&?&?&?&?&?&?&?&?\\
0&?&?&?&?&?&?&?&?&?&?&?&?&?&?&?\\
0&?&?&?&?&?&?&?&?&?&?&?&?&?&?&?\\
0&?&?&?&?&?&?&?&?&?&?&?&?&?&?&?
\end{array}\right)
\overset{\pi}{\longleftarrow}
\left(\begin{array}{cccc|cccc|cccc|cccc}
0&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta\\
0&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta\\
0&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta\\
0&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta
\end{array}\right)\\
\overset{\kappa_5\circ\gamma^{-1}\circ\theta_2^{-1}}{\longleftarrow}
\left(\begin{array}{cccc|cccc|cccc|cccc}
0&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta\\
\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&0&\varDelta&\varDelta&\varDelta\\
\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&0&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta\\
\varDelta&\varDelta&\varDelta&\varDelta&0&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta&\varDelta
\end{array}\right)
\end{gathered}
\tag{14}
$$

Distinguisher (14) can be used to recover $\kappa_0$ in an attack on 5.75-round 3D, by placing (14) in the last 4.75 rounds. The attack would proceed as follows:

(a) choose a pool $2^{32}$ texts with all possible values in positions 0, 5, 10, 15 of the state, and arbitrary constants in the remaining byte positions. From one pool, one can generate $2^{32}(2^{32}-1)/2 \approx 2^{63}$ pairs with nonzero difference in these four byte positions, and zero difference in the remaining positions;

(b) from (14), about $2^{63} \cdot 2^{-32} = 2^{31}$ pairs satisfy the four zero byte differences at the ciphertext;

(c) guess 32 subkey bits in byte positions 0, 5, 10, 15 of $\kappa_0$, and partially decrypt the first round $\pi \circ \theta_1 \circ \gamma \circ \kappa_0$ for the pairs in item (b); filter those pairs that have a single nonzero byte difference in the leftmost column of the first vertical slice of the state after $\pi$; this is a 24-bit filtration condition, since it holds with probability $2^{-3*8}$; so, $2^{32-24} = 2^8$ wrong key are suggested by (14) per text pair;

(d) due to collisions, the number of wrong subkeys surviving, using one text pool, is $2^{32}(1-2^8/2^{32})^{2^{31}} = 2^{32}(1-2^{-24})^{2^{31}} \approx 2^{32}/e^{128} < 1$, so the correct subkey can be uniquely identified;

The attack complexity is $2^{32}$ CP, about $2^{32} \cdot 2^{31} = 2^{63}$ one-round computations to recover 32 subkey bits. To recover the full first round subkey requires repeating the attack sixteen times, yielding $16 \cdot 2^{63}/5.75 \approx 2^{65.5}$ 5.75-round computations, and $2^{32}$ memory.

## 5 Software Performance

Since 3D and AES/Rijndael share very similar components, it is natural to compare them. Due to the large block size, each 3D encryption roughly equals four AES encryptions, with bytes interleaved due to $\theta_1$ and $\theta_2$. Note that 3D iterates 22 rounds, and the AES has at most 14 rounds (for 256-bit keys). Thus, the latter has a better performance than the former. Although 3D provides more opportunities for parallelism than AES or Rijndael, this feature has not been exploited in performance comparisons.

## 6 Conclusions

This paper described a new secret-key block cipher called 3D, aimed at secure and fast encryption of large volumes of data. The design of 3D was inspired by the AES, in which text and key blocks are represented by a 2-dimensional state matrix of bytes. The main innovation of 3D is the $4 \times 4 \times 4$ 3-dimensional state of bytes, that led to improvements in design, security and potential applications (hash functions, MACs, stream ciphers, pseudorandom number generators).

Table 1 lists the attack complexities of our security evaluation of 3D.

**Table 1.** Attack complexities on reduced-round 3D cipher

| Attack | Time | Data | Memory | #Rounds | Comments |
|---|---|---|---|---|---|
| Multiset | $2^{19.5}$ | $2^9$ CP | $2^8$ | 4.75 | Sect. 4.8 |
| ID | $2^{65.5}$ | $2^{36}$ CP | $2^{32}$ | 5.75 | Sect. 4.9 |
| Multiset | $2^{139}$ | $2^{129}$ CP | $2^{128}$ | 5.75 | Sect. 4.8 |

The block size of 3D can be parameterized. For instance, if the underlying cipher operations were performed over GF($2^{16}$) instead of over GF($2^8$), then the block size would double to $64 \cdot 16 = 1024$ bits, but the storage of a $16 \times 16$ S-box becomes prohibitive. We have chosen the field GF($2^8$) since it is adequate even for smartcard processing, and because it avoids endianness issues.

Alternatively, keeping the bytewise operations, larger states could also be constructed with dimensions $5 \times 5 \times 5$ or $6 \times 6 \times 6$, for instance, but requiring new and larger MDS matrices. Mini-cipher versions of 3D can use a $3 \times 3 \times 3$ state of bytes, but a new $3 \times 3$ MDS matrix is needed. For analysis purposes, mini versions of 3D could use 4-bit words instead of bytes, leading to a 256-bit block cipher.

In [1], Barkan and Biham described the concept of dual ciphers, which means an isomorphism between the original cipher framework and another instance with isomorphic mappings for the plaintext, ciphertext and key. As an example, they described duals of the AES, which also exists for Rijndael and 3D. It is an open problem how to exploit dual ciphers in an effective attacks against AES/Rijndael, 3D and similar ciphers. Analogously, the algebraic attacks described by Courtois and Pieprzyk [10] against the AES (and 3D) still remain as open research problems.

## References

1. Barkan, E., Biham, E.: In How Many Ways Can You Write Rijndael. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 160–175. Springer, Heidelberg (2002)
2. Barreto, P.S.L.M., Rijmen, V.: The ANUBIS Block Cipher. In: 1st NESSIE Workshop, Heverlee, Belgium (2000)
3. Biham, E.: How to decrypt or even substitute DES-encrypted messages in $2^{28}$ steps. Information Processing Letters 3(84), 117–124 (2002)
4. Biham, E.: New Types of Cryptanalytic Attacks using Related Keys. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 398–409. Springer, Heidelberg (1994)
5. Biham, E., Shamir, A.: Differential Cryptanalysis of DES-like Cryptosystems. Journal of Cryptology 1(4), 3–72 (1991)
6. Biryukov, A., Shamir, A.: Structural Cryptanalysis of SASAS. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 394–405. Springer, Heidelberg (2001)
7. Biryukov, A., Wagner, D.: Advanced Slide Attacks. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 589–606. Springer, Heidelberg (2000)
8. Biryukov, A., Wagner, D.: Slide Attacks. In: Knudsen, L.R. (ed.) FSE 1999. LNCS, vol. 1636, pp. 245–259. Springer, Heidelberg (1999)
9. Coppersmith, D.: The Data Encryption Algorithm and its Strength Against Attacks. IBM Journal on Research and Development 3(38), 243–250 (1994)
10. Courtois, N.T., Pieprzyk, J.: Cryptanalysis of Block Ciphers with Overdefined Systems of Quadratic Equations. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 267–287. Springer, Heidelberg (2002)
11. Daemen, J., Govaerts, R., Vandewalle, J.: Weak Keys for IDEA. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 224–231. Springer, Heidelberg (1994)
12. Daemen, J., Knudsen, L.R., Rijmen, V.: The Block Cipher SQUARE. In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 149–165. Springer, Heidelberg (1997)
13. Davies, D.W., Murphy, S.: Pairs and Triplets of DES S-Boxes. Journal of Cryptology 1(8), 1–25 (1995)
14. Ferguson, N., Schroeppel, R., Whiting, D.: A Simple Algebraic Representation of Rijndael. In: Vaudenay, S., Youssef, A.M. (eds.) SAC 2001. LNCS, vol. 2259, pp. 103–111. Springer, Heidelberg (2001)
15. FIPS 180-2: Secure Hash Standard, SHS (2002), `http://csrc.nist.gov/`
16. FIPS197: Advanced Encryption Standard (AES), FIPS PUB 197 Federal Information Processing Standard Publication 197, U.S. Department of Commerce (2001)
17. Jakobsen, T., Knudsen, L.R.: The Interpolation Attack on Block Ciphers. In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 28–40. Springer, Heidelberg (1997)
18. Kaliski Jr, B.S., Robshaw, M.J.B.: Linear Cryptanalysis Using Multiple Approximations. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 26–39. Springer, Heidelberg (1994)

19. Kelsey, J., Schneier, B., Wagner, D.: Mod n Cryptanalysis, with Applications against RC5P and M6. In: Knudsen, L.R. (ed.) FSE 1999. LNCS, vol. 1636, pp. 139–155. Springer, Heidelberg (1999)
20. Kelsey, J., Schneier, B., Wagner, D.: Related-Key Cryptanalysis of 3-Way, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA. In: Han, Y., Quing, S. (eds.) ICICS 1997. LNCS, vol. 1334, pp. 233–246. Springer, Heidelberg (1997)
21. Knudsen, L.R.: Block Ciphers – A Survey. In: Preneel, B., Rijmen, V. (eds.) State of the Art in Applied Cryptography. LNCS, vol. 1528, pp. 18–48. Springer, Heidelberg (1998)
22. Knudsen, L.R.: DEAL – a 128-bit Block Cipher, Technical Report #151, University of Bergen, Dept. of Informatics, Norway (1998)
23. Knudsen, L.R.: Truncated and Higher Order Differentials. In: Preneel, B. (ed.) FSE 1994. LNCS, vol. 1008, pp. 196–211. Springer, Heidelberg (1995)
24. Knudsen, L.R., Meier, W.: Correlations in RC6 with a Reduced Number of Rounds. In: Schneier, B. (ed.) FSE 2000. LNCS, vol. 1978, pp. 94–108. Springer, Heidelberg (2001)
25. Lai, X.: Higher Order Derivatives and Differential Cryptanalysis. In: Proceedings of Symposium on Communication, Coding and Cryptography, Monte Verita, Switzerland, pp. 227–233 (1994)
26. Lenstra, H.W.: Rijndael for Algebraists (2002), http://math.berkeley.edu/~hwl/papers
27. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North-Holland Mathematical Library 16 (1977)
28. Matsui, M.: Linear Cryptanalysis Method for DES Cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (1994)
29. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, Boca Raton
30. Merkle, R.C.: A Software Encryption Function, posted to sci.crypt USENET newsgroup (1989)
31. Nyberg, K.: Linear Approximation of Block Ciphers. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 439–444. Springer, Heidelberg (1995)
32. Rijmen, V., Daemen, J., Preneel, B., Bosselaers, A., De Win, E.: The Cipher SHARK. In: Gollmann, D. (ed.) FSE 1996. LNCS, vol. 1039, pp. 99–112. Springer, Heidelberg (1996)
33. Rijmen, V., Preneel, B., De Win, E.: On Weaknesses of Non-Surjective Round Functions. Design, Codes and Cryptography 3(12), 253–266 (1997)
34. Rosenthal, J.: A Polynomial Description of the Rijndael Advanced Encryption Standard. Journal Algebra Appl. 2(2), 223–236 (2003)
35. Shannon, C.E.: Communication Theory of Secrecy Systems. Bell System Technical Journal 28, 656–715 (1949)
36. Wu, H.: Related-Cipher Attacks. In: Deng, R. (ed.) ICICS 2002. LNCS, vol. 2513, pp. 447–455. Springer, Heidelberg (2002)

# A Appendix A

A test vector for 3D follows in hexadecimal notation.

- plaintext $P_1$:

$$\left(\begin{array}{cccc|cccc|cccc|cccc}
\mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} \\
\mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} \\
\mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} \\
\mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}}
\end{array}\right)$$

- key $K_1$:

$$\left(\begin{array}{cccc|cccc|cccc|cccc}
\mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} \\
\mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} \\
\mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} \\
\mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}} & \mathtt{00}_{\mathtt{x}}
\end{array}\right)$$

- ciphertext $C_1 = E(K_1, P_1)$:

$$\left(\begin{array}{cccc|cccc|cccc|cccc}
\mathtt{ef}_{\mathtt{x}} & \mathtt{93}_{\mathtt{x}} & \mathtt{49}_{\mathtt{x}} & \mathtt{10}_{\mathtt{x}} & \mathtt{67}_{\mathtt{x}} & \mathtt{b2}_{\mathtt{x}} & \mathtt{b4}_{\mathtt{x}} & \mathtt{59}_{\mathtt{x}} & \mathtt{ad}_{\mathtt{x}} & \mathtt{01}_{\mathtt{x}} & \mathtt{4f}_{\mathtt{x}} & \mathtt{3a}_{\mathtt{x}} & \mathtt{0c}_{\mathtt{x}} & \mathtt{97}_{\mathtt{x}} & \mathtt{fe}_{\mathtt{x}} & \mathtt{e7}_{\mathtt{x}} \\
\mathtt{f3}_{\mathtt{x}} & \mathtt{ea}_{\mathtt{x}} & \mathtt{f5}_{\mathtt{x}} & \mathtt{8c}_{\mathtt{x}} & \mathtt{03}_{\mathtt{x}} & \mathtt{46}_{\mathtt{x}} & \mathtt{33}_{\mathtt{x}} & \mathtt{ed}_{\mathtt{x}} & \mathtt{b6}_{\mathtt{x}} & \mathtt{22}_{\mathtt{x}} & \mathtt{7f}_{\mathtt{x}} & \mathtt{40}_{\mathtt{x}} & \mathtt{cd}_{\mathtt{x}} & \mathtt{02}_{\mathtt{x}} & \mathtt{52}_{\mathtt{x}} & \mathtt{b3}_{\mathtt{x}} \\
\mathtt{d0}_{\mathtt{x}} & \mathtt{ee}_{\mathtt{x}} & \mathtt{f8}_{\mathtt{x}} & \mathtt{7c}_{\mathtt{x}} & \mathtt{eb}_{\mathtt{x}} & \mathtt{70}_{\mathtt{x}} & \mathtt{28}_{\mathtt{x}} & \mathtt{da}_{\mathtt{x}} & \mathtt{62}_{\mathtt{x}} & \mathtt{dd}_{\mathtt{x}} & \mathtt{29}_{\mathtt{x}} & \mathtt{67}_{\mathtt{x}} & \mathtt{84}_{\mathtt{x}} & \mathtt{53}_{\mathtt{x}} & \mathtt{14}_{\mathtt{x}} & \mathtt{1d}_{\mathtt{x}} \\
\mathtt{fe}_{\mathtt{x}} & \mathtt{58}_{\mathtt{x}} & \mathtt{54}_{\mathtt{x}} & \mathtt{33}_{\mathtt{x}} & \mathtt{2b}_{\mathtt{x}} & \mathtt{ab}_{\mathtt{x}} & \mathtt{40}_{\mathtt{x}} & \mathtt{34}_{\mathtt{x}} & \mathtt{d3}_{\mathtt{x}} & \mathtt{66}_{\mathtt{x}} & \mathtt{d5}_{\mathtt{x}} & \mathtt{4c}_{\mathtt{x}} & \mathtt{5f}_{\mathtt{x}} & \mathtt{63}_{\mathtt{x}} & \mathtt{0b}_{\mathtt{x}} & \mathtt{0a}_{\mathtt{x}}
\end{array}\right)$$