Mixture Integral Attacks on Reduced-Round AES with a Known/Secret S-Box

Lorenzo Grassi and Markus Schofnegger

IAIK, Graz University of Technology, Austria
firstname.lastname@iaik.tugraz.at

Abstract. In this work, we present new low-data secret-key distinguishers and key-recovery attacks on reduced-round AES.
The starting point of our work is "Mixture Differential Cryptanalysis" recently introduced at FSE/ToSC 2019, a way to turn the "multiple-of-8" 5-round AES secret-key distinguisher presented at Eurocrypt 2017 into a simpler and more convenient one (though, on a smaller number of rounds). By reconsidering this result on a smaller number of rounds, we present as our main contribution a new secret-key distinguisher on 3-round AES with the smallest data complexity in the literature (that does not require adaptive chosen plaintexts/ciphertexts), i.e. approximately half of the data necessary to set up a 3-round truncated differential distinguisher (which is currently the distinguisher in the literature with the lowest data complexity). E.g. for a probability of success of 95%, our distinguisher requires just 10 chosen plaintexts versus 20 chosen plaintexts necessary to set up the truncated differential one.
Besides that, we present new competitive low-data key-recovery attacks on 3- and 4-round AES, both in the case in which the S-Box is known and in the case in which it is secret.

Keywords: AES, Mixture Differential Cryptanalysis, Secret-Key Distinguisher, Low-Data Attack, Secret S-Box

1 Introduction

AES (Advanced Encryption Standard) [6] is probably the most used and studied block cipher, and many constructions employ reduced-round AES as part of their design. Determining its security is therefore one of the most important problems in cryptanalysis. Since there is no known attack which can break the full AES significantly faster than via exhaustive search, researchers had concentrated on attacks which can break reduced-round versions of AES. Especially within the last couple of years, new cryptanalysis results on the AES have appeared regularly (e.g., [11,13,9,1]). While those papers do not pose any practical threat 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.

Among many others, a new technique called "Mixture Differential Cryptanalysis" [9] has been recently presented at FSE/ToSC 2019, which is a way to

Table 1. *Secret-key distinguishers on 3-round AES which are independent of the secret key.* The data complexity corresponds to the minimum number of chosen plaintexts/ciphertexts (CP/CC) and/or *adaptive* chosen plaintexts/ciphertexts (ACP/ACC) which are needed to distinguish the AES permutation from a random permutation with a *success probability* denoted by *Prob*. Distinguishers proposed in our work are in bold.

Property	*Prob*	Data	Reference
Imp. Mixt. Integral	≈ 65%	**6 CP**	**Section 4.3**
Trunc. Differential	≈ 65%	12 CP	[10]
Imp. Mixt. Integral	≈ 95%	**10 CP**	**Section 4.3**
Trunc. Differential	≈ 95%	20 CP	[10]
Integral	≈ 100%	$256 = 2^8$ CP	[5,12]
Yoyo	≈ 100%	2 CP + 2 ACC	[13]

translate the (complex) “multiple-of-8” 5-round distinguisher [11] 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 initial pair of plaintexts. As proved in [9], for 4-round AES the corresponding ciphertexts of the initial 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 a secret-key distinguisher – which is also independent of the details of the S-Box and of the MixColumns matrix – has been reconsidered in [4], where authors showed that it is an immediate consequence of an equivalence relation on the input pairs, under which the difference at the output of the round function is invariant. Moreover, it is also the starting point for *practical* and competitive key-recovery attacks on 5-round AES-128 and 7-round AES-192 [1], breaking the record for such attacks which was obtained 18 years ago by the classical Square attack.

In this paper, we reconsider this distinguisher on a smaller number of rounds in order to set up new (competitive) *low-data* distinguishers and key-recovery attacks on reduced-round AES.

Our Contribution and Related Work

Low-data distinguishers/attacks on reduced-round ciphers have recently gained renewed interest in the literature. This is motivated by the following reason.

In one direction, cryptanalysis of block ciphers has focused on maximizing the number of rounds that can be broken without exhausting the full code book and key space. This often leads to attacks marginally close to that of pure brute force. Even if these attacks are very important in order to e.g. determine the security margin of a cipher (that is, the ratio between the number of rounds which can be successfully attacked and the number of rounds in the full cipher), they are obviously not practical.

For this reason, it seems desirable to consider also other approaches, such as restricting the resources available to the adversary in order to adhere to “real-

Table 2. *Attacks on reduced-round AES-128.* The data complexity corresponds to the number of required chosen plaintexts (CP). The time complexity is measured in reduced-round AES encryption equivalents (E), while the memory complexity is measured in plaintexts (16 bytes). Precomputation is given in parentheses. The case in which the final MixColumns operation is omitted is denoted by "r.5 rounds", that is, r full rounds and the final round. "Key sched." highlights whether the attack exploits the details of the key schedule of AES. Attacks proposed in our work are in bold. Note that the details of the S-Box have to be known to the attacker.

Attack	Rounds	Data (CP)	Cost	Memory	Key sched.	Reference
TrD	2.5 - 3	2	$2^{31.6}$	2^{8}	No	[10]
G&D-MitM	2.5	2	2^{24}	2^{16}	Yes	[3]
G&D-MitM	3	2	2^{16}	2^{8}	Yes	[3]
TrD	2.5 - 3	3	$2^{11.2}$	–	No	[10]
G&D-MitM	3	3	2^{8}	2^{8}	Yes	[3]
TrD	2.5 - 3	3	$2^{5.7}$	2^{12}	No	[10]
MixInt	$\mathbf{2.5 - 3}$	**4**	$\mathbf{2^{8.1}}$	–	**No**	**Section 5.1**
MixInt	$\mathbf{2.5 - 3}$	**4**	$\mathbf{< 1}$ $(+2^{36.1})$	$\mathbf{2^{28}}$	**No**	**Section 5.1**
TrD (EE)	3.5 - 4	2	2^{96}	–	Yes	[10]
G&D-MitM	4	2	2^{88}	2^{8}	Yes	[3]
G&D-MitM	4	3	2^{72}	2^{8}	Yes	[3]
TrD (EE)	3.5 - 4	3	$2^{69.7}$	2^{12}	Yes	[10]
G&D-MitM	4	4	2^{32}	2^{24}	Yes	[3]
MixInt	$\mathbf{3.5 - 4}$	**6**	$\mathbf{2^{45.3}}$	–	**No**	**Section 5.2**
MixInt	$\mathbf{3.5 - 4}$	**6**	$\mathbf{2^{33.3}}$ $(+2^{35.7})$	$\mathbf{2^{28}}$	**No**	**Section 5.2**
ImpPol	3.5 - 4	8	2^{38}	2^{15}	No	[15]

G&D: Guess & Det., MitM: Meet-in-the-Middle, TrD: Truncated Differential, ImpPol: Imp. Polytopic, EE: Extension at End, EB: Extension at Beginning.

life" scenarios. In this case, the time complexity of the attack is not restricted (besides the natural bound of exhaustive search), but the data complexity is restricted to only a few known or chosen plaintexts. Attacks in this scenario have been studied explicitly in a number of papers, which include low-data Guess-and-Determine and Meet-in-the-Middle techniques (see [3]), low-data truncated differential cryptanalysis (see [10]), polytopic cryptanalysis (see [15]), and – if adaptive chosen plaintexts/ciphertexts are allowed – yoyo-like attacks (see [13]).

"Mixture Integral" Key-Recovery Attacks. In Sect. 4.2 and Sect. 5 we show that "Mixture Differential Cryptanalysis" [9] can be exploited in order to set up low-data attacks on reduced-round AES. Given a set of chosen plaintexts defined as in [9], our attacks are based on the fact that the XOR sum of the corresponding texts after 2-round AES encryptions is equal to zero with prob. 1. Using the same strategy proposed in a classical square/integral attack [5,12], this zero-sum property can be exploited to set up competitive attacks on 3- and 4-round

AES, which require only 4 and 6 chosen plaintexts, respectively. A comparison of all known low-data attacks on AES and our attacks is given in Table 2. Since *(1)* the pairs of plaintexts used to set up the attacks share the same generating variables – which are mixed in the same way proposed by the Mixture Differential Distinguisher – and since *(2)* such attacks exploit the zero-sum property (instead of a differential one), we call this attack a "Mixture Integral" attack.

"Impossible Mixture Integral" Secret-Key Distinguisher. In Section 4.3, we show that the previous distinguishers/attacks can also be exploited to set up a new 3-round secret-key distinguisher on AES, which is independent of the key, of the details of the S-Box and of the MixColumns operation. For a probability of success of $\approx 95\%$, *such a distinguisher requires only 10 chosen plaintexts (or ciphertexts), that is half of the data required by the most competitive distinguisher currently present in the literature* (which does not require adaptive chosen plaintexts/ciphertexts).

What is the property exploited by this new distinguisher? Consider a zero-sum key-recovery attack on 3-round AES (based on a 2-round zero-sum distinguisher). The assumption of an integral attack is that the zero-sum property is always satisfied when decrypting under the secret key. As a result, if there is no key for which the zero-sum property is satisfied, one can deduce that the ciphertexts have been generated by a random permutation, and not by AES. Such a strategy can be used as a distinguisher, but requires key guessing and is thus not independent of the secret key. In Section 4.3, we show *how to check this property without guessing any key material* by providing a property – which is independent of the secret key – that holds on the ciphertexts only in the case in which the key-recovery (mixture integral) attack just proposed "fails". The obtained 3-round distinguisher – which is independent of the secret key – can also be used to set up new key-recovery attacks on reduced-round AES.

AES with a Single Secret S-Box. Finally, in Appendix B we show that a competitive "Mixture Integral" attack can also be set up on reduced-round AES with a single secret S-Box, i.e., the case in which the AES S-Box is replaced by a secret 8-bit one while keeping everything else unchanged. In the literature, two possible strategies are considered to set up the attack:

Strategy S1: The attacker first determines the secret S-Box up to additive constants (that is, S-Box$(\cdot \oplus a) \oplus b$ for unknown a and b), and then they use this knowledge and apply attacks present in the literature (e.g., the integral one) to derive the whitening key.

Strategy S2: The attacker exploits a particular property of the MixColumns matrix (i.e., the fact that two elements for each row of the matrix are equal) in order to *directly* find the secret key (no information of the secret S-Box is found or used).

Examples for attacks based on the first strategy are given in [16], while examples for attacks based on the second strategy are given in [14,10,8]. In Appendix B we exploit the first strategy in order to set up a competitive attack on 3-round AES

Table 3. *Comparison of attacks on reduced-round AES with secret S-Box.* The data complexity corresponds to the number of required chosen plaintexts/ciphertexts (CP/CC). The time complexity is measured in reduced-round AES encryption equivalents (E), in memory accesses (M), or XOR operations (XOR). The memory complexity is measured in plaintexts (16 bytes). The case in which the final MixColumns operation is omitted is denoted by "r.5 rounds", that is, r full rounds and the final round. New attacks are in bold. Strategy 1 (S1) denotes an attack that requires to find the details of the S-Box, while Strategy 2 (S2) denotes an attack that directly finds the key.

Attack	**Rounds**	**S1**	**S2**	**Data**	**Computation**	**Memory**	**Reference**
MixInt	$\mathbf{2.5-3}$	✓		$2^{11.6}$ CP	2^8 E + $2^{22.6}$ XOR	$2^{10.6}$	**Appendix B**
TrD	2.5 - 3		✓	$2^{13.6}$ CP	$2^{13.2}$ XOR	small	[10]
I	2.5 - 3		✓	$2^{19.6}$ CP	$2^{19.6}$ XOR	small	[10]
I	3.5 - 4	✓		2^{16} CC	$2^{17.7}$ E	2^{16}	[16]
I	3.5 - 4	✓		2^{16} CP	$2^{28.7}$ E	2^{16}	[16, Sect. 3.5]
TrD	3.5 - 4		✓	2^{30} CP	2^{36} M $\approx 2^{29.7}$ E	2^{30}	[10]

TrD: Truncated Differential, I: Integral, ImD: Impossible Differential.

with a single secret S-Box. A comparison of all known attacks on reduced-round AES with a single secret S-Box and our attack is given in Table 3.

Practical Verification. We implemented most of our distinguishers and attacks in practice and could verify the theoretical results. We also implemented a method to find the affine equivalent of a secret S-Box. All the source code files can be found on GitHub[1].

2 Preliminary - Brief Description of AES

The Advanced Encryption Standard [6] is a *Substitution-Permutation network* that supports key sizes of 128, 192, and 256 bits. The 128-bit plaintext initializes the internal state as a 4×4 matrix of bytes as values in the finite field $\mathbb{F}_{2^8}$, 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, where $N_r = 10$ 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 to the left.
- *MixColumns* (MC) – multiplication of each column by a constant 4×4 MDS matrix (SR and MC provide diffusion in the cipher).
- *AddRoundKey* (ARK) – XORing the state with a 128-bit subkey.

[1] `https://github.com/mschof/aes-mixint-analysis`

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.

The Notation Used in this 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 R one round[2] of AES, while we denote r rounds of AES by R^r. Finally, in the paper we often use the term "partial collision" (or "*collision*") when two texts belong to the same coset of a given subspace $\mathcal{X}$. We recall that given a subspace X, the cosets $X \oplus a$ and $X \oplus b$ (where $a \neq b$) are *equal* (that is, $X \oplus a \equiv X \oplus b$) if and only if $a \oplus b \in X$.

3 Subspace Trail Cryptanalysis

The concept of *trails of subspaces* has been introduced in [10] as a generalization of invariant subspace.

Definition 1. *Let F denote a round function in an iterative block cipher and 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 exists a_{i+1} s.t. $F(V_i \oplus a_i) \subseteq V_{i+1} \oplus a_{i+1}$, then $(V_1, V_2, ..., V_{r+1})$ is subspace trail of length r for the function F.*

This means that if F^t denotes the application of t rounds with fixed keys, then $F^t(V_1 \oplus a_1) = V_{t+1} \oplus a_{t+1}$.

Subspace Trails of AES. Here we briefly recall the subspace trails of AES presented in [10] – we refer to Appendix A for more details. In the following, we only work with vectors and vector spaces over $\mathbb{F}_{2^8}^{4\times 4}$, and we denote by $\{e_{0,0}, ..., e_{3,3}\}$ the unit vectors of $\mathbb{F}_{2^8}^{4\times 4}$ (e.g., $e_{i,j}$ has a single 1 in row i and column j).

Definition 2. *For each $i \in \{0, 1, 2, 3\}$:*

- *The column spaces $\mathcal{C}_i$ are defined as $\mathcal{C}_i = \langle e_{0,i}, e_{1,i}, e_{2,i}, e_{3,i}\rangle$.*
- *The diagonal spaces $\mathcal{D}_i$ are defined as $\mathcal{D}_i = SR^{-1}(\mathcal{C}_i)$. Similarly, the inverse-diagonal spaces $\mathcal{ID}_i$ are defined as $\mathcal{ID}_i = SR(\mathcal{C}_i)$.*
- *The i-th mixed spaces $\mathcal{M}_i$ are defined as $\mathcal{M}_i = MC(\mathcal{ID}_i)$.*

Definition 3. *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 [10]:

- For any coset $\mathcal{D}_I \oplus a$ there exists a unique $b \in \mathcal{C}_I^{\perp}$ s.t. $R(\mathcal{D}_I \oplus a) = \mathcal{C}_I \oplus b$.

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

- For any coset $\mathcal{C}_I \oplus a$ there exists a unique $b \in \mathcal{M}_I^{\perp}$ s.t. $R(\mathcal{C}_I \oplus a) = \mathcal{M}_I \oplus b$.

Theorem 1 ([10]). *For each $I \subseteq \{0,1,2,3\}$ and for each $a \in \mathcal{D}_I^{\perp}$, there exists one and only one $b \in \mathcal{M}_I^{\perp}$ s.t. $R^2(\mathcal{D}_I \oplus a) = \mathcal{M}_I \oplus b$.*

Observe that if *(1)* X is a subspace, *(2)* $X \oplus a$ is a coset of X and *(3)* x and y are two elements of the (same) coset $X \oplus a$, then $x \oplus y \in X$. It follows that:

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

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

We remark that all these results can be redescribed using a more "classical" truncated differential notation. For example, if two texts t^1 and t^2 are equal except for the bytes in the i-th diagonal[3] 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 $\mathcal{ID}_I$ if the difference of the bytes that lie in the i-th anti-diagonal for each $i \notin I$ is equal to zero. Similar considerations hold for the column space $\mathcal{C}_I$ and the mixed space $\mathcal{M}_I$.

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

Definition 4 ([9]). *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 $n = 4 \cdot |I|$. 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 $t = a \oplus \bigoplus_{i=0}^{n} t^i \cdot x_i$.*

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

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

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 $p \equiv a \oplus p^0 \cdot e_{0,0} \oplus p^1 \cdot e_{1,0} \oplus p^2 \cdot e_{2,0} \oplus p^3 \cdot e_{3,0}$.

4 Mixture Integral Distinguisher on 2-Round AES

4.1 Mixture Differential Secret-Key Distinguisher

In order to present our result, we recall the "mixture differential distinguisher" [9] on reduced-round AES proposed at FSE/ToSC 2019.

[3] 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 $r + c = i \bmod 4$.

Theorem 2 ([9]). *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 $p^2 \equiv (z^2, w^2)$ (where $z^i, w^i \in \mathbb{F}_{2^8}$ for $i = 1, 2$). Let $\tilde{p}^1, \tilde{p}^2 \in \mathcal{C}_0 \oplus a \equiv \langle e_{0,0}, e_{1,0}, e_{2,0}, e_{3,0}\rangle \oplus a$ be two other plaintexts generated by*

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

where Ψ and Φ can take any possible value in $\mathbb{F}_{2^8}$. Then

$$R^4(p^1) \oplus R^4(p^2) \in \mathcal{M}_J \iff R^4(\tilde{p}^1) \oplus R^4(\tilde{p}^2) \in \mathcal{M}_J$$

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.

For completeness, we mention that such a result has been revisited recently in [4], where authors show that the above property is an immediate consequence of an equivalence relation on the input pairs, under which the difference at the output of the round function is invariant.

Proof Using the "Super-Sbox" Notation. We briefly recall the proof provided in [9] using the "super-Sbox" notation introduced in [7], where

$$\textit{super-Sbox}(\cdot) = \text{S-Box} \circ ARK \circ MC \circ \text{S-Box}(\cdot). \tag{2}$$

In order to prove the result, it is sufficient to show that

$$R^2(p^1) \oplus R^2(p^2) = R^2(\tilde{p}^1) \oplus R^2(\tilde{p}^2). \tag{3}$$

Indeed, due to the fact that $Prob(R^2(x) \oplus R^2(y) \in \mathcal{M}_I \,|\, x \oplus y \in \mathcal{D}_I) = 1$, Theorem 2 follows immediately – we refer to [9] for all details.

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

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

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

$$\textit{super-Sbox}(q^1) \oplus \textit{super-Sbox}(q^2) = \textit{super-Sbox}(\hat{q}^1) \oplus \textit{super-Sbox}(\hat{q}^2), \tag{4}$$

where $q^i = (z^i, w^i)$ and $\hat{q}^i = (z^i, w^{3-i})$, or equivalently

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

for $i \in \{1, 2\}$ (note that $SR(\mathcal{D}_{0,3} \cap \mathcal{C}_0) = \mathcal{C}_{0,3} \cap \mathcal{ID}_0$ by definition). *Since (1) each column of q^1 and q^2 depends on different and independent variables, (2) the super-Sbox works independently on each column, (3) the XOR sum is commutative, and (4) the difference in the second and in the third column is equal to zero independently of the values of Ψ and Φ*, it follows that Equation (4) is satisfied, which implies the thesis.

4.2 Mixture Integral Distinguisher on 2-Round AES

The result proposed in Theorem 2 can be rewritten as a zero-sum (or integral) distinguisher and serves as a starting point for our distinguisher and key-recovery attacks on reduced-round AES.

Lemma 2. *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*

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

where Ψ and Φ can take any possible value in $\mathbb{F}_{2^8}$. Then

$$R^2(p^1) \oplus R^2(p^2) \oplus R^2(\tilde{p}^1) \oplus R^2(\tilde{p}^2) = 0 \tag{5}$$

holds with prob. 1 for 2-round AES, independently of the secret key, of the details of the S-Box, and of the MixColumns matrix.

We highlight that, since the previous event occurs with prob. 2^{-128} if the ciphertexts are generated by a random permutation, it is potentially possible to distinguish 2-round AES from a random permutation by exploiting the previous result[4].

4.3 Impossible Mixture Integral Distinguisher on 3-Round AES

The property just proposed in the previous section is independent of the secret key and of the S-Box, and it can be used to set up a key-recovery attack on reduced-round AES. In particular, consider $p^1, p^2, \tilde{p}^1, \tilde{p}^2$ as in Lemma 2 and the corresponding ciphertexts $c^1 = R^3(p^1), c^2 = R^3(p^2), \tilde{c}^1 = R^3(\tilde{p}^1), \tilde{c}^2 = R^3(\tilde{p}^2)$ after 3-round AES encryptions. Assuming the last MixColumns operation is omitted and since

$$R^2(p^1) \oplus R^2(p^2) \oplus R^2(\tilde{p}^1) \oplus R^2(\tilde{p}^2) = 0,$$

it follows that the secret key k *must satisfy*

$$\begin{aligned} &\text{S-Box}^{-1}(c^1_{j,l} \oplus k_{j,l}) \oplus \text{S-Box}^{-1}(c^2_{j,l} \oplus k_{j,l}) \\ \oplus\ &\text{S-Box}^{-1}(\tilde{c}^1_{j,l} \oplus k_{j,l}) \oplus \text{S-Box}^{-1}(\tilde{c}^2_{j,l} \oplus k_{j,l}) = 0 \end{aligned} \tag{6}$$

for each $j, l = 0, ..., 3$ as for a classical "integral attack" [5,12] (all the details of the attack are given in the next section). The crucial point here is that *there exists at least one key (namely, the secret key) that satisfies the previous equivalence.*

Lemma 3. *Let $\{c^1, c^2, \tilde{c}^1, \tilde{c}^2\}$ be the set of ciphertexts corresponding to the 3-round encryptions of $p^1, p^2, \tilde{p}^1, \tilde{p}^2$. With prob. 1 there exists (at least) one key k that satisfies Equation* (6)*.*

[4] However, note that a truncated differential distinguisher is more competitive, since it requires only 2 chosen plaintexts instead of 4.

Proof. Let $[R^2(p)]_{i,j}$ be the byte in position (i,j) of the 2-round encryption of p. Let $\hat{k}$ be the secret key, and let k be the guessed key. Due to Lemma 2, we know that

$$\begin{aligned} &\text{S-Box}^{-1}\Big[\text{S-Box}\big([R^2(p^1)]_{j,l}\oplus \hat{k}_{j,l}\big)\oplus k_{i,j}\Big]\oplus \text{S-Box}^{-1}\Big[\text{S-Box}\big([R^2(p^2)]_{j,l}\oplus \hat{k}_{j,l}\big)\oplus k_{i,j}\Big]\\ \oplus\, &\text{S-Box}^{-1}\Big[\text{S-Box}\big([R^2(\tilde{p}^1)]_{j,l}\oplus \hat{k}_{j,l}\big)\oplus k_{i,j}\Big]\oplus \text{S-Box}^{-1}\Big[\text{S-Box}\big([R^2(\tilde{p}^2)]_{j,l}\oplus \hat{k}_{j,l}\big)\oplus k_{i,j}\Big]=0, \end{aligned}$$

where

$$c^1 = R^3(p^1) \equiv \text{S-Box}\big(R^2(p^1)\oplus \hat{k}\big)$$

and similarly for the other texts. Due to Lemma 2, the equality is always satisfied for $\hat{k}_{j,l} = k_{i,j}$, which means that there exists at least one key that satisfies Equation (6). □

What happens if there is no key for which the previous condition is satisfied? It turns out that *if this is the case, then the set of ciphertexts $\{c^1, c^2, \tilde{c}^1, \tilde{c}^2\}$ is not generated by 3-round AES, but by a random permutation.* That is, if there is no key $k_{j,l}$ that satisfies Equation (6), then the ciphertexts $\{c^1, c^2, \tilde{c}^1, \tilde{c}^2\}$ are not the 3-round AES encryptions of $p^1, p^2, \tilde{p}^1, \tilde{p}^2$, but they are generated by a random permutation.

However, if we want to set up a distinguisher which is independent of the secret key, we need a way to check this property without checking the existence of a key. So the problem is to rewrite this property in order to avoid key guessing. To solve this issue, the idea is to look for values of $\{c^1, c^2, \tilde{c}^1, \tilde{c}^2\}$ for which Equation (6) does not admit any solution k. As a result, we are going to show that *a particular property – which is independent of the secret key – holds on the ciphertexts only in the case in which the key-recovery (mixture integral) attack just proposed "fails".*

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 $p^3, p^4 \in \mathcal{C}_0 \oplus a$ be two other plaintexts generated by*

$$p^3 \equiv (z^1, w^1, \Psi, \Phi),\ p^4 \equiv (z^2, w^2, \Psi, \Phi) \quad \text{or} \quad p^3 \equiv (z^1, w^2, \Psi, \Phi),\ p^4 \equiv (z^2, w^1, \Psi, \Phi),$$

where Ψ and Φ can take any possible value in $\mathbb{F}_{2^8}$. For all $i, j = 0, ..., 3$ and for all pairwise distinct[5] *$\alpha, \beta, \gamma, \delta \in \{1, 2, 3, 4\}$, the condition*

$$\big[R^3(p^\alpha) \oplus R^3(p^\beta)]_{i,j} = 0 \quad \text{and} \quad \big[R^3(p^\gamma) \oplus R^3(p^\delta)]_{i,j} \neq 0, \tag{7}$$

where $[\cdot]_{i,j}$ denotes the byte in row i and column j, can never hold for 3-round AES (without the final MixColumns operation), independently of the secret key, of the details of the S-Box, and of the MixColumns matrix.

Note that the same event occurs with probability

$$1 - \big[1 - 2^{-8} \cdot (1 - 2^{-8})\big]^{16\cdot 6} \approx 2^{-1.65} \approx 31.87\%$$

[5] More precisely, we assume that $\alpha \neq \beta$, $\alpha \neq \gamma$, $\alpha \neq \delta$, $\beta \neq \gamma$, $\beta \neq \delta$ and $\gamma \neq \delta$.

in the case in which the ciphertexts are generated by a random permutation (there are 16 bytes and 6 possible combinations of $\alpha, \beta, \gamma, \delta \in \{1,2,3,4\}$). As a result, this property can be exploited to set up a secret-key distinguisher which is independent of the secret key.

Proof. We prove this result by contradiction. Assume there exist $j, k \in \{0, ..., 3\}$ and there exist pairwise distinct $\alpha, \beta, \gamma, \delta \in \{1,2,3,4\}$ such that

$$\big[R^3(p^\alpha) \oplus R^3(p^\beta)\big]_{i,j} = 0 \qquad \text{and} \qquad \big[R^3(p^\gamma) \oplus R^3(p^\delta)\big]_{i,j} \neq 0.$$

According to Lemma 3, there exists at least one key k for 3-round AES that satisfies Equation (6). Since $R^3(p^\alpha)]_{i,j} = [R^3(p^\beta)]_{i,j}$, it turns out that $c^\alpha_{i,j} = c^\beta_{i,j}$, which implies

$$\text{S-Box}^{-1}(c^\alpha_{i,j} \oplus k_{i,j}) \oplus \text{S-Box}^{-1}(c^\beta_{i,j} \oplus k_{i,j}) = 0.$$

It follows that Equation (6) reduces to

$$\text{S-Box}^{-1}(c^\gamma_{i,j} \oplus k_{i,j}) \oplus \text{S-Box}^{-1}(c^\delta_{i,j} \oplus k_{i,j}) = 0.$$

Since $\big[R^3(p^\gamma) \oplus R^3(p^\delta)\big]_{i,j} \neq 0$, that is, $c^\gamma_{i,j} \neq c^\delta_{i,j}$, it follows that

$$\forall k_{j,l}: \qquad \text{S-Box}^{-1}(c^\gamma_{i,j} \oplus k_{i,j}) \neq \text{S-Box}^{-1}(c^\delta_{i,j} \oplus k_{i,j}),$$

which contradicts Lemma 3. As a result, for all pairwise distinct $\alpha, \beta, \gamma, \delta \in \{1,2,3,4\}$, the condition

$$\forall i, j = 0, ..., 3: \quad \big[R^3(p^\alpha) \oplus R^3(p^\beta)\big]_{i,j} = 0 \qquad \text{and} \qquad \big[R^3(p^\gamma) \oplus R^3(p^\delta)\big]_{i,j} \neq 0$$

can never hold for 3-round AES. □

What about the name? We decided to call this an "Impossible Mixture Integral" distinguisher because it exploits a property which holds with prob. 0 and because it extends the Mixture Integral distinguisher presented before.

Notation. For the follow-up, we introduce a notation in order to easily explain the costs of the distinguisher and of the attacks based on the impossible zero-sum property just proposed. Let $x^1, y^1, x^2, y^2 \in \mathbb{F}_{2^8}$ s.t. $x^1 \neq x^2$ and $y^1 \neq y^2$ arbitrary but fixed. Let $\mathfrak{T}^{x,y}_{\Psi,\Phi}$ be the set of a pair of plaintexts (i.e., two plaintexts) defined as

$$\mathfrak{T}^{x,y}_{\Psi,\Phi} := \{p^1 = (x^1, y^1, \Psi, \Phi), p^2 = (x^2, y^2, \Psi, \Phi)\} \tag{8}$$

where $\Psi, \Phi \in \mathbb{F}_{2^8}$ and where $p^1, p^2 \in \mathcal{C}_0 \oplus a$. Since x^1, y^1, x^2, y^2 are fixed, we usually denote $\mathfrak{T}^{x,y}_{\Psi,\Phi}$ by $\mathfrak{T}_{\Psi,\Phi}$, that is, $\mathfrak{T}^{x,y}_{\Psi,\Phi} \equiv \mathfrak{T}_{\Psi,\Phi}$.

Let $\mathfrak{S}$ be the union of two sets $\mathfrak{T}^{x,y}$ (i.e., as set of four plaintexts) defined as

$$\begin{aligned} \mathfrak{S} = \mathfrak{T}_{\Psi,\Phi} \cup \mathfrak{T}_{\psi,\phi} \equiv \big\{ & p^1 = (x^1, y^1, \Psi, \Phi), p^2 = (x^2, y^2, \Psi, \Phi), \\ & p^3 = (x^1, y^1, \psi, \phi), p^4 = (x^2, y^2, \psi, \phi)\big\}, \end{aligned} \tag{9}$$

where $(\Psi, \Phi) \neq (\psi, \phi)$. Note that given $p^1, p^2, p^3, p^4 \in \mathfrak{S}$, the corresponding ciphertexts after 3-round AES encryptions satisfy Theorem 3.

```
Data: 5 different sets $\mathfrak{T}_{\Psi,\Phi} = \{p^1 = (x^1, y^1, \Psi, \Phi), p^2 = (x^2, y^2, \Psi, \Phi)\}$ where
      $p^1, p^2 \in \mathcal{C}_0 \oplus a$ s.t. $p^1 \equiv (x^1, y^1, \Phi, \Psi), p^2 \equiv (x^2, y^2, \Phi, \Psi)$ defined as in
      Equation (8), and corresponding ciphertexts after 3 rounds
Result: 1 if AES permutation - 0 if random permutation (with prob. 95%)
for each pair of couples $[R^3(p^1), R^3(p^2)]$ and $[R^3(q^1), R^3(q^2)]$ where
 $\mathfrak{T}^1 \equiv \{p^1, p^2\}$ and $\mathfrak{T}^2 \equiv \{q^1, q^2\}$ do
    for each $i, j = 0, ..., 3$ do
        if "$\big[a \oplus b\big]_{i,j} = 0$ and $\big[c \oplus d\big]_{i,j} \neq 0$" where
         $(a, b, c, d) \in \{R^3(p^1), R^3(p^2), R^3(q^1), R^3(q^2)\}$ are all distinct (i.e.
         $a \neq b,\ a \neq c,\ \ldots,\ c \neq d$) then
            return Random Perm
        end
    end
end
return 3-round AES
```

Algorithm 1: *Impossible Mixture Integral Distinguisher on 3-round AES*

Data Cost of the Distinguisher. If the goal is to distinguish 3-round AES from a random permutation with a probability higher than 95%, one needs at least 8 different sets $\mathfrak{S}$ defined as before, since $1-(1-2^{-1.65})^N \geq 0.95$ if $N \geq 8$. In order to generate such 8 sets $\mathfrak{S}$, one needs at least 5 different sets $\mathfrak{T}$, since $\binom{5}{2} = 10 \geq 8$. As a result, the cost of such a distinguisher is of $5 \cdot 2 = 10$ chosen plaintexts.

We emphasize that the corresponding 3-round encryptions of

$$p^1 \equiv (z^1, w^1, \Psi, \Phi),\ p^2 \equiv (z^2, w^2, \Psi, \Phi), p^3 \equiv (z^1, w^2, \Psi', \Phi'),\ p^4 \equiv (z^2, w^1, \Psi', \Phi')$$

satisfy Lemma 3 even if $\Psi \neq \Psi'$ and $\Phi \neq \Phi'$.

For completeness, if a success probability of 65% is sufficient, then one needs only 6 chosen plaintexts (by analogous computation, $1-(1-2^{-1.65})^N \geq 0.65$ if $N \geq 3$, which implies that $N \geq 3$ sets $\mathfrak{S}$ are required, or equivalently $\binom{3}{2} = 3$ sets $\mathfrak{T}$, that is $2 \cdot 3 = 6$ chosen plaintexts).

Computational Cost of the Distinguisher. The property needs to be tested for each pair of the 5 input sets, and for each of the 16 state bytes. Testing the property requires $\binom{4}{2} \cdot 2$ table lookups and XOR operations. Hence, the total computational cost consists of at most

$$\binom{5}{2} \cdot \binom{4}{2} \cdot 2 \cdot 16 = 1920 \approx 2^{10.9}$$

table lookups and XOR operations, that is, approximately 2^5 3-round AES encryptions (assuming[6] 20 S-Boxes ≈ 1-round).

[6] Even if this approximation is not formally correct – the size of the table of an S-Box lookup is smaller than the size of the table used for our proposed distinguisher – it allows to give a comparison between our distinguishers and the others currently present in the literature. This approximation is largely used in literature (assuming that the linear/affine operations of each AES round are negligible in terms of costs).

Before going on, we mention that this cost is *roughly of the same order* of the one required to set up a 3-round AES distinguisher based on the truncated differential property (see e.g. [10, Sect. 4.3] for more details[7]).

Remark. In Appendix C, we consider the possibility to set up a similar "Impossible Mixture Integral" distinguisher on 4-round AES (by extending the 3-round integral distinguisher). However, as we show there, it seems that a trivial application of such a distinguisher on 4 rounds requires more than the full code book. An *open future problem* is to study the possibility to set up a similar distinguisher on 4 (or even more) rounds of AES.

Practical Verification. We implemented and practically verified[8] the distinguisher just presented on 3-round AES. By practical tests, we found that using 10 chosen plaintexts, the distinguisher always recovers the 3-round AES permutation when the ciphertexts are generated by such a permutation. In the case in which the ciphertexts are generated by a random permutation – given (in our case) by 21-round AES – the distinguisher is able to recover it with a success probability of 94.4%, close to 95% used before (number of tests: $250\,000 \approx 2^{18}$). In the other cases, the distinguisher is not able to distinguish the 3-round AES permutation from the random one. Moreover, we found out that

- using 6 or 8 chosen plaintexts instead of 10, this probability decreases to 61% and 82.9%, respectively;
- using 12, 14 or 16 chosen plaintexts instead of 10, this probability increases to 98.5%, 99.7% and 99.95%, respectively.

5 Mixture Integral Attacks on Reduced-Round AES

5.1 Mixture Integral Key-Recovery Attack on 3-Round AES

The previous secret-key distinguisher on 2-round AES proposed in Section 4.2 is the starting point for a key-recovery attack on 3- and 4-round AES.[9] The attack works in the same way as a classical integral key-recovery attack [5,12], with the crucial difference that it has a data cost of only 4 chosen plaintexts.

[7] For comparison, for the case of 6 chosen plaintexts, the computational cost of our distinguisher is of $\binom{3}{2} \cdot \binom{4}{2} \cdot 2 \cdot 16 = 576 \approx 2^{9.2}$ table look-ups, which amounts to approximately 2^3 3-round AES encryptions.

[8] The source codes of the distinguishers and attacks are available as supplementary material. They will be made public together with the publication of the paper.

[9] Potentially, it is also possible to set up an attack on 5-round AES by extending the 4-round one at the beginning. However, such attack is not in the low-data scenario and it would not be competitive w.r.t. other attacks present in the literature. For this reason, we do not present it.

```
Data: 4 chosen plaintexts $p^1, p^2, \tilde{p}^1, \tilde{p}^2 \in (\mathcal{D}_{0,3} \cap \mathcal{C}_0) \oplus a$ s.t.
      $p^1 \equiv (z^1, w^1), p^2 \equiv (z^2, w^2)$ and $\tilde{p}^1 \equiv (z^1, w^2), \tilde{p}^2 \equiv (z^2, w^1)$, and
      corresponding ciphertexts after 3 rounds
Result: secret key $k$
for each $i, j = 0, ..., 3$ do
    for each $k^*_{i,j}$ from $0x00$ to $0xFF$ do
        if Equation (10) is satisfied                    // prob. $2^{-8}$ then
            store $k^*_{i,j}$ as candidate for byte $(i, j)$ of the last round key;
        end
    end
end
if more than a single candidate $k^*$ passed the test then
    do a brute-force attack on the possible candidates for the respective
      master keys (filter wrongly guessed candidates);
end
return secret key $k$.
```

Algorithm 2: *Mixture Integral Key-Recovery Attack on 3-round AES*

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

$$\tilde{p}^1 \equiv (z^1, w^1, \Psi, \Phi),\ \tilde{p}^2 \equiv (z^2, w^2, \Psi, \Phi) \quad \text{or} \quad \tilde{p}^1 \equiv (z^1, w^2, \Psi, \Phi),\ \tilde{p}^2 \equiv (z^2, w^1, \Psi, \Phi),$$

where Ψ and Φ can take any possible value in $\mathbb{F}_{2^8}$. Moreover, let $c^1, c^2, \tilde{c}^1, \tilde{c}^2$ denote the corresponding ciphertexts after 3-round AES:

$$c^1 = R^3(p^1), \qquad c^2 = R^3(p^2), \qquad \tilde{c}^1 = R^3(\tilde{p}^1), \qquad \tilde{c}^2 = R^3(\tilde{p}^2).$$

Assume that the final MixColumns operation has been omitted[10]. Due to the zero-sum distinguisher just proposed and working at byte level, we know that the secret key $k_{i,j}$ for each $i, j = 0, ..., 3$ satisfies

$$\begin{aligned} &\text{S-Box}^{-1}(c^1_{i,j} \oplus k_{i,j}) \oplus \text{S-Box}^{-1}(c^2_{i,j} \oplus k_{i,j}) \\ &\oplus \text{S-Box}^{-1}(\tilde{c}^1_{i,j} \oplus k_{i,j}) \oplus \text{S-Box}^{-1}(\tilde{c}^2_{i,j} \oplus k_{i,j}) = 0 \end{aligned} \tag{10}$$

with prob. 1 independently of the S-Box. Since a wrongly guessed key satisfies the previous equality with prob. 2^{-8}, it is possible to find the right one.

A complete pseudo-code of the attack is given in Algorithm 2. The data cost of the attack is of 4 chosen plaintexts, while the computational cost is of

$$\underbrace{16}_{\text{number of bytes}} \cdot \underbrace{2^8}_{\text{number of } k_{i,j}} \cdot \underbrace{4}_{\text{number of S-Boxes}} = 2^{14} \text{ S-Box operations,}$$

that is, $2^{8.1}$ 3-round AES encryption (assuming 20 S-Box $\approx$ 1-round).

[10] If not, since the MixColumns is a linear operation, it is sufficient to swap the final MixColumns and the final AddRoundKey operation: $k \oplus MC(\cdot) = MC(k' \oplus \cdot)$, where $k' = MC^{-1}(\cdot)$. When k' is given, one can find k using the relation $k = MC(k')$.

```
Data: 4 chosen plaintexts $p^1, p^2, \tilde{p}^1, \tilde{p}^2 \in (\mathcal{D}_{0,3} \cap \mathcal{C}_0) \oplus a$ s.t.
      $p^1 \equiv (z^1, w^1), p^2 \equiv (z^2, w^2)$ and $\tilde{p}^1 \equiv (z^1, w^2), \tilde{p}^2 \equiv (z^2, w^1)$, and
      corresponding ciphertexts after 3 rounds
Result: secret key $k$
Let $A$ be an array of $2^{32}$ entries;
for each $(i, j, h, l)$ from $(0, 0, 0, 0)$ to $(0xFF, 0xFF, 0xFF, 0xFF)$ do
    if $i \neq j$ and $i \neq h$ and $i \neq l$ and $j \neq h$ and $j \neq l$ and $h \neq l$ then
        for each $\kappa$ from $0$ to $0xFF$ do
            if $S\text{-}Box^{-1}(i \oplus \kappa) \oplus S\text{-}Box^{-1}(j \oplus \kappa) \oplus S\text{-}Box^{-1}(h \oplus \kappa) \oplus$
              $\oplus S\text{-}Box^{-1}(l \oplus \kappa) = 0$ then
                $A[i + 2^8 \times j + 2^{16} \times h + 2^{24} \times l] \leftarrow \kappa$;
            end
        end
    end
end
for each $i, j = 0, ..., 3$ do
    if $c^1_{i,j} \neq c^2_{i,j}$ and $c^1_{i,j} \neq \tilde{c}^1_{i,j}$ and $c^1_{i,j} \neq \tilde{c}^2_{i,j}$ and $c^2_{i,j} \neq \tilde{c}^1_{i,j}$ and $c^2_{i,j} \neq \tilde{c}^2_{i,j}$ and
      $\tilde{c}^1_{i,j} \neq \tilde{c}^2_{i,j}$    // approximately prob. 99.991% - see main text! then
        $k_{i,j} \leftarrow A[c^1_{i,j} + 2^8 \times c^2_{i,j} + 2^{16} \times \tilde{c}^1_{i,j} + 2^{24} \times \tilde{c}^2_{i,j}]$;
    end
    else
        $k_{i,j}$ can take any possible value;
    end
end
if more than a single key passed the test then
    do a brute-force attack on the possible candidates (filter wrongly guessed
      candidates);
end
return secret-key $k$.
```

Algorithm 3: *Mixture Integral Key-Recovery Attack on 3-round AES* (2nd Version)

An Optimal Implementation of the Attack. The previous attack does not require any memory cost. However, another possible version of the attack can be considered. This second version requires precomputation and it has a memory cost, but the computational cost is negligible. The idea is simply to generate a table with the solutions κ of the equation

$$\text{S-Box}^{-1}(i \oplus \kappa) \oplus \text{S-Box}^{-1}(j \oplus \kappa) \oplus \text{S-Box}^{-1}(h \oplus \kappa) \oplus \text{S-Box}^{-1}(l \oplus \kappa) = 0 \quad (11)$$

for each $i, j, h, l \in \mathbb{F}_{2^8}$. A complete pseudo code is given in Algorithm 3.

What is the number of solutions of the previous equality? If $i = j = h = l$ or if $i = j$ and $h = l$ (where $j \neq h$ – analogous for the other 6 cases), then the previous equality is always satisfied for each κ. Moreover, by Section 4.3 we know that the case[11] $i = j$ and $h \neq l$ (analogous for the other 6 cases) is not

[11] Note that the case $i = j = h$ and $h \neq l$ is included here.

possible. Finally, consider the case[12] $i \neq j$ and $i \neq h$ and $i \neq l$ and $j \neq h$ and $j \neq l$ and $h \neq l$. In such a case, the number of solutions is on average 1. Indeed, Equation (11) can be split into

$$\text{S-Box}^{-1}(i \oplus \kappa) \oplus \text{S-Box}^{-1}(j \oplus \kappa) \equiv \text{S-Box}^{-1}(x \oplus \Delta_I^x) \oplus \text{S-Box}^{-1}(x) = \Delta_O^x,$$
$$\text{S-Box}^{-1}(h \oplus \kappa) \oplus \text{S-Box}^{-1}(l \oplus \kappa) \equiv \text{S-Box}^{-1}(y \oplus \Delta_I^y) \oplus \text{S-Box}^{-1}(y) = \Delta_O^y,$$

where $\Delta_O^x = \Delta_O^y$ and where $x = j \oplus \kappa$, $\Delta_I^x = i \oplus j$, $y = l \oplus \kappa$ and $\Delta_I^y = h \oplus l$. Since Δ_O^x can take any possible value while Δ_I^x is fixed, there are on average 256 values of x (and so of κ since j is fixed) that satisfy the first equation (analogous for the second equation). Since $\Delta_O^x = \Delta_O^y$, the probability that κ satisfies both equations is 2^{-8}. As a result, the average number of κ values that satisfy both equations is $256/256 = 1$.

Once such a table is generated and the ciphertexts $c^1, c^2, \tilde{c}^1, \tilde{c}^2$ are given, the attacker needs only 16 table lookups to find the secret key (one lookup for each byte of the key), which is much less than a single encryption. An overall estimation for the *precomputation cost* is given by $2^{32} \cdot 2^8 \cdot 4 = 2^{42}$ S-Box operations, that is, $2^{36.1}$ 3-round AES encryptions.

5.2 Mixture Integral Key-Recovery Attack on 4-Round AES

The previous attack can be extended to 4-round AES using the technique proposed in [5,12] in order to extend an integral attack at the end. The idea is to guess the final anti-diagonal of the key, partially decrypt one round and to use the previous attack on 3 rounds to filter out wrongly guessed keys:

$$(p^1, p^2, q^1, q^2) \xrightarrow[\text{prob. 1}]{R^2(\cdot)} \text{Zero Sum} \xleftarrow[\text{key guess (byte)}]{R^{-1}(\cdot)} \xleftarrow[\text{key guess (anti-diag.)}]{R^{-1}(\cdot)} (c^1, c^2, d^1, d^2)$$

where $c^i = R^4(p^i), d^i = R^4(q^i)$ for $i = 1, 2$ and where $\mathfrak{S} = \{p^1, p^2, q^1, q^2\}$ is defined as in Equation (9). We refer to Algorithm 4 for a complete pseudo code and details.

Data Cost. We consider 2 pairs of texts, that is, (p^1, q^1) and (p^2, q^2) generated by mixing variables. The attacker guesses 4 bytes of the final key – that is, $(k^4_{0,0}, k^4_{1,3}, k^4_{2,2}, k^4_{3,1})$ – and 4 bytes of the second to final key – that is, $(k^3_{0,0}, k^3_{1,0}, k^3_{2,0}, k^3_{3,0})$ (analogous for the other four cases). Since she can verify the zero-sum property "only" on four bytes and using only 4 chosen plaintexts, the number of remaining keys is 2^{32} ($\equiv$ values of k^4) $\cdot 2^{32}$ ($\equiv$ values of k^3) $\cdot 2^{-32} = 2^{32}$. As a result and without using the details of the key schedule, she needs at least another pair of texts (p^3, q^3) to filter out all wrongly guessed keys (without using brute force), for a total of 6 chosen plaintexts.

[12] The probability of this case is

$$\frac{2^8 \cdot (2^8-1) \cdot (2^8-2) \cdot (2^8-3)}{2^8 \cdot (2^8-1) \cdot (2^8-2) \cdot (2^8-3) + \underbrace{6 \cdot 2^8 \cdot (2^8-1)}_{\text{case: } i=j \text{ and } h=l} + \underbrace{2^8}_{\text{case: } i=j=h=l}} = 99.991\%$$

```
Data: 6 chosen plaintexts $p^i, q^i \in \mathcal{C}_0 \oplus a$ for $i = 1, 2, 3$ s.t.
      $p^i \equiv (x, y, \phi^i, \psi^i), q^i \equiv (z, w, \phi^i, \psi^i)$ for $x \neq z$ and $y \neq w$, and
      corresponding ciphertexts $c^i = R^4(p^i), d^i = R^4(q^i)$ after 4 rounds
Result: $(k^3_{0,0}, k^3_{1,0}, k^3_{2,0}, k^3_{3,0})$ and $(k^4_{0,0}, k^4_{1,3}, k^4_{2,2}, k^4_{3,1})$
Let $A$ be an array of $2^{32}$ entries;
for each $(i, j, h, l)$ from $(0, 0, 0, 0)$ to $(0xFF, 0xFF, 0xFF, 0xFF)$ such that (1st)
 $i \neq j$, (2nd) $i \neq h$, (3rd) $i \neq l$, (4th) $j \neq h$, (5th) $j \neq l$ and (6th) $h \neq l$ do
    for each $\kappa$ from $0$ to $0xFF$ do
        if $S\text{-}Box^{-1}(i \oplus \kappa) \oplus S\text{-}Box^{-1}(j \oplus \kappa) \oplus S\text{-}Box^{-1}(h \oplus \kappa) \oplus$
         $\oplus S\text{-}Box^{-1}(l \oplus \kappa) = 0$ then
            $A[i + 2^8 \times j + 2^{16} \times h + 2^{24} \times l] \leftarrow \kappa$;
        end
    end
end
$(k^3_{0,0}, k^3_{1,0}, k^3_{2,0}, k^3_{3,0})$ and $(k^4_{0,0}, k^4_{1,3}, k^4_{2,2}, k^4_{3,1})$ for each $k^4_{0,0}, k^4_{1,3}, k^4_{2,2}, k^4_{3,1}$ from
 (0,0,0,0) to $(0xFF, 0xFF, 0xFF, 0xFF)$ do
    for each $i = 1, 2, 3$ do
        Compute 1-round decryption w.r.t. guessed key $k^4$:
        $$\begin{bmatrix} \tilde{c}^i_{0,0} \\ \tilde{c}^i_{1,0} \\ \tilde{c}^i_{2,0} \\ \tilde{c}^i_{3,0} \end{bmatrix} \leftarrow MC^{-1} \cdot \begin{bmatrix} \text{S-Box}^{-1}(c^i_{0,0} \oplus k^4_{0,0}) \\ \text{S-Box}^{-1}(c^i_{3,1} \oplus k^4_{3,1}) \\ \text{S-Box}^{-1}(c^i_{2,2} \oplus k^4_{2,2}) \\ \text{S-Box}^{-1}(c^i_{1,3} \oplus k^4_{1,3}) \end{bmatrix}$$
        (similar for $[\tilde{d}^i_{0,0}, \tilde{d}^i_{1,0}, \tilde{d}^i_{2,0}, \tilde{d}^i_{3,0}]^T$)
    end
    let $g(\cdot, \cdot, \cdot, \cdot) : \mathbb{N}^4 \rightarrow \mathbb{N}$ defined as $g(x, y, z, w) := x + 2^8 \cdot y + 2^{16} \cdot z + 2^{24} \cdot w$
     where $x, y, z, w \in [0, 255]$;
    if $A[g(\tilde{c}^1_{0,0}, \tilde{d}^1_{0,0}, \tilde{c}^2_{0,0}, \tilde{d}^2_{0,0}] = A[g(\tilde{c}^1_{0,0}, \tilde{d}^1_{0,0}, \tilde{c}^3_{0,0}, \tilde{d}^3_{0,0}]$     // prob. $2^{-8}$ then
        if $A[g(\tilde{c}^1_{1,0}, \tilde{d}^1_{1,0}, \tilde{c}^2_{1,0}, \tilde{d}^2_{1,0}] = A[g(\tilde{c}^1_{1,0}, \tilde{d}^1_{1,0}, \tilde{c}^3_{1,0}, \tilde{d}^3_{1,0}]$ then
            if $A[g(\tilde{c}^1_{2,0}, \tilde{d}^1_{2,0}, \tilde{c}^2_{2,0}, \tilde{d}^2_{2,0}] = A[g(\tilde{c}^1_{2,0}, \tilde{d}^1_{2,0}, \tilde{c}^3_{2,0}, \tilde{d}^3_{2,0}]$ then
                if $A[g(\tilde{c}^1_{3,0}, \tilde{d}^1_{3,0}, \tilde{c}^2_{3,0}, \tilde{d}^2_{3,0}] = A[g(\tilde{c}^1_{3,0}, \tilde{d}^1_{3,0}, \tilde{c}^3_{3,0}, \tilde{d}^3_{3,0}]$ then
                    $\hat{k}^3_{0,0} \leftarrow A[\tilde{c}^1_{0,0} + 2^8 \times \tilde{c}^2_{0,0} + 2^{16} \times \tilde{d}^1_{0,0} + 2^{24} \times \tilde{d}^2_{0,0}]$;
                    $\hat{k}^3_{1,0} \leftarrow A[\tilde{c}^1_{1,0} + 2^8 \times \tilde{c}^2_{1,0} + 2^{16} \times \tilde{d}^1_{1,0} + 2^{24} \times \tilde{d}^2_{1,0}]$;
                    $\hat{k}^3_{2,0} \leftarrow A[\tilde{c}^1_{2,0} + 2^8 \times \tilde{c}^2_{2,0} + 2^{16} \times \tilde{d}^1_{2,0} + 2^{24} \times \tilde{d}^2_{2,0}]$;
                    $\hat{k}^3_{3,0} \leftarrow A[\tilde{c}^1_{3,0} + 2^8 \times \tilde{c}^2_{3,0} + 2^{16} \times \tilde{d}^1_{3,0} + 2^{24} \times \tilde{d}^2_{3,0}]$;
                end
            end
        end
    end
end
$[k^3_{0,0}, k^3_{1,0}, k^3_{2,0}, k^3_{3,0}]^T \leftarrow MC \cdot [\hat{k}^3_{0,0}, \hat{k}^3_{1,0}, \hat{k}^3_{2,0}, \hat{k}^3_{3,0}]^T$;
return $(k^3_{0,0}, k^3_{1,0}, k^3_{2,0}, k^3_{3,0})$ and $(k^4_{0,0}, k^4_{1,3}, k^4_{2,2}, k^4_{3,1})$
```

Algorithm 4: *Mixture Integral Key-Recovery Attack on 4-round AES* (repeat (an analogous) attack to find the full key – if more than a single key passes the distinguisher, do a brute-force attack on the possible candidates)

Computational Cost. First of all, a 1-round decryption costs

$$\underbrace{2^{32}}_{\text{anti-diagonal of } k^4} \cdot \underbrace{4}_{\text{number of S-Boxes}} \cdot \underbrace{6}_{\text{number of texts}} = 3 \cdot 2^{35}$$

S-Box lookups. For each anti-diagonal of k^4, the cost to find 4 bytes of k^3 is of $2 \cdot (1 + 2^{-8} + 2^{-16} + 2^{-24}) = 2$ table lookups. Since the probability that the first condition (namely, $A[g(\tilde{c}^1_{0,0}, \tilde{d}^1_{0,0}, \tilde{c}^2_{0,0}, \tilde{d}^2_{0,0}] = A[g(\tilde{c}^1_{0,0}, \tilde{d}^1_{0,0}, \tilde{c}^3_{0,0}, \tilde{d}^3_{0,0}]$ in Algorithm 4) is satisfied is 2^{-8}, it is 2^{-16} for both the first and the second condition and so on. As a result, in order to find the full key, the cost is of $4 \cdot 3 \cdot 2^{35} \cdot 2 = 3 \cdot 2^{38}$ S-Box lookups, that is, $2^{33.3}$ 4-round encryptions. Note that *no details of the key schedule* have been used.

For completeness, we mention that the same attack can be performed without precomputation and table lookups. Using the same computation proposed before for the 3-round attack, the cost in such a case would be of

$$4 \cdot 3 \cdot 2^{35} \cdot (\underbrace{4}_{\text{number of bytes}} \cdot \underbrace{2^8}_{k^3_{i,j}} \cdot \underbrace{4 \cdot 2}_{\text{number of S-Boxes}}) = 3 \cdot 2^{50}$$

S-Box lookups, that is, $2^{45.3}$ 4-round encryptions.

Practical Verification. We implemented and practically verified Algorithm 2 in C++, which allows us to find the last secret round key almost instantly on our tested machine (Intel i7-8550U @ 4.00 GHz).

6 Impossible Mixture Integral Attack on 4-Round AES

In Section 4.3, we presented a new 3-round AES secret-key distinguisher which is independent of the key. Using the techniques just described, here we exploit this distinguisher in order to set up an attack[13] on 4-round AES.

The *low-data* attack works as follows. Assuming that the final MixColumns operation is omitted, the attacker partially guesses one anti-diagonal of the final key, partially decrypts one round, computes the inverse MC operation, and exploits the 3-round distinguisher of Theorem 3 to partially decrypt

$$(p^1, p^2, p^3, p^4) \xrightarrow[\text{prob. } 1]{R^3_f(\cdot)} \text{Distinguisher (Theorem 3)} \xleftarrow[\text{key-guess (4 bytes)}]{MC^{-1} \circ (R^{-1}(\cdot))} (c^1, c^2, c^3, c^4),$$

where $\mathfrak{S} = \{p^1, p^2, p^3, p^4\}$ is defined as in Equation (9) and where $R^3_f(\cdot)$ denotes a 3-round encryption of AES without the final MixColumns operation.

By exploiting the distinguisher presented in Theorem 3, the attacker can filter wrongly guessed keys, since for wrongly guessed keys the behavior is similar to that of a random permutation. That is, due to the "Wrong-Key Randomization

[13] Potentially, the 4-round attack can be extended at the beginning, in order to set up a 5-round AES attack. However, as we show in details in Appendix D, such an attack is not competitive w.r.t. other attacks in the literature.

```
Data: $24 \simeq 2^{4.58}$ different sets $\mathfrak{T}_{\Psi,\Phi} = \{p^1 = (x^1, y^1, \Psi, \Phi), p^2 = (x^2, y^2, \Psi, \Phi)\}$
      where $p^1, p^2 \in \mathcal{C}_0 \oplus a$ s.t. $p^1 \equiv (x^1, y^1, \Psi, \Phi), p^2 \equiv (x^2, y^2, \Psi, \Phi)$ defined
      as in Equation (8), and corresponding ciphertexts after 4 rounds
Result: (final) secret key $k$
given $p^1, p^2 \in \mathfrak{T}$, let $c^1 = R^4(p^1)$ and $c^2 = R^4(p^2)$;
for each $k^4_{0,0}, k^4_{1,3}, k^4_{2,2}, k^4_{3,1}$ from (0,0,0,0) to $(0xFF, 0xFF, 0xFF, 0xFF)$ do
    (partially) compute the 1-round decryption of $\mathfrak{T}_{\Psi,\Phi}$ for each $\Psi, \Phi$ w.r.t. the
      guessed key $k^4$, that is:
        $\mathfrak{T}' \leftarrow \left\{ \forall i = 1, 2 : \quad \begin{bmatrix} t^i_{0,0} \\ t^i_{1,0} \\ t^i_{2,0} \\ t^i_{3,0} \end{bmatrix} \leftarrow MC^{-1} \cdot \begin{bmatrix} \text{S-Box}^{-1}(c^i_{0,0} \oplus k^4_{0,0}) \\ \text{S-Box}^{-1}(c^i_{1,3} \oplus k^4_{1,3}) \\ \text{S-Box}^{-1}(c^i_{2,2} \oplus k^4_{2,2}) \\ \text{S-Box}^{-1}(c^i_{3,1} \oplus k^4_{3,1}) \end{bmatrix} \right\}$
        // Note: $\mathfrak{T}'$ contains a pair of 4 bytes, not a pair of texts!
    $flag \leftarrow 0$;
    for each $\mathfrak{T}'_{\Psi,\Phi}$ and $\mathfrak{T}'_{\psi,\phi}$ (where $(\Psi, \Phi) \neq (\psi, \phi)$) do
        for each $i = 0, ..., 3$ do
            if "$a \oplus b = 0$ and $c \oplus d \neq 0$" where $(a, b, c, d) \in [\mathfrak{T}'_{\Psi,\Phi} \cup \mathfrak{T}'_{\psi,\phi}]_{i,0}$ are
              all distinct (where $[\cdot]_{i,0}$ denotes the byte in position $(i, 0)$) then
                $(k^4_{0,0}, k^4_{1,3}, k^4_{2,2}, k^4_{3,1})$ is wrong: check next 4-byte value;
                $flag \leftarrow 1$;
            end
        end
    end
    if $flag = 0$ then
        Return $(k^4_{0,0}, k^4_{1,3}, k^4_{2,2}, k^4_{3,1})$ as a possible candidate for the key;
    end
end
Repeat the same procedure for the next 3 anti-diagonals of the final key;
if more than a single key passed the test then
    do a brute-force attack on the possible candidates (filter wrongly guessed
      candidates);
end
return (final) secret key $k$
```

Algorithm 5: *Impossible Mixture Integral Attack on 4-round AES*

Hypothesis"[14], given the ciphertexts (c^1, c^2, c^3, c^4) and for a wrongly guessed key $\hat{k}$, the texts $MC^{-1} \circ (R^{-1}(c^i \oplus \hat{k}))$ for $i = 1, ..., 4$ satisfy the property of Theorem 3 with prob. $1 - (1 - 2^{-8})^{4 \cdot 6}$, while such a property is never satisfied by the secret key.

Remark. We emphasize that the distinguisher on 3 rounds (Section 4.3) works independently on each byte of the ciphertexts *only in the case* in which the final MixColumns operation is omitted. If it is not omitted, it is sufficient to swap it with the AddRoundKey operation (since both operations are linear). However,

[14] This hypothesis states that when decrypting one or several rounds with a wrong key guess creates a function that behaves like a random function.

if such a property is exploited to set up a key-recovery attack on 4 (or more) rounds of AES (by extending the distinguisher at the end), one has to work on an entire column (namely, 4 bytes) in order to check it instead of checking each byte independently. As a result, the attacker has to guess one anti-diagonal (4 bytes) of the final key, because she has to apply the inverse MixColumns operation in order to check if the required property is satisfied or not.

Data Cost. Assume the goal is to filter all wrong keys with probability at least 95%. Working independently on each column/anti-diagonal of the key, 4 random texts $\{t^1, t^2, t^3, t^4\}$ satisfy the required property with prob. $1-(1-2^{-8})^{4\cdot 6}$ (using the same argumentation provided for the corresponding distinguisher).

Since there are 4 columns/anti-diagonals and each one of them can take 2^{32} different values (which are all independent), we ask that for each 4-byte key guess there exists at least one set $\mathfrak{S}$ that satisfies the required property with prob. $0.95^{1/(4\cdot 2^{32})}$. As a result, we need N different sets of $\mathfrak{S}$ defined as in Equation (9) such that

$$1-(1-2^{-8})^{24\cdot N} \geq 0.95^{\frac{1}{4\cdot 2^{32}}}$$

in order to find the *entire* key with prob. higher than 95%, that is, $N \geq 284$. It follows that we need n different $\mathfrak{T}$ defined as in Equation (8) in order to construct N sets $\mathfrak{S}$ s.t. $\binom{n}{2} \geq 284$, that is, $n \geq 24 \simeq 2^{4.58}$. In conclusion, we need approximately $2 \cdot 24 = 48 \simeq 2^{5.6}$ pairs of texts $(p^1, p^2) \in \mathfrak{T} \subseteq \mathcal{C}_0 \oplus a$.

Computational Cost. The 1-round decryption requires $4 \cdot 4 \cdot 2^{32} \cdot 2^{4.6} = 2^{41.6}$ S-Box lookups. We store these (partially decrypted) values in a table. In order to check the required property of Theorem 3, one has to construct all possible sets of 4 texts for each possible guessed key. As a result, the total cost of the attack is of

$$\underbrace{2^{41.6}}_{\text{partially decrypt}} + \underbrace{4\cdot 2^{32}}_{\text{number of keys}} \underbrace{\cdot 4 \cdot 2 \cdot \binom{24}{2}}_{\text{check property}} \approx 2^{41.6} + (4\cdot 2^{32}) \cdot (4 \cdot 2 \cdot 2^{8.11}) \approx 2^{45.23}$$

table and S-Box lookups, which corresponds to $2^{38.91}$ 4-round encryptions.

Note that this is the computational cost in the worst case. Indeed, on average $N = 2^5$ different sets $\mathfrak{S}$ are sufficient to discard a wrongly guessed key, since $1-(1-2^{-8})^{4\cdot 6\cdot 2^5} \geq 0.95$. In particular, when the attacker finds a set $\mathfrak{S}$ for which the required property is not satisfied, she can simply discard such a wrongly guessed key (that is, she does not need to verify the required property for the other sets $\mathfrak{S}$). As a result, the *average* cost of the attack is well approximated by $2^{41.6} + (4 \cdot 2^{32}) \cdot (4 \cdot 2 \cdot 2^7) = 2^{44.25}$ table and S-Box lookups, which corresponds to $2^{37.9}$ 4-round encryptions.

Practical Verification. We implemented and practically verified the Impossible Mixture Integral Attack on 4-round AES (Algorithm 5) in C++ on our machine, and are able to find 4 bytes of the final round key in under one hour.

Acknowledgements. This work has been partially supported by IOV42.

References

1. Bar-On, A., Dunkelman, O., Keller, N., Ronen, E., Shamir, A.: Improved Key Recovery Attacks on Reduced-Round AES with Practical Data and Memory Complexities. In: Advances in Cryptology - CRYPTO 2018. LNCS, vol. 10992, pp. 185–212 (2018)
2. Biham, E., Keller, N.: Cryptanalysis of Reduced Variants of Rijndael (2001), unpublished, `http://csrc.nist.gov/archive/aes/round2/conf3/papers/35-ebiham.pdf`
3. Bouillaguet, C., Derbez, P., Fouque, P.A.: Automatic Search of Attacks on Round-Reduced AES and Applications. In: Advances in Cryptology - CRYPTO 2011. LNCS, vol. 6841, pp. 169–187 (2011)
4. Boura, C., Canteaut, A., Coggia, D.: A General Proof Framework for Recent AES Distinguishers. IACR Transactions on Symmetric Cryptology **2019**(1), 170–191 (2019)
5. Daemen, J., Knudsen, L.R., Rijmen, V.: The Block Cipher Square. In: Fast Software Encryption - FSE 1997. LNCS, vol. 1267, pp. 149–165 (1997)
6. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Information Security and Cryptography, Springer (2002)
7. Daemen, J., Rijmen, V.: Understanding two-round differentials in AES. In: SCN. LNCS, vol. 4116, pp. 78–94 (2006)
8. Grassi, L.: MixColumns Properties and Attacks on (Round-Reduced) AES with a Single Secret S-Box. In: Topics in Cryptology - CT-RSA 2018. LNCS, vol. 10808, pp. 243–263 (2018)
9. Grassi, L.: Mixture differential cryptanalysis: a new approach to distinguishers and attacks on round-reduced AES. IACR Trans. Symmetric Cryptol. **2018**(2), 133–160 (2018)
10. Grassi, L., Rechberger, C., Rønjom, S.: Subspace trail cryptanalysis and its applications to AES. IACR Trans. Symmetric Cryptol. **2016**(2), 192–225 (2016)
11. Grassi, L., Rechberger, C., Rønjom, S.: A new structural-differential property of 5-round AES. In: Advances in Cryptology - EUROCRYPT 2017. LNCS, vol. 10211, pp. 289–317 (2017)
12. Knudsen, L.R., Wagner, D.A.: Integral Cryptanalysis. In: Fast Software Encryption – FSE 2002. LNCS, vol. 2365, pp. 112–127 (2002)
13. Rønjom, S., Bardeh, N.G., Helleseth, T.: Yoyo Tricks with AES. In: Advances in Cryptology - ASIACRYPT 2017. LNCS, vol. 10624, pp. 217–243 (2017)
14. Sun, B., Liu, M., Guo, J., Qu, L., Rijmen, V.: New Insights on AES-Like SPN Ciphers. In: Advances in Cryptology - CRYPTO 2016. LNCS, vol. 9814, pp. 605–624 (2016)
15. Tiessen, T.: Polytopic Cryptanalysis. In: Advances in Cryptology - EUROCRYPT 2016. LNCS, vol. 9665, pp. 214–239 (2016)
16. Tiessen, T., Knudsen, L.R., Kölbl, S., Lauridsen, M.M.: Security of the AES with a Secret S-Box. In: Fast Software Encryption - FSE 2015. LNCS, vol. 9054, pp. 175–189 (2015)
17. Tunstall, M.: Improved ”Partial Sums”-based Square Attack on AES. In: International Conference on Security and Cryptography – SECRYPT 2012. pp. 25–34. SciTePress (2012)

A Subspace Trail Cryptanalysis for AES

In this section, we give all the details about the subspace trails of AES presented in [10] and briefly recalled in Section 3.

Here we only work with vectors and vector spaces over $\mathbb{F}_{2^8}^{4\times 4}$, and we denote by $\{e_{0,0}, ..., e_{3,3}\}$ the unit vectors of $\mathbb{F}_{2^8}^{4\times 4}$ (e.g., $e_{i,j}$ has a single 1 in row i and column j).

Definition 5. *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 example, $\mathcal{C}_0$ corresponds to the symbolic matrix

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

Definition 6. *The diagonal spaces* $\mathcal{D}_i$ *are defined as* $\mathcal{D}_i = SR^{-1}(\mathcal{C}_i)$. *Similarly, the inverse-diagonal spaces* $\mathcal{ID}_i$ *are defined as* $\mathcal{ID}_i = SR(\mathcal{C}_i)$.

For example, $\mathcal{D}_0$ and $\mathcal{ID}_0$ correspond to the 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 all $x_1, x_2, x_3, x_4 \in \mathbb{F}_{2^8}$.

Definition 7. *The i-th mixed spaces* $\mathcal{M}_i$ *are defined as* $\mathcal{M}_i = MC(\mathcal{ID}_i)$.

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

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

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

Definition 8. *For* $I \subseteq \{0, 1, 2, 3\}$, *let* $\mathcal{C}_I$, $\mathcal{D}_I$, $\mathcal{ID}_I$ *and* $\mathcal{M}_I$ *be defined as*

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

B Mixture Integral Attacks on Reduced-Round AES with a Single Secret S-Box

The 3-round mixture integral attack proposed in Section 5.1 exploits a property which holds with prob. 1 and which is independent of the secret key and of the details of the S-Box. For this reason, we are going to show that a similar attack can be set up on 3-round AES with a single secret S-Box, exploiting an idea similar to the one proposed in [16]. The strategy consists of two steps:

1. The attacker finds the S-Box up to additive constants, i.e., they find $\text{S-Box}^{-1}(\cdot \oplus a) \oplus b$.
2. The attacker exploits the previous information in order to find the key up to 2^8 equivalents, like $(k_0, k_1 \oplus k_0, ..., k_{15} \oplus k_0)$.

B.1 Strategy of the Attack

Finding the S-Box (Up to Additive Constants). In order to find $S' = \text{S-Box}^{-1}(\cdot \oplus a) \oplus b$, we use the equality

$$\begin{aligned} &\text{S-Box}^{-1}(c^1_{0,0} \oplus k_{0,0}) \oplus \text{S-Box}^{-1}(c^2_{0,0} \oplus k_{0,0}) \\ \oplus\ &\text{S-Box}^{-1}(\tilde{c}^1_{0,0} \oplus k_{0,0}) \oplus \text{S-Box}^{-1}(\tilde{c}^2_{0,0} \oplus k_{0,0}) = 0. \end{aligned} \tag{12}$$

This is similar to what is done in [16], where the authors exploit the fact that

$$\bigoplus_{x \in (\mathcal{D}_0 \cap \mathcal{C}_0)} \text{S-Box}^{-1}([R^4(x)]_{0,0} \oplus k_{0,0}) = 0,$$

which is a well-known property of the integral attack on 4-round AES. We emphasize that this equality involves 256 different texts, while the one exploited in this paper requires only 4 texts (even if on a smaller number of rounds).

Working as in [16], taking different sets $(c^1, c^2, \tilde{c}^1, \tilde{c}^2)$ of ciphertexts corresponding to the 3-round encryptions of plaintexts that share the same generating variables, we can now try to generate enough linear equations to be able to determine $\text{S-Box}^{-1}(\cdot \oplus a) \oplus b$. However, we are only able to determine

$$\{L \circ \text{S-Box}^{-1}(\cdot \oplus a) \oplus b \,|\, L : \mathbb{F}_{2^8} \to \mathbb{F}_{2^8} \text{ linear permutation } \& \ a, b \in \mathbb{F}_{2^8}\},$$

which is of size $2^{70.2}$. In the following, $L \circ \text{S-Box}^{-1}(\cdot \oplus a) \oplus b \equiv A \circ \text{S-Box}^{-1}(\cdot \oplus a)$, where A is an affine permutation.

As each linear equation gives us one byte of information and as we can only determine the S-Box up to $2^{70.2} \leq 2^{72} = 2^{9 \cdot 8}$ variants, there can at most be $256 - 9 = 247$ linearly independent equations like Equation (12). By practical experiments, we found that using $3 \cdot 256 = 768 \approx 2^{9.6}$ different sets of pairs of texts are sufficient in most cases to generate a set of equations with rank 247 (for a cost of $4 \cdot 2^{9.6} = 2^{11.6}$ chosen plaintexts).

Given such a set of equations and working as in [16], it is now easy to determine one representative from the set of affine equivalents to $\text{S-Box}(k_0 \oplus \cdot)$: We incrementally start assigning linearly independent values to variables in order to potentially increase the rank of the corresponding coefficient matrix to 256. However, when assigning a value, it might happen that we do not increase the rank of the matrix. If this is the case, we remove the assignment for this variable and instead try a different one, in order to filter out variable assignments which result in a system with no solutions (this countermeasure is sufficient due to the Rouché-Capelli theorem[15]). We repeat this approach until we have found 9 variables, such that fixing them to linearly independent values in $\mathbb{F}_{2^8}$ results in a rank-256 coefficient matrix, and we note that such a set of variable assignments can easily be found after a small number of trials. When choosing a different approach (e.g., assigning random values to the variables), we might find a solution to the original equation system which is not a permutation.

Let the representative we found be denoted as $A \circ \text{S-Box}^{-1}(\cdot \oplus a)$ for an invertible (and unknown) affine transformation A.

Note that this is sufficient in order to find the secret key, since

$$\bigoplus_{x \in \mathcal{X}} \text{S-Box}^{-1}(x \oplus a) = 0 \iff \bigoplus_{x \in \mathcal{X}} A \circ \text{S-Box}^{-1}(x \oplus a) = 0,$$

where A is an affine operation (that is, $A(x \oplus y) = A(x) \oplus A(y)$ for each x, y).

Finally, we refer to [16, Sect. 3.2] if one aims to determine more information about $\text{S-Box}(\cdot \oplus a) \oplus b$. We highlight that such information is not necessary in order to find the key, and that the strategy proposed there applies here as well.

Finding the Key Given the S-Box. Assume the attacker knows $A \circ \text{S-Box}^{-1}(\cdot \oplus k_{0,0})$ for some unknown A and $k_{0,0}$. This information can be used to find $k_{0,0} \oplus k_{i,j}$ for $0 \le i \le 3, 0 \le j \le 3$ (except where $i = j = 0$), where k denotes the third round key. Indeed, given p^1, p^2, q^1, q^2 as before and the corresponding ciphertexts $c^1, c^2, \tilde{c}^1, \tilde{c}^2$ after 3 rounds, we know that Equation (6), which can be rewritten as

$$\left[\text{S-Box}^{-1}\biggl([c^1_{i,j} \oplus (k_{i,j} \oplus k_{0,0})] \oplus k_{0,0}\biggr)\right] \oplus \left[\text{S-Box}^{-1}\biggl([c^2_{i,j} \oplus (k_{i,j} \oplus k_{0,0})] \oplus k_{0,0}\biggr)\right]$$
$$\oplus \left[\text{S-Box}^{-1}\biggl([\tilde{c}^1_{i,j} \oplus (k_{i,j} \oplus k_{0,0})] \oplus k_{0,0}\biggr)\right] \oplus \left[\text{S-Box}^{-1}\biggl([\tilde{c}^2_{i,j} \oplus (k_{i,j} \oplus k_{0,0})] \oplus k_{0,0}\biggr)\right] = 0,$$

is satisfied. Since $A \circ \text{S-Box}^{-1}(\cdot \oplus k_{0,0})$ is known (note that the affine layer $A(\cdot)$ plays no role), it is possible to exploit such information in order to find $k_{i,j} \oplus k_{0,0}$ by guessing $k_{i,j} \oplus k_{0,0}$ (2^8 values) and verifying that Equation (6) is fulfilled.

[15] The Rouché-Capelli theorem states that a system of linear equations in n variables has a solution if and only if the rank of its coefficient matrix is equal to the rank of its augmented matrix. Since we are assigning linearly independent values to the new variables and since the rank of the whole matrix is at least 247, the rank of the augmented matrix is always larger than or equal to the rank of the coefficient matrix. Thus, verifying that the rank of the coefficient matrix *(1)* increases when assigning a variable and *(2)* reaches 256 is sufficient for our purposes.

B.2 Computational Cost

We need the ciphertexts of $768 \cdot 4$ chosen plaintexts for our attack to work with high probability. Thus, the data complexity is $2^{\log_2(768\cdot 4)} \approx 2^{11.6}$. Finding the S-Box consists of matrix rank calculations and the solving step for the system of linear equations. The rank of an $m \times n$ matrix with coefficients in $\mathbb{F}_2$ can be found in $\mathcal{O}(mn^2)$ XOR operations involving single bits using Gaussian elimination. In our case, $n = 256$ and $m = 786$, therefore we need at most $2^{\log_2(768\cdot 256^2)} \approx 2^{25.6}$ single-bit XOR operations for the initial rank calculation, which amounts to $\approx 2^{22.6}$ 8-bit XOR operations. For the additional variable assignments, where we need to calculate the rank again for each assignment, note that we assign values to variables such that we can *(1)* reuse the previous matrix (which is now in row echelon form) and *(2)* minimize the number of operations needed by choosing variables efficiently. For example, when 10 trials are needed to find 9 suitable variables (which is sufficient in most cases according to our practical tests), we can choose them such that at most $2^{\log_2(10^3\cdot 10)} \approx 2^{13.3}$ XOR operations are needed (note that these potentially include 8-bit XOR operations now, since we are adding arbitrary elements of $\mathbb{F}_{2^8}$ to our augmented matrix).

We still need to account for the cost of solving the final system of linear equations. However, note that this is almost free, since our final matrix is already in row echelon form due to the previous computations.

In order to find the 15 key relations $k_{i,j} \oplus k_{0,0}$, we need $15\cdot 256\cdot 4 = 2^{13.91}$ table lookups, which amounts to about 2^8 3-round AES encryptions (assuming that the cost of one encryption round is approximately the same as the cost of 20 table lookups). Hence, the complexity of the whole attack is given by $\approx 2^{25.6}$ single-bit XOR operations (in order to find the rank of the first 768×256 matrix) and of 2^8 3-round AES encryptions (in order to find the 15 key relations $k_{i,j} \oplus k_{0,0}$).

Practical Verification. We implemented the attack in practice and both finding an S-Box representative and finding the key relations take less than 0.2 seconds on our tested machine (note that we do not count the time needed for the encryption oracle). We note that the bounds for XOR operations given in our theoretical estimation above are actually upper bounds. We expect the real number of operations to be lower, mainly because our initial systems are relatively sparse.

B.3 Note on an Attack on 4-Round AES with a Single Secret S-Box

As done in [16, Section 3.4], the previous attack can potentially be used against 4 rounds of AES instead of 3 rounds. Instead of finding the secret S-Box up to an affine equivalence, the idea is to find the secret *super-Sbox* up to an affine equivalence. In particular, the last 2 rounds can be written as a combination of a *super-Sbox* operation and affine operations (we refer to Section 4.1 for more details). When the secret *super-Sbox* operation is found (up to an affine equivalence), one can determine the secret S-Box and repeat the attack as before. We refer to [16, Section 3.4] for all the details.

Due to the same computation given in [16, Section 3.4], a system of linear equations for one *super-Sbox* now involves 2^{32} variables instead of 2^8 ones. This means that we need at least 2^{32} chosen plaintexts (or ciphertexts) to find it. Since the integral attack on 4-round AES with a single secret S-Box proposed in [16, Sect. 3.2] requires only 2^{16} chosen plaintexts, our attack cannot be more competitive than that. For this reason, we have decided not to present our attack in details. We remark that, while in the attack proposed in [16, Section 3.2] the attacker looks for an equivalence representation of the secret S-Box, in our attack the attacker must look for an equivalence representation of the secret *super-Sbox*.

C An Impossible Mixture Integral Distinguisher on 4-Round AES?

In Section 4.3, we proposed a new distinguisher on 3-round AES. Here we show that it does not seem to be possible to set up a similar distinguisher on 4-round AES.

As it is well-known from integral cryptanalysis, the relation

$$\begin{bmatrix} A & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \end{bmatrix} \xrightarrow{R^3(\cdot)} \begin{bmatrix} B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \end{bmatrix}$$

holds with prob. 1. Equivalently,

$$\bigoplus_{x \in (\mathcal{D}_0 \cap \mathcal{C}_0) \oplus a} R^3(x) = 0.$$

This property can be used to set up an integral key-recovery attack on 4-round AES. In particular, consider $p^i \in (\mathcal{D}_0 \cap \mathcal{C}_0) \oplus a$ for $i = 0, ..., 2^8 - 1$ and the corresponding ciphertexts $c^i = R^4(p^i)$ after 4 rounds. Assuming that the last MixColumns operation is omitted, it is well-known that the secret key k *must satisfy*

$$\bigoplus_{i=0}^{2^8-1} \text{S-Box}^{-1}(c^i_{j,l} \oplus k_{j,l}) = 0. \tag{13}$$

The crucial point here is that the secret key satisfies the previous equivalence with prob. 1. In other words, if the set of ciphertexts $\{c^i\}_i$ corresponds to the 4-round encryptions of $p^i \in (\mathcal{D}_0 \cap \mathcal{C}_0) \oplus a$, then there exists (at least) one key k that satisfies the previous equivalence.

If, however, there is no key for which the previous zero sum is satisfied, that is,

$$\forall k_{j,l} : \qquad \bigoplus_{i=0}^{2^8-1} \text{S-Box}^{-1}(c^i_{j,l} \oplus k_{j,l}) \neq 0,$$

then the ciphertexts $\{c^i\}_i$ are not generated by 4-round AES, but by a random permutation. However, to check this property we need to check the existence of a key. So the problem is to rewrite this property in order not to depend on the existence of the key.

To solve this issue, the idea is to look for values of $c^i_{j,l}$ for which Equation (13) does not have any solution k. This result is given by the following theorem.

Theorem 4. *Consider 2^8 chosen plaintexts p^i in $(\mathcal{D}_0 \cap \mathcal{C}_0) \oplus a$ and the corresponding ciphertexts $c^i = R^4(p^i)$ after 4 rounds for $i \in \{0, 1, \ldots, 2^8 - 1\}$. Then, for any $i, j \in \{0, 1, 2, 3\}$ the conditions*

1. *There exist $\alpha, \beta \in \{0, 1, \ldots, 2^8 - 1\}$, where $\alpha \neq \beta$ and s.t. $c^\alpha \oplus c^\beta \neq 0$.*
2. *For each $\gamma \in \{0, 1, \ldots, 2^8 - 1\} \setminus \{\alpha, \beta\}$, there exists a $\delta \in \{0, 1, \ldots, 2^8 - 1\} \setminus \{\alpha, \beta, \gamma\}$ s.t. $c^\gamma \oplus c^\delta = 0$.*

can never hold for 4-round AES, independently of the key, of the details of the S-Box, and of the MixColumns matrix.

Since the previous event can occur for a random permutation, it is possible to use it to distinguish 4-round AES from a random permutation.

Proof. We prove the previous result by contradiction. Assume there exist $j, k \in \{0, 1, 2, 3\}$ such that *(1)* there exists $\alpha, \beta \in \{0, 1, \ldots, 2^8 - 1\}$ s.t. $\alpha \neq \beta$ and s.t. $c^\alpha \oplus c^\beta \neq 0$ and *(2)* for each $\gamma \in \{0, 1, \ldots, 2^8 - 1\} \setminus \{\alpha, \beta\}$ there exists $\delta \in \{0, 1, \ldots, 2^8 - 1\} \setminus \{\alpha, \beta, \gamma\}$ s.t. $c^\gamma \oplus c^\delta = 0$. As we have just seen, it is not possible for 4-round AES that

$$\forall k_{j,l} : \qquad \bigoplus_{i=0}^{2^8-1} \text{S-Box}^{-1}(c^i_{j,l} \oplus k_{j,l}) \neq 0.$$

Due to the second assumption, it turns out that

$$\bigoplus_{i=0}^{2^8-1} \text{S-Box}^{-1}(c^i_{j,l} \oplus k_{j,l}) = \text{S-Box}^{-1}(c^\alpha_{j,l} \oplus k_{j,l}) \oplus \text{S-Box}^{-1}(c^\beta_{j,l} \oplus k_{j,l}),$$

since

$$c^\gamma \oplus c^\delta = 0 \quad \rightarrow \quad \text{S-Box}^{-1}(c^\gamma_{j,l} \oplus k_{j,l}) \oplus \text{S-Box}^{-1}(c^\delta_{j,l} \oplus k_{j,l}) = 0.$$

The results follow from the fact that

$$c^\alpha \oplus c^\beta \neq 0 \quad \rightarrow \quad \text{S-Box}^{-1}(c^\alpha_{j,l} \oplus k_{j,l}) \neq \text{S-Box}^{-1}(c^\beta_{j,l} \oplus k_{j,l})$$

for each $k_{j,l}$. As a consequence, there is no key that satisfies Equation (13), which is not possible for 4-round AES. □

In order to give details about the data complexity, we need to estimate how many different independent pairs we are able to construct.

Proposition 1. *Given $2N$ texts, it is possible to construct*

$$\prod_{i=1}^{N} (2 \cdot i - 1)$$

different independent pairs.

Proof. We prove this result by induction. For $2N = 2$, it is possible to construct just 1 set.

Assume that the result is true for $2N$. We prove the results for $2(N+1)$. Given texts $\{t^{(1)}, t^{(2)}, ..., t^{(2N+1)}, t^{(2N+2)}\}$, it is possible to construct $2N+1$ different pairs of texts that contain $t^{(1)}$, that is $(t^{(1)}, t^{(i)})$ for $i = 2, ..., 2N+2$. Now, given the texts $\{t^{(2)}, ..., t^{(2N+1)}, t^{(2N+2)}\} \setminus \{t^{1}\}$ of $2N+1$ texts, it is possible to construct

$$\prod_{i=1}^{N} (2 \cdot i - 1)$$

different pairs. As a result, it is possible to construct

$$(2N+1) \cdot \prod_{i=1}^{N} (2 \cdot i - 1) = \prod_{i=1}^{N+1} (2 \cdot i - 1)$$

different sets, which concludes the proof. □

Note that

$$\prod_{i=1}^{N} (2 \cdot i - 1) = (2 \cdot N)! \cdot \left(\prod_{i=1}^{N} (2 \cdot i) \right)^{-1} = \frac{(2 \cdot N)!}{2^N \cdot N!}.$$

Using the previous result, it turns out that the number of different sets of independent pairs of bytes that is possible to construct is given by

$$\prod_{i=1}^{2^7} (2 \cdot i - 1) = \frac{2^8!}{2^{128} \cdot 2^7!} \approx 2^{841.27},$$

where we use Stirling's Formula, i.e., $n! \approx n^n / e^n \cdot \sqrt{2\pi \cdot n}$.

Hence, for a fixed byte in row j and column l and for a *random permutation*, the probability of the event given in Theorem 4 is approximately

$$1 - \left[1 - (1 - 2^{-8}) \cdot (2^{-8})^{2^7 - 1} \right]^{2^{841.27}} \approx 1 - \left(1 - 2^{-1\,016}\right)^{2^{841.27}} \approx 1 - e^{-\frac{1}{2^{174.73}}}.$$

Indeed, two bytes are equal with prob. 2^{-8} and $2^7 - 1 = 127$ pairs of bytes must be equal in order to satisfy the assumption of Theorem 4. Note that we used the definition of Euler's number $\lim_{x \to \infty} (1 - x^{-1})^x = e^{-1}$ (where $\lim_{x \to \infty} (1 - x^{-1})^x \approx \lim_{x \gg 1} (1 - x^{-1})^x$).

Since there are 16 bytes, it follows that one needs to repeat the test at least $2^{174.73}/16 \simeq 2^{170.73}$ times in order to distinguish 4-round AES from a random permutation. This means that one needs more than the full code book to set up the distinguisher.

Open Problem. An open problem regards the possibility to "improve" Theorem 4, in order to consider more cases for which it is possible to distinguish 4-round AES from a random permutation without guessing the key.

D Impossible Mixture Integral Attack on 5-Round AES

As already mentioned in Section 6, the impossible mixture key-recovery attack on 4-round AES can be extended at the beginning to set up a 5-round attack.

The idea is very simple. As recalled in [10], each coset of a diagonal space $\mathcal{D}_0$ is mapped into a coset of a column space $\mathcal{C}_0$, independently of the secret key. Thus, the attacker chooses plaintexts in the same coset of $\mathcal{D}_0$ which is mapped into a coset of another subspace $\mathcal{C}_0$ after one round. Here, the attacker guesses the first diagonal of the secret key k^0, such that they can compute $R_{k^0}(\mathcal{D}_0 \oplus a)$. Then, the attacker divides the texts in sets $\mathfrak{T}$ and $\mathfrak{S}$, repeats the attack on 4-round AES, and filters wrongly guessed keys. Similar to what is done in mixture differential cryptanalysis [9,1], the crucial points of the attack are that *(1)* the way in which the couples of two (plaintext, ciphertext) pairs are divided in sets $\mathfrak{T}$ and $\mathfrak{S}$ depends on the (partially) guessed key and *(2)* the behavior of a set for a wrongly guessed key is (approximately) the same as in a random permutation, thus the attacker can filter wrong candidates for the key and finally finds the right one.

This means that for a wrongly guessed key, the texts are divided in sets $\mathfrak{T}$ and $\mathfrak{S}$ in a *random* way. As a result, for a wrongly guessed key, the property presented in Theorem 4 has probability different from zero, and one can exploit it to filter wrong key guesses.

Details of the Attack

In order to set up the attack, the attacker chooses $24 \simeq 2^{4.6}$ different sets $\mathfrak{T}_{\Psi,\Phi} = \{t^1 = (x^1, y^1, \Psi, \Phi), t^2 = (x^2, y^2, \Psi, \Phi)\}$ where $t^1, t^2 \in \mathcal{C}_0 \oplus b$ s.t. $t^1 \equiv (x^1, y^1, \Phi, \Psi), t^2 \equiv (x^1, y^2, \Phi, \Psi)$ as defined in Equation (8), exactly in the same way as proposed in Section 6.

Then, for each possible value of the first diagonal of the secret key $(k^0_{0,0}, k^0_{1,1}, k^0_{2,2}, k^0_{3,3})$, they partially decrypt the texts

$$p^i_{\Psi,\Phi} = a \oplus \underbrace{\begin{bmatrix} k^0_{0,0} & 0 & 0 & 0 \\ 0 & k^0_{1,1} & 0 & 0 \\ 0 & 0 & k^0_{2,2} & 0 \\ 0 & 0 & 0 & k^0_{3,3} \end{bmatrix}}_{\text{guessed key}} \oplus \text{S-Box}^{-1} \left[SR^{-1} \left(MC^{-1} \times \underbrace{\begin{bmatrix} x^i \; 0 \; 0 \; 0 \\ y^i \; 0 \; 0 \; 0 \\ \Psi \; 0 \; 0 \; 0 \\ \Phi \; 0 \; 0 \; 0 \end{bmatrix}}_{\equiv t^i_{\Psi,\Phi}} \right) \right]$$

for a random constant a and for $i = 1, 2$. Note that the corresponding texts p^i belong to a coset of $\mathcal{D}_0$, so it is sufficient for the attacker to guess just one diagonal of the secret key (we highlight that both the integral [5] and the impossible differential attack [2] on reduced-round AES work in the same way).

Then the attacker asks for the corresponding ciphertexts after 6 rounds. Using the key-recovery attack proposed in Section 6, they can find the final key k^5. In order to filter out wrongly guessed keys, they simply use these keys and the key schedule to compute the first diagonal of the first key, denoted by $\hat{k}^0$. If

$$diag^0(k^0) \neq diag^0(\hat{k}^0)$$

(where $diag^0(\cdot)$ denotes the first diagonal), the guessed keys are wrong. Since the previous condition is satisfied with prob. $1 - 2^{-32}$, all wrongly guessed keys are filtered.

Data and Computational Costs. Since the 4-round attack proposed in Section 6 must be repeated for every possible diagonal of the secret key k^0, the cost of the attack is roughly given by

$$\underbrace{2^{32}}_{\text{diag of } k^0} \cdot \Big(\underbrace{4 \cdot 24}_{\text{partial decryption}} \cdot \underbrace{2^{44.25}}_{\text{cost of the attack}} \Big) = 2^{82.8}$$

S-Box and table look-ups, which corresponds to $2^{76.2}$ 5-round AES encryptions.

A rough approximation of the data cost is given by 2^{32} chosen plaintexts. Even if this cost can be improved using the technique proposed in [1], these results are not very interesting on their own sake, as they are clearly inferior to, e.g., the improved Square attack on the same variant of AES [17], which has a data complexity of 2^8 chosen plaintexts and a computational cost of 2^{38} 5-round AES encryptions.