IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2018, No. 2, pp. 133–160. DOI:10.13154/tosc.v2018.i2.133-160

Mixture Differential Cryptanalysis: a New Approach to Distinguishers and Attacks on round-reduced AES

Lorenzo Grassi

Institute of Applied Information Processing and Communications (IAIK), Graz University of Technology, Graz, Austria
lorenzo.grassi@iaik.tugraz.at

Abstract. At Eurocrypt 2017 the first secret-key distinguisher for 5-round AES - based on the "multiple-of-8" property - has been presented. Although it allows to distinguish a random permutation from an AES-like one, it seems rather hard to implement a key-recovery attack different than brute-force like using such a distinguisher.
In this paper we introduce "*Mixture Differential Cryptanalysis*" on round-reduced AES-like ciphers, a way to translate the (complex) "multiple-of-8" 5-round distinguisher into a simpler and more convenient one (though, on a smaller number of rounds). Given a pair of chosen plaintexts, the idea is to construct new pairs of plaintexts by *mixing* the generating variables of the original pair of plaintexts. Here we theoretically prove that for 4-round AES the corresponding ciphertexts of the original pair of plaintexts lie in a particular subspace if and only if the corresponding pairs of ciphertexts of the new pairs of plaintexts have the same property. Such secret-key distinguisher - which is independent of the secret-key, of the details of the S-Box and of the MixColumns matrix (except for the branch number equal to 5) - can be used as starting point to set up new key-recovery attacks on round-reduced AES. Besides a theoretical explanation, we also provide a practical verification both of the distinguisher and of the attack.

Keywords: AES · Secret-Key Distinguisher · Key-Recovery Attack · Mixture Differential Cryptanalysis · Truncated Differential · Subspace Trail Cryptanalysis

1 Introduction

Block ciphers are certainly among the most important cryptographic primitives. They are designed by iterating an efficiently implementable round function many times in the hope that the resulting composition behaves like a randomly drawn permutation. In the compromise, a round function is iterated enough times to make sure that any symmetries and structural properties that might exist in the round function vanish.

One of the most important tools that a cryptanalyst has at hand when trying to evaluate the security of ciphers or hash functions is - without doubt - differential cryptanalysis. Since its conception by Biham and Shamir [BS90, BS91] in their effort to break the Data Encryption Standard (DES), it has been successfully applied in many cases such that any modern cipher is expected to have strong security arguments against this attack.

The methodology of differential cryptanalysis has been extended several times with a number of attack vectors, most importantly truncated differentials [Knu95], impossible differentials [Knu98, BBS99], higher-order differentials [Knu95], boomerang attacks [Wag99] and differential-linear attacks [LH94].

Received: 2018-03-01, Accepted: 2018-05-01, Published: 2018-06-07

With today's knowledge, designing a secure block cipher is a problem that is largely considered solved. Especially with the AES we have at hand a very well analyzed and studied cipher that, after more than 20 years of investigation still withstands all cryptanalytic attacks. However, new results on the AES still appear regularly, especially within the last couple of years (e.g. polytopic cryptanalysis [Tie16], "multiple-of-8" distinguisher [GRR17a] and yoyo distinguisher [RBH17]). While those papers do not pose any practical thread to the AES, they do give new insights into the internals of what is arguably the cipher that is responsible for the largest fraction of encrypted data worldwide.

"Multiple-of-8" distinguisher [GRR17a] proposed at Eurocrypt 2017 by Grassi, Rechberger and Rønjom is the first 5-round secret-key distinguisher for AES that exploits a property which is independent of the secret key and of the details of the S-Box. This distinguisher is based on a new structural property for up to 5 rounds of AES: by appropriate choices of a number of input pairs it is possible to make sure that the number of times that the difference of the resulting output pairs lie in a particular subspace is *always* a multiple of 8. This distinguisher allows to distinguish an AES permutation from a random one with a success probability greater than 99% using 2^{32} chosen texts and a computational cost of $2^{35.6}$ look-ups. On the other hand, as this distinguisher is based on a property of the whole state in the output of AES, it makes it challenging to convert it into a key-recovery attack over more rounds, since e.g. it requires guessing the whole subkey in the last round.

In this paper we introduce "*mixture differential cryptanalysis*" on round-reduced AES-like ciphers, a way to translate the (complex) "multiple-of-8" 5-round distinguisher [GRR17a] into a simpler and more convenient one (though, on a smaller number of rounds). As we are going to show, such new proposed technique leads to a new distinguisher and key-recovery attacks on 4- and 5-round AES (respectively) with data and computational complexity similar than other attacks in literature.

Such distinguisher and attack - fully practically verified - are also general enough to be applied to any AES-like cipher, and they might be valuable as a reference framework. In particular, many constructions employ reduced round AES as part of their design (e.g. among many others, AEGIS [WP] - one of the finalist of the on-going CAESAR competition [CAE] - uses five AES round-functions in the state update functions). Reduced versions of AES have nice and well-studied properties that can be favorably as components of larger designs (see for instance Simpira [GM16]). As a result, distinguishers and attacks on 4-/5-round AES can be also useful in analyzing those primitives. To give a concrete example, in [BEK16] authors exploit - in a new way - known properties of round-reduced AES to set up a new attack on ELmD [DN], another finalist of the on-going CAESAR competition.

Related Work

To the best of our knowledge, the concept of mixture differential cryptanalysis is new and has not been used in cryptanalysis before. Nonetheless there are other works that share some similarities with mixture differential cryptanalysis.

Before going on, as first thing we recall the notion of secret-key distinguisher, one of the weakest attacks that can be launched against a secret-key cipher. In this attack, there are two oracles: one that simulates the cipher for which the cryptographic key has been chosen at random and one that simulates a truly random permutation. The adversary can query both oracles and her task is to decide which oracle is the cipher and which is the random permutation. The attack is considered to be successful if the number of queries required to make a correct decision is below a well defined level.

Differential Attacks. Differential attacks [BS90] exploit the fact that couples of plaintexts with certain differences yield other differences in the corresponding ciphertexts with a non-uniform probability distribution. The resulting pair of differences is called a

Table 1: *Secret-Key Distinguishers for 4-round AES.* The complexity is measured in minimum number of chosen plaintexts/ciphertexts (CP/CC) or/and adaptive chosen plaintexts/ciphertexts (ACP/ACC) which are needed to distinguish the AES permutation from a random one with probability higher than 95% (all distinguishers work both in the encryption and in the decryption mode). Time complexity is measured in equivalent encryptions (E), memory accesses (M) or XOR operations (XOR) - using the common approximation 20 M $\approx$ 1 Round of Encryption. The distinguisher of this paper is in bold.

Property	**Data**	**Cost**	**Ref.**
Yoyo Game	2 CP + 2 ACC	2 XOR	[RBH17]
Impossible Differential	$2^{16.25}$ CP	$2^{22.3}$ M $\approx 2^{16}$ E	[BK01]
Mixture Diff.	$\mathbf{2^{17}}$ **CP**	$\mathbf{2^{23.1}}$ **M** $\approx \mathbf{2^{16.75}}$ **E**	**Sect. 4**
Integral	2^{32} CP	2^{32} XOR	[DKR97]
Multiple-of-8	2^{33} CP	2^{40} M $\approx 2^{33.7}$ E	[GRR17a]

differential. Such a property can be used both to distinguish a cipher permutation from a random one, and to recover the secret key. Possible variants of this attack/distinguisher are the truncated differential attack [Knu95], in which the attacker considers only part of the difference between pairs of texts (i.e. a differential attack where only part of the difference in the ciphertexts can be predicted), and impossible differential attack [Knu98, BBS99], in which the attacker considers differential with zero-probability.

In the original version of differential cryptanalysis [BS90], a unique differential is exploited. A generalization of such attack is multiple differential cryptanalysis [BG11], where several input differences are considered together and the corresponding output differences can be different from an input difference to another, that is the set of considered differentials has no particular structure.

The common feature of all these distinguishers/attacks is the fact that - in all these cases - the attacker focuses on the probability that a single pair of plaintexts with a certain input difference yield other difference in the corresponding pair of ciphertexts, working *independently* on each pair of texts.

Recent Results. Recently, new differential distinguishers have been proposed in the literature, precisely the polytopic cryptanalysis [Tie16] at Eurocrypt 2016 and the yoyo distinguisher on SPN constructions [RBH17] at Asiacrypt 2017, which present an important difference with respect to the previously recalled attacks. Instead of working on each couple[1] of two (plaintext, ciphertext) pairs independently of the others as in the previous scenario, in these cases the attacker works on the relations that hold among the couples of pairs of texts. In other words, *given a couple of two (plaintext, ciphertext) pairs with a certain input/output differences, one focuses and studies how such couple influences other couples of two (plaintext, ciphertext) pairs to satisfy particular input/output differences.*

More precisely, polytopic cryptanalysis is similar to multiple differential cryptanalysis. However, as opposed to assuming independence of the differentials (which does not hold in general, as showed in [Mur11]), the authors explicitly take their correlation into account and use it in their framework, considering interdependencies between larger sets of texts and as they traverse through the cipher.

The strategy exploited by the yoyo game on SPN constructions proposed at Asiacrypt 2017 is similar to the one that we are going to exploit to set up our new distinguisher. Given a pair of chosen plaintexts and the corresponding ciphertexts, the attacker constructs new pair of ciphertexts related to the other ones by linear and differential relations. Authors

[1] *Notation:* we use the term "*pair*" to denote a plaintext and its corresponding ciphertext. A "*couple*" denotes a set of two such pairs.

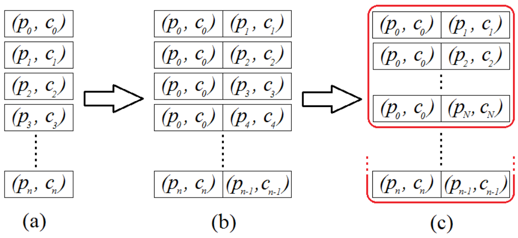

Figure 1: *New Differential Secret-Key Distinguishers for round-reduced AES.* Consider N (plaintext, ciphertext) pairs (a). In a “classical" differential attack (b), one works independently on each couple of two (plaintext, ciphertext) pairs and exploits the probability that it satisfies a certain differential trail. In our attack (c), one divides the couples into non-random sets, and exploits particular relationships (based on differential trails) that hold among the couples that belong to the same set in order to set up a distinguisher.

prove that the corresponding new pair of plaintexts of this new second pair of ciphertexts satisfies - with prob. 1 - a difference related “in some sense” to the input difference of the original pair of plaintexts, independently of the secret-key. This allows to distinguish e.g. round-reduced AES from a random permutation, or to set up key-recovery attack.

Our Contribution

In this paper, we present “mixture differential cryptanalysis” on 4-round AES. This 4-round secret-key distinguisher - proposed in Sect. 4 - is similar in nature to polytopic cryptanalysis and the yoyo distinguishers just recalled.

Given plaintexts in the same coset of a subspace $\mathcal{C}$, the attacker first divides the couples of two (plaintext, ciphertext) pairs into sets of $N \geq 2$ *non-independent* couples. These sets are defined such that particular relationships (that involve differential and linear relationships) hold among the plaintexts of the couples that belong to the same set. Due to the particular way - explained in detail the following - in which these sets are defined, we call our new technique as *Mixture Differential Cryptanalysis*. As already pointed out, the way in which these sets are constructed resemble the “multiple-of-8” distinguisher [GRR17a] recently proposed at Eurocrypt 2017.

Such sets have the property that the two ciphertexts of a certain couple belong to the same coset[2] of a particular subspace $\mathcal{M}$ if and only if the two ciphertexts of all the other couples in that set have the same property. In other words, it is not possible that two ciphertexts of some couples belong to the same coset of $\mathcal{M}$, and that two ciphertexts of other couples don’t have this property. Since this last event can occur for a random permutation, it is possible to distinguish 4-round AES from a random permutation.

In more detail and referring to Fig. 1, given n chosen (plaintext, ciphertext) pairs, in a “classical” (differential) attack one works on each couple of two (plaintext, ciphertext) pairs independently of the others - case (b). In our distinguishers/attacks instead, one first divides the couples in (non-random) sets of $N \geq 2$ couples - case (c), and then she works on each set of couples independently of the other sets, exploiting the property just given.

We remark that our new mixture differential distinguisher *is independent of the secret key (and of the key-schedule), of the details of the S-Box and of the MixColumns matrix.* Such distinguisher works both in the encryption and in the decryption process, and it is general enough to be applied to any AES-like cipher. Compared to the yoyo distinguisher proposed at Asiacrypt 2017 that requires adaptive chosen plaintexts/ciphertexts, ours

[2] *A pair of texts has a certain difference if and only if the texts belong to the same coset of a particular subspace $\mathcal{X}$.*

Table 2: *Comparison of Attacks on 5-round AES-128.* Data complexity is measured in number of required chosen plaintexts/ciphertexts (CP/CC) and/or adaptive chosen plaintexts/ciphertexts (ACP/ACC). Time complexity is measured in round-reduced AES encryption equivalents (E) - the number in the brackets denotes the precomputation cost (if not negligible). Memory complexity is measured in texts (16 bytes). Attack presented in this paper is in bold.

Attack	Data	Computation	Memory	Ref.
MitM	8 CP	2^{64}	2^{56}	[Der13, Sec. 7.5.1]
Imp. Polytopic	15 CP	2^{70}	2^{41}	[Tie16]
Partial Sum	2^{8} CP	2^{38}	small	[Tun12]
Integral (EE)	2^{11} CP	$2^{45.7}$	small	[DR02]
Yoyo Game	$2^{11.3}$ ACC	2^{31}	small	[RBH17]
Imp. Differential	$2^{31.5}$ CP	2^{33} $(+2^{38})$	2^{38}	[BK01]
Integral (EE)	2^{32} CP	$2^{25.4}$	2^{5}	App. C
Integral (EB)	2^{33} CP	$2^{37.7}$	2^{32}	[DR02]
Mixture Diff.	**$2^{33.6}$ CP**	**$2^{33.3}$**	**2^{34}**	**Sect. 5**

MitM: Meet-in-the-Middle, EE: Extension at End, EB: Extension at Beginning

requires only chosen plaintexts. A complete and detailed comparison among our new proposed distinguisher and the other ones present in the literature is proposed in Sect. 4.3.

Furthermore, we highlight that our 4-round distinguisher might be used as starting point in order to set up new 5-round secret-key distinguishers on AES, as suggested in detail in [Gra17].

A New Key-Recovery Attack on 5-round AES-128. Finally, we show that mixture differential cryptanalysis is not only theoretically intriguing, but indeed relevant for practical cryptanalysis. In particular, in Sect. 5 we propose an attack on 5-round AES that exploits the distinguisher on 4 rounds proposed in Sect. 4. Such attack has then been improved in [BODK+18], becoming the one with the *lowest computational cost* among the attacks currently present in the literature (that don't use adaptive chosen plaintexts/ciphertexts). In this attack, the attacker chooses plaintexts in the same coset of a particular subspace $\mathcal{D}$ which is mapped after one round into a coset of another subspace $\mathcal{C}$. Using the mixture differential distinguisher just introduced and the facts that

- the way in which the couples of two (plaintext, ciphertext) pairs are divided in sets depends on the (partially) guessed key
- the behavior of a set for a wrongly guessed key is (approximately) the same as the case of a random permutation,

she can filter wrong candidates of the key, and finally finds the right one.

2 Preliminary

2.1 Description of AES

The Advanced Encryption Standard [DR02] is a *Substitution-Permutation network* that supports key sizes of 128, 192 and 256 bits. The 128-bit plaintext initializes the internal state represented by a 4×4 matrix of bytes seen as values in the finite field $\mathbb{F}_{256}$, defined using the irreducible polynomial $x^8 + x^4 + x^3 + x + 1$. Depending on the version of AES, N_r rounds are applied to the state: $N_r = 10$ for AES-128, $N_r = 12$ for AES-192 and $N_r = 14$ for AES-256. An AES round applies four operations to the state matrix:

- *SubBytes* (S-Box) - applying the same 8-bit to 8-bit invertible S-Box 16 times in parallel on each byte of the state (provides non-linearity in the cipher);
- *ShiftRows* (SR) - cyclic shift of each row (i-th row is shifted by i bytes to the left);
- *MixColumns* (MC) - multiplication of each column by a constant 4×4 invertible matrix over the field $GF(2^8)$ (together with the ShiftRows operation, it provides diffusion in the cipher);
- *AddRoundKey* (ARK) - XORing the state with a 128-bit subkey.

One round of AES can be described as $R(x) = K \oplus MC \circ SR \circ$ S-Box(x). In the first round an additional AddRoundKey operation (using a whitening key) is applied, and in the last round the MixColumns operation is omitted.

Notation Used in the Paper. Let x denote a plaintext, a ciphertext, an intermediate state or a key. Then $x_{i,j}$ with $i, j \in \{0, ..., 3\}$ denotes the byte in the row i and in the column j. We denote by k^r the subkey of the r-th round (where k^0 is the secret key for AES-128). If only one subkey is used (e.g. the first subkey k^0), then we denote it by k to simplify the notation. Finally, we denote by R one round[3] of AES, while we denote r rounds of AES by R^r. As last thing, in the paper we often use the term "partial collision" (or "*collision*") when two texts belong to the same coset of a given subspace $\mathcal{X}$.

2.2 Subspace Trails

Let F denote a round function in an iterative block cipher and let $V \oplus a$ denote a coset of a vector space V. Then if $F(V \oplus a) = V \oplus a$ we say that $V \oplus a$ is an *invariant coset* of the subspace V for the function F. This concept can be generalized to *trails of subspaces* [GRR17b], which has been recently introduced as generalization of the invariant subspace cryptanalysis.

Definition 1. Let $(V_1, V_2, ..., V_{r+1})$ denote a set of $r + 1$ subspaces with $\dim(V_i) \leq \dim(V_{i+1})$. If for each $i = 1, ..., r$ and for each a_i, there exist a_{i+1} such that $F(V_i \oplus a_i) \subseteq V_{i+1} \oplus a_{i+1}$, then $(V_1, V_2, ..., V_{r+1})$ is *subspace trail* of length r for the function F. If all the previous relations hold with equality, the trail is called a *constant-dimensional subspace trail.*

This means that if F^t denotes the application of t rounds with fixed keys, then $F^t(V_1 \oplus a_1) = V_{t+1} \oplus a_{t+1}$. We refer to [GRR17b] for more details about the concept of subspace trails. Our treatment here is however meant to be self-contained.

Subspace Trails of AES

Here we recall the subspace trails of AES presented in [GRR17b], working with vectors and vector spaces over $\mathbb{F}_{2^8}^{4\times4}$. For the following, we denote by $\{e_{0,0}, ..., e_{3,3}\}$ the unit vectors of $\mathbb{F}_{2^8}^{4\times4}$ (e.g. $e_{i,j}$ has a single 1 in row i and column j). We recall that given a subspace $\mathcal{X}$, the cosets $\mathcal{X} \oplus a$ and $\mathcal{X} \oplus b$ (where $a \neq b$) are *equal* (that is $\mathcal{X} \oplus a \equiv \mathcal{X} \oplus b$) if and only if $a \oplus b \in \mathcal{X}$.

Definition 2. The *column spaces* $\mathcal{C}_i$ are defined as $\mathcal{C}_i = \langle e_{0,i}, e_{1,i}, e_{2,i}, e_{3,i} \rangle$.

For instance, $\mathcal{C}_0$ corresponds to the symbolic matrix

$$\mathcal{C}_0 = \left\{ \begin{bmatrix} x_1 & 0 & 0 & 0 \\ x_2 & 0 & 0 & 0 \\ x_3 & 0 & 0 & 0 \\ x_4 & 0 & 0 & 0 \end{bmatrix} \middle| \forall x_1, x_2, x_3, x_4 \in \mathbb{F}_{2^8} \right\} \equiv \begin{bmatrix} x_1 & 0 & 0 & 0 \\ x_2 & 0 & 0 & 0 \\ x_3 & 0 & 0 & 0 \\ x_4 & 0 & 0 & 0 \end{bmatrix}.$$

[3] Sometimes we use the notation R_k instead of R to highlight the round key k.

Definition 3. The *diagonal spaces* $\mathcal{D}_i$ and the *inverse-diagonal spaces* $\mathcal{ID}_i$ are defined as $\mathcal{D}_i = SR^{-1}(\mathcal{C}_i)$ and $\mathcal{ID}_i = SR(\mathcal{C}_i)$.

For instance, $\mathcal{D}_0$ and $\mathcal{ID}_0$ correspond to symbolic matrices

$$\mathcal{D}_0 \equiv \begin{bmatrix} x_1 & 0 & 0 & 0 \\ 0 & x_2 & 0 & 0 \\ 0 & 0 & x_3 & 0 \\ 0 & 0 & 0 & x_4 \end{bmatrix}, \qquad \mathcal{ID}_0 \equiv \begin{bmatrix} x_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 \\ 0 & 0 & x_3 & 0 \\ 0 & x_4 & 0 & 0 \end{bmatrix}$$

for each $x_1, x_2, x_3, x_4 \in \mathbb{F}_{2^8}$.

Definition 4. The *i-th mixed spaces* $\mathcal{M}_i$ are defined as $\mathcal{M}_i = MC(\mathcal{ID}_i)$.

For instance, $\mathcal{M}_0$ corresponds to symbolic matrix

$$\mathcal{M}_0 \equiv \begin{bmatrix} \text{0x02} \cdot x_1 & x_4 & x_3 & \text{0x03} \cdot x_2 \\ x_1 & x_4 & \text{0x03} \cdot x_3 & \text{0x02} \cdot x_2 \\ x_1 & \text{0x03} \cdot x_4 & \text{0x02} \cdot x_3 & x_2 \\ \text{0x03} \cdot x_1 & \text{0x02} \cdot x_4 & x_3 & x_2 \end{bmatrix}.$$

Definition 5. For $I \subseteq \{0, 1, 2, 3\}$, let $\mathcal{C}_I$, $\mathcal{D}_I$, $\mathcal{ID}_I$ and $\mathcal{M}_I$ be defined as

$$\mathcal{C}_I = \bigoplus_{i \in I} \mathcal{C}_i, \qquad \mathcal{D}_I = \bigoplus_{i \in I} \mathcal{D}_i, \qquad \mathcal{ID}_I = \bigoplus_{i \in I} \mathcal{ID}_i, \qquad \mathcal{M}_I = \bigoplus_{i \in I} \mathcal{M}_i.$$

As shown in detail in [GRR17b][4]:

- for any coset $\mathcal{D}_I \oplus a$ there exists unique $b \in \mathcal{C}_I^\perp$ such that $R(\mathcal{D}_I \oplus a) = \mathcal{C}_I \oplus b$;
- for any coset $\mathcal{C}_I \oplus a$ there exists unique $b \in \mathcal{M}_I^\perp$ such that $R(\mathcal{C}_I \oplus a) = \mathcal{M}_I \oplus b$.

Theorem 1 ([GRR17b]). *For each I and for each $a \in \mathcal{D}_I^\perp$, there exist unique $b \in \mathcal{C}_I^\perp$ and $c \in \mathcal{M}_I^\perp$ (which depend on a and on the secret key k) such that*

$$R^2(\mathcal{D}_I \oplus a) = R(\mathcal{C}_I \oplus b) = \mathcal{M}_I \oplus c. \tag{1}$$

We refer to [GRR17b] for a complete proof of the Theorem. Moreover, note that if $\mathcal{X}$ is a generic subspace, $\mathcal{X} \oplus a$ is a coset of $\mathcal{X}$ and x and y are two elements of the (same) coset $X \oplus a$, then $x \oplus y \in \mathcal{X}$. It follows that:

Lemma 1 ([GRR17b]). *For all x, y and for all $I \subseteq \{0, 1, 2, 3\}$:*

$$Prob(R^2(x) \oplus R^2(y) \in \mathcal{M}_I \,|\, x \oplus y \in \mathcal{D}_I) = 1. \tag{2}$$

We finally recall that for each $I, J \subseteq \{0, 1, 2, 3\}$:

$$\mathcal{M}_I \cap \mathcal{D}_J = \{0\} \qquad \text{if and only if} \qquad |I| + |J| \leq 4, \tag{3}$$

as demonstrated in [GRR17b]. It follows that:

Proposition 1 ([GRR17b]). *Let $I, J \subseteq \{0, 1, 2, 3\}$ such that $|I| + |J| \leq 4$. For all x, y with $x \neq y$:*

$$Prob(R^4(x) \oplus R^4(y) \in \mathcal{M}_I \,|\, x \oplus y \in \mathcal{D}_J) = 0. \tag{4}$$

[4] *Remark.* Observe that $\mathcal{D}_I \oplus \mathcal{D}_I^\perp = \mathcal{C}_I \oplus \mathcal{C}_I^\perp = \mathcal{ID}_I \oplus \mathcal{ID}_I^\perp = \mathcal{M}_I \oplus \mathcal{M}_I^\perp = \mathbb{F}_{2^8}^{4\times 4}$ for each $I \subseteq \{0, 1, 2, 3\}$. As a result, the complements of the subspaces $\mathcal{C}_I$, $\mathcal{D}_I$, $\mathcal{ID}_I$, $\mathcal{M}_I$ are simply the (respective) *orthogonal* complements $\mathcal{C}_I^\perp$, $\mathcal{D}_I^\perp$, $\mathcal{ID}_I^\perp$, $\mathcal{M}_I^\perp$.

We remark that all these results can be re-described using a more "classical" truncated differential notation[5], as formally pointed out in [BLN17]. To be more concrete, if two texts t^1 and t^2 are equal except for the bytes in the i-th diagonal[6] for each $i \in I$, then they belong to the same coset of $\mathcal{D}_I$. A coset of $\mathcal{D}_I$ corresponds to a set of $2^{32\cdot|I|}$ texts with $|I|$ active diagonals. Again, two texts t^1 and t^2 belong to the same coset of $\mathcal{M}_I$ if the bytes of their difference $MC^{-1}(t^1 \oplus t^2)$ in the i-th anti-diagonal for each $i \notin I$ are equal to zero. Similar considerations hold for the column space $\mathcal{C}_I$ and the inverse-diagonal space $\mathcal{ID}_I$.

We finally introduce some notation that we largely use in the following.

Definition 6. Given two different texts $t^1, t^2 \in \mathbb{F}_{2^8}^{4\times 4}$, we say that $t^1 \le t^2$ if $t^1 = t^2$ or if there exists $i, j \in \{0,1,2,3\}$ such that (1) $t^1_{k,l} = t^2_{k,l}$ for all $k,l \in \{0,1,2,3\}$ with $k + 4\cdot l < i + 4 \cdot j$ and (2) $t^1_{i,j} < t^2_{i,j}$. Moreover, we say that $t^1 < t^2$ if $t^1 \le t^2$ (with respect to the definition just given) and $t^1 \neq t^2$.

Definition 7. Let $\mathcal{X}$ be one of the previous subspaces, that is $\mathcal{C}_I$, $\mathcal{D}_I$, $\mathcal{ID}_I$ or $\mathcal{M}_I$. Let $x_0, ..., x_{n-1} \in \mathbb{F}_{2^8}^{4\times 4}$ be a basis of $\mathcal{X}$ - i.e. $\mathcal{X} \equiv \langle x_0, x_1, ..., x_{n-1}\rangle$ where $n = 4\cdot|I|$ - s.t. $x_i < x_{i+1}$ for each $i = 0, ..., n-1$. Let t be an element of an arbitrary coset of $\mathcal{X}$, that is $t \in \mathcal{X} \oplus a$ for arbitrary a. We say that t is "generated" by the *generating variables* $(t^0, ..., t^{n-1})$ - for the following, $t \equiv (t^0, ..., t^{n-1})$ - if and only if

$$t \equiv (t^0, ..., t^{n-1}) \quad \text{iff} \quad t = a \oplus \bigoplus_{i=0}^{n-1} t^i \cdot x_i.$$

As an example, let $\mathcal{X} = \mathcal{M}_0 \equiv \langle MC(e_{0,0}), MC(e_{3,1}), MC(e_{2,2}), MC(e_{1,3})\rangle$, and let $p \in \mathcal{M}_0 \oplus a$. Then $p \equiv (p^0, p^1, p^2, p^3)$ if and only if

$$p \equiv p^0 \cdot MC(e_{0,0}) \oplus p^1 \cdot MC(e_{1,3}) \oplus p^2 \cdot MC(e_{2,2}) \oplus p^3 \cdot MC(e_{3,1}) \oplus a. \tag{5}$$

Similarly, let $\mathcal{X} = \mathcal{C}_0 \equiv \langle e_{0,0}, e_{1,0}, e_{2,0}, e_{3,0}\rangle$, and let $p \in \mathcal{C}_0 \oplus a$. Then $p \equiv (p^0, p^1, p^2, p^3)$ if and only if $p \equiv a \oplus p^0 \cdot e_{0,0} \oplus p^1 \cdot e_{1,0} \oplus p^2 \cdot e_{2,0} \oplus p^3 \cdot e_{3,0}$.

3 "Multiple-of-8" Secret-Key Distinguisher for 5-round AES

The starting point of our secret-key distinguisher is the property proposed and exploited in [GRR17a] to set up the first 5-round secret-key distinguisher of AES (independent of the secret key). For this reason, in this section we recall the main idea of that paper, and we refer to [GRR17a] for a complete discussion.

Consider a set of plaintexts in the same coset of the diagonal space $\mathcal{D}_I$, that is $2^{32\cdot|I|}$ plaintexts with $|I|$ active diagonals, and the corresponding ciphertexts after 5 rounds. The 5-round AES distinguisher proposed in [GRR17a] exploits the fact that the number of different pairs of ciphertexts that belong to the same coset of $\mathcal{M}_J$ for a fixed J (that is, the number of different pairs of ciphertexts that are equal on $|J|$ fixed anti-diagonals, omitting the final MixColumns operation) is always a multiple of 8 with probability 1 independently of the secret key, of the details of the S-Box and of the MixColumns matrix. In more details, given a set of plaintexts/ciphertexts (p^i, c^i) for $i = 0, ..., 2^{32\cdot|I|} - 1$ (where all the plaintexts belong to the same coset of $\mathcal{D}_I$), the number of different pairs[7] of ciphertexts (c^i, c^j) that satisfy $c^i \oplus c^j \in \mathcal{M}_J$ for a certain fixed $J \subset \{0,1,2,3\}$ has the special property to be a multiple of 8 with prob. 1. Since for a random permutation the same number

[5] Our choice to use the subspace trail notation to present our new distinguisher and attack is motivated by the fact that it allows to describe them in a more formal way than using the "classical" notation.

[6] The i-th diagonal of a 4×4 matrix A is defined as the elements that lie on row r and column c such that $r - c = i \bmod 4$. The i-th anti-diagonal of a 4×4 matrix A is defined as the elements that lie on row r and column c such that $r + c = i \bmod 4$.

[7] Two pairs (c^i, c^j) and (c^j, c^i) are considered equivalent.

doesn't have any special property (e.g. it has the same probability to be even or odd), this allows to distinguish 5-round AES from a random permutation.

Since each coset of $\mathcal{D}_I$ is mapped into a coset of $\mathcal{M}_I$ after 2 rounds with prob. 1 - see Theorem 1 - and vice-versa, in order to prove the result given in [GRR17a] it is sufficient to show that given plaintexts in the same coset of $\mathcal{M}_I$, then the number of collisions after one round in the same coset of $\mathcal{D}_J$ is a multiple of 8 (see [GRR17a] for details).

Theorem 2 ([GRR17a])**.** *Let $\mathcal{M}_I$ and $\mathcal{D}_J$ be the subspaces defined as before for certain fixed I and J with $1 \leq |I| \leq 3$. Given an arbitrary coset of $\mathcal{M}_I$ - that is $\mathcal{M}_I \oplus a$ for a fixed $a \in \mathcal{M}_I^{\perp}$, let (p^i, c^i) for $i = 0, ..., 2^{32\cdot|I|} - 1$ be the $2^{32\cdot|I|}$ plaintexts in $\mathcal{M}_I \oplus a$ (i.e. $p^i \in \mathcal{M}_I \oplus a$ for each i) and the corresponding ciphertexts after 1 round (i.e. $c^i = R(p^i)$).*

The number n of different pairs of ciphertexts (c^i, c^j) for $i \neq j$ such that $c^i \oplus c^j \in \mathcal{D}_J$ (i.e. c^i and c^j belong to the same coset of $\mathcal{D}_J$) is always a multiple of 8 with prob. 1.

We refer to [GRR17a] for a detailed proof, and we limit here to recall and to highlight the main concepts that are useful for the following.

Without loss of generality (w.l.o.g.), we focus on the case $|I| = 1$ and we assume $I = \{0\}$. Given two texts p and q in $\mathcal{M}_0 \oplus a$, by definition there exist $p^0, p^1, p^2, p^3 \in \mathbb{F}_{2^8}$ and $q^0, q^1, q^2, q^3 \in \mathbb{F}_{2^8}$ such that

$$p = a \oplus \begin{bmatrix} 2\cdot p^0 & p^1 & p^2 & 3\cdot p^3 \\ p^0 & p^1 & 3\cdot p^2 & 2\cdot p^3 \\ p^0 & 3\cdot p^1 & 2\cdot p^2 & p^3 \\ 3\cdot p^0 & 2\cdot p^1 & p^2 & p^3 \end{bmatrix}, \qquad q = a \oplus \begin{bmatrix} 2\cdot q^0 & q^1 & q^2 & 3\cdot q^3 \\ q^0 & q^1 & 3\cdot q^2 & 2\cdot q^3 \\ q^0 & 3\cdot q^1 & 2\cdot q^2 & q^3 \\ 3\cdot q^0 & 2\cdot q^1 & q^2 & q^3 \end{bmatrix}$$

where $2 \equiv$ 0x02 and $3 \equiv$ 0x03, or equivalently $p \equiv (p^0, p^1, p^2, p^3)$ and $q \equiv (q^0, q^1, q^2, q^3)$ - see (5). As first thing, we recall that if $1 \leq r \leq 3$ generating variables are equal, then the two texts cannot belong to the same coset of $\mathcal{D}_J$ for $|J| \leq r$ after one round - this is due to the branch number of the MixColumns matrix (which is 5).

Case: Different Generating Variables. If the two texts p and q defined as before are generated by different variables (i.e. $p^i \neq q^i$ for each $i = 0, ..., 3$), then they can belong to the same coset of $\mathcal{D}_J$ for a certain J with $|J| \geq 1$ after one round. It is possible to prove that $p \equiv (p^0, p^1, p^2, p^3)$ and $q \equiv (q^0, q^1, q^2, q^3)$ satisfy $R(p) \oplus R(q) \in \mathcal{D}_J$ for $|J| \geq 1$ if and only if others pairs of texts generated by different combinations of the previous variables have the same property. A formal statement is provided in Lemma 2.

Lemma 2. *Let p and q be two different elements in $\mathcal{M}_I \oplus a$ (i.e. a coset of $\mathcal{M}_I$) for $I \subseteq \{0, 1, 2, 3\}$ and $|I| = 1$, with $p \equiv (p^0, p^1, p^2, p^3)$ and $q \equiv (q^0, q^1, q^2, q^3)$, such that $p^i \neq q^i$ for each $i = 0, ..., 3$. Independently of the secret key, of the details of the S-Box and of the MixColumns matrix, $R(p)$ and $R(q)$ belong to the same coset of a particular subspace $\mathcal{D}_J$ for $J \subseteq \{0, 1, 2, 3\}$ (that is $R(p) \oplus R(q) \in \mathcal{D}_J$) if and only if the pairs of texts in $\mathcal{M}_I \oplus a$ generated by the following combinations of variables*

1. (p^0, p^1, p^2, p^3) *and* (q^0, q^1, q^2, q^3);
2. (q^0, p^1, p^2, p^3) *and* (p^0, q^1, q^2, q^3);
3. (p^0, q^1, p^2, p^3) *and* (q^0, p^1, q^2, q^3);
4. (p^0, p^1, q^2, p^3) *and* (q^0, q^1, p^2, q^3);
5. (p^0, p^1, p^2, q^3) *and* (q^0, q^1, q^2, p^3);
6. (q^0, q^1, p^2, p^3) *and* (p^0, p^1, q^2, q^3);
7. (q^0, p^1, q^2, p^3) *and* (p^0, q^1, p^2, q^3);
8. (q^0, p^1, p^2, q^3) *and* (p^0, q^1, q^2, p^3).

have the same property.

Case: Equal Generating Variables. Similar results can be obtained if one or two variables are equal. For the following, we focus on the case in which two variables are equal (the case of one equal variable is analogous).

Lemma 3. *Let p and q be two different elements in $\mathcal{M}_I \oplus a$ for $I \subseteq \{0,1,2,3\}$ and $|I| = 1$, with $p \equiv (p^0, p^1, p^2, p^3)$ and $q \equiv (q^0, q^1, q^2, q^3)$, such that $p^i \neq q^i$ for $i = 0, 1$ and $p^i = q^i$ for $i = 2, 3$ (similar for the other cases). Independently of the secret key, of the details of the S-Box and of the MixColumns matrix, $R(p)$ and $R(q)$ belong to the same coset of a particular subspace $\mathcal{D}_J$ for $J \subseteq \{0,1,2,3\}$ if and only if the pairs of texts in $\mathcal{M}_I \oplus a$ generated by the following combinations of variables*

$$1.\, (p^0, p^1, z, w) \;\; \textit{and} \;\; (p^0, p^1, z, w); \qquad 2.\, (p^0, q^1, z, w) \;\; \textit{and} \;\; (q^0, p^1, z, w);$$

where z and w can take any possible value in $\mathbb{F}_{2^8}$, have the same property.

Case $|\mathbf{I}| = 2$ and $|\mathbf{I}| = 3$. Finally, we mention that similar considerations can be done for the cases $|I| \geq 2$. W.l.o.g consider $|I| = 2$ and assume $I = \{0,1\}$ (the other cases are analogous). Given two texts p and q in the same coset of $\mathcal{M}_I$, that is $\mathcal{M}_I \oplus a$ for a given $a \in \mathcal{M}_I^{\perp}$, there exist $p_0', p_0'', p_1', p_1'', p_2', p_2'', p_3', p_3'' \in \mathbb{F}_{2^8}$ and $q_0', q_0'', q_1', q_1'', q_2', q_2'', q_3', q_3'' \in \mathbb{F}_{2^8}$ such that:

$$p = a \oplus MC \cdot \begin{bmatrix} p_0' & p_1'' & 0 & 0 \\ p_0'' & 0 & 0 & p_3' \\ 0 & 0 & p_2' & p_3'' \\ 0 & p_1' & p_2'' & 0 \end{bmatrix}, \qquad q = a \oplus MC \cdot \begin{bmatrix} q_0' & q_1'' & 0 & 0 \\ q_0'' & 0 & 0 & q_3' \\ 0 & 0 & q_2' & q_3'' \\ 0 & q_1' & q_2'' & 0 \end{bmatrix}.$$

As for the case $|I| = 1$, the idea is to consider all the possible combinations of the variables $p_0 \equiv (p_0', p_0''), p_1 \equiv (p_1', p_1''), p_2 \equiv (p_2', p_2''), p_3 \equiv (p_3', p_3'')$ and $q_0 \equiv (q_0', q_0''), q_1 \equiv (q_1', q_1''), q_2 \equiv (q_2', q_2''), q_3 \equiv (q_3', q_3'')$. In other words, the idea is to consider variables in $(\mathbb{F}_{2^8})^2 \equiv \mathbb{F}_{2^8} \times \mathbb{F}_{2^8}$ and not in $\mathbb{F}_{2^8}$. For $|I| = 3$, the idea is to work with variables in $(\mathbb{F}_{2^8})^3$.

For the following, given texts in the same cosets of $\mathcal{C}_I$ or $\mathcal{M}_I$ for $I \subseteq \{0,1,2,3\}$, we recall that the number of couples of texts with n "equal generating variable(s) in $(\mathbb{F}_{2^8})^{|I|}$" (as just defined) for $0 \leq n \leq 3$ is given by

$$\binom{4}{n} \cdot 2^{32\cdot|I|-1} \cdot (2^{8\cdot|I|} - 1)^{4-n} \tag{6}$$

as proved in App. A.

Why is it (rather) *hard* to set up key-recovery attacks that exploit such distinguisher?

Given this 5-round distinguisher, a natural question regards the possibility to exploit it in order to set up a key-recovery attack on 6-round AES-128 which is better than a brute force one. A possible way is the following. Consider 2^{32} chosen plaintexts in the same coset of a diagonal space $\mathcal{D}_i$, and the corresponding ciphertexts after 6 rounds. A possibility is to guess the final key, decrypt the ciphertexts and check if the number of collisions in the same coset of $\mathcal{M}_J$ is a multiple of 8. If not, the guessed key is wrong. However, since a coset of $\mathcal{M}_J$ is mapped into the full space, it seems hard to check this property one round before without guessing the entire key. It follows that it is rather hard to set up an attack different than a brute force one that exploits directly the 5-round distinguisher proposed in [GRR17a]. For comparison, note that such a problem doesn't arise for the other distinguishers for up to 4-round AES (e.g. the impossible differential or the integral ones) present in the literature, for which it is sufficient to guess only part of the secret key in order to verify if the required property is satisfied or not.

4 New 4-round Secret-Key Distinguisher for AES

In this section, we re-exploit the property proposed in [GRR17a] to set up a *new* 4-round secret-key distinguisher for AES. Before we go into the details, we present the general idea.

As we have just seen, given 2^{32} plaintexts in the same coset of $\mathcal{M}_I$ for $|I| = 1$ and the corresponding ciphertexts after 1 round, that is (p^i, c^i) for $i = 0, ..., 2^{32} - 1$ where $p^i \in \mathcal{M}_I \oplus a$ and $c^i = R(p^i)$, then the number n of different pairs of ciphertexts (c^i, c^j) for $i \neq j$ that satisfy $c^i \oplus c^j \in \mathcal{D}_J$ is always a multiple of 8. This is due to the fact that if one pair of texts belong to the same coset of $\mathcal{D}_J$ after one round, then other pairs of texts have the same property.

Thus, consider a pair of plaintexts p^1 and p^2 such that the corresponding texts after one round belong (or not) to the same coset of $\mathcal{D}_J$. As we have seen, there exist other pairs of plaintexts $\hat{p}^1$ and $\hat{p}^2$ whose ciphertexts after one round have the same property. *The crucial point is that the pairs (p^1, p^2) and $(\hat{p}^1, \hat{p}^2)$ are not independent in the sense that the variables that generate the first pair of texts are the same that generate the other pairs, but in a different combination.* The idea is to exploit this property in order to set up a new distinguisher for round-reduced AES. In other words, *instead of limiting to count the number of collisions and check that it is a multiple of* 8 *as in [GRR17a], the idea is to check if these relationships between the variables that generate the plaintexts* (whose ciphertexts belong or not the same coset of a given subspace $\mathcal{M}_J$) *hold or not.*

4.1 *Mixture Differential* Distinguisher for 4-round AES

A formal description of the proposed Mixture Differential Distinguisher for 4-round AES is given in the following Lemma[8].

Lemma 4. *Given the subspace $\mathcal{C}_0 \cap \mathcal{D}_{0,3} \equiv \langle e_{0,0}, e_{1,0} \rangle \subseteq \mathcal{C}_0$, consider two plaintexts p^1 and p^2 in the same coset $(\mathcal{C}_0 \cap \mathcal{D}_{0,3}) \oplus a$ generated by $p^1 \equiv (z^1, w^1)$ and $p^2 \equiv (z^2, w^2)$. Let $\hat{p}^1, \hat{p}^2 \in (\mathcal{D}_{0,3} \cap \mathcal{C}_0) \oplus a$ be two other plaintexts generated by $\hat{p}^1 \equiv (z^1, w^2)$ and $\hat{p}^2 \equiv (z^2, w^1)$. The following event*

$$R^4(p^1) \oplus R^4(p^2) \in \mathcal{M}_J \quad \text{if and only if} \quad R^4(\hat{p}^1) \oplus R^4(\hat{p}^2) \in \mathcal{M}_J$$

holds with prob. 1 for 4-round AES, independently of the secret key, of the details of the S-Box and of the MixColumns matrix (except for the branch number equal to 5).

Since for a random permutation the same event happens with approximately probability $2^{-32 \cdot (4-|J|)}$ - i.e close to 0 (note that this probability is maximized by $|J| = 3$), it is possible to exploit this fact to set up a 4-round distinguisher. Due to the fact that the variables of p^1 and p^2 are "mixed" in order to generate $\hat{p}^1$ and $\hat{p}^2$, we name this distinguisher as *Mixture Differential* distinguisher.

4.1.1 Proof using the "super-Sbox" Notation

As first thing, we prove the previous result using the "super-Sbox" notation - introduced in [DR06] by the designers of AES, where

$$\textit{super-Sbox}(\cdot) = \text{S-Box} \circ ARK \circ MC \circ \text{S-Box}(\cdot) \tag{7}$$

Consider two pairs of texts (p^1, p^2) and $(\hat{p}^1, \hat{p}^2)$ in a coset of $\mathcal{C}_0 \cap \mathcal{D}_{0,3}$ - that is $(\mathcal{C}_0 \cap \mathcal{D}_{0,3}) \oplus a$ for a fixed a, such that

$$p^i \equiv a \oplus \begin{bmatrix} z^i & 0 & 0 & 0 \\ w^i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \text{and} \qquad \hat{p}^i \equiv a \oplus \begin{bmatrix} z^i & 0 & 0 & 0 \\ w^{3-i} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

[8] We mention that the result proposed in Lemma 4 is already included - as a special case - in the results proposed in Theorem 3.

for $i = 1, 2$, that is $p^i \equiv (z^i, w^i)$ and $\hat{p}^i \equiv (z^i, w^{3-i})$.

The goal is to prove that

$$R^4(p^1) \oplus R^4(p^2) \in \mathcal{M}_J \quad \textit{if and only if} \quad R^4(\hat{p}^1) \oplus R^4(\hat{p}^2) \in \mathcal{M}_J.$$

Since $Prob(R^2(x) \oplus R^2(y) \in \mathcal{M}_I \,|\, x \oplus y \in \mathcal{D}_I) = 1$ (see (2)), this is equivalent to prove that

$$R^2(p^1) \oplus R^2(p^2) \in \mathcal{D}_J \quad \textit{if and only if} \quad R^2(\hat{p}^1) \oplus R^2(\hat{p}^2) \in \mathcal{D}_J.$$

First of all, observe that $p^1 \oplus p^2 \in \big(\mathcal{C}_0 \cap \mathcal{D}_{0,3}\big) \subseteq \mathcal{D}_{0,3}$, and that $R^2(p^1) \oplus R^2(p^2) \in \mathcal{M}_{0,3}$. Since $\mathcal{M}_{0,3} \cap \mathcal{D}_J \neq \{0\}$ if and only if $|J| = 3$ (see (3) for details), $R^2(p^1) \oplus R^2(p^2) \in \mathcal{D}_J$ can occur if and only if $|J| = 3$.

As it is well known, 2-round encryption can be rewritten using the super-Sbox notation

$$R^2(\cdot) = ARK \circ MC \circ SR \circ \textit{super-Sbox} \circ SR(\cdot).$$

Since ShiftRows and MixColumns operations are linear, it is sufficient to prove that

$$\textit{super-Sbox}(q^1) \oplus \textit{super-Sbox}(q^2) \in \mathcal{W}_J \quad \text{iff} \quad \textit{super-Sbox}(\hat{q}^1) \oplus \textit{super-Sbox}(\hat{q}^2) \in \mathcal{W}_J$$

where

$$q^i = SR(p^i) \equiv SR(a) \oplus \begin{bmatrix} z^i & 0 & 0 & 0 \\ 0 & 0 & 0 & w^i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \hat{q}^i = SR(\hat{p}^i) \equiv SR(a) \oplus \begin{bmatrix} z^i & 0 & 0 & 0 \\ 0 & 0 & 0 & w^{3-i} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

for $i = 1, 2$ (note that $SR(\mathcal{D}_{0,3} \cap \mathcal{C}_0) = \mathcal{C}_{0,3} \cap \mathcal{ID}_0$ by definition) and where the subspace $\mathcal{W}_J$ is defined as

$$\mathcal{W}_J := SR^{-1} \circ MC^{-1}(\mathcal{D}_J). \tag{8}$$

Since each column of q^1 and q^2 depends on different and independent variables, since the super-Sbox works independently on each column and since the XOR-sum is commutative, it follows that

$$\textit{super-Sbox}(q^1) \oplus \textit{super-Sbox}(q^2) = \textit{super-Sbox}(\hat{q}^1) \oplus \textit{super-Sbox}(\hat{q}^2)$$

which implies the thesis.

4.1.2 Data and Computational Cost

Data Cost. Since a coset of $\mathcal{C}_0 \cap \mathcal{D}_{0,3}$ contains 2^{16} plaintexts, it is possible to construct $2^{15} \cdot (2^{16} - 1) \simeq 2^{31}$ different couples. For our goal, we consider only the pairs of texts $p^1 \equiv (z^1, w^1)$ and $p^2 \equiv (z^2, w^2)$ with different generating variables, that is $z^1 \neq z^2$ and $w^1 \neq w^2$ (if $z^1 = z^2$ or $w^1 = w^2$, then $p^1 \oplus p^2 \in \big(\mathcal{C}_0 \cap \mathcal{D}_k\big) \subseteq \mathcal{D}_k$ for a certain $k \in \{0, 3\}$, which implies that $R^4(p^1) \oplus R^4(p^2) \notin \mathcal{M}_J$ for each J due to Prop. 1). Using formula (6), the number of pairs with two different generating variables is given by $2^{15} \cdot (2^8 - 1)^2 \simeq 2^{30.989}$. As we have just seen, only half of them - that is, $2^{29.989}$ - are independent.

In order to distinguish 4-round AES from a random permutation, one has to check that

$$c^1 \oplus c^2 = R^4\big(p^1 \equiv (z^1, w^1)\big) \oplus R^4\big(p^2 \equiv (z^2, w^2)\big) \in \mathcal{M}_J$$

if and only if

$$\hat{c}^1 \oplus \hat{c}^2 = R^4\big(\hat{p}^1 \equiv (z^1, w^2)\big) \oplus R^4\big(\hat{p}^2 \equiv (z^2, w^1)\big) \in \mathcal{M}_J.$$

If this property is not satisfied for at least one couple, then it is possible to conclude that the analyzed permutation is a random one.

```
Data: 2 cosets of $\mathcal{D}_{0,3} \cap \mathcal{C}_0$ (e.g. $(\mathcal{C}_0 \cap \mathcal{D}_{0,3}) \oplus a_i$ for $a_0, a_1 \in (\mathcal{D}_{0,3} \cap \mathcal{C}_0)^{\perp}$) and
      corresponding ciphertexts after 4 rounds
Result: $0 \equiv$ Random permutation or $1 \equiv$ 4-round AES - Prob. 95%
for each coset $(\mathcal{D}_{0,3} \cap \mathcal{C}_0) \oplus a_x$ for $x = 0, 1$ do
    for each $I \subseteq \{0, 1, 2, 3\}$ with $|I| = 3$ do
        let $(p^i, c^i)$ for $i = 0, ..., 2^{16} - 1$ be the $2^{16}$ (plaintexts, ciphertexts) of
          $(\mathcal{D}_{0,3} \cap \mathcal{C}_0) \oplus a_x$;
        re-order this set of elements w.r.t. the partial order $\preceq$ described in Def. 8
          s.t. $c^k \preceq c^{k+1}$ for each $k$;                    // $\preceq$ depends on $I$
        $i \leftarrow 0$;
        while $i < 2^{16} - 1$ do
            $j \leftarrow i$;
            while $c^j \oplus c^{j+1} \in \mathcal{M}_I$ do
                $j \leftarrow j + 1$;
            end
            for each $k$ from $i$ to $j$ do
                for each $l$ from $k + 1$ to $j$ do
                    given $p^k \equiv (z^1, w^1)$ and $p^l \equiv (z^2, w^2)$, let $q^1 \equiv (z^1, w^2)$ and
                      $q^2 \equiv (z^2, w^1)$ in $(\mathcal{D}_{0,3} \cap \mathcal{C}_0) \oplus a_i$;
                    if $R^4(q^1) \oplus R^4(q^2) \notin \mathcal{M}_I$// Remember that $R^4(p^k) \oplus R^4(p^l) \in \mathcal{M}_I$
                      then
                        return 0.                               // Random permutation
                    end
                end
            end
            $i \leftarrow j + 1$;
        end
    end
end
return 1.                               // 4-round AES permutation - Prob. 95%
```

Algorithm 1: *Secret-Key Distinguisher for 4-round of AES.*

Given *a random permutation* $\Pi(\cdot)$*, what is the probability that* $c^1 \oplus c^2 \equiv \Pi(p^1) \oplus \Pi(p^2) \in \mathcal{M}_J$ *and* $\hat{c}^1 \oplus \hat{c}^2 \equiv \Pi(\hat{p}^1) \oplus \Pi(\hat{p}^2) \notin \mathcal{M}_J$ *- or vice-versa - for a certain* $J \subset \{0, 1, 2, 3\}$ *with* $|J| = 3$*?* Since there are 4 different indexes J with $|J| = 3$ and since $Prob(t \in \mathcal{M}_J) = 2^{-32 \cdot (4 - |J|)}$, this event happens with probability (approximately) equal to

$$2 \cdot 4 \cdot 2^{-32} \cdot (1 - 2^{-32}) \simeq 2^{-29}.$$

As a result, in order to distinguish a random permutation from 4-round AES with probability higher than *pr*, it is sufficient that the previous event occurs for at least one couple of two pairs of texts with probability higher than *pr* (in order to recognize the random permutation). It follows that one needs approximately n different *independent* pairs of texts such that $pr \geq 1 - (1 - 2^{-29})^n$, that is

$$n \geq \frac{\log(1 - pr)}{\log(1 - 2^{-29})} \approx -2^{29} \cdot \log(1 - pr).$$

For $pr = 95\%$, one needs approximately $n \geq 2^{30.583}$ different *independent* pairs of texts, that is approximately 2 different cosets $\mathcal{C}_0 \cap \mathcal{D}_{0,3}$ for a total data cost of $2^{16} \cdot 2 = 2^{17}$ chosen plaintexts.

Computational Cost. We limit here to report the computational cost of the distinguisher, and we refer to App. B for all the details. In order to implement the distinguisher, the idea is to re-order the ciphertexts using a particular partial order $\preceq$ as defined in Def. 8, and to work in the way described in Algorithm 1.

Instead of checking the previous property for all possible couples of texts, the idea is to check it only for the couples of texts for which the two ciphertexts belong to the same coset of $\mathcal{M}_J$. In other words, if $c^1 \oplus c^2 \in \mathcal{M}_J$, then one checks that $\hat{c}^1 \oplus \hat{c}^2 \in \mathcal{M}_J$ (prob. 1 for 4-round AES *vs* prob. 2^{-32} for a random permutation). Instead, if $c^1 \oplus c^2 \notin \mathcal{M}_J$, then one doesn't check that $\hat{c}^1 \oplus \hat{c}^2 \notin \mathcal{M}_J$. Note that the probability of this last event is very close for the AES and for the random permutation (prob. 1 for 4-round AES *vs* prob. $1-2^{-32}$ for a random permutation). In other words, checking that "if $c^1 \oplus c^2 \in \mathcal{M}_J$ then $\hat{c}^1 \oplus \hat{c}^2 \in \mathcal{M}_J$" is sufficient to distinguish 4-round AES from a random permutation.

The reason of this strategy is that it allows to save and minimize the computational cost, which is well approximated by $2^{23.09}$ table look-ups, or approximately $2^{16.75}$ four-round encryptions (assuming[9] 20 table look-ups $\approx$ 1 round of encryption), where we limit to remember that *the cost to re-order a set of n texts w.r.t. a given partial order is*

$$\mathcal{O}(n \cdot \log n) \qquad \textit{table look-ups.}$$

Definition 8. Let $I \subset \{0,1,2,3\}$ with $|I| = 3$ and let $l \in \{0,1,2,3\} \setminus I$. Let $t^1, t^2 \in \mathbb{F}_{2^8}^{4\times 4}$ with $t^1 \neq t^2$. Text t^1 is less or equal than text t^2 w.r.t. the partial order $\preceq$ (i.e. $t^1 \preceq t^2$) if and only if one of the two following conditions is satisfied (indexes are taken modulo 4):

- there exists $j \in \{0,1,2,3\}$ s.t. $MC^{-1}(t^1)_{i,l-i} = MC^{-1}(t^2)_{i,l-i}$ for all $i < j$ and $MC^{-1}(t^1)_{j,l-j} < MC^{-1}(t^2)_{j,l-j}$;
- $MC^{-1}(t^1)_{i,l-i} = MC^{-1}(t^2)_{i,l-i}$ for all $i = 0, ..., 3$, and $MC^{-1}(t^1) < MC^{-1}(t^2)$ where $<$ is defined in Def. 6.

4.1.3 Practical Verification

Using a C/C++ implementation[10], we have practically verified the distinguishers just described both for full size AES and a small scale variant of AES, as presented in [CMR05]. While for full size AES each word is composed of 8 bits, in the small scale variant each word is composed of 4 bits (we refer to [CMR05] for a complete description of this small scale AES). We highlight that the previous results hold exactly in the same way also for this small scale variant of AES, since the previous argumentation is independent of the fact that each word of AES is of 4 or 8 bits.

The distinguisher just presented works in the same way for full and small scale AES, and it is able to distinguish AES from a random permutation using $2 \cdot (2^8)^2 = 2^{17}$ chosen plaintexts in the first case and $2 \cdot (2^4)^2 = 2^9$ in the second one (i.e. 2 cosets of $\mathcal{C}_0 \cap \mathcal{D}_{0,3}$, each one of size 2^{16} and 2^8 respectively for full and small scale AES[11]) as expected. For full size AES, while the theoretical computational cost is of 2^{23} table look-ups, the practical one is on average 2^{22} in the case of a random permutation and 2^{24} in the case of an AES permutation. We emphasize that for a random permutation, it is sufficient to find *one* couple of two pairs of texts that doesn't satisfy the required property (to recognize the

[9] We highlight that even if this approximation is not formally correct - the size of the table of an S-Box look-up is lower than the size of the table used for ours distinguisher, it allows to give a comparison between our distinguishers and the others currently present in the literature. This approximation is largely used in the literature.

[10] The source codes of the distinguishers/attacks are available at https://github.com/Krypto-iaik/Attacks_AES

[11] Following the same analysis proposed in Sect. 4.1, here we show that 2 initial cosets are necessary to set up the attack also for the small scale case. Using the same notation of Sect. 4.1.1, the probability that $R^4(p^1) \oplus R^4(p^2) \in \mathcal{M}_J$ and $R^4(\hat{p}^1) \oplus R^4(\hat{p}^2) \notin \mathcal{M}_J$ (or vice-versa) for a (small scale) random permutation is $2 \cdot 4 \cdot 2^{-16} \cdot (1-2^{-16}) = 2^{-13}$. It follows that one needs $n \geq 2^{14.583}$ different *independent* pairs of texts to set up the attack with probability higher than 95%, that is approximately 2 different cosets $\mathcal{C}_0 \cap \mathcal{D}_{0,3}$ (note that for each coset it is possible to construct $\frac{1}{2} \cdot \binom{2^8}{2} \approx 2^{14}$ independent pairs of texts).

random permutation). In the case of the AES permutation, the difference between the theoretical and the practical cases (i.e. a factor 2) is due to the fact that the cost of the merge sort algorithm is $O(n \cdot \log n)$ and by the definition of the big $O(\cdot)$ notation[12].

For the small scale AES, using 2 different initial cosets of $\mathcal{C}_0 \cap \mathcal{D}_{0,3}$, the theoretical computational cost is well approximated by $2 \cdot 4 \cdot 2^8 \cdot (\log 2^8 + 1) \simeq 2^{14.2}$ table look-ups. The practical cost is approximately $2^{13.5}$ for the case of a random permutation and 2^{15} for the AES case.

4.2 Generic Mixture Differential Distinguishers for 4-round AES

Using results presented in [GRR17a] and recalled in detail in Sect. 3, it is possible to set up *alternative* 4-round "mixture differential" distinguishers also for any pair of plaintexts p^1 and p^2 that have different generating variables or that belong to the same coset of a subspace $\mathcal{C}_I$ for each $I \subseteq \{0, 1, 2, 3\}$. For sake of simplicity, we don't list all possible cases, but we limit to (formally) present two cases that are used to set up new secret-key distinguishers - see [Gra17] for details - and new key-recovery attacks for AES - see Sect. 5. The proof of the following distinguishers is based on the one just proposed, adapted to the analyzed case. As before, also the following distinguishers work in both the decryption and encryption direction[13].

Starting Point for 5-round Distinguisher proposed in [Gra17]. As first case, we present a generalization of the result proposed in Lemma 4.

Theorem 3. *Given the subspace $\mathcal{C}_0 \cap \mathcal{D}_{0,3} \equiv \langle e_{0,0}, e_{1,0} \rangle \subseteq \mathcal{C}_0$, consider two plaintexts p^1 and p^2 in the same coset $(\mathcal{C}_0 \cap \mathcal{D}_{0,3}) \oplus a$ generated by $p^1 \equiv (z^1, w^1)$ and $p^2 \equiv (z^2, w^2)$. Let $\tilde{p}^1, \tilde{p}^2 \in \mathcal{C}_0 \oplus a$ be two other plaintexts generated by*

$$\tilde{p}^1 \equiv (z^1, w^1, x, y),\ \tilde{p}^2 \equiv (z^2, w^2, x, y) \quad \textit{or} \quad \tilde{p}^1 \equiv (z^1, w^2, x, y),\ \tilde{p}^2 \equiv (z^1, w^2, x, y)$$

where x and y can take any possible value in $\mathbb{F}_{2^8}$. The following event

$$R^4(p^1) \oplus R^4(p^2) \in \mathcal{M}_J \quad \textit{if and only if} \quad R^4(\tilde{p}^1) \oplus R^4(\tilde{p}^2) \in \mathcal{M}_J$$

holds with prob. 1 for 4-round AES, independently of the secret key, of the details of the S-Box and of the MixColumns matrix (except for the branch number equal to 5).

The proof of this result is equivalent to the one proposed in Sect. 4.1.1. In particular, let $q^1 = SR(p^1)$ and $q^2 = SR(p^2)$ as before. *If a column of q^1 is equal to the corresponding column of q^2*, it follows that *the difference super-Sbox$(q^1) \oplus$ super-Sbox(q^2) is independent of the value of such column.* As a result, the difference $R^2(p^1) \oplus R^2(p^2)$ is independent of the generating variables which are equal for p^1 and p^2. It follows that

$$\begin{aligned} R^2(p^1) \oplus R^2(p^2) = R^2(\tilde{p}^1 \equiv (z^1, w^1, x, y)) \oplus R(\tilde{p}^2 \equiv (z^2, w^2, x, y)) = \\ = R^2(\tilde{p}^1 \equiv (z^1, w^2, x, y)) \oplus R(\tilde{p}^2 \equiv (z^2, w^1, x, y)), \end{aligned}$$

which implies the result since $Prob(R^2(x) \oplus R^2(y) \in \mathcal{M}_J \,|\, x \oplus y \in \mathcal{D}_J) = 1$.

Such result is the starting point for new 5-round secret-key distinguisher of AES, as proposed in [Gra17]. We finally emphasize that the previous result is based on Lemma 3 (proposed in [GRR17a]).

Starting Point for Key-Recovery Attack of Sect. 5. As second case, we consider two plaintexts in a coset of $\mathcal{C}_0$ (or more generally $\mathcal{C}_I$ for $|I| = 1$) generated by different generating variables.

[12] A similar difference among the theoretical and the practical cases was present also in [GRR17a].

[13] This is due to the fact that such distinguishers re-exploit the one proposed in [GRR17a], which has the property to work in both directions.

Theorem 4. *Given the subspace* $\mathcal{C}_0 \equiv \langle e_{0,0}, e_{1,0}, e_{2,0}, e_{3,0}\rangle$*, consider two plaintexts* p^1 *and* p^2 *in the same coset* $\mathcal{C}_0 \oplus a$ *generated by* $p^1 \equiv (x^1, y^1, z^1, w^1)$ *and* $p^2 \equiv (x^2, y^2, z^2, w^2)$*. Let* $\tilde{p}^1, \tilde{p}^2 \in \mathcal{C}_0 \oplus a$ *be two other plaintexts generated by*

1. (x^2, y^1, z^1, w^1) *and* (x^1, y^2, z^2, w^2);
2. (x^1, y^2, z^1, w^1) *and* (x^2, y^1, z^2, w^2);
3. (x^1, y^1, z^2, w^1) *and* (x^2, y^2, z^1, w^2);
4. (x^1, y^1, z^1, w^2) *and* (x^2, y^2, z^2, w^1);
5. (x^2, y^2, z^1, w^1) *and* (x^1, y^1, z^2, w^2);
6. (x^2, y^1, z^2, w^1) *and* (x^1, y^2, z^1, w^2);
7. (x^2, y^1, z^1, w^2) *and* (x^1, y^2, z^2, w^1).

The following event

$$R^4(p^1) \oplus R^4(p^2) \in \mathcal{M}_J \quad \textit{if and only if} \quad R^4(\tilde{p}^1) \oplus R^4(\tilde{p}^2) \in \mathcal{M}_J$$

holds with prob. 1 for 4-round AES, independently of the secret key, of the details of the S-Box and of the MixColumns matrix (except for the branch number equal to 5).

The proof of this result is equivalent to the one proposed in Sect. 4.1.1. In particular, let $q^1 = SR(p^1)$ and $q^2 = SR(p^2)$ as before. Since *(1st)* the *super-Sbox*$(\cdot)$ works independently on each column of q^1 and q^2, *(2nd)* the columns of q^1 and q^2 depend on different and independent variables and *(3rd)* the XOR sum is commutative, it follows that

$$\textit{super-Sbox}(q^1) \oplus \textit{super-Sbox}(q^2) = \textit{super-Sbox}(\tilde{q}^1) \oplus \textit{super-Sbox}(\tilde{q}^2)$$

where $\tilde{q}^i = SR(\tilde{p}^i)$ and where $\tilde{p}^i$ are defined as before. Thus, $R^2(p^1) \oplus R^2(p^2) = R^2(\tilde{p}^1) \oplus R^2(\tilde{p}^2)$, which implies the result since $Prob(R^2(x) \oplus R^2(y) \in \mathcal{M}_J \,|\, x \oplus y \in \mathcal{D}_J) = 1$.

Such result is used in Sect. 5 to set up a new key-recovery attack on 5-round AES. We finally emphasize that the previous result is based on Lemma 2 (proposed in [GRR17a]).

4.3 Comparison with Other 4-round Secret-Key Distinguishers

Here we highlight the major differences with respect to the other 4-round AES secret-key distinguishers present in the literature. Omitting the integral one (which exploits a completely different property), we focus on the impossible and the truncated differential distinguishers, on the polytopic cryptanalysis, on the "multiple-of-8" distinguisher (adapted - in a natural way - to the 4-round case) and on the yoyo distinguisher.

Impossible Differential. The *impossible differential distinguisher* is based on Prop. 1, that is it exploits the property that $\mathcal{M}_I \cap \mathcal{D}_J = \{0\}$ for $|I| + |J| \leq 4$. In our case, we consider plaintexts in the same coset of $\mathcal{C}_0 \cap \mathcal{D}_I \subseteq \mathcal{D}_I$ where $|I| \geq 2$ (e.g. $I = \{0, 3\}$) and looks for collisions in $\mathcal{M}_J$ with $|J| = 3$. Since $|I| + |J| \geq 5$, the property exploited by the impossible differential distinguisher cannot be applied here.

Truncated Differential. The *truncated differential distinguisher* has instead some aspects in common with our distinguisher. In this case, given pairs of plaintexts with certain difference on certain bytes (i.e. that belong to the same coset of a subspace $\mathcal{X}$), one considers the probability that the corresponding ciphertexts belong to the same coset of a subspace $\mathcal{Y}$. For 2-round AES it is possible to exploit truncated differential trails with probability 1, while for the 3-round case there exist truncated differential trails with probability lower than 1 but higher than for the random case[14] (in both cases, $\mathcal{X} \equiv \mathcal{D}_I$ and $\mathcal{Y} \equiv \mathcal{M}_J$).

[14] For 3-round AES it holds that $Prob(R^3(x) \oplus R^3(y) \in \mathcal{M}_J \,|\, x \oplus y \in \mathcal{D}_I) = \left(2^{8\cdot|I|}\right)^{-4+|J|}$, while for a random permutation $\Pi(\cdot)$ it holds that $Prob(\Pi(x) \oplus \Pi(y) \in \mathcal{M}_J \,|\, x \oplus y \in \mathcal{D}_I) = \left(2^{32}\right)^{-4+|J|}$.

Our distinguisher works in a similar way and exploits a similar property. However, instead of working with a single couple of texts independently of the others, in our distinguisher one basically considers sets of 2 "non-independent" couples of texts and exploits the relationships that hold among the couples of texts that belong to the same set.

Polytopic Cryptanalysis. *Polytopic cryptanalysis* [Tie16] has been introduced by Tiessen at Eurocrypt 2016, and it can be viewed as a generalization of standard differential cryptanalysis. Consider a set of $d \geq 2$ couples of plaintexts $(p^0, p^0 \oplus \alpha^1), (p^0, p^0 \oplus \alpha^2), ...(p^0, p^0 \oplus \alpha^d)$ with one plaintext in common (namely p^0), called d-poly. The idea of polytopic cryptanalysis is to exploit the probability that the input set of differences $\boldsymbol{\alpha} \equiv (\alpha^1, \alpha^2, ..., \alpha^d)$ is mapped into an output set of differences $\boldsymbol{\beta} \equiv (\beta^1, \beta^2, ..., \beta^d)$ after r rounds. If this probability[15] - *which depends on the S-Box details* - is different from the corresponding probability in the case of a random permutation, it is possible to set up distinguishers or key-recovery attacks. Impossible polytopic cryptanalysis focuses on the case in which the probability of the previous event is zero. In [Tie16], an impossible 8-polytopic is proposed for 2-round AES, which allows to set up key-recovery attacks on 4- and 5-round AES.

Our proposed distinguisher works in a similar way, since also in our case we consider sets of "non-independent" couples of texts and we focus on the input/output differences. However, instead of working with a set of couples of plaintexts with one plaintext in common, we consider sets of couples of texts for which particular relationships between the generating variables of the texts hold. Moreover, instead of considering the probability that "generic" input differences $\boldsymbol{\alpha}$ are mapped into output differences $\boldsymbol{\beta}$, the way in which the texts are divided in sets guarantees the two ciphertexts of *all* couples satisfy or not an output (truncated) difference *independently of the S-Box details* (that is, it is not possible that some of them satisfy this output difference and some others not).

"Multiple-of-8" Distinguisher. The *"multiple-of-8" distinguisher* [GRR17a] can be adapted to the 4-round case, e.g. considering plaintexts in the same coset of $\mathcal{C}_J$, counting the number of collisions of the ciphertexts in the same coset of $\mathcal{M}_I$ and checking if it is (or not) a multiple of 8. *Since our distinguisher exploits more information* (that is, the relationships that hold among the generating variables of the couples of plaintexts in the same set, beside the fact that the previous number is a multiple of 8), its data and computational costs are lower than [GRR17a], in particular 2^{17} chosen plaintexts/ciphertexts instead of 2^{33} and approximately 2^{23} table look-ups instead of 2^{40}.

Yoyo Distinguisher. The basic idea exploited by the *yoyo distinguisher* [RBH17] proposed at Asiacrypt 2017 is similar to the one exploited by our distinguisher. Consider 4-round AES, where the initial and the final ShiftRows and the final MixColumns operations are omitted[16]. Given a pair of plaintexts in the same coset of a column space $\mathcal{C}_I$ - that is $p^1, p^2 \in \mathcal{C}_I \oplus a$, consider the corresponding ciphertexts c^1 and c^2 after 4 rounds. In the yoyo game, the idea is to construct a new pair of ciphertexts $\hat{c}^1$ and $\hat{c}^2$ by *swapping the columns* of c^1 and c^2. E.g., if $c^i \equiv (c^i_0, c^i_1, c^i_2, c^i_3)$ for $i = 1, 2$ where c^i_j denotes the j-th column of c^i, one can define the new pair of ciphertexts as $\hat{c}^1 \equiv (c^2_0, c^1_1, c^1_2, c^1_3)$ and $\hat{c}^2 \equiv (c^1_0, c^2_1, c^2_2, c^2_3)$. As proved in [RBH17], the corresponding plaintexts $\hat{p}^1 = R^{-4}(\hat{c}^1)$ and $\hat{p}^2 = R^{-4}(\hat{c}^2)$ belong to the same coset of $\mathcal{C}_I$ with prob.

[15] We mention that the probability of polytopic trails is usually much lower than the probability of trails in differential cryptanalysis, that is simple polytopic cryptanalysis can not in general outperform standard differential cryptanalysis - see Sect. 2 of [Tie16] for details.

[16] The distinguisher works as well also in the case in which these linear operations are not omitted. We refer to [RBH17] for all the details.

1 for 4-round AES (that is, $\hat{p}^1 \oplus \hat{p}^2 \in \mathcal{C}_I$ with prob. 1), while this happens with prob. $2^{-32\cdot(4-|I|)}$ for a random permutation.

Our distinguisher and the yoyo one are very similar. Both ones exploit particular relationships that hold among the generating variables of a pair of texts and particular properties which depend on such relations to distinguish 4-round AES from a random permutation. However, we emphasize that while the yoyo distinguisher requires *adaptive* chosen ciphertexts in order to construct new pairs of texts related to the original one, in our case such new pairs of texts are constructed directly from the chosen plaintexts. In other words, ours distinguisher doesn't require adaptive chosen plaintexts/ciphertexts.

5 New Key-Recovery Attack on 5-round AES

The modified version of the previous 4-round secret-key distinguisher proposed in Sect. 4.2 can be used as starting point to set up a new (practical verified) key-recovery attack on 5-round AES.

W.l.o.g. consider two plaintexts p^1 and p^2 in the same coset of $\mathcal{D}_0$, e.g. $\mathcal{D}_0 \oplus a$ for $a \in \mathcal{D}_0^\perp$, such that $p^i = x^i \cdot e_{0,0} \oplus y^i \cdot e_{1,1} \oplus z^i \cdot e_{2,2} \oplus w^i \cdot e_{3,3} \oplus a$ or equivalently $p^i \equiv (x^i, y^i, z^i, w^i)$. By Theorem 1, there exists $b \in \mathcal{C}_0^\perp$ such that

$$R(p^i) = \begin{bmatrix} \hat{x}^i & 0 & 0 & 0 \\ \hat{y}^i & 0 & 0 & 0 \\ \hat{z}^i & 0 & 0 & 0 \\ \hat{w}^i & 0 & 0 & 0 \end{bmatrix} \oplus b \equiv MC \cdot \begin{bmatrix} \text{S-Box}(x^i \oplus k_{0,0}) & 0 & 0 & 0 \\ \text{S-Box}(y^i \oplus k_{1,1}) & 0 & 0 & 0 \\ \text{S-Box}(z^i \oplus k_{2,2}) & 0 & 0 & 0 \\ \text{S-Box}(w^i \oplus k_{3,3}) & 0 & 0 & 0 \end{bmatrix} \oplus b$$

for $i = 1, 2$, that is

$$R(p^i) \equiv (\hat{x}^i, \hat{y}^i, \hat{z}^i, \hat{w}^i) \equiv \hat{x}^i \cdot e_{0,0} \oplus \hat{y}^i \cdot e_{1,0} \oplus \hat{z}^i \cdot e_{2,0} \oplus \hat{w}^i \cdot e_{3,0} \oplus b.$$

The idea is to filter wrongly guessed keys of the first round by exploiting the previous distinguisher.

In particular, given plaintexts in the same coset of $\mathcal{D}_0$, the idea of the attack is simply to guess 4 bytes of the first diagonal of the secret key k, that is $k_{i,i}$ for each $i \in \{0, 1, 2, 3\}$, to (partially) compute $R_k(p^1)$ and $R_k(p^2)$ and to exploit the following consideration: *if the guessed key is the right one*, then

$$R^4\big[R_k(p^1)\big] \oplus R^4\big[R_k(p^2)\big] \in \mathcal{M}_J$$

if and only if there exist other pairs of texts $R_k(q^1)$ and $R_k(q^2)$ with the same property, that is

$$R^4\big[R_k(q^1)\big] \oplus R^4\big[R_k(q^2)\big] \in \mathcal{M}_J$$

where $R_k(q^1)$ and $R_k(q^2)$ are defined by a different combination of the generating variables of $R_k(p^1)$ and $R_k(p^2)$. If this property is not satisfied and due to the distinguisher just proposed, then it is possible to claim that the guessed key is a wrong candidate. As we are going to show, *this attack works because the variables that define the (other) pairs of texts $R_k(q^1)$ and $R_k(q^2)$ depend on the guessed key* (besides on the texts p^1 and p^2).

Details of the Attack

In the following we give all the details of the attack. As for the distinguisher just presented, consider a pair of texts p^1 and p^2 in the same coset of $\mathcal{D}_0$ such that

- $c^1 \oplus c^2 \equiv R^5(p^1) \oplus R^5(p^2) \in \mathcal{M}_J$ (observe that this condition is independent of the (partially) guessed key);

- $R(p^i) \equiv (\hat{x}^i, \hat{y}^i, \hat{z}^i, \hat{w}^i)$ for $i = 1, 2$ as before, s.t. $\hat{x}^1 \neq \hat{x}^2$, $\hat{y}^1 \neq \hat{y}^2$, $\hat{z}^1 \neq \hat{z}^2$ and $\hat{w}^1 \neq \hat{w}^2$.

For completeness, we emphasize that the attack works even if one or two generating variables of $R(p^1)$ and $R(p^2)$ are equal (e.g. if two generating variables are equal, in the following it is sufficient to exploit Lemma 3 instead of Lemma 2). We limit to discuss the case in which the generating variables are all different *only* for sake of simplicity, and since this is the event that happens with highest probability (the probability that all the generating variables are different is $[(256 \cdot 255)/256^2]^4 = \frac{255^4}{256^4} \simeq 98.45\%$). Due to the definition of $\hat{x}^i, \hat{y}^i, \hat{z}^i, \hat{w}^i$

$$[\hat{x}^i,\, \hat{y}^i\; \hat{z}^i\; \hat{w}^i]^T \equiv MC \cdot [\text{S-Box}(x^i \oplus k_{0,0}),\, \text{S-Box}(y^i \oplus k_{1,1}),\, \text{S-Box}(z^i \oplus k_{2,2}),\, \text{S-Box}(w^i \oplus k_{3,3})]^T,$$

note that the second condition depends on the (partially) guessed key.

Given p^1 and p^2 as before, we have to define $R_k(q^1)$ and $R_k(q^2)$ in order to set up the distinguisher. Using Lemma 2 and the "super-Sbox" argumentation given in Sect. 4.1.1, it is possible to construct 7 different pairs of - intermediate - texts $R_k(q^1)$ and $R_k(q^2)$ in $\mathcal{C}_0 \oplus b$ defined by the following combinations of generating variables

1. $(\hat{x}^2, \hat{y}^1, \hat{z}^1, \hat{w}^1)$ and $(\hat{x}^1, \hat{y}^2, \hat{z}^2, \hat{w}^2)$;
2. $(\hat{x}^1, \hat{y}^2, \hat{z}^1, \hat{w}^1)$ and $(\hat{x}^2, \hat{y}^1, \hat{z}^2, \hat{w}^2)$;
3. $(\hat{x}^1, \hat{y}^1, \hat{z}^2, \hat{w}^1)$ and $(\hat{x}^2, \hat{y}^2, \hat{z}^1, \hat{w}^2)$;
4. $(\hat{x}^1, \hat{y}^1, \hat{z}^1, \hat{w}^2)$ and $(\hat{x}^2, \hat{y}^2, \hat{z}^2, \hat{w}^1)$;
5. $(\hat{x}^2, \hat{y}^2, \hat{z}^1, \hat{w}^1)$ and $(\hat{x}^1, \hat{y}^1, \hat{z}^2, \hat{w}^2)$;
6. $(\hat{x}^2, \hat{y}^1, \hat{z}^2, \hat{w}^1)$ and $(\hat{x}^1, \hat{y}^2, \hat{z}^1, \hat{w}^2)$;
7. $(\hat{x}^2, \hat{y}^1, \hat{z}^1, \hat{w}^2)$ and $(\hat{x}^1, \hat{y}^2, \hat{z}^2, \hat{w}^1)$

that must satisfy the required property

$$R^4\big[R_k(p^1)\big] \oplus R^4\big[R_k(p^2)\big] \in \mathcal{M}_J \qquad \textit{iff} \qquad R^4\big[R_k(q^1)\big] \oplus R^4\big[R_k(q^2)\big] \in \mathcal{M}_J.$$

Using this observation, it is possible to filter all the wrong keys. Again, since $R^5(p^1) \oplus R^5(p^2) \in \mathcal{M}_J$, all these pairs of - intermediate - texts $(R_k(q^1), R_k(q^2))$ must belong to the same coset of $\mathcal{M}_J$ after 4 rounds if the guessed key is the right one. If this property is not satisfied, then one can simply deduce that the guessed key is wrong (for a wrong guessed key, the behavior is similar to the one of a random permutation).

Why does the attack work? Wrong-Key Randomization Hypothesis! One of the assumption required by the proposed attack is the "wrong-key randomization hypothesis". This hypothesis states that when decrypting one or several rounds with a wrong key guess creates a function that behaves like a random function. For our setting, we formulate it as following:

Wrong-key randomization hypothesis. When the pairs of - intermediate - texts $R_k(q^1)$ and $R_k(q^2)$ are generated using a wrongly guessed key, the probability that the resulting pairs of ciphertexts satisfy the required property is equal to the probability given for the case of a random permutation.

In the following we show that such assumption holds. The crucial point is that *the new pairs of texts $R_k(q^1)$ and $R_k(q^2)$ (and the way in which they are constructed) depend on the guessed key.*

In the proposed attack, the wrong-key randomization hypothesis follows immediately from the definition of the generating variables and from the fact that the S-Box is a non-linear operation. To have more evidence of this fact, let k be the secret key and $\tilde{k}$ be a guessed key. Given $R_k(p^1) \equiv (x^1, y^1, z^1, w^1)$ and $R_k(p^2) \equiv (x^2, y^2, z^2, w^2)$ in $\mathcal{C}_0 \oplus b$ as before, the generating variables of $R_{\tilde{k}}(q^1) \equiv (\tilde{x}^1, \tilde{y}^1, \tilde{z}^1, \tilde{w}^1)$ and $R_{\tilde{k}}(q^2) \equiv (\tilde{x}^2, \tilde{y}^2, \tilde{z}^2, \tilde{w}^2)$

```
Data: 1 coset of $\mathcal{D}_0$ (e.g. $\mathcal{D}_0 \oplus a$ for $a \in \mathcal{D}_0^{\perp}$) and corresponding ciphertexts after
      5 rounds - more generally a coset of $\mathcal{D}_i$ for $i \in \{0, 1, 2, 3\}$
Result: 4 bytes of the secret key - $(k_{0,0}, k_{1,1}, k_{2,2}, k_{3,3})$
let $(p^i, c^i)$ for $i = 0, ..., 2^{32} - 1$ be the $2^{32}$ (plaintexts, ciphertexts) of $\mathcal{D}_0 \oplus a$;
do
    find indexes $j$ and $h$ s.t. $c^j \oplus c^h \in \mathcal{M}_I$;
    for each one of the $2^{32}$ combinations of $\hat{k} = (k_{0,0}, k_{1,1}, k_{2,2}, k_{3,3})$ do
        (partially) compute $R_{\hat{k}}(p^j)$ and $R_{\hat{k}}(p^h)$;
        $flag \leftarrow 0$;
        for each couple $(q^1, R^5(q^1))$ and $(q^2, R^5(q^2))$ where $R_{\hat{k}}(q^1)$ and $R_{\hat{k}}(q^2)$ are
          constructed by a different combination of the generating variables of $R_{\hat{k}}(p^j)$
          and $R_{\hat{k}}(p^h)$ do
            if $R^5(q^1) \oplus R^5(q^2) \notin \mathcal{M}_I$ then
                $flag \leftarrow 1$;
                next combination of $(k_{0,0}, k_{1,1}, k_{2,2}, k_{3,3})$;
            end
        end
        if $flag = 0$ then
            identify $(k_{0,0}, k_{1,1}, k_{2,2}, k_{3,3})$ as candidate of the key;
        end
    end
while more than a single candidate of the key is found - Repeat the procedure for
  different indexes $j, h$ (and $I$) // usually not necessary - only one candidate is found;
return $(k_{0,0}, k_{1,1}, k_{2,2}, k_{3,3})$
```

Algorithm 2: *5-round AES Key-Recovery Attack.* The attack exploits the 4-round distinguisher presented in Sect. 4.2. For sake of simplicity, in this pseudo-code we limit to describe the attack of 4 bytes - 1 diagonal of the secret key (the same attack can be used to recover the entire key).

in $\mathcal{C}_0 \oplus b$ are given by

$$\begin{bmatrix} \tilde{x}^i \\ \tilde{y}^i \\ \tilde{z}^i \\ \tilde{w}^i \end{bmatrix} = MC \circ \text{S-Box} \circ \left(\begin{bmatrix} \tilde{k}_{0,0} \oplus k_{0,0} \\ \tilde{k}_{1,1} \oplus k_{1,1} \\ \tilde{k}_{2,2} \oplus k_{2,2} \\ \tilde{k}_{3,3} \oplus k_{3,3} \end{bmatrix} \oplus \text{S-Box}^{-1} \circ MC^{-1} \circ \begin{bmatrix} x^h \\ y^j \\ z^k \\ w^l \end{bmatrix} \right)$$

for certain $h, j, k, l \in \{1, 2\}$. For a wrongly guessed key $\tilde{k} \neq k$, the relations among the generating variables $[\tilde{x}^i, \tilde{y}^i, \tilde{z}^i, \tilde{w}^i] = [x^h, y^j, z^k, w^l]$ do *not* hold[17]. It follows that if $k \neq \tilde{k}$, then the attacker is considering *random* pairs of texts, which implies that the required property is - in general - not satisfied (as for the case of a random permutation).

Before going on, we emphasize that this result also implies the *impossibility to set up a 5-round distinguisher similar to the one just presented in this section* choosing plaintexts in the same coset of a diagonal space $\mathcal{D}_I$ instead of a column space $\mathcal{C}_I$. Indeed, given p^1 and p^2 as before in the same coset of $\mathcal{D}_I$ (instead of $\mathcal{C}_I$), since the key k is secret and the S-Box is non-linear, *there is no way to find $\hat{p}^1$ and $\hat{p}^2$ in the coset of $\mathcal{D}_I$ s.t. $R^5(p^1) \oplus R^5(p^2) \in \mathcal{M}_J$ if and only if $R^5(\hat{p}^1) \oplus R^5(\hat{p}^2) \in \mathcal{M}_J$ without guessing the secret key.*

[17] Note that if $k = \tilde{k}$, then $\tilde{x}^i = x^h$, $\tilde{y}^i = y^j$, $\tilde{z}^i = z^k$ and $\tilde{w}^i = w^l$ (which implies that the required property is satisfied) as expected.

5.1 Data and Computational Costs

Data Cost. First of all, since the cardinality of a coset of $\mathcal{D}_I$ for $|I| = 1$ is 2^{32} and since $Prob(t \in \mathcal{M}_J) = 4 \cdot 2^{-32} = 2^{-30}$ for $|J| = 3$, the average number of collisions for each coset of $\mathcal{D}_I$ is approximately $2^{-30} \cdot \binom{2^{32}}{2} \simeq 2^{-30} \cdot 2^{63} \simeq 2^{33}$, so it's very likely that two (plaintexts, ciphertexts) pairs (p^1, c^1) and (p^2, c^2) exist such that $c^1 \oplus c^2 \in \mathcal{M}_J$ and for which the two plaintexts have different generating variables.

Given a couple of plaintexts p^1 and p^2 for which the corresponding ciphertext c^1 and c^2 belong to the same coset of $\mathcal{M}_J$, consider the other 7 couples of plaintexts q^1 and q^2 defined as before (that is, such that $R(q^1)$ and $R(q^2)$ are defined by a different combination of the generating variables of $R(p^1)$ and $R(p^2)$). For a wrong key, the probability that the two ciphertexts of each one of the other 7 couples belong to the same coset of $\mathcal{M}_J$ for a fixed J (that is, the probability that a wrong key passes the test) is $(2^{-32})^7 = 2^{-224}$.

Since there are $2^{32} - 1$ wrong candidates for the diagonal of the key, the probability that at least one of them passes the test is approximately $1 - (1 - 2^{-224})^{2^{32}-1} \simeq 2^{-192}$. Thus, one couple of plaintexts p^1 and p^2 (for which the corresponding ciphertexts belong to the same coset of $\mathcal{M}_J$) together with the corresponding other 7 couples of texts q^1 and q^2 are (largely) sufficient to discard all the wrong candidates for a diagonal of the key. Actually, in general only two different couples q^1 and q^2 (that is, two couples of texts given by two different combinations of the generating variables) are sufficient to discard all the wrong candidates, so it is not necessary to consider all the 7 pairs of texts q^1 and q^2. Indeed, given two couples, the probability that at least one wrong key passes the test is approximately $1 - (1 - 2^{-32 \cdot 2})^{2^{32}-1} \simeq 2^{-32} \ll 1$, which means that all the wrong candidates are discarded with high probability.

As a result, the attack requires $2^{33.6}$ chosen plaintexts.

Computational Cost. Each coset of $\mathcal{D}_I$ with $|I| = 1$ is composed of 2^{32} texts, thus on average $2^{63} \cdot 2^{-32} = 2^{31}$ different pairs of ciphertexts belong to the same coset of $\mathcal{M}_J$ for a fixed J with $|J| = 3$. However, it is sufficient to find one collision in order to implement the attack and to find the key.

In order to find it, the best strategy is to re-order the ciphertexts with respect to the partial order $\preceq$ and then to work on consecutive elements, as done in Sect. 4.1.2. For each initial coset of $\mathcal{D}_I$ and for a fixed J, the cost to re-order the ciphertexts with respect to the partial order $\preceq$ (for $\mathcal{M}_J$ with J fixed - $|J| = 3$) and to find a collision is approximately of $2^{32} \cdot (\log 2^{32} + 1) = 2^{37}$ table look-ups.

When such a collision is found, one has to guess 4 bytes of the key and to construct - at least - two other different couples given by a different combination of the generating variables of $R(p^1)$ and $R(p^2)$ (observe that the condition $\hat{x}^1 \neq \hat{x}^2$, $\hat{y}^1 \neq \hat{y}^2$, $\hat{z}^1 \neq \hat{z}^2$ and $\hat{w}^1 \neq \hat{w}^2$ is satisfied with probability $(255/256)^4 \approx 1$). In order to perform this step efficiently, the idea is to re-order - and to store separately *a second copy of* - the (plaintexts, ciphertexts) pairs w.r.t. the partial order $\leq$ as defined in Def. 6 s.t. $p^i \leq p^{i+1}$ for each i. Using the same strategy proposed for the 4-round distinguisher (see App. B for all details), this allows to construct these two new different couples (and to check if the corresponding ciphertexts satisfy or not the required property) with only 4 table look-ups. As a result, the cost of this step is of $2^{32} \cdot 2 \cdot 4 = 2^{35}$ S-Box and of $2^{32} \cdot 4 = 2^{34}$ table look-ups.

It follows that the cost to find one diagonal of the key is well approximated by 2^{35} S-Box look-ups and $2^{37.17}$ table look-ups, that is approximately $2^{30.95}$ five-round encryptions. The idea is to use this approach for three different diagonals, and to find the last one by brute force. As a result, the total computational cost is of $2^{32} + 3 \cdot 2^{30.95} = 2^{33.28}$ five-round encryptions, while the data cost is of $3 \cdot 2^{32} = 2^{33.6}$ chosen plaintexts.

Summary. As a result, the attack - practical verified on a small scale AES - requires $2^{33.6}$ chosen plaintexts and has a computational cost of $2^{33.28}$ five-round encryptions. The pseudo-code of the attack is given in Algorithm 2. We remark for completeness that the same attack works also in the decryption/reverse direction, using chosen ciphertexts instead of plaintexts.

5.2 Practical Verification

Using a C/C++ implementation, we have practically verified the attack just described[18] on the small scale AES [CMR05]. We emphasize that since the proposed attack is independent of the fact that each word of AES is composed of 4 or 8 bits, our verification on the small scale variant of AES is strong evidence for it to hold for the real AES.

Practical Results. For simplicity, we limit to report the result for a single diagonal of the key. First of all, a single coset of a diagonal space $\mathcal{D}_i$ is largely sufficient to find one diagonal of the key. In more detail, given two (plaintexts, ciphertexts) pairs (p^1, c^1) and (p^2, c^2), then other two different couples q^1 and q^2 (out of the seven possible ones) are sufficient to discard all the wrong candidates of the diagonal of the key, as predicted.

About the computational cost, the theoretical cost for the small scale AES case is well approximated by $4 \cdot 2^{16} \cdot (\log 2^{16} + 1) + 2^{16} \cdot 4 = 2^{21}$ table look-ups and $2^{16} \cdot 4 \cdot 3 = 2^{19.6}$ S-Box look-ups, for a total of $2^{19.6} + 2^{21} = 2^{21.5}$ table look-ups (assuming that the cost of 1 S-Box look-up is approximately equal to the cost of 1 table look-up). The average practical computational cost is of $2^{21.5}$ table look-ups, approximately the same as the theoretical one.

Acknowledgements

The author thanks Christian Rechberger for fruitful discussions and comments that helped to improve the quality of the paper. The author also thanks the anonymous reviewers and Anne Canteaut for their valuable comments and suggestions.

References

[BBS99] Eli Biham, Alex Biryukov, and Adi Shamir. Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. In *Advances in Cryptology — EUROCRYPT 1999*, volume 1592 of *LNCS*, pages 12–23, 1999.

[BEK16] Asli Bay, Oguzhan Ersoy, and Ferhat Karakoç. Universal Forgery and Key Recovery Attacks on ELmD Authenticated Encryption Algorithm. In *Advances in Cryptology - ASIACRYPT 2016*, volume 10031 of *LNCS*, pages 354–368, 2016.

[BG11] Céline Blondeau and Benoît Gérard. Multiple Differential Cryptanalysis: Theory and Practice. In *Fast Software Encryption - FSE 2011*, volume 6733 of *LNCS*, 2011.

[BK01] Eli Biham and Nathan Keller. Cryptanalysis of Reduced Variants of Rijndael. unpublished, 2001. http://csrc.nist.gov/archive/aes/round2/conf3/papers/35-ebiham.pdf.

[BLN17] Céline Blondeau, Gregor Leander, and Kaisa Nyberg. Differential-Linear Cryptanalysis Revisited. *Journal of Cryptology*, 30(3):859–888, 2017.

[18] The source codes of the distinguishers/attacks are available at https://github.com/Krypto-iaik/Attacks_AES

[BODK$^+$18] Achiya Bar-On, Orr Dunkelman, Nathan Keller, Eyal Ronen, and Adi Shamir. Improved Key-Recovery Attacks on AES with Practical Data and Memory Complexities, 2018. Accepted at CRYPTO 2018.

[BS90] Eli Biham and Adi Shamir. Differential Cryptanalysis of DES-like Cryptosystems. In *Advances in Cryptology - CRYPTO 1990*, volume 537 of *LNCS*, pages 2–21. Springer, 1990.

[BS91] Eli Biham and Adi Shamir. Differential Cryptanalysis of DES-like Cryptosystems. *Journal of Cryptology*, 4(1):3–72, 1991.

[CAE] CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness. http://competitions.cr.yp.to/caesar.html.

[CMR05] Carlos Cid, Sean Murphy, and Matthew J. B. Robshaw. Small Scale Variants of the AES. In *Fast Software Encryption - FSE 2005*, volume 3557 of *LNCS*, pages 145–162, 2005.

[Der13] Patrick Derbez. Meet-in-the-middle attacks on AES. PhD Thesis, Ecole Normale Supérieure de Paris - ENS Paris, 2013. https://tel.archives-ouvertes.fr/tel-00918146.

[DKR97] Joan Daemen, Lars R. Knudsen, and Vincent Rijmen. The Block Cipher Square. In *Fast Software Encryption - FSE 1997*, volume 1267 of *LNCS*, pages 149–165, 1997.

[DN] Nilanjan Datta and Mridul Nandi. ELmD. https://competitions.cr.yp.to/round1/elmdv10.pdf.

[DR02] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.

[DR06] Joan Daemen and Vincent Rijmen. Understanding Two-Round Differentials in AES. In *Security and Cryptography for Networks - SCN 2006:, 5th International Conference, Italy. Proceedings*, volume 4116 of *LNCS*, pages 78–94, 2006.

[GM16] Shay Gueron and Nicky Mouha. Simpira v2: A Family of Efficient Permutations Using the AES Round Function. In *Advances in Cryptology - ASIACRYPT 2016*, volume 10031 of *LNCS*, pages 95–125, 2016.

[GR17] Lorenzo Grassi and Christian Rechberger. New and Old Limits for AES Known-Key Distinguishers. Cryptology ePrint Archive, Report 2017/255, 2017. https://eprint.iacr.org/2017/255.

[Gra17] Lorenzo Grassi. Mixture Differential Cryptanalysis and Structural Truncated Differential Attacks on round-reduced AES. Cryptology ePrint Archive, Report 2017/832, 2017. https://eprint.iacr.org/2017/832.

[GRR17a] Lorenzo Grassi, Christian Rechberger, and Sondre Rønjom. A New Structural-Differential Property of 5-Round AES. In *Advances in Cryptology - EUROCRYPT 2017*, volume 10211 of *LNCS*, pages 289–317. Springer, 2017.

[GRR17b] Lorenzo Grassi, Christian Rechberger, and Sondre Rønjom. Subspace Trail Cryptanalysis and its Applications to AES. *IACR Transactions on Symmetric Cryptology*, 2016(2):192–225, 2017.

[Knu95] Lars R. Knudsen. Truncated and higher order differentials. In *Fast Software Encryption - FSE 1994*, volume 1008 of *LNCS*, pages 196–211, 1995.

[Knu98] Lars Ramkilde Knudsen. DEAL - a 128-bit block cipher. Technical Report 151, Department of Informatics, University of Bergen, Norway, Feb. 1998.

[LH94] Susan K. Langford and Martin E. Hellman. Differential-Linear Cryptanalysis. In *Advances in Cryptology - CRYPTO 1994*, volume 839 of *LNCS*, pages 17–25, 1994.

[Mur11] Sean Murphy. The Return of the Cryptographic Boomerang. *IEEE Trans. Information Theory*, 57(4):2517–2521, 2011.

[RBH17] Sondre Rønjom, Navid Ghaedi Bardeh, and Tor Helleseth. Yoyo Tricks with AES. In *Advances in Cryptology - ASIACRYPT 2017*, volume 10624 of *LNCS*, pages 217–243, 2017.

[Tie16] Tyge Tiessen. Polytopic Cryptanalysis. In *Advances in Cryptology - EUROCRYPT 2016*, volume 9665 of *LNCS*, pages 214–239, 2016.

[Tun12] Michael Tunstall. Improved “Partial Sums"-based Square Attack on AES. In *International Conference on Security and Cryptography - SECRYPT 2012*, volume 4817 of *LNCS*, pages 25–34, 2012.

[Wag99] David A. Wagner. The Boomerang Attack. In *Fast Software Encryption - FSE 1999*, volume 1636 of *LNCS*, pages 156–170, 1999.

[WP] Hongjun Wu and Bart Preneel. A Fast Authenticated Encryption Algorithm. http://competitions.cr.yp.to/round1/aegisv11.pdf.

A How to Compute the Number of Pairs with n Equal Generating Variables

Given texts in the same cosets of $\mathcal{C}_I$ (and similar for $\mathcal{M}_I$) for $I \subseteq \{0,1,2,3\}$, the number of couples of texts with n equal “generating variable(s) in $(\mathbb{F}_{2^8})^{|I|}$” for $0 \le n \le 3$ is given by

$$\binom{4}{n} \cdot 2^{32\cdot|I|-1} \cdot (2^{8\cdot|I|} - 1)^{4-n}.$$

Here we prove this result.

W.l.o.g. consider for simplicity the case $|I| = 1$. First of all, note that there are $\binom{4}{n}$ different combinations of $n \le 4$ variables. If $n \ge 1$, the n variables that must be equal for the two texts of the couple can take $(2^8)^n$ different values. For each one of the remaining $4-n$ variables, the variables must be different for the two texts of each couple. Thus, these $4-n$ variables can take exactly $\left[(2^8)^{4-n} \cdot (2^8-1)^{4-n}\right]/2$ different values. The result follows immediately. In particular, for $|I| = 1$ there are:

- $2^{63} \cdot (2^8 - 1)^4$ couples for which the two texts have different generating variables;
- $2^{33} \cdot (2^8 - 1)^3$ couples for which the two texts have one equal generating variable;
- $3 \cdot 2^{32} \cdot (2^8 - 1)^2$ couples for which the two texts have two equal generating variables;
- $2^{33} \cdot (2^8 - 1)$ couples for which the two texts have three equal generating variables.

Note that the total number of all the possible couples is $2^{31} \cdot (2^{32} - 1)$.

The formula for the other cases is obtained in an analogous way.

Remark. The proposed formula is used in the context of the 5-round distinguisher presented in [GRR17a], and for the distinguishers proposed in this paper. As explained in Sect. 3, in these distinguishers we work with "generating variables" in $(\mathbb{F}_{2^8})^{|I|}$. If $|I| = 1$, then this corresponds to work independently on each variable. In the other cases, this means to work with *sets of variables* in $(\mathbb{F}_{2^8})^{|I|}$ for $|I| \geq 2$.

For example, given $p^1, p^2 \in \mathcal{C}_{0,1} \oplus a$ s.t. $p^i = a \oplus \bigoplus_{j=0}^{3} \bigoplus_{k=0}^{1} p^i_{j,k} \cdot e_{j,k}$, one works with the following sets of variables: $(p^i_{0,0}, p^i_{1,1}), (p^i_{1,0}, p^i_{2,1}), (p^i_{2,0}, p^i_{3,1}), (p^i_{3,0}, p^i_{0,1})$ (and not independently on each variable).

As a result, *this formula doesn't apply if one works independently on each generating variable also in the cases* $|I| \geq 2$, that is with generating variables in $\mathbb{F}_{2^8}$ also for $|I| \geq 2$. In this last case, the required formula becomes

$$\binom{4}{n} \cdot 2^{32 \cdot |I| - 1} \cdot (2^8 - 1)^{(4-n) \cdot |I|}.$$

(note that the two formulæ are identical only for $|I| = 1$).

B 4-round Secret-Key Distinguisher for AES - Details

In this section, we give all the details about the computational cost of the 4-round secret-key distinguisher for AES presented in Sect. 4. We refer to Sect. 4 for all the details about the distinguisher.

Given 2^{16} chosen plaintexts in the same coset of $(\mathcal{C}_0 \cap \mathcal{D}_{0,3}) \oplus a$ and the corresponding ciphertexts, a first possibility is to construct all the possible pairs, to divide them in sets S of *non-independent* pairs defined as

$$S = \left\{ (p^1, p^2), (\hat{p}^1, \hat{p}^2) \in \big(\mathcal{C}_0 \cap \mathcal{D}_{0,3} \oplus a\big)^2 \,\middle|\, \big[\big(p^1 \equiv (x^1, x^2), c^1 = R^4(p^1)\big), \big(p^2 \equiv (y^1, y^2), c^2 = R^4(p^1)\big)\big]; \big[\big(\hat{p}^1 \equiv (y^1, x^2), \hat{c}^1 = R^4(\hat{p}^1)\big), \big(\hat{p}^2 \equiv (x^1, y^2), \hat{c}^2 = R^4(\hat{p}^2)\big)\big] \right\},$$

where $\big(\mathcal{C}_0 \cap \mathcal{D}_{0,3} \oplus a\big)^2 \equiv ((\mathcal{C}_0 \cap \mathcal{D}_{0,3}) \oplus a) \times ((\mathcal{C}_0 \cap \mathcal{D}_{0,3}) \oplus a)$, and to check for each set if the required property is satisfied (or not).

The cost to check if the property

$$c^1 \oplus c^2 \in \mathcal{M}_J \qquad \textit{if and only if} \qquad \hat{c}^1 \oplus \hat{c}^2 \in \mathcal{M}_J$$

is satisfied (or not) is equal to 2 XOR and 2 MixColumns operations[19], which is negligible with respect to the total cost to construct all the couples of two pairs of texts. For this reason, we focus on the cost to construct the sets S. Using the previous strategy, since the number of pairs is approximately 2^{31} for each coset, the cost is of approximately $2 \cdot 2^{31} = 2^{32}$ table look-ups.

In order to reduce the computational cost, a possibility is to re-order the ciphertexts with respect to a partial order $\preceq$ as defined in Def. 8 (see also [GRR17a]). Note that $\preceq$ depends on an index J. Using a merge-sort algorithm, the cost to re-order n texts is of $O(n \cdot \log n)$ table look-ups. When the ciphertexts have been re-ordered, it is no more necessary to construct all the possible pairs and it is sufficient to work only on consecutive texts with respect to $\preceq$.

[19] Given x, y, then $x \oplus y \in \mathcal{M}_I$ if and only if $MC^{-1}(x \oplus y) \in \mathcal{ID}_I$ for each I.

In more detail, first one stores all the plaintext/ciphertext pairs twice, (1) once in which the plaintexts are ordered with respect to the partial order $\leq$ defined in Def. 6 and (2) once in which the ciphertexts are ordered with respect to the partial order $\preceq$ defined in Def. 8. Then, working on this second set, one focuses only on consecutive ciphertexts c^i and c^{i+1} for each i, and checks if $c^i \oplus c^{i+1} \in \mathcal{M}_J$ or not. Assume that $c^i \oplus c^{i+1} \in \mathcal{M}_J$ for a certain J fixed previously. The idea is to take the corresponding plaintexts $p^i \equiv (x^1, y^1)$ and $p^{i+1} \equiv (x^2, y^2)$, to construct the corresponding set S and to check if the ciphertexts $\hat{c}^1$ and $\hat{c}^2$ of the corresponding plaintexts $\hat{p}^1 \equiv (x^1, y^2)$ and $\hat{p}^2 \equiv (x^2, y^1)$ satisfy the condition $\hat{c}^1 \oplus \hat{c}^2 \in \mathcal{M}_J$ for the same J. If not, by previous observations one can simply deduce that this is a random permutation. Note that if there are r consecutive ciphertexts $c^i, c^{i+1}, ..., c^{i+r-1}$ such that $c^j \oplus c^l \in \mathcal{M}_J$ for $i \leq j, l < i + r$, then one has to repeat the above procedure for all these $\binom{r}{2} = r \cdot (r-1)/2$ possible pairs[20].

To optimize the computational cost, note that the plaintexts $\hat{p}^1$ and $\hat{p}^2$ are respectively in positions $x^1 + 2^8 \cdot y^2$ and $x^2 + 2^8 \cdot y^1$ in the first set of plaintext/ciphertext pairs (i.e. in the set where the plaintexts are ordered with respect to the partial order $\leq$). Thus, the cost to get these two elements is only of 2 table look-ups. Moreover, we emphasize that it is sufficient to work only on (consecutive) ciphertexts c^i and c^j such that $c^i \oplus c^j \in \mathcal{M}_J$. Indeed, consider the case in which the two ciphertexts c^i and c^j don't belong to the same coset of $\mathcal{M}_J$, i.e. $c^i \oplus c^j \notin \mathcal{M}_J$. If the corresponding ciphertexts $\hat{c}^1$ and $\hat{c}^2$ - defined as before - don't belong to the same coset of $\mathcal{M}_J$, then the property is (obviously) verified. Instead if $\hat{c}^1 \oplus \hat{c}^2 \in \mathcal{M}_J$, then this case is surely analyzed. The pseudo-code of such strategy can be found in Algorithm 1.

Using this procedure, the memory cost is well approximated by $4 \cdot 2^{17} \cdot 16 = 2^{23}$ bytes - the same plaintext/ciphertext pairs in two different ways. The cost to order the ciphertexts for each possible J with $|J| = 3$ and for each one of the two cosets is approximately of $2 \cdot 4 \cdot 2^{16} \cdot \log 2^{16} \simeq 2^{23}$ table look-ups, while the cost to construct all the possible pairs of consecutive ciphertexts is of $2 \cdot 4 \cdot 2^{16} = 2^{19}$ table look-ups. Since the probability that a pair of ciphertexts belong to the same coset of $\mathcal{M}_J$ for $|J| = 3$ is 2^{-30} and since each coset contains approximately 2^{31} different pairs, then one has to do on average $2 \cdot 4 \cdot 2^{-30} \cdot 2^{31} = 2^4$ table look-ups in the plaintext/ciphertext pairs ordered with respect to the plaintexts. Thus, the total cost of this distinguisher is well approximated by $2^{23} + 2^{19} + 16 \simeq 2^{23.09}$ table look-ups, or approximately $2^{16.75}$ four-round encryptions (using the approximation 20 table look-ups $\approx$ 1 round of encryption).

C Integral Attack on 5-round AES

Here we recall a strategy that allows to improve the computational cost of an integral attack on 5-round AES. Such a strategy has been proposed by an anonymous reviewer[21].

Integral Attack [DR02] on 5-round AES.

It is well known that given 2^{32} chosen plaintexts with one active diagonal (i.e. a coset of a diagonal space $\mathcal{D}_I \oplus a$), then the XOR-sum of the corresponding texts after four encryption is equal 0 independently of the value of the secret key, that is

$$\bigoplus_{p^i \in \mathcal{D}_I \oplus a} R^4(p^i) = 0$$

for each $I \in \{0, 1, 2, 3\}$.

[20] Since $\mathcal{M}_J$ is a subspace, given a, b, c such that $a \oplus b \in \mathcal{M}_J$ and $b \oplus c \in \mathcal{M}_J$, then $a \oplus c \in \mathcal{M}_J$.

[21] A similar strategy has also been exploited e.g. in [GR17] in order to set up a "known-key" distinguisher for 12-round AES.

Such a property can be used to set up an integral attack on 5-round AES, by guessing - byte per byte - the final key and checking that the XOR-sum is equal to zero one round before. In particular, assume the final MixColumns is omitted, and let $k_{j,l}$ be the guessed byte in row j and column l of the final subkey. Due to the previous property, such guessed byte of the key is certainly *wrong if*

$$\bigoplus_{p^i \in \mathcal{D}_I \oplus a} \text{S-Box}^{-1}(c^i_{j,l} \oplus k_{j,l}) \neq 0 \tag{9}$$

where $c^i \equiv R^5(p^i)$ and $I \in \{0, 1, 2, 3\}$. By simple computation, the attack requires 2^{32} chosen plaintexts and $16 \cdot 2^8 \cdot 2^{32} = 2^{44}$ S-Box operations, that is approximately $2^{37.36}$ 5-round encryptions (assuming 20 S-Box $\equiv$ 1-round encryption).

Note that the probability that a wrong byte key satisfies the zero-sum property is 2^{-8}. As a result, if a wrong key survives the test, then it can simply filtered using a brute force attack. Finally, if the final MixColumns is not omitted, one can simply repeat the previous attack by swapping the final MixColumns operation and the final subkey (we refer to [DR02] for more details) - remember that the MixColumns operation is linear.

Improved Integral Attack on 5-round AES.

Here a way to improve the previous attack is proposed. The crucial point is the following. Working at byte level, note that Eq. (9) is the sum of 2^{32} bytes. Since each byte can take "only" 2^8 different values, it turns out that many bytes of the form S-Box$^{-1}(c^i_{j,l} \oplus k_{j,l})$ in (9) are equal. Thus, the total computational cost can be reduced using a precomputation process, as showed in the following.

```
Data: $2^{32}$ texts $p^i \in \mathcal{D}_I \oplus a$ for $i = 0, ..., 2^{32} - 1$ and for $I \in \{0, 1, 2, 3\}$, and the
      corresponding ciphertexts $c^i \equiv R^5(p^i)$
Result: One byte of $k$ - e.g. $k_{0,0}$ - s.t. $\bigoplus_i$ S-Box$^{-1}(c^i_{0,0} \oplus k_{0,0}) = 0$
Let $A[0, ..., 2^8 - 1]$ an array initialized to zero;
for $i$ from 0 to $2^{32} - 1$ do
    $A[c^i_{0,0}] \leftarrow (A[c^i_{0,0}] + 1)$ mod 2; // $A[x]$ denotes the value stored in the $x$-th address
     of the array $A$
end
for $k$ from 0x00 to 0xff do
    $x \leftarrow 0$;
    for $i$ from 0 to 255 do
        $x \leftarrow x \oplus A[i]\cdot$ S-Box$^{-1}(i \oplus k)$;            // $A[i]$ can only be 0 or 1
    end
    if $x = 0$ then
        identify $k$ as candidate for $k_{0,0}$;
    end
end
return candidates for $k_{0,0}$.
```

Algorithm 3: *Integral attack on 5-round AES:* working on each byte of the key independently of the others, filter wrong key candidates using zero-sum property. Other bytes of the key can be found in a similar way.

The pseudo-code of the attack is given in Algorithm 3. Given 16 tables (one for each byte of the state) of size 2^8 bits, for each ciphertext the idea is to record in each hash table the value of the corresponding byte. The crucial point is that in order to compute the XOR-sum (9), we are interested only in the *parity of the number of times each value appears.* As a result, given a guessed byte of the key, the cost to check if the XOR-sum (9) is zero or not is of 2^8 S-Box operations instead of 2^{32}. It follows that the overall cost is dominated by the cost to prepare the array $A[\cdot]$. It turns out that the attack requires 2^{32} chosen plaintexts, 2^{32} memory look-ups, that is approximately $2^{25.4}$ 5-round encryptions (assuming 20 memory look-ups $\equiv$ 1-round encryption) and $16 \cdot 2^8 = 2^{12}$ bits of memory.