A Simplified Representation of AES

Henri Gilbert*

ANSSI, France
henri.gilbert@ssi.gouv.fr

Abstract. We show that the so-called *super S-box representation* of AES – that provides a simplified view of two consecutive AES rounds – can be further simplified. In the *untwisted representation* of AES presented here, two consecutive AES rounds are viewed as the composition of a non-linear transformation S and an affine transformation R that respectively operate on the four 32-bit columns and on the four 32-bit rows of their 128-bit input. To illustrate that this representation can be helpful for analysing the resistance of AES-like ciphers or AES-based hash functions against some structural attacks, we present some improvements of the known-key distinguisher for the 7-round variant of AES presented by Knudsen and Rijmen at ASIACRYPT 2007. We first introduce a known-key distinguisher for the 8-round variant of AES which constructs a 2^{64}-tuple of (input,output) pairs satisfying a simple integral property. While this new 8-round known-key distinguisher is outperformed for 8 AES rounds by known-key differential distinguishers of time complexity 2^{48} and 2^{44} presented by Gilbert and Peyrin at FSE 2010 and Jean, Naya-Plasencia, and Peyrin at SAC 2013, we show that one can take advantage of its specific features to mount a known-key distinguisher for the 10-round AES with independent subkeys and the full AES-128. The obtained 10-round distinguisher has the same time complexity 2^{64} as the 8-round distinguisher it is derived from, but the highlighted input-output correlation property is more intricate and therefore its impact on the security of the 10-round AES when used as a known key primitive, e.g. in a hash function construction, is questionable. The new known-key distinguishers do not affect at all the security of AES when used as a keyed primitive, for instance for encryption or message authentication purposes.

1 Introduction

In this paper we present an alternative representation of AES. More precisely we show that AES can be viewed as the composition of other elementary transformations than those originally used for the specification of its round function. While one might wonder whether selecting any of the equivalent descriptions of a cipher is more than an arbitrary convention, numerous examples illustrate that the choice of an appropriate description may be very useful for highlighting some

* This work was partially supported by the French National Research Agency through the BLOC project (contract ANR-11-INS-011)

of its structural features and serve as a starting point for its cryptanalysis or for optimised implementions. To take a simple example, it is well known that while the so-called ladder representation of the Feistel scheme is strictly equivalent to its more traditional twisted representation for any even number of rounds, it is helpful for understanding some attacks against DES and DES-like ciphers, for instance the Davies-Murphy attack [8].

In the case of AES, several alternative representations have been proposed [9, 20] to highlight some aspects of its algebraic structure. These representations respectively allow to relate the ciphertext to the plaintext using continued fractions, resp. algebraic equations over $GF(2^8)$. In [2] it was shown that numerous dual ciphers of AES - *i.e.* equivalent descriptions of AES up to fixed, easy to compute and to invert bijective mappings on the plaintexts, the ciphertexts, and the keys - can be obtained by applying appropriately chosen modifications to the irreducible polynomial used to represent $GF(2^8)$, the affine transformation in the S-box, the coefficients of MixColumns, etc. This observation was further extended in [3]. While these dual ciphers can be considered as equivalent representations of AES, these representations essentially preserve the structure of the round function of the AES up to small variations on the exact parameter of each elementary transformation. They are therefore closer to the original AES than the equivalent representations we consider in this paper.

The starting point for the AES representation introduced here is the so-called super S-box (or super-box) representation of two AES rounds which allows to describe two consecutive AES rounds as the composition of one single non-linear operation, namely a range of four parallel 32-bit to 32-bit key-dependent S-boxes and several affine transformations. This representation was introduced in [7] by the designers of AES as a useful notion for the analysis of AES differentials over two rounds. It was subsequently reused in [11, 12] and [18] in order to extend so-called rebound attacks on AES-like permutations by at least one round: this improved rebound technique, sometimes referred to as super S-box cryptanalysis, was shown to be applicable in two related contexts, the cryptanalysis of AES-like hash functions and the investigation of so-called known-key distinguishers for AES-like block ciphers. Many recent improved distinguishers for reduced-round versions of AES-like hash functions such as the SHA-3 candidates Grøstl and ECHO are using super S-boxes, e.g. [19, 16, 15].

We introduce a novel representation of two consecutive AES rounds that results from an extra simplification of the super S-box representation. The simplification relates to the description of the affine transformations that surround the 32-bit super S-boxes. We show that all these transformations can be replaced by one simple 32-bit oriented affine transformation that operates on the rows of the 4×4 matrix of bytes representing the current state. We propose to name the resulting view of two or more generally r AES rounds the *untwisted* representation since it avoids viewing the affine transformations that surround the super S-boxes as column-oriented operations "twisted" by the action the ShiftRows transformation. The untwisted representation thus provides an equivalent description of two consecutive AES rounds as the composition of:

– *a non-linear transformation* denoted by S (a shorthand for "super S-boxes") that consists of the parallel application of four non-linear bijective mappings which operate on the four 32-bit columns of the AES state. These four mappings are essentially super S-boxes up to permutations of the four input bytes and the four output bytes of each column;
– *an affine transformation* denoted by R (a shorthand for "MixRows") that consists of the parallel application of four affine mappings which operate on the four 32-bit rows of the AES state.

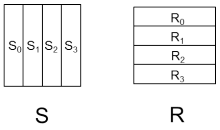

Fig. 1. Equivalent representation of two AES rounds as the composition $R \circ S$ of four parallel non-linear bijections of the columns and four parallel affine bijections of the rows of the input state.

As shown in Figure 1, two consecutive AES rounds can thus be viewed as one "super-round" that is the composition $R \circ S$ of S and R. As will be shown more in detail in the sequel, the small price to pay for this simplified view is that in the resulting equivalent representation of $2r$ AES rounds as the composition of r super-rounds, the first (resp. last) super-round is preceded (resp. followed) by a simple affine permutation.

While an alternative representation of a cipher can obviously be regarded in itself neither as a design nor as a cryptanalysis result, we believe that the simplicity of the new representation can play a significant heuristic role in the investigation of structural attacks on reduced-round versions of AES. Indeed, the new representation pushes the advantage of the super S-box representation of highlighting the 32-bit structure underlying the AES transformation one step further.

To illustrate this alternative representation, we present extensions of the known cryptanalytic results on reduced-round versions of AES in the so-called known-key model. The known-key model was first introduced by Knudsen and Rijmen in [17]. Attacks in this model are most often named known-key distinguishers and we will use this terminology in the sequel.[1] An integral known-key distinguisher for the 7-round AES was introduced by Knudsen and Rijmen in [17]. We first present an improvement of this distinguisher whose idea was inspired by the use of the untwisted representation of AES. This provides a known-key distinguisher against the 8-round AES. While this distinguisher is outperformed by the differential known-key distinguishers for the 8-round AES

[1] This terminology may seem a bit confusing since known-key distinguishers have little to do with the notion of distinguisher one considers in more traditional security models, namely a testing algorithm with an oracle access capability. But on the other hand the wording known-key distinguisher conveys probably less risks of misinterpretation than the wording known-key attack.

of [12] and [14], whose respective complexities are 2^{48} and 2^{44}, we show that one can take advantage of its specific features, that reflect integral properties of the 8-round AES, to extend it by one outer round at both sides. We thus obtain the first known-key distinguisher for the full 10-round AES. This known-key distinguisher has the same time complexity 2^{64} (now measured as an equivalent number of 10-round AES encryptions) as the one of the 8-round distinguisher it is derived from, but the highlighted input-output correlation property is more intricate. We nevertheless provide some evidence that unlike some generic known distinguishers that are known to exist for block ciphers if the key size is sufficiently small, the obtained distinguisher can reasonably be considered meaningful. While in this paper we will only investigate the security of AES in the known-key model, it is worth mentioning a recent result on the security of AES in a related but even stronger security model, namely the chosen-key distinguisher on the 9-round AES-128 of [10].

The rest of this paper is organized as follows. In Section 2, we introduce the novel representation of two consecutive AES rounds and of $2r$ AES rounds. In Section 3, we propose a definition of the known-key model, *i.e.* we define the adversaries considered in this model and we remind known impossibility results on the resistance of block ciphers to all known-key distinguishers. In Section 4, we show how to use the untwisted representation of AES to mount known-key distinguishers for the 8-round AES and its extension to the full 10-round AES and why the latter distinguisher can be considered meaningful.

2 A new representation of AES

Notational conventions and usual representation of AES. Throughout this paper we most often denote the composition of two mappings F and G multiplicatively by $F \cdot G$ instead of using the more classical notation $G \circ F$. The advantage of this notation in the context considered here is that when read from left to right it describes the successive transformations that are applied to the input value.

Let us briefly recall the AES features that will be useful for the sequel and the associated notation. Each AES block is represented by a four times four matrix of bytes. While there are three standard versions of AES, of respective key lengths k= 128, 192, and 256 bits and respective number of rounds 10, 12, and 14 rounds, for the purpose of this paper we restrict ourselves for the sake of simplicity to the full 10-round AES-128 and reduced-round versions of this cipher.[2] For $r \leq 10$, the r-round version of the AES-128 encryption function is denoted by AES_r and is parametrized by $(r+1)$ 128-bit subkeys denoted by K_0 to K_r. These subkeys are derived from a k-bit key K by the key schedule; since the exact features of the AES-128 key schedule are not relevant for the analysis

[2] However, since the AES properties we are investigating do not relate to the key schedule but to the data encryption part of the block cipher that is the same for all AES versions, all the presented results are also applicable to reduced-round versions of AES-192 and AES-256.

presented here, we do not detail them and refer to the full specification of AES for their description. Each round of the encryption function AES_r is the composition $SB \cdot SR \cdot MC \cdot AK$ of four transformations named SubBytes or SB, ShiftRows or SR, MixColumns or MC, and AddRoundKey or AK. SubBytes applies a fixed 8-bit to 8-bit bijective S-box to each input byte, ShiftRows circularly shifts each of the four 4-byte rows of the input state by 0, 1, 2, and 3 bytes to the left, MixColumns applies to each of the four-byte columns of the input state, viewed as a 4-coordinate vector with $GF(2^8)$ coefficients, a left multiplication by a fixed 4×4 matrix M with $GF(2^8)$ coefficients, and at round $i \in [1; r]$, AddRoundKey or AK consists of a bytewise exclusive or of the input block with subkey K_i.[3] The first round of AES_r is preceded by a key addition with the subkey K_0 and the MixColumns operation is omitted in the last round. In the sequel we will sometimes also have to refer to the variant of AES_r where the MixColumns transformation is kept in the last round: we will denote this variant by AES_{r^+}. At the end of Section 4, we will also have to refer to the r-round variant of AES parametrized by $r+1$ independent subkeys. Depending whether the MixColumns transformation is omitted or kept in the last round, we will denote this variant by AES^*_r or $\text{AES}^*_{r^+}$.

Super S-box representation of 2 consecutive AES rounds. The super S-box representation allows to view two consecutive AES rounds as the parallel invocation of four 32-bit to 32 bit mappings named super S-boxes - which are applied to the four columns of the AES state - surrounded by affine applications. More in detail, since the transformations SB and SR commute and the composition of transformations is associative, the composition of two consecutive rounds:

$$SB \cdot SR \cdot MC \cdot AK \cdot SB \cdot SR \cdot MC \cdot AK$$

can be rewritten as:

$$SR \cdot (SB \cdot MC \cdot AK \cdot SB) \cdot SR \cdot MC \cdot AK.$$

We can notice that the middle term in brackets, *i.e.* $SuperSB = (SB \cdot MC \cdot AK \cdot SB)$, where $SuperSB$ stands for "Super S-boxes", is the composition of transformations that all preserve the column-wise structure of the AES state. Thus $SuperSB$ splits up into 4 parallel key-dependent bijective transformations of one column of the input state. It is surrounded by the left, resp right affine transformations SR, resp $SR \cdot MC \cdot AK$. Each super S-box applies its 4-byte input column the composition of 4 parallel S-box invocations, a left multiplication by the MixColumn matrix M, a xor with a 32-bit subkey column, and 4 final parallel S-box invocations.

[3] Since AddRoundKey is parametrized by a subkey the use of the notation AK, that suggests a fixed transformation, is a slight abuse of notation, but this notation is convenient in the context of this paper: in the sequel AK just stands for a xor with some constant — whose value does not affect the properties we consider.

Moving to the untwisted representation of 2 consecutive AES rounds. We now show how to move from the super S-box representation of two consecutive rounds to their untwisted representation as the composition $S \cdot R$ of four parallel column-wise non-linear transformations and four parallel row-wise affine transformations. We first observe that the periodic repetition, in r iterations, of the 2-round pattern associated with the super S-box representation:

$$SR \cdot SuperSB \cdot SR \cdot MC \cdot AK$$

can be equivalently viewed as the periodic repetition in r iterations of the cyclically shifted periodic 2-round pattern:

$$SuperSB \cdot SR \cdot MC \cdot AK \cdot SR$$

up to a minor correction, namely the left composition of the first iteration with SR and the right composition of the last iteration with SR^{-1}. Now in order to move to the aimed 2-round representation the conducting idea is to left and right-compose the $SuperSB$ and $SR \cdot MC \cdot AK \cdot SR$ transformations using well chosen byte permutations P and Q and their inverses P^{-1} and Q^{-1}. Due to the cancellation effect produced by the alternate use of these permutations and their inverse, r iterations of the obtained 2-round description:

$$(Q^{-1} \cdot SuperSB \cdot P^{-1}) \cdot (P \cdot SR \cdot MC \cdot AK \cdot SR \cdot Q)$$

gives, for any choice of the two byte permutations, exactly the same product as r iterations of the 2-round transformation it is derived from, up to a left composition of the first iteration by Q^{-1} and a right composition of the last iteration by Q. In order for the byte permutations P and Q to provide the desired untwisted representation, they must satisfy the two following extra requirements:

- (i) the non-linear transformation $S = Q^{-1} \cdot SuperSB \cdot P^{-1}$ must operate on columns;
- (ii) the affine transformation $R = P \cdot SR \cdot MC \cdot AK \cdot SR \cdot Q$ must operate on rows.

In order to describe the byte permutations satisfying the above requirements that we found, we introduce the following auxiliary byte permutations:

- we denote by T the matrix transposition that operates on 4×4 matrices of bytes as follows:

$$T : \begin{pmatrix} a_0 & a_4 & a_8 & a_{12} \\ a_1 & a_5 & a_9 & a_{13} \\ a_2 & a_6 & a_{10} & a_{14} \\ a_3 & a_7 & a_{11} & a_{15} \end{pmatrix} \mapsto \begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 & a_7 \\ a_8 & a_9 & a_{10} & a_{11} \\ a_{12} & a_{13} & a_{14} & a_{15} \end{pmatrix}$$

- we denote by SC (or SwapColumns) the swapping of the second and fourth columns of the input state:

$$SC : \begin{pmatrix} a_0 & a_4 & a_8 & a_{12} \\ a_1 & a_5 & a_9 & a_{13} \\ a_2 & a_6 & a_{10} & a_{14} \\ a_3 & a_7 & a_{11} & a_{15} \end{pmatrix} \mapsto \begin{pmatrix} a_0 & a_{12} & a_8 & a_4 \\ a_1 & a_{13} & a_9 & a_5 \\ a_2 & a_{14} & a_{10} & a_6 \\ a_3 & a_{15} & a_{11} & a_7 \end{pmatrix}$$

Proposition 1. *The byte permutations*

$$P = SR \cdot T \cdot SR^{-1} \quad : \begin{pmatrix} a_0 & a_4 & a_8 & a_{12} \\ a_1 & a_5 & a_9 & a_{13} \\ a_2 & a_6 & a_{10} & a_{14} \\ a_3 & a_7 & a_{11} & a_{15} \end{pmatrix} \mapsto \begin{pmatrix} a_0 & a_5 & a_{10} & a_{15} \\ a_3 & a_4 & a_9 & a_{14} \\ a_2 & a_7 & a_8 & a_{13} \\ a_1 & a_6 & a_{11} & a_{12} \end{pmatrix}$$

and

$$Q = SR^{-1} \cdot T \cdot SR \cdot SC \quad : \begin{pmatrix} a_0 & a_4 & a_8 & a_{12} \\ a_1 & a_5 & a_9 & a_{13} \\ a_2 & a_6 & a_{10} & a_{14} \\ a_3 & a_7 & a_{11} & a_{15} \end{pmatrix} \mapsto \begin{pmatrix} a_0 & a_7 & a_{10} & a_{13} \\ a_1 & a_4 & a_{11} & a_{14} \\ a_2 & a_5 & a_8 & a_{15} \\ a_3 & a_6 & a_9 & a_{12} \end{pmatrix}$$

satisfy the requirements (i) *and* (ii) *and thus result in the desired untwisted representation.*

Proof sketch.
(i): It is easy to see that P, Q, and their inverses operate on columns. Therefore $S = Q^{-1} \cdot SuperSB \cdot P^{-1}$ also operates on columns.
(ii): We can simplify the expression of R:

$$\begin{aligned} R &= P \cdot SR \cdot MC \cdot AK \cdot SR \cdot Q \\ &= SR \cdot T \cdot SR^{-1} \cdot SR \cdot MC \cdot AK \cdot SR \cdot SR^{-1} \cdot T \cdot SR \cdot SC \\ &= SR \cdot T \cdot MC \cdot AK \cdot T \cdot SR \cdot SC \end{aligned}$$

Since $T \cdot MC \cdot T$ and therefore $T \cdot MC \cdot AK \cdot T$ operates on rows and SR and SC also operate on rows, R operates on rows. □

The linear part of the row-wise affine transformation R determined by P and Q is described by the four following circulant matrices R_i, $i = 0$ to 3. Each matrix R_i operates on a 4-byte row vector x_i that represents row i of the input block of R and produces the 4-byte row vector $y_i = x_i \cdot R_i$ that represents row i of the linear part of the image of the input block by R. The coefficients of the R_i are those of the MixColumns matrix M (in a different order).

$$R_0 = R_2 = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 3 & 1 & 1 & 2 \\ 1 & 1 & 2 & 3 \\ 1 & 2 & 3 & 1 \end{pmatrix} \qquad R_1 = R_3 = \begin{pmatrix} 1 & 1 & 2 & 3 \\ 1 & 2 & 3 & 1 \\ 2 & 3 & 1 & 1 \\ 3 & 1 & 1 & 2 \end{pmatrix} \qquad M = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix}$$

Remark. (P, Q) is not the unique pair of byte permutations that satisfy requirements (i) and (ii). Given any permutations σ and τ of the set $\{0, 1, 2, 3\}$, let us denote by C_σ, resp. D_τ the associated column and row permutations, that on input a 4-tuple (x_0, x_1, x_2, x_3) of columns, resp. of rows produces the permuted 4-tuple (x'_0, x'_1, x'_2, x'_3) of columns, resp. of rows given by $x'_{\sigma(i)} = x_i$, resp. $x'_\tau(i) = x_i$, *i.e.* $x'_i = x_{\sigma^{-1}(i)}$ resp. $x'_i = x_\tau^{-1}(i)$, $i = 0$ to 3. It is easy to see that all the pairs of byte permutations $(P_{\sigma,\tau}, Q_{\sigma,\tau}) = (C_\sigma \cdot D_\tau \cdot P, Q \cdot D_{\tau^{-1}} \cdot C_{\sigma^{-1}})$ also satisfy requirements (i) and (ii). We will however only use (P, Q) in the sequel.

Resulting untwisted representation of $\mathbf{AES_{2r^+}}$ and $\mathbf{AES_{2r}}$. The former 2-round untwisted representation of two consecutive AES rounds immediately results in the following equivalent untwisted description of the $2r$-round version AES_{2r^+} of the encryption function of AES (in which the MixColumns transformation is kept in the last round).

$$AES_{2r^+} = AK \cdot IP \cdot (S \cdot R)^r \cdot FP,$$

where the initial and final permutations IP and FP are the byte permutations given by:

$$\begin{array}{l} IP = \quad SR \cdot Q \quad = T \cdot SR \cdot SC; \\ FP = Q^{-1} \cdot SR^{-1} = IP^{-1}. \end{array}.$$

This representation AES_{2r^+} is illustrated on Figure 2. To confirm the equivalence of the above representation of AES_{2r^+} with its usual representation using SB, SR, MC, and AK, implementations based on both representations were checked to provide equal output values on a few input values.

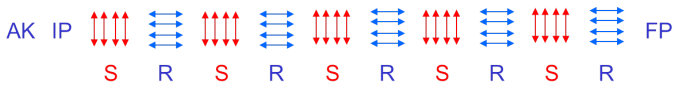

Fig. 2. Equivalent representation of AES_{10^+}. IP and FP are permutations of the byte positions.

The former representation of AES_{2r^+} can be used to derive a first representation of AES_{2r}, that will be used in the sequel to mount a known-key distinguisher for AES_8. The right composition of AES_{2r} with an appropriate conjugate of MC^{-1} is required in order to cancel out the MixColumns operation in the last round. If one "develops" the last occurrence of R and simplifies the obtained expression, one obtains the equality:

$$AES_{2r} = AK \cdot IP \cdot (S \cdot R)^{r-1} \cdot S \cdot P \cdot SR \cdot AK.$$

We also introduce a second equivalent representation of AES_{2r} that will be used in the sequel to mount a known-key distinguisher for AES_{10}: we start from an equivalent representation of the $2(r-1)$-round version $\mathrm{AES}_{2(r-1)^+}$ of AES, apply a left composition with a full round and a right composition with a last round without MixColumns, and simplify the obtained expression using the equality $R = P \cdot SR \cdot MC \cdot AK \cdot SR \cdot Q$.

$$\begin{aligned} AES_{2r} &= (AK \cdot SB \cdot SR \cdot MC) \cdot AES_{2(r-1)^+} \cdot (SB \cdot SR \cdot AK) \\ &= AK \cdot SB \cdot SR \cdot MC \cdot AK \cdot SR \cdot Q \cdot (S \cdot R)^{r-1} \cdot Q^{-1} \cdot SR^{-1} \cdot SB \cdot SR \cdot AK \\ &= AK \cdot SB \cdot P^{-1} \cdot R \cdot (S \cdot R)^{r-1} \cdot Q^{-1} \cdot SB \cdot AK \\ &= AK \cdot P^{-1} \cdot SB \cdot R \cdot (S \cdot R)^{r-1} \cdot SB \cdot Q^{-1} \cdot AK \end{aligned}$$

Thus AES_{2r} can be equivalently viewed as a middle transformation $R \cdot (S \cdot R)^{r-1}$ preceded and followed by simplified initial and final "external rounds", namely $AK \cdot P^{-1} \cdot SB$ and $SB \cdot Q^{-1} \cdot AK$.

3 The known-key model

We believe that the untwisted AES representation introduced above can potentially help analysing known *structural attacks* of reduced-round versions of AES, AES-like ciphers, or AES-based hash functions.[4] In the next section we will present two "attacks" that substantiate this belief. They both happen to belong to a quite specific class of structural attacks, the so-called known-key distinguishers, and respectively relate to a reduced-round version of AES and the full 10-round AES-128. In this section we introduce the underlying security model, that is named the *known-key* model. This model was inspired from the cryptanalysis of hash functions and first introduced by Knudsen and Rijmen in [17]. The difference between the known-key model and the usual security model considered for block ciphers can be outlined as follows.

- In the usual model, the adversary is given a *black box* (oracle) access to an instance of the encryption function associated with a random *secret* key and its inverse and must find the key or more generally efficiently distinguish the encryption function from a perfect random permutation;
- In the known-key model, the adversary is given a *white box* (*i.e.* full) access to an instance of the encryption function associated with a *known* random key and its inverse and her purpose is to simultaneously control the inputs and the outputs of the primitive, *i.e.* to achieve input-output correlations she could not efficiently achieve with the inputs and outputs of a perfect random permutation to which she would have an oracle access.

We now propose a more detailed definition of the known-key model – *i.e.* of the adversaries considered in this model, that are named *known-key distinguishers*. In order to capture the idea that the goal of such adversaries is to derive an N-tuple of input blocks of the considered block cipher E that is "abnormally correlated" with the corresponding N-tuple of output blocks, we first introduce the notion of T-intractable relation on N-tuples of E blocks. This notion (that is independent of E up to the fact that for the sake of simplicity we are using the time complexity of E as the unit for quantifying time complexities) is closely related to the notion of correlation intractable relation proposed in [6]. It essentially expresses that it is difficult to derive from oracle queries to a random permutation and its inverse an N-tuple of input/output pairs satisfying the relation.

Definition 1 (T-intractable relation). *Let $E : (K, X) \in \{0,1\}^k \times \{0,1\}^n \mapsto E_K(X) \in \{0,1\}^n$ denote a block cipher of block size n bits. Let $N \geq 1$ and $\mathcal{R}$ denote an integer and any relation*[5] *over the set S of N-tuples of n-bit blocks. $\mathcal{R}$*

[4] By structural attacks we mean here attacks that unlike statistical attacks, e.g. differential and linear cryptanalysis, do not consider the detail of the algorithm's elementary ingredients such as the S-boxes, but put more emphasis on their overall construction, their use of transformations that preserve the byte structure or the 32-bit structure of the data, etc.

[5] Let us remind that for any set S, a relation $\mathcal{R}$ over S can be defined as a subset of the cartesian product $S \times S$ and that for any pair (a, b) of $S \times S$, $a\mathcal{R}b$ means that (a, b) belongs to this subset.

is said to be T-intractable relatively to E if, given any algorithm $\mathcal{A}'$ that is given an oracle access to a perfect random permutation Π of $\{0,1\}^n$ and its inverse, it is impossible for $\mathcal{A}'$ to construct in time $T' \leq T$ two N-tuples $\mathcal{X}' = (X'_i)$ and $\mathcal{Y}' = (Y'_i)$ such that $Y'_i = \Pi(X'_i)$, $i = 1 \cdots N$ and $\mathcal{X}' \; \mathcal{R} \; \mathcal{Y}'$ with a success probability $p' \geq \frac{1}{2}$ over Π and the random choices of $\mathcal{A}'$. The computing time T' of $\mathcal{A}'$ is measured as an equivalent number of computations of E, with the convention that the time needed for one oracle query to Π or Π^{-1} is equal to 1. Thus if q' denotes the number of queries of $\mathcal{A}'$ to Π or Π^{-1}, $q' \leq T'$.

Definition 2 (known-key distinguisher). *Let $E : (K, X) \in \{0,1\}^k \times \{0,1\}^n \mapsto E_K(X) \in \{0,1\}^n$ denote a block cipher of block size n bits. A known-key distinguisher $(\mathcal{R}, \mathcal{A})$ of order $N \geq 1$ consists of (1) a relation $\mathcal{R}$ over the N-tuples of n-bit blocks (2) an algorithm $\mathcal{A}$ that on input a k-bit key K produces in time $T_{\mathcal{A}}$, i.e. in time equivalent with $T_{\mathcal{A}}$ computations of E, an N-tuple $\mathcal{X} = (X_i)_{i=1\cdots N}$ of plaintext blocks and an N-tuple $\mathcal{Y} = (Y_i)_{i=1\cdots N}$ of ciphertext blocks related by $Y_i = E_K(X_i)$, The two following conditions must be met:*
(i) The relation $\mathcal{R}$ must be $T_{\mathcal{A}}$-intractable relatively to E.
(ii) The validity of $\mathcal{R}$ must be efficiently checkable: we formalize this requirement by incorporating the time for checking whether two N-tuples are related by $\mathcal{R}$ in the computing time $T_{\mathcal{A}}$ of algorithm $\mathcal{A}$.[6]

It is important for the sequel to notice that in the former definition, while the algorithm $\mathcal{A}$ takes a random key K as input, the relation $\mathcal{R}$ satisfied by the N-tuples of input and output blocks constructed by $\mathcal{A}$ is the same for all values of K and must be efficiently checkable without knowing K.

Example 1. The following example of a known-key distinguisher of order $N = 2$ illustrates the link between the use of block ciphers for hashing purposes and their security in the known-key model. Let E denote a block cipher of key length k bits and block length n bits and (X_1, X_2) and (Y_1, Y_2) denote two pairs of n-bit blocks. We define the relation $(X_1, X_2) \; \mathcal{R} \; (Y_1, Y_2)$ by the conditions $X_1 \neq X_2$ and $X_1 \oplus Y_1 = X_2 \oplus Y_2$. The definition of relation $\mathcal{R}$ obviously implies that if E is vulnerable to a known-key distinguisher $(\mathcal{R}, \mathcal{A})$ of complexity $T \ll 2^{\frac{n}{2}}$, then the compression function $h : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n : (K, X) \mapsto X \oplus E_K(X)$ derived from E using the Matyas-Meyer-Oseas construction is vulnerable to a collision attack of complexity T that is more powerful than any generic collision attack against h.[7]

In the next example and throughout the rest of this paper, we are using the following notation to describe integral properties of partial AES encryptions and decryptions.

Notation. *Let $F : \{0,1\}^n \to \{0,1\}^n$ denote any mapping over the block space and let us consider the transformation by F of a structure $\mathcal{X}$ of $N = 2^{8m}$ blocks,*

[6] This avoids specifying an explicit upper bound on the time complexity for checking whether two N-tuples are related by $\mathcal{R}$. In practice one typically expects the time complexity for checking $\mathcal{R}$ to be at most the one of N computations of E.

[7] It could be shown that if $T \ll 2^{\frac{n}{2}}$, $\mathcal{R}$ is T-intractable.

$m \leq 16$. *An input or output byte* b_i, $i \in \{0, \cdots, 15\}$ *of* F *is said to be constant and marked* C *if it takes one constant value. It is said to be uniform and marked* U *if it takes each of the* 2^8 *possible values exactly* $2^{8(m-1)}$ *times. A* s*-tuple* $(b_{i_1}, \cdots, b_{i_s})$*, where* $s \leq m$ *and* $i_1, \cdots i_s \in \{0, \cdots, 15\}$*, of input or output bytes of* F *is said to be uniform and marked* $U_1, \cdots U_s$ *if* $(b_{i_1}, \cdots, b_{i_s})$ *takes each of the* 2^{8s} *possible* s*-tuple values exactly* $2^{8(m-s)}$ *times.*

Example 2. The known-key distinguisher for AES_7 introduced in [17] uses a relation $\mathcal{R}$ of order $N = 2^{56}$ that exploits integral properties of partial AES encryptions and decryptions. The following integral properties are used:

$$\begin{pmatrix} U_1 & C & C & C \\ C & U_2 & C & C \\ C & C & U_3 & C \\ C & C & C & U_4 \end{pmatrix} \overset{+4\text{r}}{\rightarrow} \begin{pmatrix} U & U & U & U \\ U & U & U & U \\ U & U & U & U \\ U & U & U & U \end{pmatrix} \text{ and } \begin{pmatrix} U_1 & C & C & C \\ U_2 & C & C & C \\ U_3 & C & C & C \\ U_4 & C & C & C \end{pmatrix} \overset{-3\text{r}}{\rightarrow} \begin{pmatrix} U & U & U & U \\ U & U & U & U \\ U & U & U & U \\ U & U & U & U \end{pmatrix}$$

where $4r$ denotes 4 consecutive AES encryption rounds without MixColumns in the last round and $-3r$ denotes 3 full AES decryption rounds. These properties imply that if a middle structure $\mathcal{Z}$ of $N = 2^{56}$ blocks is chosen as to satisfy the properties of the intermediate block of the scheme below, then by applying 4 forward encryption rounds and 3 backward decryption rounds to this structure one obtains a N-tuple of (plaintext, ciphertext) pairs that satisfy the relation $\mathcal{R}$ that (1) the N input blocks are pairwise distinct and (2) each of the 16 input bytes and each of the 16 output bytes is uniformly distributed.

$$\begin{pmatrix} U & U & U & U \\ U & U & U & U \\ U & U & U & U \\ U & U & U & U \end{pmatrix} \overset{-3\text{r}}{\leftarrow} \overset{\mathcal{Z}}{\begin{pmatrix} U_1 & C & C & C \\ U_5 & U_2 & C & C \\ U_6 & C & U_3 & C \\ U_7 & C & C & U_4 \end{pmatrix}} \overset{+4\text{r}}{\rightarrow} \begin{pmatrix} U & U & U & U \\ U & U & U & U \\ U & U & U & U \\ U & U & U & U \end{pmatrix}$$

While $\mathcal{R}$ could be shown to be N-intractable by the same kind of arguments as those used in the next section, we do not give a detailed proof here. The authors of [17] do not use exactly the same notion of T-intractable relation, but conjecture the related – somewhat stronger – property that "for a randomly chosen 128-bit permutation, finding a collection of 2^{56} texts in similar time, using similar (little) memory and with similar properties as in the case of 7-round AES has a probability of succeeding which is very close to zero".

Example 3. In [12] a known-key distinguisher of order $N = 2$ for AES_8 of time complexity $T = 2^{48}$, memory about 2^{32}, and success probability close to 1 is described. The associated relation $\mathcal{R}$ is differential in nature. It is defined as follows: $(X_1, X_2)\mathcal{R}(Y_1, Y_2)$ if and only if $X_1 \neq X_2$, the single non-zero bytes of the input difference $X_1 \oplus X_2$ are the diagonal bytes, *i.e.* the bytes numbered 0, 5, 10, and 15, and the single non-zero bytes of the output difference $Y_1 \oplus Y_2$ are the four bytes numbered 0, 7, 10, and 13. It was shown in [13] that given a perfect random permutation Π, the best method to get an input pair (X_1, X_2) and an output pair $(Y_1, Y_2) = (\Pi(X_1), \Pi(X_2))$ satisfying $(X_1, X_2)\mathcal{R}(Y_1, Y_2)$ is the so-called limited birthday technique, that requires about 2^{65} oracle queries for a target success probability of about $\frac{1}{2}$. With only $T = 2^{48}$ oracle queries, the success probability of this best method would decrease to about 2^{-17}.

Example 4. When applied to block ciphers, so-called zero-sum distinguishers [1, 4, 5], that thanks to higher order differential properties produce structures $(X_i, Y_i)_{i=1\cdots N}$ of N (input, output) pairs such that $\bigoplus_{i=1}^{N} X_i = \bigoplus_{i=1}^{N} Y_i = 0$ also represent examples of known-key distinguishers.

Impossiblity results on the resistance of block ciphers to all known-key distinguishers. Specifying the requirements on the resistance of a block cipher E against known-key distinguishers is a notoriously difficult issue because of an impossibility result that was first pointed out by Canetti, Goldreich, and Halevi in [6]. While the notion of correlation intractability was originally used to state this result, the related notion of resistance against known-key distinguishers can be used to reformulate it as follows:

Proposition 2. *Every block cipher of key length k bits and block length n bits such that $k \leq n$ is vulnerable to a known-key distinguisher of order 1 and complexity about one computation of E.*

Proof sketch. In order to give the intuition of the proof, let us restrict ourselves to the situation where $k = n$. It suffices to use the whole specification of E in the definition of R to get the claimed result. Let us define $X \ \mathcal{R} \ Y$, where X and Y are any n-bit blocks, by the condition $Y = E_X(X)$. Given any known k-bit key K, the easy to compute values $X = K$ and $Y = E_K(K)$ are related by E_K and satisfy $X \ \mathcal{R} \ Y$. However, for any adversary $\mathcal{A}'$ that makes $q << 2^n$ queries to a perfect random permutation Π of the block space, finding X such that $X \ \mathcal{R} \ \Pi(X)$, *i.e.* $\Pi(X) = E_X(X)$ is very unlikely to succeed: by separately considering the cases where $\mathcal{A}'$ outputs a value X that belongs or does not belong to a queried pair it can indeed be shown that the success probability of $\mathcal{A}'$ is upper bounded by $\frac{q}{2^n} + \frac{1}{2^n - q}$, and is therefore negligible if $q << 2^n$. □

The former proposition can be easily extended as follows.

Proposition 3. *Every block cipher of block length n bits and key length $k = Nn$ bits is vulnerable to a known-key distinguisher of order N and complexity about N computations of E.*[8]

Proof sketch. We just need to replace the relation $\mathcal{R}$ used in the former proof by the following relation $\mathcal{R}_N$ over the N-tuples of blocks: if $\mathcal{X} = (X_i)_{i=1\cdots N}$ and $\mathcal{Y} = (Y_i)_{i=1\cdots N}$, $\mathcal{X}\mathcal{R}_N\mathcal{Y}$ iff $\forall i \in [1; N] E_{\mathcal{X}}(X_i) = Y_i$, where $E_{\mathcal{X}}$ denotes the block cipher E parametrized by the Nn-bit key $X_1||X_2||\cdots||X_N$. □

To summarize the above impossibility results, for a block cipher E of block and key lengths n and k, generic known-key distinguishers of order N are known to exist iff $k \leq Nn$.

Discussion. If $k > Nn$, any known-key distinguisher of order at most N can be reasonably conjectured to be *meaningful*, *i.e.* to reflect, unlike the *artificial* generic known-key distinguishers of Propositions 2 and 3, a meaningful correlation property of E. Now in the frequently encountered case where $k \leq Nn$,

[8] One can generalize the former result a bit further by noticing that if $k \leq Nn$, then given any easy to compute and easy to invert function $f : \{0,1\}^{Nn} \to \{0,1\}^k$, a simple variant of the known-key distinguisher of Proposition 3 can be obtained by replacing $\mathcal{R}_N$ by the relation $\mathcal{R}'_N$ defined by $\mathcal{X}\mathcal{R}'_N\mathcal{Y}$ iff $\forall i \in [1; N]$ $E_{f(\mathcal{X})}(X_i) = Y_i$.

that is met for instance for the known-key distinguisher of [17] where $k = 128$ and $Nn = 2^{56} \times 128$, characterizing which known-key distinguishers of order N should be considered *meaningful* and which ones should be considered *artificial* is a very complex issue. Finding a complete characterization remains an open problem that even lacks a rigorous statement and we will not attempt to solve it here. We will limit ourselves to propose informal criteria allowing to identify two classes of known key distinguishers that have little to do with artificial distinguishers identified so far and can be both reasonably considered meaningful.
– *Informal criterion 1.* One heuristic argument in favour of the view that the known-key distinguisher of Example 2 [17] for AES_7 is *meaningful* is the observation that while the description of the generic relations $\mathcal{R}$ and $\mathcal{R}_N$ used in Propositions 2 and 3 involve the specification of E itself, the relation $\mathcal{R}$ used in [17] has no obvious connection with the specification of E. More generally, if a known-key distinguisher uses an intractable relation $\mathcal{R}$ whose specification does not extensively reuse operations of E, this provides some heuristic evidence that it can be considered meaningful.[9]
– *Informal criterion 2.* While the informal criterion 1 sounds like a reasonable sufficient condition, we think it should not be considered as a necessary condition. In other words, known-key distinguishers that do not satisfy it, *i.e.* whose relation $\mathcal{R}$ re-uses some operations of E, should not be systematically ruled out as if they were all *artificial*. We informally state an alternative criterion for highlighting that independently of whether their relation $\mathcal{R}$ reuses operations of E or not, some known-key distinguishers have little to do with existing *artificial* distinguishers. One can observe that in the *artificial* distinguishers $(\mathcal{A}, \mathcal{R})$ of Propositions 2 and 3 and of the generalisation of Proposition 3 in the remark above, algorithm $\mathcal{A}$ produces an N-tuple $\mathcal{X}$ of input blocks from which the value of the whole key can be easily derived: in other words, one exploits the fact that $\mathcal{X}$ "encodes" the value of the entire key. If for a given known-key distinguisher $(\mathcal{A}, \mathcal{R})$ the entire key can neither be derived from the N-tuples of input values $\mathcal{X}$ nor from the N-tuples of output values $\mathcal{Y}$ produced by $\mathcal{A}$ one is brought back to a situation somewhat similar to the case where $k > Nn$ (a condition that obviously prevents $\mathcal{X}$ and $\mathcal{Y}$ from encoding the entire key) and this provides some evidence that $(\mathcal{A}, \mathcal{R})$ has little to do with the *artificial* distinguishers identified so far. We will use this informal criterion at the end of the next section.

4 Application: improved known-key distinguishers for $\mathbf{AES_8}$ and $\mathbf{AES_{10}}$

4.1 A known-key distinguisher for $\mathbf{AES_8}$

Let us now show how to use the first untwisted representation of AES_{2r} introduced in Section 2 in order to mount a known-key distinguisher of order $N = 2^{64}$ for AES_8. The distinguisher starts from a suitably chosen middle N-block structure and exploits the forward and backward properties of the final

[9] Giving a rigorous definition of the former informal criterion seems difficult. One might perhaps express that the verification of $\mathcal{R}$ is not substantially sped up by oracle accesses to E.

rounds, resp. the initial rounds of the AES_8, that are illustrated on Figure 3. These properties result from the fact that the initial and final rounds essentially consist of the composition $S \cdot R \cdot S$, up to simple initial and final transformations.

Property 1. *For any structure* $\mathcal{X}_{(a,b,c,d)} = \{(x \oplus a, b, c, d), x \in \{0,1\}^{32}\}$ *of* 2^{32} *input blocks — where* (a, b, c, d) *denotes an AES block of columns* a*,* b*,* c*, and* d *— each of the four 4-byte columns of the image of* $\mathcal{X}_{(a,b,c,d)}$ *by* $S \cdot R \cdot S$ *is uniformly distributed.*

This can be easily seen by following the column-wise transitions through transformations S, R, and S on the top of Figure 3 and by observing (1) that S transforms each column bijectively and (2) that if one fixes the second, third, and fourth input columns of R, each of the four output columns of R is a bijective affine function of the first input column. Since moreover $P \cdot SR \cdot AK$ is just a permutation of the byte positions followed by a key addition, each of the 16 bytes of the image of $\mathcal{X}_{(a,b,c,d)}$ by $S \cdot R \cdot S \cdot P \cdot SR \cdot AK$ is uniformly distributed and can be marked "U".

Property 2. *For any structure* $\mathcal{Y}_{(e,f,g,h)} = \{(y \oplus e, f, g, h), y \in \{0,1\}^{32}\}$ *of* 2^{32} *blocks, each four-byte column of the preimage of* $\mathcal{Y}_{(e,f,g,h)}$ *by* $S \cdot R \cdot S$ *is uniformly distributed.*

This can be easily seen by following the column-wise transitions through transformations S^{-1}, R^{-1}, and S^{-1} on the bottom of Figure 3 and observing (1) that S^{-1} transforms each column bijectively and that (2) if one fixes the second, third, and fourth input columns of R^{-1}, each of the four output columns of R^{-1} is a bijective affine function of the first input column. Since moreover IP^{-1} is a permutation of the byte positions, each of the 16 bytes of the preimage of $\mathcal{Y}_{(e,f,g,h)}$ by $AK \cdot IP \cdot S \cdot R \cdot S$ is uniformly distributed and can be marked "U".

$$\overset{\mathcal{X}_{(a,b,c,d)}}{\begin{pmatrix} U_1 & C & C & C \\ U_2 & C & C & C \\ U_3 & C & C & C \\ U_4 & C & C & C \end{pmatrix}} \overset{S}{\to} \begin{pmatrix} U_1 & C & C & C \\ U_2 & C & C & C \\ U_3 & C & C & C \\ U_4 & C & C & C \end{pmatrix} \overset{R}{\to} \begin{pmatrix} U_1^1 & U_1^2 & U_1^3 & U_1^4 \\ U_2^1 & U_2^2 & U_2^3 & U_2^4 \\ U_3^1 & U_3^2 & U_3^3 & U_3^4 \\ U_4^1 & U_4^2 & U_4^3 & U_4^4 \end{pmatrix} \overset{S}{\to} \begin{pmatrix} U_1^1 & U_1^2 & U_1^3 & U_1^4 \\ U_2^1 & U_2^2 & U_2^3 & U_2^4 \\ U_3^1 & U_3^2 & U_3^3 & U_3^4 \\ U_4^1 & U_4^2 & U_4^3 & U_4^4 \end{pmatrix} \overset{P \cdot SR \cdot AK}{\longrightarrow} \begin{pmatrix} U & U & U & U \\ U & U & U & U \\ U & U & U & U \\ U & U & U & U \end{pmatrix}$$

$$\overset{\mathcal{Y}_{(e,f,g,h)}}{\begin{pmatrix} U_1 & C & C & C \\ U_2 & C & C & C \\ U_3 & C & C & C \\ U_4 & C & C & C \end{pmatrix}} \overset{S^{-1}}{\to} \begin{pmatrix} U_1 & C & C & C \\ U_2 & C & C & C \\ U_3 & C & C & C \\ U_4 & C & C & C \end{pmatrix} \overset{R^{-1}}{\to} \begin{pmatrix} U_1^1 & U_1^2 & U_1^3 & U_1^4 \\ U_2^1 & U_2^2 & U_2^3 & U_2^4 \\ U_3^1 & U_3^2 & U_3^3 & U_3^4 \\ U_4^1 & U_4^2 & U_4^3 & U_4^4 \end{pmatrix} \overset{S^{-1}}{\to} \begin{pmatrix} U_1^1 & U_1^2 & U_1^3 & U_1^4 \\ U_2^1 & U_2^2 & U_2^3 & U_2^4 \\ U_3^1 & U_3^2 & U_3^3 & U_3^4 \\ U_4^1 & U_4^2 & U_4^3 & U_4^4 \end{pmatrix} \overset{(AK \cdot IP)^{-1}}{\longrightarrow} \begin{pmatrix} U & U & U & U \\ U & U & U & U \\ U & U & U & U \\ U & U & U & U \end{pmatrix}$$

Fig. 3. Forward and backward properties of $S \cdot R \cdot S$

We are using these properties to mount the known-key distinguisher of order $N = 2^{64}$ for AES_8 illustrated on Figure 4, *i.e.* an algorithm $\mathcal{A}$ allowing to efficiently derive from any known key a N-tuple $\big((X_i, Y_i)\big)_{i=1\cdots N}$ of AES_8 (input, output) pairs that satisfy the relation $\mathcal{R}$ defined as follows.

Relation $\mathcal{R}$: $(X_i)_{i=1\cdots N} \mathcal{R} (Y_i)_{i=1\cdots N}$ *iff the* N *blocks* X_i *are pairwise distinct and for each byte position* $j \in \{0, \cdots, 15\}$*, the* j*-th byte of the* X_i *and the* j*-th byte of the* Y_i *are uniformly distributed.*

Algorithm $\mathcal{A}$: The conducting idea is that in the untwisted representation of AES_8 in Figure 4, the initial and final rounds of Figure 3 are linked together by the transformation R, that is affine. This allows to construct a structure that simultaneously achieves the requirements on the intput and the output of R^{-1} in order to apply Properties 1 and 2. More in detail, we are using the 2^{64} chosen middle blocks structure $\mathcal{Z} = \mathcal{X}_{\mathbf{0}} \oplus R\mathcal{Y}_{\mathbf{0}}$, where $\mathcal{X}_{\mathbf{0}}$ and $\mathcal{Y}_{\mathbf{0}}$ are shorthands for $\mathcal{X}_{(0,0,0,0)}$ and $\mathcal{Y}_{(0,0,0,0)}$ and $\mathcal{X}_{\mathbf{0}} \oplus R\mathcal{Y}_{\mathbf{0}}$ denotes the set $\{X \oplus R(Y), X \in \mathcal{X}_{\mathbf{0}}, Y \in \mathcal{Y}_{\mathbf{0}}\}$. It directly results from the definition of $\mathcal{Z}$ that it can be partitioned into 2^{32} structures $\mathcal{X}_{\mathbf{0}} \oplus R(y,0,0,0) = \mathcal{X}_{R(y,0,0,0)}$ of 2^{32} blocks each, one for each value $y \in \{0,1\}^{32}$. In other words, $\mathcal{Z}$ can be partitioned into 2^{32} structures of the form $\mathcal{X}_{(a,b,c,d)}$. Therefore, due to Property 1, each byte of the image of $\mathcal{Z}$ by $S \cdot R \cdot S \cdot P \cdot SR \cdot AK$ satisfies property U. Let us denote by L and C the linear and constant parts of the affine mapping R, *i.e.* the linear mapping and the constant such that $\forall X \in \{0,1\}^{128} R(X) = L(X) \oplus C$. Since the linear mapping and the constant associated with R^{-1} are $L' = L^{-1}$ and $C' = L^{-1}(C)$, the preimage of $\mathcal{Z}$ by R is $R^{-1}(\mathcal{Z}) = L^{-1}(\mathcal{X}_{\mathbf{0}} \oplus L(\mathcal{Y}_{\mathbf{0}}) \oplus C) \oplus C' = L^{-1}(\mathcal{X}_{\mathbf{0}}) \oplus \mathcal{Y}_{\mathbf{0}}$. Therefore $R^{-1}(\mathcal{Z})$ can be partitioned into 2^{32} structures $\mathcal{Y}_{\mathbf{0}} \oplus L^{-1}(x,0,0,0) = \mathcal{Y}_{L^{-1}(x,0,0,0)}$ of 2^{32} blocks each[10] – one for each value $x \in \{0,1\}^{32}$. In other words, $R^{-1}(\mathcal{Z})$ can be partitioned into 2^{32} structures of the form $\mathcal{Y}_{(e,f,g,h)}$ and the application of Property 2 to $R^{-1}(\mathcal{Z})$ shows that each byte of the preimage of $R^{-1}(\mathcal{Z})$ by $AK \cdot IP \cdot S \cdot R \cdot S$, *i.e.* each byte of the preimage of $\mathcal{Z}$ by $AK \cdot IP \cdot S \cdot R \cdot S \cdot R$, satisfies property U.

$$\begin{pmatrix} U & U & U & U \\ U & U & U & U \\ U & U & U & U \\ U & U & U & U \end{pmatrix} \xleftarrow{(AK \cdot IP)^{-1}} \begin{pmatrix} U_1^1 & U_1^2 & U_1^3 & U_1^4 \\ U_2^1 & U_2^2 & U_2^3 & U_2^4 \\ U_3^1 & U_3^2 & U_3^3 & U_3^4 \\ U_4^1 & U_4^2 & U_4^3 & U_4^4 \end{pmatrix} \xleftarrow{(SRS)^{-1}} \overset{\substack{R^{-1}(\mathcal{Z}) = \mathcal{Y}_{\mathbf{0}} \oplus L^{-1}\mathcal{X}_{\mathbf{0}} \\ \equiv 2^{32} \times}}{\begin{pmatrix} U_1 & C & C & C \\ U_2 & C & C & C \\ U_3 & C & C & C \\ U_4 & C & C & C \end{pmatrix}} \xleftarrow{R^{-1}} \overset{\substack{\mathcal{Z} = \mathcal{X}_{\mathbf{0}} \oplus R\mathcal{Y}_{\mathbf{0}} \\ \equiv 2^{32} \times}}{\begin{pmatrix} U_1 & C & C & C \\ U_2 & C & C & C \\ U_3 & C & C & C \\ U_4 & C & C & C \end{pmatrix}} \cdots$$

$$\cdots \xrightarrow{SRS} \begin{pmatrix} U_1^1 & U_1^2 & U_1^3 & U_1^4 \\ U_2^1 & U_2^2 & U_2^3 & U_2^4 \\ U_3^1 & U_3^2 & U_3^3 & U_3^4 \\ U_4^1 & U_4^2 & U_4^3 & U_4^4 \end{pmatrix} \xrightarrow{P \cdot SR \cdot AK} \begin{pmatrix} U & U & U & U \\ U & U & U & U \\ U & U & U & U \\ U & U & U & U \end{pmatrix}$$

Fig. 4. A known-key distinguisher for AES_8

In summary, we derived from the middle structure $\mathcal{Z}$ a N-tuple $\big((X_i, Y_i)\big)_{i=1\cdots N}$ of AES_8 (input, output) pairs that satisfy relation $\mathcal{R}$. The time complexity of the derivation of such an N-tuple is $T = N = 2^{64}$ AES_8 computations. To complete the proof that we have mounted a known-key distinguisher for AES_8, we just have to show that property $\mathcal{R}$ is T-intractable, *i.e.* that the success probability of any oracle algorithm $\mathcal{A}^{(\Pi,\Pi^{-1})}$ of overall time complexity upper bounded by N (and therefore of number q of queries also upper bounded by N) is negligible.

[10] One can notice that the above partitions of $\mathcal{Z}$ and $R^{-1}(\mathcal{Z})$ do not map into eachother through R.

Proposition 4. *For any oracle algorithm $\mathcal{A}$ that makes $q \leq N = 2^{64}$ oracle queries to a perfect random permutation Π of $\{0,1\}^n$ (where $n = 128$) or its inverse, the probability that $\mathcal{A}$ successfully outputs a N-tuple $\big((X_i, Y_i)\big)_{i=1\cdots N}$ of (input, output) pairs of Π that satisfy $\mathcal{R}$ is upper bounded by $\frac{1}{2^n-(N-1)}$ and hence by $\frac{1}{2^{n-1}}$.*

Proof. If at least one of the N pairs (X_i, Y_i) output by $\mathcal{A}$ does not result from the query X_i to Π or the query Y_i to Π^{-1}, then the probability that for this pair $Y_i = \Pi(X_i)$ and thus the success probability of $\mathcal{A}$ is upper bounded by $\frac{1}{2^n-(N-1)}$. In the opposite case, *i.e.* if $q = N$ and all the (X_i, Y_i) result from queries to Π or Π^{-1}, we can assume w.l.o.g. that (X_N, Y_N) results from the N-th query X_N or Y_N of $\mathcal{A}$ to Π or Π^{-1}. But given any pairs $(X_i, Y_i)_{i=1\cdots N-1}$ at most one value of the block Y_N, resp. X_N is such that each of the 16 bytes of $(Y_i)_{i=1\cdots N}$, resp. $(X_i)_{i=1\cdots N}$ be uniformly distributed.[11] However the oracle answer Y_N, resp. X_N is uniformly drawn from $\{0,1\}^n \setminus \{Y_1, \cdots Y_{N-1}\}$, resp. $\{0,1\}^n \setminus \{X_1, \cdots X_{N-1}\}$. Therefore the probability that the answer to the N-th query allows the output of $\mathcal{A}$ to satisfy property $\mathcal{R}$ is also upper bounded by $\frac{1}{2^n-(N-1)}$ in this case. □

Discussion. The known-key distinguisher of order $N = 2^{64}$ for AES_8 presented above has a time complexity of about 2^{64}. It is obviously applicable without modification to the AES_8 variant parametrized by independent subkeys AES_8^*. In both cases, the fact that informal criterion of Section 3 is met, *i.e.* that the relation $\mathcal{R}$ used by the distinguisher has no obvious connection with the AES specification suggests that the obtained known-key distinguisher can be considered meaningful. While the presented 8-round known-key distinguisher is outperformed by the differential known-key distinguishers for AES_8 of complexities 2^{48} and 2^{44} of [12, 14], the strong property expressed by relation $\mathcal{R}$ that each input and output byte is not only balanced as in zero-sum distinguishers, but uniformly distributed turns out to be convenient for further extending the known key distinguisher by two rounds in a provable manner, as will be shown in the rest of this section.

Strengthening Proposition 4 under a heuristic assumption. Let us give some partial evidence that $\mathcal{R}$ is actually T-intractable in a stronger sense than in Proposition 4 above, namely that the success probability of any adversary $\mathcal{A}$ who makes $M > N$ oracle queries to Π or Π^{-1} remains negligible if $M - N$ is not too large. While a rigorous proof requiring no unproven assumptions could be easily derived along the same lines as Proposition 4 for values of M marginally larger than N, *e.g.* $N + 3$, for larger values of M we make the heuristic assumption that querying both Π and Π^{-1} does not improve the performance of $\mathcal{A}$ over an adversary who only queries one of these oracles. Therefore, we consider an adversary $\mathcal{A}$ who only makes queries to an oracle permutation Π not its inverse, and aims at finding an N-tuple of (input, output) pairs that satisfy the relation

[11] This can for instance be deduced from the fact that the X_i and the Y_i must satisfy $\bigoplus_{i=1}^{N} X_i = \bigoplus_{i=1}^{N} Y_i = 0$.

$\mathcal{R}$ of Section 4.1. To upper bound the success probability of such an adversary, we observe that given any N-tuple of distinct input blocks X_i and any output byte position $j \in [0;15]$, the 256-tuple $(N_0, \cdots, N_{255})$ of numbers of occurrences of the values 0, 1, $\cdots$ 255 for byte j of the blocks $Y_i = \Pi(X_i)$ is nearly governed by a multinomial law. For any 256-tuple $(N_0, \cdots, N_{255})$ such that $\sum_{i=0}^{255} N_i = N$, we denote the multinomial coefficient $\frac{N!}{N_0!N_1!\cdots N_{255}!}$ by $\binom{N}{N_0,\cdots N_{255}}$.

Proposition 5. *For any N-tuple $(X_i)_{i=1\cdots N}$ of distinct inputs to Π an upper bound on the probability p that for byte positions $j = 0$ to 15, the 256-tuple of numbers of occurrence of the values of byte j of $\Pi(X_i)$ be $(N_0^j, \cdots, N_{255}^j)$ — where for j=0 to 15 the 256-tuple $(N_0^j, \cdots, N_{255}^j)$ satisfies $\sum_0^{255} N_i^j = N$ — is given by:*

$$p \leq \prod_{j=0}^{15} \binom{N}{N_0^j, \cdots N_{255}^j} \times (\frac{1}{2^{128} - N + 1})^N.$$

An upper bound on the success probability p_A of an adversary $\mathcal{A}$ who makes $M > N$ queries to Π and no query to Π^{-1} is given by:

$$p_A \leq \binom{M}{N} \times \binom{N}{\frac{N}{256}, \frac{N}{256} \cdots \frac{N}{256}}^{16} \times (\frac{1}{2^{128} - N + 1})^N.$$

Since $N = 2^{64}$, Proposition 5 provides very small upper bounds $p_A \ll \frac{1}{2}$ for values of M of up to $M \approx N + 2^{11}$. But it provides no bound $p_A < \frac{1}{2}$ for slightly larger values, *e.g.* $M \approx N + 2^{12}$. We do not know whether the bounds of Proposition 5, that relate to the probability that the (input, output) pairs provided by M queries contain one N-tuple, can be significantly improved. Since even in a situation where such N-tuples exist it can be computationally difficult to find one in time T, a potential approach might consist in establishing upper bounds that hold for higer values of M under computational assumptions.

4.2 A known-key distinguisher for the 10-round AES

In this section we show that the former known-key distinguisher for AES_8 can be extended by two rounds without significant complexity increase. The price to pay for this extension is that the relation $\mathcal{R}$ of the new distinguisher is much less simple and that its description involves operations of the first and last rounds. This raises the question whether the new known-key distinguisher reflects a *meaningful* correlation property of the cipher. Since we can provide more simple arguments supporting this view for AES_{10}^* (*i.e.* the 10-round AES parametrized by 11 independent subkeys), we first describe the application of the new known-key distinguisher to AES_{10}^* and then discuss how this transposes to AES-128.

As shown at the end of Section 2, AES_{10}^* can be equivalently represented by the sequence of transformations

$$AK \cdot P^{-1} \cdot SB \cdot R \cdot (S \cdot R)^4 \cdot SB \cdot Q^{-1} \cdot AK$$

The properties we are using to build a known-key distinguisher on AES_{10}^* are illustrated on Figure 5.

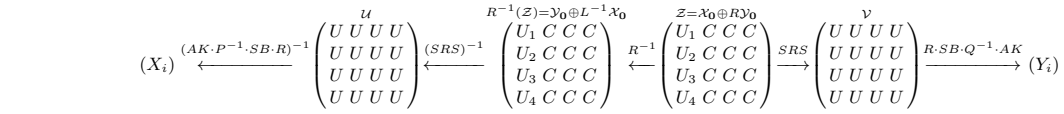

Fig. 5. Derivation of the N AES_{10} (input,output) pairs used in our known-key distinguisher

Algorithm $\mathcal{A}$: We reuse the same structure $\mathcal{Z}$ of $N = 2^{64}$ intermediate blocks as for the known-key distinguisher on AES_8 presented above, but extend the forward computation and backward computations $S \cdot R \cdot S$ and $(S \cdot R \cdot S \cdot R)^{-1}$, by two outer transformations whose structures are symmetric of each other, namely $(AK \cdot P^{-1} \cdot SB \cdot R)^{-1}$ (backward) and $R \cdot SB \cdot Q^{-1} \cdot AK$ (forward) to get an N-tuple of related AES_{10}^* inputs and outputs. As shown in the former subsection the inputs to the forward and backward outer transformations each consist of four columns that are uniformly distributed and therefore each of the 16 bytes of each of these two states $\mathcal{U}$ and $\mathcal{V}$ is uniformly distributed and can be marked U. However, these states are related to the AES_{10}^* inputs X_i and to the AES_{10}^* outputs Y_i by the outer transformations.

This implies that if we denote by α and β the 128-bit states $P^{-1}(K_0)$ and $Q(K_{10})$ the N-tuple $\mathcal{X} = (X_i)_{i=1\cdots N}$ and $\mathcal{Y} = (Y_i)_{i=1\cdots N}$ are related by the key-dependent relation $\mathcal{R}_{\alpha,\beta}$ defined as follows: $\mathcal{X}\mathcal{R}_{\alpha,\beta}\mathcal{Y}$ if and only if each byte of $R \circ SB(P^{-1}(X_i) \oplus \alpha)$ and each byte of $R^{-1} \circ SB^{-1}(Q(Y_i) \oplus \beta)$ is uniformly distributed. We can now define the following relation $\mathcal{R}$ over the N-tuples of blocks:

Relation $\mathcal{R}$: *Given two N-tuples $\mathcal{X}' = (X'_i)_{i=1\cdots N}$ and $\mathcal{Y}' = (Y'_i)_{i=1\cdots N}$ $\mathcal{X}'\mathcal{R}\mathcal{Y}'$ if and only if all the X'_i, $i = 1 \cdots N$ are pairwise distinct and there exists a pair α', β' of 128-bit states such that $\mathcal{X}'\mathcal{R}_{\alpha',\beta'}\mathcal{Y}'$.*

It is important to understand that though relation $\mathcal{R}$ reflects the existence of values α' and β' that can be conveniently interpreted as subkeys, ckecking $\mathcal{R}$ does not take as input any key or subkey: given two N-tuples $\mathcal{X}'$ and $\mathcal{Y}'$ that can be possibly derived from a random key value K by algorithm $\mathcal{A}$, whether $\mathcal{X}'\mathcal{R}\mathcal{Y}'$ must be efficiently checkable without providing the verifyer with K or any other side information about suitable values of α' and β'.

It immediately results from the definition of $\mathcal{R}$ that the N-tuples $\mathcal{X}$ and $\mathcal{Y}$ derived as described in Figure 5 satisfy property $\mathcal{R}$ and the complexity of the derivation algorithm $\mathcal{A}$ is $T = N = 2^{64}$. To complete the proof that $(\mathcal{R}, \mathcal{A})$ is a known-key distinguisher for AES_{10}^*, we just have to show that $\mathcal{R}$ is efficiently checkable and T-intractable.

$\mathcal{R}$ is efficiently checkable. Though the involvement in $\mathcal{R}$ of 128-bit constants α' and β' might suggest that checking $\mathcal{R}$ has a huge complexity, this is not the case because the existence of 128-bit states α', β' such that $\mathcal{X}'\mathcal{R}_{\alpha',\beta'}\mathcal{Y}'$ can be split into independent conditions. Let us denote by $sb : \{0,1\}^{32} \to \{0,1\}^{32}$ a parallel application of four AES S-boxes that from a four-byte row produces a four-byte output row. For j= 0 to 3 let us denote by row_j the mapping that

from a 128-bit state outputs the row numbered j of this state, and by R_j the linear transformation of row j introduced in Section 2. It is easy to see that the existence of α' and β' is equivalent to the existence of eight 32-bit constants α'_j, $j = 0 \cdots 3$ and $\beta'_j, j = 0 \cdots 3$ (representing the rows of α' and β') such that for $j = 0 \cdots 3$ each of the four bytes of $R_j \circ sb \circ row_j(P^{-1}(X_i) \oplus \alpha'_j)$ and $R_j^{-1} \circ sb^{-1} \circ row_j(Q(Y_i) \oplus \alpha'_j)$ is uniformly distributed. This can be easily done by first computing in a first step the number of occurrences of each of the 2^{32} possible values of the 32-bit words $row_j(P^{-1}(X_i))$ and $row_j(Q(Y_i))$, $j = 0 \cdots 3$, and then using the obtained distributions of frequencies in a second step for computing, for $j = 0$ to 3 and each of the 2^{32} possible values of α'_j, resp. β'_j the resulting distribution of frequencies of $R_j \circ sb \circ row_j(P^{-1}(X_i) \oplus \alpha'_j)$, resp $R_j^{-1} \circ sb^{-1} \circ row_j(Q(Y_i) \oplus \alpha'_j)$ and checking that at least one of them induces a balanced distribution for each byte position. Since the first step requires 2^{64} very simple operations that are much less complex that one operation of AES^*_{10} and the second step again requires 8 times 2^{64} very simple operations, the overall complexity of checking $\mathcal{R}$ is strictly smaller than $N = 2^{64}$ AES^*_{10} operations.

Remark. The reader might wonder whether the technique we used to derive a known-key distinsguisher for the 10-round AES from a known-key distinguisher for the 8-round AES, by expressing that the 10-round inputs and outputs are related (by one outer round at each side) to intermediate blocks that satisfy the relation used by the 8-round distinguisher does not allow to extend this 8-round known distinguisher by an arbitrary number of rounds. If this was the case, this would of course render this technique highly suspicious. It is easy however to see that the argument showing that 10-round relation $\mathcal{R}$ is efficiently checkable does not transpose for showing that the relations over $r > 10$ rounds one could derive from the 8-round relation by expressing that the r-round inputs and outputs are related by $r-8 > 2$ outer rounds to intermediate blocks that satisfy the 8-round relation are efficiently checkable. To complete this remark, we explain at the end of this section why the 2-round extension technique we used is not generically applicable to extend any r-round known-key distinguisher to a $r+2$-round distinguisher.

$\mathcal{R}$ is T-intractable. In order to show that relation $\mathcal{R}$ is T-intractable, we now have to prove that the success probability of any oracle algorithm of overall time complexity upper bounded by $N = 2^{64}$ (and therefore of number q of queries also upper bounded by N) is negligible.

Proposition 6. *For any oracle algorithm $\mathcal{A}$ that makes $q \leq N = 2^{64}$ oracle queries to a perfect random permutation Π of $\{0,1\}^{128}$ or Π^{-1}, the probability that $\mathcal{A}$ outputs a N-tuple $(X_i, Y_i)_{i=1\cdots N}$ of Π that satisfies and $\forall i \in [1; N]$ $Y_i = \Pi(X_i)$ and also satisfies $\mathcal{R}$ is upper bounded by $2^{256} \times (\frac{5^{16}}{2^{128}-(N-5)})^3 \approx 2^{-16.5}$.*

Proof. If at least one of the N pairs (X_i, Y_i) output by $\mathcal{A}$ does not result from a query X_i to Π or a query Y_i to Π^{-1}, then the probability that for this pair $Y_i = \Pi(X_i)$ and consequently the success probability of $\mathcal{A}$ is upper bounded by

$\frac{1}{2^n-(N-1)}$. So from now on we only consider the opposite case, *i.e.* $q = N$ and all the (X_i, Y_i) result from queries to Π or Π^{-1}. Given any two 128-bit words α and β, let us upper bound the probability that $\mathcal{A}$ outputs an N-tuple (X_i, Y_i) that satisfies $\forall i \in [1; N]\ Y_i = \Pi(X_i)$ and the relation $\mathcal{R}_{\alpha,\beta}$. The conducting idea is that the constraints on the very last queries to the oracle (Π, Π^{-1}) in order for $\mathcal{R}_{\alpha,\beta}$ to hold are so strong this is extremely unlikely to happen. For the sake of simplicity of this proof, we consider the consider the last 5 queries of $\mathcal{A}$ to the oracle (Π, Π^{-1}): indeed, while considering the d last queries, $d > 5$, might have lead to a tighter upper bound, the chosen value of 5 is sufficient for establishing a suitable upper bound. Since the 5 last queries contain at least 3 queries to either Π or Π^{-1} we can assume *w.l.o.g.* that they contain at least 3 queries X, X', and X'' to Π and we denote the corresponding responses by Y, Y', and Y''. In order for the property $\mathcal{R}_{\alpha,\beta}$ to be satisfied, for each byte position $j \in [0; 15]$, the set of byte values $B_j = \{b \in [0; 255] \mid \sharp\{i \in [1; N-5] \mid R^{-1} \circ SB^{-1}(Q(Y_i) \oplus \beta)[j] = b\} \neq \frac{N}{256}\}$ must contain at most 5 elements (since the last 5 queries can affect the number of occurrences of at most 5 of the 256 byte values and all the unaffected numbers of occurrences must already be $\frac{N}{256}$). Furthermore, in order for property $\mathcal{R}_{\alpha,\beta}$ to be satisfied, one must have $\forall i \in [N-4; N]\ R^{-1} \circ SB^{-1}(Q(Y_i) \oplus \beta)[j] \in B_j$, *i.e.* $\forall i \in [N-4; N]\ Y_i \in \mathcal{S} = Q^{-1} \circ SB \circ R(\prod_{j=0}^{15} B_j) \oplus \beta$. Since Q, SB, R, and the xor with β are bijective, the set $\mathcal{S}$ defined above contains $\sharp\mathcal{S} = \sharp \prod_{j=0}^{15} B_j$ elements (where $\prod_{j=0}^{15} B_j$ denotes the Cartesian product of the B_j). Since for j=0 to 15 $\sharp B_j \leq 5$, $\sharp \prod_{j=0}^{15} B'_j \leq 5^{16}$ and hence $\sharp\mathcal{S} \leq 5^{16}$. Therefore the probability that the three blocks Y, Y', and Y'' all belong to $\mathcal{S}$ is upper bounded by $(\frac{5^{16}}{2^{128}-(N-5)})^3$. By summing the obtained upper bound over all the 2^{256} possible values of α, β, one gets the claimed upper bound $2^{256} \times (\frac{5^{16}}{2^{128}-(N-5)})^3 \approx 2^{-16.5}$ on the probability that $\mathcal{R}$ be satisfied. □

In order to give partial evidence that $\mathcal{R}$ is not only N-intractable as shown in Proposition 6 above, but remains M-intractable for $M > N$ if $M - N$ is not too large, we can make the heuristic assumption that the success probabilities of adversaries who are allowed to make oracle queries to both Π and Π^{-1} and adversaries who are allowed to make oracle queries to Π only have the same upper limit. Proposition 5 can be transposed to the 10-round relation $\mathcal{R}$, up to a multiplication of the upper bounds obtained for p and p_A by 2^{256}. This multiplicative factor does not strongly affect the values of $M-N$ one can reach and one still gets very small upper bounds $p_A \ll \frac{1}{2}$ for values of M of up to $M \approx N+2^{11}$.

The former 2-round extension technique is not generic. The reader might wonder why the two-round extension technique introduced above does not allow to extend any r-round known-key distinguisher to an $r + 2$-round known-key distinguisher. There are two reasons that can make such an extension fail: firstly, unlike the r-round relation it is derived from, the $r + 2$-round relation may not be efficiently ckeckable; secondly, unlike the r-round relation it is derived from, the $r+2$-round relation may be insufficiently intractable to mount a $r+2$-round distinguisher. This second situation occurs in the case of the 8-round differential

relation $\mathcal{R}_8$ of order 2 used in [12]. In the full version of this paper we show that unlike $\mathcal{R}_8$, that is T-intractable for $T = 2^{48}$, the 10-round relation $\mathcal{R}_{10}$ derived from $\mathcal{R}_8$ is not intractable at all for $T = 2^{48}$, but simple to achieve with a probabillity about 0.97 with only two queries to a perfect random permutation Π and no extra operation. In other words, the transposition of our technique to the 8-round distinguisher of [12] does not allow to derive a valid 10-round distinguisher.

In the full version of this paper, we also show that while we do not preclude that the use of the stronger property (reflected by a higher-order relation than $\mathcal{R}_8$) that several pairs satisfying the differential relation of [12] can be derived might potentially result in a 10-round distinguisher that outperforms the 10-round distinguisher presented above, giving a rigorous proof (as was done in Proposition 6) seems technically difficult. We leave the investigation of improved 10-round known-key distinguishers and associated proofs – or even plausible heuristic arguments if rigorous proofs turn out to be too difficult to obtain – as an open issue.

Discussion. The known-key distinguisher $(\mathcal{R}, \mathcal{A})$ of order $N = 2^{64}$ for AES^*_{10} presented above has a time complexity of about 2^{64}. Unlike in the former 8-round known-key distinguishers the relation $\mathcal{R}$ involves operations of the AES. However, it is easy to show that the alternative criterion at the end of Section 3 for differentiating certain known-key distinguishers from the artificial known-key distinguishers that result from generic impossibility results is applicable. Indeed, the derivation by $\mathcal{A}$ of the input N-tuple $(X_i)_{i=1\cdots N}$ from the intermediate structure $\mathcal{Z}$ only involves the 6 first subkeys K_0 to K_5 and the derivation $\mathcal{A}$ of the output N-tuple $(Y_i)_{i=1\cdots N}$ from the same structure only involves the 5 last subkeys K_6 to K_{11}. Consequently the 5 last subkeys cannot be derived from $(X_i)_{i=1\cdots N}$ and thus the input N-tuples do not "encode" the entire key. Similarly, the 6 first subkeys cannot be derived from $(Y_i)_{i=1\cdots N}$ and thus the output N-tuples do not "encode" the entire key. This suggests that the obtained known-key distinguisher for AES^*_{10} can reasonably be considered meaningful.

While the former known-key distinguisher is obviously applicable without any modification to AES_{10}, *i.e.* the full AES-128, the former argument vanishes in this case because all subkeys are related by the key schedule: the first subkey, resp. the last subkey can actually be derived from the input, resp. the output N-tuple and because of the key schedule relations this determines the entire key. This does not mean that when applied to AES_{10} the former distinguisher becomes artificial. Actually, the fact that the very same distinguisher is applicable to AES^*_{10} gives a hint that it can still be considered meaningful.[12]

[12] Since the input N-tuple now encodes the entire key, there might exist *artificial* variants of the former known-key distinguisher that produce the same input N-tuples (or the same output N-tuples) but can be extended to AES_r for any value of r. We conjecture however that unlike the known-key distinguisher presented here, such variants would not be applicable to AES^*_r.

5 Conclusion

As said before, the untwisted representation of AES introduced in this paper is not exclusively intended for the analysis of the security of AES in the known-key model. We think however that the fact that this represention was used to find the two known-key distinguishers presented in Section 4 provides some evidence that this representation is well suited for analysing the resistance of (a reduced-round version of) AES against some structural attacks.

Whether there exists a more simple 10-round known-key or even chosen-key distinguisher for AES than the 10-round known key distinguisher presented in this paper – allowing to highlight a less tenuous deviation from the behaviour of a perfect random permutation, resp. of an ideal cipher remains an interesting open question.

Acknowledgements. We would like to thank Yannick Seurin for helpful discussions and insights.

References

1. Jean-Philippe Aumasson and Willi Meier. Zero-sum distinguishers for reduced Keccak-f and for the core functions of Luffa and Hamsi, 2009. Comment on the NIST SHA-3 Hash Competition.
2. Elad Barkan and Eli Biham. In How Many Ways Can You Write Rijndael ? In *Advances in Cryptology - ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 160–175. Springer, 2002.
3. Alex Biryukov, Christophe De Cannière, An Braeken, and Bart Preneel. A Toolbox for Cryptanalysis: Linear and Affine Equivalence Algorithms. In *Advances in Cryptology - EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 33–50. Springer, 2003.
4. Christina Boura and Anne Canteaut. Zero-Sum Distinguishers for Iterated Permutations and Application to Keccak-f and Hamsi-256. In *Selected Areas in Cryptography – SAC 2010*, volume 6544 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2011.
5. Christina Boura, Anne Canteaut, and Christophe De Cannière. Higher-Order Differential Properties of Keccak and *Luffa*. In *Fast Software Encryption - FSE 2011*, volume 6733 of *Lecture Notes in Computer Science*, pages 252–269. Springer, 2011.
6. Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.
7. Joan Daemen and Vincent Rijmen. Understanding Two-Round Differentials in AES. In *Security and Cryptography for Networks - SCN 2006*, volume 4116 of *Lecture Notes in Computer Science*, pages 78–94. Springer, 2006.
8. Donald W. Davies and Sean Murphy. Pairs and Triplets of DES S-Boxes. *Journal of Cryptology*, 8(1):1–25, 1995.
9. Niels Ferguson, Richard Schroeppel, and Doug Whiting. A Simple Algebraic Representation of Rijndael. In *Selected Areas in Cryptography - SAC 2001*, volume 2259 of *Lecture Notes in Computer Science*, pages 103–111. Springer, 2001.
10. Pierre-Alain Fouque, Jérémy Jean, and Thomas Peyrin. Structural Evaluation of AES and Chosen-Key Distinguisher of 9-Round AES-128. In *Advances in Cryptology - CRYPTO 2013*, volume 8042 of *Lecture Notes in Computer Science*, pages 183–203, 2013.

11. Henri Gilbert and Thomas Peyrin. Super-Sbox Cryptanalysis: Improved Attacks for AES-like permutations. *IACR Cryptology ePrint Archive*, 2009:531, 2009.
12. Henri Gilbert and Thomas Peyrin. Super-Sbox Cryptanalysis: Improved Attacks for AES-Like Permutations. In *Fast Software Encryption - FSE 2010*, volume 6147 of *Lecture Notes in Computer Science*, pages 365–383. Springer, 2010.
13. Mitsugu Iwamoto, Thomas Peyrin, and Yu Sasaki. Limited-Birthday Distinguishers for Hash Functions - Collisions beyond the Birthday Bound. In *Advances in Cryptology - ASIACRYPT 2013*, volume 8270 of *Lecture Notes in Computer Science*, pages 504–523. Springer, 2013.
14. Jérémy Jean, María Naya-Plasencia, and Thomas Peyrin. Multiple Limited-Birthday Distinguishers and Applications.
15. Jérémy Jean, María Naya-Plasencia, and Thomas Peyrin. Improved Rebound Attack on the Finalist Grøstl. 7549:110–126, 2012.
16. Jérémy Jean, María Naya-Plasencia, and Martin Schläffer. Improved Analysis of ECHO-256. In *Selected Areas in Cryptography - SAC 2011*, volume 7118 of *Lecture Notes in Computer Science*, pages 19–36. Springer, 2012.
17. Lars R. Knudsen and Vincent Rijmen. Known-Key Distinguishers for Some Block Ciphers. In *Advances in Cryptology - ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 315–324. Springer, 2007.
18. Mario Lamberger, Florian Mendel, Christian Rechberger, Vincent Rijmen, and Martin Schläffer. Rebound Distinguishers: Results on the Full Whirlpool Compression Function. In *Advances in Cryptology - ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 126–143. Springer, 2009.
19. Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen. Rebound Attacks on the Reduced Grøstl Hash Function. In *CT-RSA 2010*, volume 5985 of *Lecture Notes in Computer Science*, pages 350–365. Springer, 2010.
20. Sean Murphy and Matthew J. B. Robshaw. Essential Algebraic Structure within the AES. In *Advances in Cryptology - CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2002.