Impossible-Differential Attacks on block-cipher based Hash and Compression Functions using 3D and Whirlpool

Daniel Santana de Freitas[1], **Jorge Nakahara Jr**

[1]Depto de Informática e Estatística (INE) – Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 – 88040-900 – Florianópolis – SC - Brazil

santana@inf.ufsc.br, jorge_nakahara@yahoo.com.br

Abstract. *In this paper, we analyse block-cipher-based hash functions, which means hash functions that use block ciphers as compression functions in a mode of operation, such as Davies-Meyer (DM), Matyas-Meyer-Oseas (MMO) and Miyaguchi-Preneel (MP), for instance. We use impossible differentials (ID) to distinguish the compression (or hash) function from an ideal primitive (a random oracle) by detecting a nonrandom behavior. We applied an ID analysis to an 8-round variant of the 3D block cipher used in MMO mode, as a compression function of a hypothetical hash function. This attack effectively improves upon the previously known distinguishing ID attacks on reduced-round 3D. We can also attack a hash function using 3D as compression function in DM mode. Finally, we attacked the compression function in Whirlpool with a 5-round W cipher in MP mode with 2^{100} time and 2^{64} memory.*

Keywords: impossible differentials, block-cipher-based hash functions, modes of operation, Whirlpool hash function.

1. Introduction

There are several properties a hash function is expected to satisfy, depending on its application. The most well-known are: collision resistance, preimage resistance and second preimage resistance [17]. But, there are also other relevant properties such as partial preimage resistance, and nonrandom properties [11, 19, 13] that although not leading to collisions nor to (second) preimages, may turn the hash function unsuitable to applications that expect it to behave as a random mapping [22], in some particular (abstract) model. Thus, detectable evidence of nonrandom behavior demonstrates that the hash function is not an ideal primitive, which is expected for a high-security cryptographic function. For instance, in the SHA-3 competition [23], NIST has requested that candidate hash function algorithms behave as close as possible to random oracle. In this paper, we apply the impossible-differential (ID) technique to detect nonrandom properties in the compression or the hash functions.

The impossible differential (ID) approach is a chosen-plaintext technique formerly described in [12]. To construct ID distinguishers, truncated differentials and the miss-in-the-middle technique [5] are typically used. In our case, a truncate differential has two types of difference: zero and nonzero (byte) difference. Moreover, the nonzero difference value is not relevant (so we do not pay in probability for any specific difference pattern). The idea of the miss-in-the-middle approach is to concatenate two truncated differentials, say, $\alpha \xrightarrow{f} \beta$ and $\epsilon \xleftarrow{g} \eta$, both of which hold with certainty, into a single differential

$\alpha \overset{g^{-1}\circ f}{\rightarrow} \eta$, where $g^{-1} \circ f$ stands for the functional composition of f and g^{-1} in this order. Nonetheless, $\beta \neq \epsilon$, that is, the differences do not match in the middle of the distinguisher, which leads to a contradiction. The terminology $\alpha \overset{f}{\rightarrow} \beta$ means that the difference α causes difference β after the transformation f in the encryption (or forward) direction; $\epsilon \overset{g}{\leftarrow} \eta$ means that the difference η causes difference ϵ after the transformation g in the decryption (or backwards) direction. Note the direction of the arrows. Consequently, $\alpha \overset{g^{-1}\circ f}{\rightarrow} \eta$ holds with probability zero, or analogously, $\alpha \overset{g^{-1}\circ f}{\not\rightarrow} \eta$ holds with certainty. The terminology $\alpha \overset{g^{-1}\circ f}{\not\rightarrow} \eta$ means that α can never cause the difference η across the transformations $g^{-1} \circ f$ in the encryption direction. Analogously, the same reasoning applies for the decryption direction. This set of differences α, β, ϵ, η characterize an ID distinguisher constructed with the miss-in-the-middle technique. The initial difference α and the final difference η will be clearly identified in the ID distinguishers (3) and (9) in this paper.

The ID technique has already been applied to several ciphers, including IDEA and Khufu [5], Rijndael [2, 6], among others.

Table 1 compares our distinguishing attacks and previous ones on reduced-round versions of the 3D cipher. Table 2 summarizes attacks on Whirlpool and its compression function.

This paper is organized as follows: Sect. 2 summarizes the contributions of this paper; Sect. 3 briefly describes the 3D block cipher; Sect. 3.1 describes a distinguishing attack on a compression function using an 8-round version of the 3D cipher in the MMO mode; Sect. 4 presents an attack on the Whirlpool hash function, whose block cipher W in the compression function is reduced to 8 rounds; Sect. 5 concludes this paper.

2. Contributions

The contributions of this paper include

- new distinguishing attacks on compression and hash functions that use block ciphers as building blocks in well-known modes of operation such as Matyas-Meyer-Oseas (MMO), Davies-Meyer (DM) and Miyaguchi-Preneel (MP) [17].
- our attacks use for the first time the impossible differential technique to demonstrate properties of the underlying block cipher that may propagate to the entire hash function, depending on the padding, the number of rounds of the block cipher and on the (hashing) mode of operation surrounding the block cipher. In our case, we attack one or two rounds beyond the ID distinguishers for the underlying primitives, showing that there is a further margin of insecurity to take into account when ciphers are used as building blocks.
- we do not aim at traditional attacks such as collisions or (second) preimage, but our findings are relevant in applications where hash functions are expected to behave as random oracles such as (pseudo-)random number generators, and taking into account NIST's requirements for the new SHA-3 algorithms [22, 23].
- our targets are a compression function using the 3D block cipher [20] (in MMO and DM modes), and the W block cipher used in Whirlpool hash function [4] (in MP mode). For 3D, we effectively improve on previous distinguishing attacks on

reduced-round 3D (see Table 1), and could attack a hash function in DM mode. In this respect, the DM mode was more vulnerable than the other modes.

3. Impossible Differential of 8-round 3D Cipher

The 3D block cipher operates on 512-bit blocks under a 512-bit user key, both of which are represented as a $4 \times 4 \times 4$ state of bytes [20]. The three-dimensional cipher state for a 64-byte data block $A = (a_0, a_1, \ldots, a_{63})$ is denoted as

$$\text{State} = \left(\begin{array}{cccc|cccc|cccc|cccc} a_0 & a_4 & a_8 & a_{12} & a_{16} & a_{20} & a_{24} & a_{28} & a_{32} & a_{36} & a_{40} & a_{44} & a_{48} & a_{52} & a_{56} & a_{60} \\ a_1 & a_5 & a_9 & a_{13} & a_{17} & a_{21} & a_{25} & a_{29} & a_{33} & a_{37} & a_{41} & a_{45} & a_{49} & a_{53} & a_{57} & a_{61} \\ a_2 & a_6 & a_{10} & a_{14} & a_{18} & a_{22} & a_{26} & a_{30} & a_{34} & a_{38} & a_{42} & a_{46} & a_{50} & a_{54} & a_{58} & a_{62} \\ a_3 & a_7 & a_{11} & a_{15} & a_{19} & a_{23} & a_{27} & a_{31} & a_{35} & a_{39} & a_{43} & a_{47} & a_{51} & a_{55} & a_{59} & a_{63} \end{array}\right) \tag{1}$$

that is, bytes are ordered column-wise. The round subkeys follow this same ordering and structure. We denote the first byte of subkey k_i as $k_{i,0}$, the second byte as $k_{i,1}$, and so on. Each set of 16 bytes in a 4×4-byte square is called a slice of the state. In total, the 3D cipher iterates 22 rounds. The round transformations in 3D have a clear correspondence with those of the AES [7, 10]. Using the terminology of [20]:

- κ_i: bit-wise xor with round subkey, equivalent to AddRoundKey in AES;
- γ: a byte-wise application of the AES S-box, equivalent to SubBytes in AES;
- θ_1, θ_2: equivalent to ShiftRows in AES but applied to each slice of the state alternately; θ_1 in the odd-numbered rounds, θ_2 in the even-numbered rounds;
- π: matrix multiplication with columns of the state, equivalent to MixColumns in AES.

Each round transformation stands for a fraction of 0.25 (a quarter) of a round. Distinguishers for 3D may sometimes cover fractions of a round, such as 5.25 rounds, for instance (see Table 1). The i-th full round of 3D, encrypting a text block X, is denoted $\tau_i(X) = \pi \circ \theta_{i \bmod 2+1} \circ \gamma \circ \kappa_i(X) = \pi(\theta_{i \bmod 2+1}(\gamma(\kappa_i(X))))$. The last round does not include π. The key schedule of 3D follows a similar framework as encryption. We refer to [20] for further details.

3.1. A distinguishing attack on the compression function

Suppose an 8-round variant of the 3D cipher, with the first round starting with θ_1, is used as compression function in a hash function in Matyas-Meyer-Oseas (MMO) mode [17]. We show that the impossible-differential technique can be used to distinguish this variant as compression function from a random oracle (RO), that is, to detect a nonrandom behavior of an 8-round 3D variant. Recall that we apply a distinguish-from-random type of attack. This nonrandom property does not imply a collision nor a (second) preimage.

Let $E : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n$ denote an n-bit block cipher parameterized by a k-bit key K. Let H_i denote the i-th chaining variable during the hash computation, with the initial value IV= H_0; let $m_i \in \{0,1\}^n$ denote the i-th block of message being hashed; and let $g : \{0,1\}^n \to \{0,1\}^k$. The MMO mode computes

$$H_i = m_i \oplus E_{g(H_{i-1})}(m_i), \tag{2}$$

that is, g adapts the $(i-1)$-th chaining variable of n bits to the key input of k bits. In our case, E is the 3D cipher, $k = n = 512$, and g becomes the identity mapping.

The setting of this attack is against the compression function only. H_0 and m_i are known, and $H_i = m_i \oplus E_{H_{i-1}}(m_i)$ in the MMO mode. Notice that the feedforward of m_i in (2) can be moved: $E_{H_{i-1}}(m_i) = H_i \oplus m_i$. Thus, for a fixed key H_{i-1}, the adversary has access to the input and output of E. Using the terminology of Sect. 1, $\alpha = m_i \oplus m_i^*$ and $\eta = E_{H_{i-1}}(m_i) \oplus E_{H_{i-1}}(m_i^*)$ because H_{i-1} is the key and is kept fixed.

The 6-round impossible differential (ID) distinguisher used in this attack is depicted in (3), where '0' denotes a zero byte difference and 'Δ' denotes an (arbitrary) nonzero byte difference. Informally, this ID distinguisher means that a single-byte text difference in any of the 64 positions in the message input m_i (input text difference is denoted α) cannot lead to a 16-byte zero difference pattern (denoted η) after 6-round 3D (a very structured difference pattern). This is a slightly modified version of the distinguisher whose construction details can be found in [21]. In summary, there is a contradiction after π of the third round: there are only Δ bytes in each column of the leftmost slice before π, but only zero byte differences in all four columns after π, which contradicts the branch number of π which is five[1]. In other words, $\Pr(\alpha \rightarrow \eta) = 0$. Notice that there are four impossible output difference patterns, by choosing one of the four patterns of rows with zero byte difference in η. Note that the output difference η contains zero byte differences only in positions (0, 4, 8, 12, 19, 23, 27, 31, 34, 38, 42, 46, 49, 53, 57, 61) of the state (1). Following this construction, let us call η_1, η_2 and η_3 the three other impossible output difference patterns with zero byte differences in positions (1, 5, 9, 13, 16, 20, 24, 28, 35, 39, 43, 47, 50, 54, 58, 62), (2, 6, 10, 14, 17, 21, 25, 29, 32, 36, 40, 44, 51, 55, 59, 63) and (3, 7, 11, 15, 18, 22, 26, 30, 33, 37, 41, 45, 48, 52, 56, 60), respectively.

The attack framework is depicted in (4), where '*'s denote chosen input byte positions before the 6-round distinguisher (3). Thus, $i = 2$ in κ_i. We use text structures to cover the first two rounds and the 6-round ID distinguisher afterwards, for a total of eight rounds. Note that starting one round before the distinguisher provides only $2^{32}(2^{32} - 1)/2 \approx 2^{63}$ text pairs for the attack, which is not enough, while starting the attack three rounds before the distinguisher would require the full codebook. So, two

[1]The branch number [7, 10] is a measure of the diffusion power of a mapping $F : \text{GF}(2^n)^m \rightarrow \text{GF}(2^n)^m$. In the case of 3D, $n = 8$ and $m = 4$. The branch number of F is $B_F = \min_{x \neq 0}\{HW(x) + HW(F(x))\}$, where HW denotes the Hamming weight. Let F be π which consists of a 4×4 matrix from an MDS code. From the theory of error correcting codes [15], the MDS code guarantees a minimum distance of 5 between two codewords, that is, $B_\pi = 5$.

rounds before the distinguisher as in (4) is the best trade-off.

$$\alpha = \begin{pmatrix} \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \xrightarrow{\pi \circ \theta_1 \circ \gamma \circ \kappa_i}$$

$$\begin{pmatrix} \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\xrightarrow{\theta_2 \circ \gamma \circ \kappa_{i+1}} \begin{pmatrix} \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \xrightarrow{\pi}$$

$$\begin{pmatrix} \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 \end{pmatrix}$$

$$\xrightarrow{\theta_1 \circ \gamma \circ \kappa_{i+2}} \begin{pmatrix} \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 \\ 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta \\ 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 \\ 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 \end{pmatrix} \xrightarrow{\pi}$$

$$\begin{pmatrix} \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \end{pmatrix} \xrightarrow{\theta_2 \circ \gamma \circ \kappa_{i+3}}$$

$$\begin{pmatrix} \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \end{pmatrix} \xrightarrow{\pi}$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ 0 & 0 & 0 & 0 & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ 0 & 0 & 0 & 0 & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ 0 & 0 & 0 & 0 & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \end{pmatrix} \xleftarrow{\kappa_{i+4} \circ \gamma^{-1} \circ \theta_1^{-1} \circ \pi^{-1}}$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ 0 & 0 & 0 & 0 & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ 0 & 0 & 0 & 0 & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ 0 & 0 & 0 & 0 & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \end{pmatrix}$$

$$\xleftarrow{\kappa_{i+5} \circ \gamma^{-1} \circ \theta_2^{-1} \circ \kappa_{i+6}} \begin{pmatrix} 0 & 0 & 0 & 0 & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & 0 & 0 & 0 & 0 \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & 0 & 0 & 0 & 0 & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & 0 & 0 & 0 & 0 & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \end{pmatrix} = \eta \quad (3)$$

$$\begin{pmatrix} * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 \\ 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 \\ 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * \end{pmatrix} \xrightarrow{\theta_1 \circ \gamma \circ \kappa_0}$$

$$\begin{pmatrix} * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 \\ * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 \\ * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 \\ * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 \end{pmatrix} \xrightarrow{\pi}$$

$$\begin{pmatrix} * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & * & 0 & 0 & 0 \end{pmatrix} \xrightarrow{\theta_2 \circ \gamma \circ \kappa_1}$$

$$\begin{pmatrix} * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\xrightarrow{\pi} \text{6-round ID distinguisher (3)} \quad (4)$$

The attack proceeds as follows: in (4) consider a (text) structure consisting of $(2^8)^{16} =$

2^{128} chosen plaintexts m_i, in which each byte, denoted by '*' in (4), in the diagonal of each slice of the state of 3D assumes all possible byte values, while the remaining state bytes are constants. Each structure of 2^{128} plaintexts contains $2^{128}(2^{128}-1)/2 = 2^{255} - 2^{127}$ text pairs $P_i \oplus P_j$, for $i \neq j$. After one full round, with probability $16 \cdot (2^{-8 \cdot 3})^4 = 2^{-92}$, there will be a single nonzero byte difference per slice of the state, that is, each column of nonzero byte difference, the '*'s, will collapse into a single nonzero byte difference. Notice that there are four possibilities for the single nonzero byte differences to be placed in specific positions for the next round so that they will align into a single column after θ_2. In summary, for the given configuration of '*'s in (4) there are four possible columns where the four '*'s can end up after the first round, and for each column there are four possible places they can be after two rounds. We expect $(2^{255} - 2^{127}) \cdot 2^{-92} = 2^{163} - 2^{35}$ pairs to survive this filtering per structure. After the second round, with probability $4 \cdot 2^{-24} = 2^{-22}$, we expect that the four nonzero byte differences lead to a single nonzero byte difference, which is exactly the input difference to the 6-round ID distinguisher (3). This filtering leaves $(2^{163} - 2^{35}) \cdot 2^{-22} = 2^{141} - 2^{13}$ pairs.

Thus, these $2^{141} - 2^{13}$ text pairs will never lead to a forbidden output difference of (3), since they lead to an impossible differential. The remaining $2^{255} - 2^{127} - 2^{141} + 2^{13}$ pairs may satisfy the output difference of (3), which has sixteen zero byte differences. We expect that a remaining pair can satisfy one of the four output differences (η, η_1, η_2, η_3) with probability $2^{-128} \times 4 = 2^{-126}$. We expect that $(2^{255} - 2^{127} - 2^{141} + 2^{13}) \cdot 2^{-126} = 2^{129} - 2 - 2^{15} + 2^{-113}$ pairs to survive this filtering. For a random permutation on the same plaintext space as 3D, we expect $(2^{255} - 2^{127}) \cdot 2^{-126} = 2^{129} - 2$ pairs to survive. We exclude from the random permutation set the dual ciphers of 3D, in a similar setting as the duals of the AES [3], which would behave as 3D under the ID distinguisher. Thus, we can distinguish 8-round 3D cipher from a random permutation mapping using 2^{128} blocks of memory (for storing the ciphertexts) and effort of 2^{128} 8-round computations. Note that unlike in conventional impossible-differential attacks, we use both the text pairs that lead to the ID distinguisher differences and those that do not.

In a block cipher setting, the data complexity is 2^{128} chosen plaintexts, but in a compression/hash function setting there is no secret key, or more precisely, the key is known, so the adversary does not depend on the legitimate parties to obtain data. Therefore, in the latter, there is no data complexity, since the adversary knows the key e.g. a chaining variable. Note that the work effort consists in checking the difference in text pairs corresponding simply to the xor of two text blocks. But this xor is much cheaper than a single encryption and is done offline by the adversary. For this attack, in particular, we are interested only in the ciphertext pairs, $C_i \oplus C_j$. Therefore, the 2^{255} text pairs do not account for the time complexity. The effort actually come from 2^{128} texts encrypted in a structure.

The difference between the number of surviving pairs for reduced-round 3D and a random permutation is just 2^{15}, which means an advantage of only $2^{15}/2^{129} = 2^{-114}$. Following [1][Sect.6.7], in a distinguish-from-random setting times, to achieve a high success rate attack, the number of queries shall be proportional to $(2^{-114})^{-2} = 2^{228}$. Since we already use 2^{128} texts in a structure, this means that the attack shall be repeated 2^{100} times which means using 2^{100} structures in total. This factor leads to an updated effort of $2^{100} \cdot 2^{128} = 2^{228}$ 8-round 3D computations, but the memory cost remains the same since

we do not need to store all structures at the same time. There is no shortage of structures to repeat the attack. Simply choose other constant values for the zero-difference bytes of the state in (3). Since there are 48 bytes with difference zero, one can choose up to $256^{48} = 2^{384}$ constants and thus up to 2^{384} structures.

We could describe other 6-round ID distinguishers for 3D, as well as attacks on 8-round 3D versions using these distinguishers simultaneously, just like (3) and (4) without increasing the attack complexities. These additional distinguishers demonstrate that there are large sets of text pairs inside 8-round 3D that never lead to a set of forbidden differences, in comparison to a random permutation. As an example of these distinguishers, note that at the minimum we only need four zero byte differences in η, for instance, in positions (0, 19, 34, 49) of the state (1). These four byte differences will lead to a single column of zeros after π of the 4th round, leading to a contradiction. Note further that there are sixteen such zero-byte difference patterns, leading to sixteen 6-round ID distinguishers for 3D: (4, 23, 38, 53), (8, 27, 42, 57), and so on.

Notice that the text structure in (4) has bytes with nonzero difference over the padding part, assuming the Merkle-Damgård paradigm and a message space with size at least 2^{64}, which implies bytes a_{56} until a_{63} of the state (1) are reserved for the original message size. Thus, the attack is restricted to the compression function only, and does not apply to the hash function.

The distinguishing attack described in this section improves on the attack described in [21], which was restricted to 6-round 3D, both in terms of number of rounds and attack complexities.

3.2. A distinguishing attack on the hash function

Consider now the same 8-round variant of the 3D cipher in Sect. 3.1, but this time the mode of operation is Davies-Meyer (DM) [17] and the target is the hash function using the Merkle-Damgård (MD) padding scheme [8, 18]. In the MD scheme the padding consists of a single '1' bit followed by '0' bits up to $512-l$ bits, so that the last l bits store the size of the original message to be hashed. For instance, if $l = 128$, then the bytes in positions a_{48} until a_{63} of the state (1) are reserved for the message size.

In the attack setting, $H_0 = IV$ and m_i are known, and the updating rule in the DM mode is

$$H_i = H_{i-1} \oplus E_{m_i}(H_{i-1}). \tag{5}$$

Notice that the feedforward of H_{i-1} can be moved: $E_{m_i}(H_{i-1}) = H_i \oplus H_{i-1}$. Thus, for a fixed key m_i, the adversary has access to the input and output of E (that is 8-round 3D).

In this section we show that this hash function can be distinguished from a random oracle (RO) using the impossible-differential technique. The attack is on 2-block messages, $M = m_0 \| m_1$, with $m_0, m_1 \in \{0,1\}^{512}$, but can be extended to larger messages M as well. The attack proceeds as follows:

- we vary the value of the first 512-bit block, m_0 to generate text pairs for the attack on the second E instance; in the DM mode, $H_1 = H_0 \oplus E_{m_0}(H_0)$ vary. Note that $E_{m_0}(H_0)$ behaves as a random function for variable m_0 and fixed H_0. We need

2^{228} distinct values of H_1 as input to the second E instance, according to the attack described in Sect. 3.1. Following [24], we expect about $2^{224} \cdot e \approx 2^{225.44}$ values of m_0 to obtain 2^{224} distinct values of H_1.

- we keep m_1 always fixed in all messages M; recall that m_1 contains the padding and message size since it is the last message block.
- since m_1 is fixed, the second E instance performs a permutation, with input H_1, from which we can generate difference pairs for the impossible differential attack in Sect. 3.1.
- we can therefore, detect differences in H_2 for the impossible differential, and the attack complexities of Sect. 3.1 hold, but this time the analysis is applied to the hash function, not to the compression function. The H_1 value can be computed and separated from H_2 to obtain the output of $E_{m_1}(H_1)$ only.

Since we have encryptions of $m_0 \| m_1$ that is, two E computations at a time, the attack effort becomes $2^{226.44}$ 8-round 3D computations.

For larger messages, M, say, t-block messages, the idea is to vary m_1, m_2, ..., m_{t-1}, but not the last block m_t, which contains the padding and message size ($512 \cdot t$ bits), forming the Merkle-Damgård strengthening.

In the impossible-differential attack setting, the DM mode is weaker than the MMO mode, since an adversary can control the key input in DM mode (which is a message block) making the underlying cipher in the compression function to behave as a permutation, which is the setting needed for an impossible-differential attack.

4. A distinguishing attack on 5-round W in Whirlpool

The Whirlpool hash function was designed by Barreto and Rijmen [4] in 2003, according to the wide-trail design strategy. Whirlpool follows the Merkle-Damgård strengthening (message size is limited to 2^{256} bits) and uses a block cipher called $W : \{0,1\}^{512} \times \{0,1\}^{512} \rightarrow \{0,1\}^{512}$ in MP mode. The cipher W is similar to the AES but operates on an 8×8 state of bytes where bytes are input row-wise, and W iterates ten rounds. An example of a state of W indicating the byte ordering is depicted in (6).

$$\text{State} = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\ a_8 & a_9 & a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{16} & a_{17} & a_{18} & a_{19} & a_{20} & a_{21} & a_{22} & a_{23} \\ a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & a_{29} & a_{30} & a_{31} \\ a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} \\ a_{40} & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} \\ a_{48} & a_{49} & a_{50} & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \\ a_{56} & a_{57} & a_{58} & a_{59} & a_{60} & a_{61} & a_{62} & a_{63} \end{pmatrix} \tag{6}$$

Following the terminology of [4], the rounds operations in W are

- γ: nonlinear layer; byte-wise application of an 8×8-bit S-box;
- π: byte permutation, i.e. column j is cyclically shifted down by j positions, where columns are numbered from 0 in left-to-right order;
- θ: diffusion layer; multiplication of each row of the state by an 8×8 MDS matrix;
- σ_k: exclusive-or between the state and the k-th round subkey (which are predefined constants).

The k-th full round of W, operating on a text block X, consists of $\tau(X) = \sigma_k \circ \theta \circ \pi \circ \gamma(X) = \sigma_k(\theta(\pi(\gamma(X))))$. There is a pre-whitening layer consisting of σ_0.

The ID distinguisher (9) for Whirlpool (in the appendix) covers four rounds, and was constructed with the miss-in-the-middle technique. It consists of two truncated differentials: one in the encryption direction covering three rounds, and the other differential in the decryption direction covering one round. The contradiction (or miss-in-the-middle) happens in the top row of the state after the third round. A row with all eight nonzero byte differences cannot cause a row with only zero byte differences because the MDS matrix used in θ has branch number nine [7]. Notice that there are $64 \cdot 8$ equivalent 4-round ID distinguishers since the initial nonzero Δ byte can be in any of the 64 positions of the state, and the output pattern of eight zero byte differences can follow eight possible patterns. The attack framework is depicted in (7) and starts one round before the ID distinguisher (9) covering a total of five full rounds of W.

$$\begin{pmatrix} * & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & * & 0 \\ 0 & 0 & 0 & 0 & 0 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & * & 0 & 0 & 0 \\ 0 & 0 & 0 & * & 0 & 0 & 0 & 0 \\ 0 & 0 & * & 0 & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \overset{\pi \circ \gamma \circ \sigma_i}{\rightarrow} \begin{pmatrix} * & * & * & * & * & * & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \overset{\theta}{\rightarrow} \text{4-round ID distinguisher (9)} \quad (7)$$

The attack setting is the following: Whirlpool uses the MP mode. Let $H_0 = IV$ denote the initial value of the chaining variable; let m_i denote the i-th message block (the input to W). We aim to find a nonrandom behaviour of the compression function.

The Miyaguchi-Preneel (MP) mode [17] is the following:

$$H_i = m_i \oplus H_{i-1} \oplus E_{H_{i-1}}(m_i), \tag{8}$$

where E is the block cipher W in this case. For Whirlpool, the text and key input of W have the same size. We keep H_{i-1} fixed as key input, so that $W_{H_{i-1}}(m_i)$ behaves as a permutation.

The attack proceeds as follows: consider a (text) structure consisting of $(2^8)^8 = 2^{64}$ chosen messages (plaintexts) m_i, in which the eight bytes in the pattern (7) of the state W denoted '*' assume all possible byte values, while the remaining state bytes are constants. Each structure of 2^{64} plaintexts contain about $2^{64}(2^{64}-1)/2 = 2^{127} - 2^{63}$ text pairs $m_i^j \oplus m_i^k$, for $k \neq j$. After one full round, with probability $8 \cdot (2^{-8})^7 = 2^{-53}$, there will be a single nonzero byte difference in the first row of the state, that is, the first row will collapse into a single nonzero byte difference in one of the first eight positions in that row. Notice that there are eight possibilities for the single nonzero byte difference Δ to be placed in specific positions after the first round. We expect $(2^{127} - 2^{63}) \cdot 2^{-53} = 2^{74} - 2^{10}$ pairs to survive this filtering per structure. This single-byte difference is exactly the input difference to the 6-round ID distinguisher (9). The remaining $2^{127} - 2^{63} - 2^{74} + 2^{10}$ pairs may satisfy an output difference of (9) which has eight zero byte differences.

The pattern η in (9) has zero byte differences in positions (0, 9, 18, 27, 36, 45, 54, 63), the main diagonal of the state (6). Notice that the eight zero byte differences could follow any of eight possible patterns, each in a sub-diagonal of the state. Thus, in total, there are eight forbidden output patterns, including η, and we expect that a remaining pair can satisfy one of these output differences with probability $8 \cdot 2^{-64} = 2^{-61}$.

Using a single structure, we expect that $(2^{127} - 2^{63} - 2^{74} + 2^{10}) \cdot 2^{-61} = 2^{66} - 2^2 - 2^{13} + 2^{-51}$ pairs survive for 5-round W as compression function. But, for a random

permutation, we expect $(2^{127} - 2^{63}) \cdot 2^{-61} = 2^{66} - 2^2$ pairs to survive. Thus, due to the deficit in the number of surviving pairs for each cipher, we can distinguish a compression function containing 5-round W from a random mapping (oracle) using 2^{64} blocks of memory and an effort of 2^{64} 5-round W computations.

The difference between the number of surviving pairs for reduced-round W and a random permutation is just 2^{13}, which means an advantage of $2^{13}/2^{63} = 2^{-50}$. Following [1][Sect.6.7], in a distinguish-from-random setting, to achieve a high success rate attack, the number of queries shall be proportional to $(2^{-50})^{-2} = 2^{100}$. Since we already use 2^{64} texts in a structure, this means that the attack shall be repeated 2^{36} times which means using 2^{36} structures. This factor leads to an updated effort of $2^{36} \cdot 2^{64} = 2^{100}$ 5-round W computations, but the memory cost remains the same since we do not need to store all structures at the same time. There is no shortage of structures to repeat the attack. To find more structures simply choose other constant values for the zero-difference bytes of the state in (7). Since there are 56 bytes with difference zero in (7), one can choose up to $256^{56} = 2^{448}$ constants and thus, up to 2^{448} structures.

Notice that the text structure in (7) has (message) bytes with nonzero difference across a sub-diagonal of the state (6). That means these bytes straddle over the padding and message length parts, assuming the Merkle-Damgård paradigm and a message space with size 2^{256}. Even if the message space were as low as 2^{64}, the attack would still be restricted to the compression function only.

5. Conclusions

This paper presented distinguishing attacks on block-cipher-based compression and hash functions using the impossible-differential technique. In particular, we analysed block-cipher-based hash functions in three classical hash modes: Davies-Meyer (DM), Matyas-Meyer-Oseas (MMO) and Miyaguchi-Preneel (MP). Our attacks focused on reduced-round versions of the 3D and W block ciphers as compression functions.

Table 1 lists the complexities of ours attacks on a compression function and previous attacks using the 3D cipher. This comparison is for completeness purposes only, since the attack settings are not the same (block cipher and compression/hash function). Our results are not a threat to the full 22-round 3D cipher, but compare quite favorably with the 6-round distinguishing attack in [21]. Note that our results apply in a distinguish-from-random setting, while other attacks operate in key-recovery settings since the latter does not make sense for compression and hash functions. Notice that in a block cipher setting there is a secret key and thus, in a chosen-plaintext attack the adversary needs the legitimate parties to encrypt/decrypt texts for him, because he does not know the key. In a compression function setting the key to the underlying block cipher is not secret, so the adversary does not depend on the legitimate parties to encrypt or decrypt. In this latter case, there is no data complexity, since the adversary can encrypt/decrypt any text (offline).

Table 2 compares the complexities of ours and previous attacks on reduced-round Whirlpool, not just the compression function. It is not straightforward to compare our results with previous attacks since we do not aim at collisions or (second) preimages. Our motivation in looking for nonrandom behavior of the compression and hash function constructions was based in [22, 23], where NIST stated the need for candidate hash functions

Table 1. Attack complexities on reduced-round versions of the 3D cipher.

Technique	Time	Data	Memory	#Rounds	Ref.	Setting
Multiset	$2^{19.5}$	2^9 CP	2^8	4.75	[20]	block cipher
ID	$2^{65.5}$	2^{36} CP	2^{32}	5.75	[20]	block cipher
Multiset	2^{139}	2^{129} CP	2^{128}	5.75	[20]	block cipher
ID	2^{256}	2^{256} CP	2^{256}	6	[21]	block cipher
Known-key	2^8	2^8 CP	negligible	7.75	[21]	block cipher
ID	2^{228}	—	2^{128}	8	Sect. 3.1	comp. function (MMO mode)
ID	$2^{226.44}$	—	2^{128}	8	Sect. 3.2	hash function (DM mode)
Known-key	2^{56}	2^{56} CP	negligible	9.75	[21]	block cipher
ID	2^{401}	2^{501} CP	2^{311}	10	[21]	block cipher
Known-key	2^{200}	2^{200} CP	2^8	15	[9]	block cipher

CP: Chosen Plaintext/messages

Table 2. Attack complexities on reduced-round W cipher of Whirlpool.

Technique	Time	Memory	#Rounds	Ref.	Comments
rebound	2^{120}	2^{16}	4.5	[16]	collision
ID	2^{100}	2^{64}	5	Sect. 4	non-randomness (comp. function)
rebound	2^{120}	2^{16}	5.5	[16]	semi-free start collision
rebound	2^{128}	2^{16}	7.5	[16]	semi-free start near-collision
rebound	2^{184}	2^8	7.5	[14]	collision
rebound	2^{176}	2^8	9.5	[14]	near-collision
rebound	2^{188}	2^8	10	[14]	distinguisher

to behave as close as possible to ideal functions (random oracles).

In [13], integral distinguishers were found for up to seven rounds of Whirlpool, but they were not applied further to derive attacks.

References

[1] Baigneres, T.: Quantitative Security of Block Ciphers: Designs and Cryptanalysis Tools. PhD Thesis, Ecole Polytechnique Federale de Lausanne (2008)

[2] Barak, B., Aref, M.R.: Impossible Differential Attack on seven-round AES-128. IET Information Security, (2):2, 28–32 (2008)

[3] Barkan,E., Biham, E.: In How Many Ways can you write Rijndael?. In: Y Zheng (Ed.), Asiacrypt 2002, Springer, LNCS 2501, 160–175 (2002)

[4] Barreto, P.S.L.M., Rijmen, V.: The Whirlpool Hashing Function. available at http://www.larc.usp.br/ pbarreto/WhirlpoolPage.html (2003)

[5] Biham, E., Biryukov, A., Shamir, A.: Miss-in-the-Middle Attacks on IDEA and Khufu. In: Knudsen, L.R. (Ed.), Fast Software Encryption (FSE), Springer, LNCS 1636, 124–138 (1999)

[6] Cheon, J.H., Kim, M., Kim, K., Lee, J.Y., Kang,S.: Improved Impossible Differential Cryptanalysis of Rijndael and Crypton. 4th International Conference on Information Security and Cryptology (ICISC), Springer, LNCS 2288, 39–49 (2001)

[7] Daemen,J., Rijmen,V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer (2002)

[8] Damgård, I.: A Design Principle for Hash Functions. In Brassard,G. (Ed.), Crypto'89, 9th Annual International Cryptology Conference, 1989. Springer, LNCS 435, 416–427 (1990)

[9] Dong, L., Wu, W., Wu, S., Zou, J.: Known-key distinguishers on round-reduced 3D block cipher. Information Security Applications (WISA), Springer, LNCS 7115, 55–69 (2012)

[10] FIPS197: Advanced Encryption Standard (AES). FIPS PUB 197 Federal Information Processing Standard Publication 197, U.S. Department of Commerce (2001)

[11] Gligoroski, D.: Narrow-pipe SHA-3 candidates differ significantly from ideal random functions defined over big domains. Faculty of Information Technology, Norwegian Univ. of Science and Technology, Trondheim, Norway.

[12] Knudsen, L.R.: DEAL – a 128-bit Block Cipher. Technical Report #151, University of Bergen, Dept. of Informatics, Norway (1998)

[13] Knudsen, L.R.: Nonrandom Properties of reduced-round Whirlpool. NES/DOC/UIB/WP5/016/2, Aug. 15 (2002)

[14] Lamberger, M., Mendel, F., Rechberger, C., Rijmen, V., Schlaffer, M.: Rebound Distinguishers: results on the full Whirlpool Compression Function. Asiacrypt 2009, 15th International Conference, Springer, LNCS 5912, 126–143 (2009)

[15] MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North-Holland Mathematical Library (1977)

[16] Mendel, F., Rechberger, C., Schlaffer,M., Thomsen, S.S.: The Rebound Attack: cryptanalysis of reduced Whirlpool and Grøstl. In: Dunkelman, O. (Ed.), Fast Software Encryption (FSE) 2009, Springer, LNCS 5665, 260–276 (2009)

[17] Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press (1997)

[18] Merkle, R.C.: One way hash functions and DES. In Brassard,G. (Ed.), Crypto'89, 9th Annual International Cryptology Conference, Springer, LNCS 435, 428–446 (1990)

[19] Mouha, N., Bjorstad, T.E., Preneel, B.: Non-randomness in the Sarmal compression function. The Ecrypt Hash Function website http://ehash.iaik.tugraz.at/wiki/Sarmal

[20] Nakahara Jr, J.: 3D: a Three-Dimensional Block Cipher. In: M.K.Franklin, L.C.K.Hui, D.S.Wong (Eds.), Cryptology and Network Security (CANS), Springer, LNCS 5339, 252–267 (2008)

[21] Nakahara Jr, J.: New Impossible Differential and Known-Key Distinguishers for the 3D Cipher. In: Bao, F., Weng, J. (Eds.), The 7th Information Practice and Experience Conference (ISPEC), Springer, LNCS 6672, 208–221 (2011)

[22] NIST: Cryptographic Hash Algorithm Competition. available at http://csrc.nist.gov/groups/ST/hash/sha-3/index.html (2007)

[23] NIST: Announcing request for candidate algorithm nominations for a new cryptographic hash algorithm (SHA-3) family. Federal Register, vol. 72, no. 212, Nov. 2 (2007)

[24] Rijmen, V., Preneel, B., De Win, E.: On weaknesses of non-surjective round functions. Design, Codes and Cryptography (12):3, 253–266 (1997)

A. Appendix

$$\alpha = \begin{pmatrix} \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \overset{\theta\circ\pi\circ\gamma\circ\sigma_i}{\rightarrow} \begin{pmatrix} \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \overset{\pi\circ\gamma\circ\sigma_{i+1}}{\rightarrow}$$

$$\begin{pmatrix} \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \Delta & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \Delta & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Delta & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Delta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \Delta & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \Delta & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta \end{pmatrix} \overset{\theta}{\rightarrow} \begin{pmatrix} \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \end{pmatrix} \overset{\theta\circ\pi\circ\gamma\circ\sigma_{i+2}}{\not\rightarrow}$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \end{pmatrix} \overset{\sigma_{i+4}\circ\gamma^{-1}\circ\pi^{-1}\circ\sigma_{i+5}}{\leftarrow} \begin{pmatrix} 0 & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & 0 & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & 0 & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & 0 & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & 0 & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & 0 & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & 0 & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & 0 \end{pmatrix} = \eta \quad (9)$$

