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

Lorenzo Grassi[1,2] and Markus Schofnegger[2]

[1] Radboud University, Nijmegen, The Netherlands
[2] IAIK, Graz University of Technology, Austria
l.grassi@cs.ru.nl, markus.schofnegger@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), namely 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). For a success probability of 95%, our distinguisher requires just 10 chosen plaintexts versus 20 chosen plaintexts necessary to set up the truncated differential attack.
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 exhaustive search, researchers have focused 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., [12,15,9,1]). 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

translate the (complex) "multiple-of-8" 5-round distinguisher [12] 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 the authors show 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 these 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

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

For this reason, low-data distinguishers/attacks on reduced-round ciphers have recently gained renewed interest in the literature. Indeed, it seems desirable to also consider other approaches, such as restricting the attacker's resources, in order to adhere to "real-life" scenarios. In this case, the time complexity of the attack is not limited (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 in various papers, which include low-data Guess-and-Determine and Meet-in-the-Middle techniques [3], low-data truncated differential cryptanalysis [11], polytopic cryptanalysis [17], and, if adaptive chosen plaintexts/ciphertexts are allowed, yoyo-like attacks [15].

"Mixture Integral" Key-Recovery Attacks. In Section 4 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 [5,14], 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

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.

Property	Prob	Data	Reference
Imp. Mixt. Integral	$\approx 65\%$	**6 CP**	**Section 3.2**
Trunc. Differential	$\approx 65\%$	12 CP	[11]
Imp. Mixt. Integral	$\approx 95\%$	**10 CP**	**Section 3.2**
Trunc. Differential	$\approx 95\%$	20 CP	[11]
Integral	$\approx 100\%$	$256 = 2^8$ CP	[5,14]
Yoyo	$\approx 100\%$	2 CP + 2 ACC	[15]

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 3.2, 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 success probability of $\approx 95\%$, *such a distinguisher requires only 10 chosen plaintexts (or ciphertexts), i.e., half of the data required by the most competitive distinguisher currently present in the literature* (which does not require adaptive chosen texts).

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). An integral attack assumes that the zero-sum property is always satisfied when decrypting under the secret key. Thus, if there is no key for which the zero-sum property is satisfied, the ciphertexts have likely 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 3.2, we show how to evaluate this property without guessing any key material by providing a property which is independent of the secret key, and which holds for the ciphertexts only in the case in which the key-recovery (mixture integral) attack just proposed fails. The obtained 3-round distinguisher can also be used to set up new key-recovery attacks on reduced-round AES.

AES with a Single Secret S-Box. Finally, in Section 5 we show that a competitive mixture integral attack can also be set up on reduced-round AES

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" (r full rounds + the final round). "Key sched." highlights whether the attack exploits the details of the key schedule of AES.

Attack	Rounds	Data (CP)	Cost	Memory	Key sched.	Reference
TrD	2.5 - 3	2	$2^{31.6}$	2^8	No	[11]
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	[11]
G&D-MitM	3	3	2^8	2^8	Yes	[3]
TrD	2.5 - 3	3	$2^{5.7}$	2^{12}	No	[11]
MixInt	**2.5 – 3**	**4**	$\mathbf{2^{8.1}}$	–	**No**	**Section 4.1**
MixInt	**2.5 – 3**	**4**	$\mathbf{< 1}$ $(+2^{36.1})$	$\mathbf{2^{28}}$	**No**	**Section 4.1**
TrD (EE)	3.5 - 4	2	2^{96}	–	Yes	[11]
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	[11]
G&D-MitM	4	4	2^{32}	2^{24}	Yes	[3]
MixInt	**3.5 – 4**	**6**	$\mathbf{2^{45.3}}$	–	**No**	**Section 4.2**
MixInt	**3.5 – 4**	**6**	$\mathbf{2^{33.3}}$ $(+2^{35.7})$	$\mathbf{2^{28}}$	**No**	**Section 4.2**
ImpPol	3.5 - 4	8	2^{38}	2^{15}	No	[17]

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

Table 3. *Comparison of attacks on reduced-round AES with a 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 finding 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	**2.5 – 3**	✓		$2^{11.6}$ CP	2^8 E + $2^{22.6}$ XOR	$2^{10.6}$	**Section 5**
TrD	2.5 - 3		✓	$2^{13.6}$ CP	$2^{13.2}$ XOR	small	[11]
I	2.5 - 3		✓	$2^{19.6}$ CP	$2^{19.6}$ XOR	small	[11]

TrD: Truncated Differential, I: Integral

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. They exploit a particular property of the MixColumns matrix (i.e., the fact that two elements for each row of the matrix are equal) to *directly* find the secret key (no information of the secret S-box is found/used).

Examples for attacks based on the first strategy are given in [7,18], while examples for attacks based on the second strategy are given in [16,11,8]. In Section 5 we exploit the first strategy in order to set up a competitive attack on 3-round AES 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 Algorithm 1, Algorithm 2, Algorithm 4, and Algorithm 5 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 online.[1]

2 Preliminaries

2.1 Brief Description of AES

AES [6] is a substitution-permutation network (SPN) 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 $\mathbb{F}_{2^8} \equiv \mathbb{F}_2[X]/(X^8 + X^4 + X^3 + X + 1)$. 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.

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.

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

The Notation Used in the Paper. Let x denote a plaintext, a ciphertext, an intermediate state, or a key. Then $x_{i,j}$ with $i, j \in \{0, \ldots, 3\}$ denotes the byte in the j-column of the i-row. We denote by R one round 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 $\mathcal{X}$, 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 $a \oplus b \in \mathcal{X}$.

2.2 Subspace Trail Notation for AES

Here we briefly recall the subspace trail notation for AES [11]. We work with vectors and vector spaces over $\mathbb{F}_{2^8}^{4\times 4}$, and we denote by $\{e_{0,0}, \ldots, e_{3,3}\}$ 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 1. *For $i \in \{0, 1, 2, 3\}$:*

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

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

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

Let $\mathcal{X}^c$ be the complementary space of $\mathcal{X}$. As shown in detail in [11],

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

Theorem 1 ([11]). *For each $I \subseteq \{0, 1, 2, 3\}$ and for each $a \in \mathcal{D}_I^c$, there exists one and only one $b \in \mathcal{M}_I^c$ 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 $x \oplus y \in X$. We hence give the following result.

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

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

All these results can be redescribed using a more "classical" truncated differential notation. E.g., if two texts t^1 and t^2 are equal except for the bytes in the i-th diagonal[2] for each $i \in I$, then they belong to the same coset of $\mathcal{D}_I$. A coset of $\mathcal{D}_I$

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

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 a notation that we largely use in the following.

Definition 3 ([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 $x_0, \ldots, x_{n-1} \in \mathbb{F}_{2^8}^{4\times 4}$ be a basis of $\mathcal{X}$, i.e., $\mathcal{X} \equiv \langle x_0, x_1, \ldots, x_{n-1}\rangle$, where $n = 4\cdot|I|$. Let t be an element of an arbitrary coset of $\mathcal{X}$, that is, $t \in \mathcal{X} \oplus a$. We say that t is "generated" by the generating variables $(t^0, \ldots, t^{n-1})$ (where $t^i \in \mathbb{F}_{2^8}$) – for the following, $t \equiv (t^0, \ldots, t^{n-1})$ – 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 $p \equiv a \oplus p^0 \cdot e_{0,0} \oplus p^1 \cdot e_{1,0} \oplus p^2 \cdot e_{2,0} \oplus p^3 \cdot e_{3,0}$.

3 Mixture Integral Distinguisher on 2-Round AES

3.1 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.

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*

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

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

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

or equivalently $R^2(p^1) \oplus R^2(p^2) \oplus R^2(\tilde{p}^1) \oplus R^2(\tilde{p}^2) = 0$ and

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

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 the 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. Moreover, we mention that a generalization of such a result, the *exchange attack*, has been presented in [2].

The property given in Eq. (1) is the starting point for our distinguisher and key-recovery attacks on reduced-round AES. Indeed, since this event occurs with prob. 2^{-128} if the ciphertexts are generated by a random permutation, it allows to set up a secret-key distinguisher for 2-round AES.

3.2 Impossible Mixture Integral Distinguisher on 3-Round AES

The zero-sum property given in Eq. (1) 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 Eq. (1) 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*

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

for each $j, l = 0, \ldots, 3$ as for a classical integral attack [5,14] (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 2. *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 Eq.* (2).

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. From Eq. (1),

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

where

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

and similarly for the other texts. Due to Eq. (1), the equality is always satisfied for $\hat{k}_{j,l} = k_{i,j}$. Hence, there exists at least one key that satisfies Eq. (2). □

If there is no key for which the previous condition is satisfied, 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 Eq. (2), then the ciphertexts $\{c^1, c^2, \tilde{c}^1, \tilde{c}^2\}$ are not the 3-round AES encryptions of $p^1, p^2, \tilde{p}^1, \tilde{p}^2$.

However, since our goal is 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 Eq. (2) 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 for 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*

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

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

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

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 event given in Eq. (3) occurs with a probability of around

$$1 - \left(\frac{256 \cdot 255 \cdot 254 \cdot 253 + 3 \cdot 256 \cdot 255 + 256}{256^4}\right)^{16} \approx 2^{-1.675} \approx 31.34\% \tag{4}$$

if the ciphertexts are generated by a random permutation[4] Π, where

(1) $256 \cdot 255 \cdot 254 \cdot 253 / 256^4$ corresponds to the probability of all bytes $[\Pi(p^\alpha)]_{i,j}$, $[\Pi(p^\beta)]_{i,j}$, $[\Pi(p^\gamma)]_{i,j}$,$[\Pi(p^\delta)]_{i,j}$ being different,
(2) $3 \cdot 256 \cdot 255 / 256^4$ corresponds to the probability of $[\Pi(p^\alpha)]_{i,j} = [\Pi(p^\beta)]_{i,j}$ and $[\Pi(p^\gamma)]_{i,j} = [\Pi(p^\delta)]_{i,j}$, where the first value is different from the second one, for all three independent combinations of $\alpha, \beta, \gamma, \delta \in \{1, 2, 3, 4\}$, and
(3) $256/256^4$ corresponds to the probability of all bytes being equal.

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 $i, j \in \{0, \ldots, 3\}$ and there exist pairwise distinct $\alpha, \beta, \gamma, \delta \in \{1, 2, 3, 4\}$ such that

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

[3] That is, we assume that $\alpha \neq \beta$, $\alpha \neq \gamma$, $\alpha \neq \delta$, $\beta \neq \gamma$, $\beta \neq \delta$, and $\gamma \neq \delta$.

[4] To be more precise, this is actually the probability for a random function. In particular, note that the event $\Pi(x) \oplus \Pi(y) = 0^{4\times 4}$ can never happen, or equivalently at least one byte of $\Pi(x) \oplus \Pi(y)$ is different from zero in the case in which $\Pi(\cdot)$ is a permutation for $x \neq y$. At the same time, since the probability that $\Pi(x) \oplus \Pi(y) = 0^{4\times 4}$ is 2^{-128}, such event just influences in a negligible way the probability given in Eq. (4).

According to Lemma 2, there exists at least one key k for 3-round AES that satisfies Eq. (2). Moreover, $[R^3(p^\alpha)]_{i,j} = [R^3(p^\beta)]_{i,j} \implies c^\alpha_{i,j} = c^\beta_{i,j}$, and hence

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

It follows that Eq. (2) 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 2. As a result, for all pairwise distinct $\alpha, \beta, \gamma, \delta \in \{1, 2, 3, 4\}$, the condition

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

can never hold for 3-round AES. □

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. $x^1 \neq x^2, y^1 \neq y^2$ arbitrary but fixed. Let $\mathfrak{T}^{x,y}_{\Psi,\Phi}$ be a set of two plaintexts defined by

$$\mathfrak{T}^{x,y}_{\Psi,\Phi} := \{p^1 = (x^1, y^1, \Psi, \Phi), p^2 = (x^2, y^2, \Psi, \Phi)\}, \tag{5}$$

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

$$\begin{aligned}\mathfrak{S} = \mathfrak{T}_{\Psi,\Phi} \cup \mathfrak{T}_{\psi,\phi} \equiv \big\{&p^1 = (x^1, y^1, \Psi, \Phi), p^2 = (x^2, y^2, \Psi, \Phi),\\ &p^3 = (x^1, y^1, \psi, \phi), p^4 = (x^2, y^2, \psi, \phi)\big\},\end{aligned} \tag{6}$$

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 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 $1 - (1 - 2^{-1.675})^N \geq 0.95$ if $N \geq 8$. In order to generate 8 sets $\mathfrak{S}$, one needs at least 5 different sets $\mathfrak{T}$, since $\binom{5}{2} = 10 \geq 8$, which results in a data cost of $5 \cdot 2 = 10$ chosen plaintexts.

Algorithm 1: Imp. mixture integral distinguisher on 3-round AES.

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 Eq. (5), and corresponding ciphertexts after 3 rounds.
Result: 1 if 3-round AES, 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* $i, j = 0, \ldots, 3$ **do**
    if $[a \oplus b]_{i,j} = 0$ **and** $[c \oplus d]_{i,j} \neq 0$ *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 *0 (random permutation)*
return *1 (3-round AES)*

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 2 even if $\Psi \neq \Psi'$ and $\Phi \neq \Phi'$. For completeness, if a success probability of 65% is sufficient, then one needs only 3 chosen plaintexts (by analogous computation, $1 - (1 - 2^{-1.675})^N \geq 0.65$ if $N \geq 3$, which implies that $N \geq 3$ sets $\mathfrak{S}$ are required, or equivalently 3 sets $\mathfrak{T}$, that is 6 chosen plaintexts).

Remark. The sets $\mathfrak{S}$ just defined are not independent. Here we explain why this is just a minor problem for the distinguisher. For simplicity, we work at byte level and consider three sets:

$$\mathfrak{X}_1 = \{a_0, a_1, b_0, b_1\}, \quad \mathfrak{X}_2 = \{a_0, a_1, c_0, c_1\}, \quad \mathfrak{X}_3 = \{b_0, b_1, c_0, c_1\},$$

where $a_0, a_1, b_0, b_1, c_0, c_1$ correspond to a (fixed position) byte of $[\Pi(p^i)]$ for several (related) plaintexts p^i.

Assume that both the sets $\mathfrak{X}_1$ and $\mathfrak{X}_2$ do *not* satisfy the event described in Eq. (3). Here we analyze the impact on $\mathfrak{X}_3$.

- If all bytes of $\mathfrak{X}_1$ and of $\mathfrak{X}_2$ are different, it is still possible that the set $\mathfrak{X}_3$ satisfies the event described in Eq. (3), e.g., if $b_0 = c_0$ and $b_1 \neq c_1$ (or vice-versa). Hence, the set $\mathfrak{X}_3$ can still satisfy the event in Eq. (3), even if both the sets $\mathfrak{X}_1$ and $\mathfrak{X}_2$ do *not* satisfy it.
- If all bytes of $\mathfrak{X}_1$ are equal (or, more generally, if $a_0 = a_1, b_0 = b_1, a_0 \neq b_0$) then $\mathfrak{X}_2$ does *not* satisfy the event described in Eq. (3) if and only if $c_0 = c_1$ as well. In this case, it follows that $\mathfrak{X}_3$ does *not* satisfy the event described in Eq. (3) with prob. 1. A similar conclusion holds if $a_0 = b_0, a_1 = b_1, a_0 \neq a_1$.

Note that the first case happens with a probability much larger than the second one (that is, with prob. $\frac{256 \cdot 255 \cdot 254 \cdot 253}{256 \cdot 255 \cdot 254 \cdot 253 + 3 \cdot 256 \cdot 255 + 256} \approx 99.99\%$ for each set $\mathfrak{X}_i$ where $i = 1, 2$). As a result, if the sets $\mathfrak{S}$ are generated as before (namely, reusing sets $\mathfrak{T}$), it follows that the real probability is actually smaller than $1 -$

$(1-2^{-1.675})^N$. However, our practical experiments show that the real probability is just slightly smaller than the approximated theoretical one given before. Hence, the approximated theoretical number of sets given before corresponds in many cases to the real number of sets required in practice (or it is just slightly smaller).

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 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, i.e., about 2^5 3-round AES encryptions (for 20 S-boxes $\approx$ 1-round[5]).

Before going on, we mention that this cost is *roughly of the same order* as the one required to set up a 3-round AES distinguisher based on the truncated differential property (see e.g. [11, Sect. 4.3] for more details).

Practical Verification. We implemented the distinguisher and found that by using 10 chosen plaintexts, it indeed successfully detects 3-round AES or a random permutation in around 95% of the cases. Our sample size was $300000 \approx 2^{18}$ and we used 21-round AES to simulate a random permutation. Moreover, by 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.

Note that using 6 chosen plaintexts, we would expect a success probability of $\approx 65\%$ using our approximated theoretical formula. As explained before, this small gap is due to the way in which the sets $\mathfrak{S}$ are constructed. A similar consideration also holds for the case in which 10 plaintexts are used, where the practical success probability is slightly smaller than the theoretical one.

A Similar Distinguisher on 4-round AES. In Appendix A, 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.

4 Mixture Integral Attacks on Reduced-Round AES

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

Eq. (1) is the starting point for a key-recovery attack on 3- and 4-round AES. The attack works in the same way as a classical integral key-recovery attack [5,14], with the crucial difference that it has a data cost of only 4 chosen plaintexts.

[5] 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 others currently present in the literature. This approximation is largely used in the literature (assuming that the linear/affine operations of each AES round are negligible in terms of costs).

Algorithm 2: Mixture integral key-recovery attack on 3-round AES.

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* κ *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**
        $A[i + 2^8 \times j + 2^{16} \times h + 2^{24} \times l] \leftarrow \kappa$.
for *each* $i, j = 0, \ldots, 3$ **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 in the main text for details)` **then**
    $k_{i,j} \leftarrow A[c^1_{i,j} + 2^8 \times c^2_{i,j} + 2^{16} \times \tilde{c}^1_{i,j} + 2^{24} \times \tilde{c}^2_{i,j}]$.
  else
    $k_{i,j}$ can take any possible value.
if *more than a single key passed the test* **then**
  Find the key by trying all remaining candidates.
return *Secret key* k

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

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

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, i.e., $c^1 = R^3(p^1), c^2 = R^3(p^2), \tilde{c}^1 = R^3(\tilde{p}^1)$ and $\tilde{c}^2 = R^3(\tilde{p}^2)$. Assume that the final MixColumns operation has been omitted[6]. Due to the proposed zero-sum distinguisher and working at byte level, we know that the secret key $k_{i,j}$ for each $i, j = 0, \ldots, 3$ satisfies

$$\begin{aligned} &\text{S-box}^{-1}(c^1_{i,j} \oplus k_{i,j}) \oplus \text{S-box}^{-1}(c^2_{i,j} \oplus k_{i,j}) \\ \oplus\ &\text{S-box}^{-1}(\tilde{c}^1_{i,j} \oplus k_{i,j}) \oplus \text{S-box}^{-1}(\tilde{c}^2_{i,j} \oplus k_{i,j}) = 0 \end{aligned} \tag{7}$$

with prob. 1 independently of the S-box. Since a wrongly guessed key satisfies the previous equality with a probability of 2^{-8}, it is possible to find the right one. The data cost of the attack is 4 chosen plaintexts, while the computational cost is 16 (= number of bytes) $\cdot\ 2^8$ (= number of $k_{i,j}$) $\cdot\ 4$ (= number of S-boxes) $= 2^{14}$ S-box operations, i.e., $2^{8.1}$ 3-round AES (assuming 20 S-boxes $\approx$ 1-round).

[6] If it is not omitted, since 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 $k = MC(k')$.

An Optimal Implementation of the Attack. The previous attack does not require any memory cost. In order to reduce the computational cost, another version of the attack can be considered. The idea is simply to generate a table with the solutions κ of the equation

$$\text{S-box}^{-1}(i \oplus \kappa) \oplus \text{S-box}^{-1}(j \oplus \kappa) \oplus \text{S-box}^{-1}(h \oplus \kappa) \oplus \text{S-box}^{-1}(l \oplus \kappa) = 0 \quad (8)$$

for each $i, j, h, l \in \mathbb{F}_{2^8}$.

We briefly analyze the number of solutions of Eq. (8). If $i = j = h = l$ or if $i = j$ and $h = l$ (where $j \neq h$ – analogous for the other six cases), then the previous equality is always satisfied for each κ. Moreover, due to the result presented in Section 3.2 we know that the case[7] $i = j$ and $h \neq l$ (analogous for the other six cases) can never occur for 3-round AES (where i, j, h, l correspond to $[R^3(p^\alpha)]_{x,y}, [R^3(p^\beta)]_{x,y}, [R^3(p^\gamma)]_{x,y}, [R^3(p^\delta)]_{x,y}$ in Theorem 3). Hence, in the case $i \neq j, i \neq h, i \neq l, j \neq h, j \neq l, h \neq l$ (useful for the attack), the average number of solutions is 1 (as shown in [10, Sect. 5.2] and by practical experiments).

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 – a complete pseudo code is given in Algorithm 2. 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.

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

The previous attack can be extended to 4-round AES using the same technique proposed in [5,14] 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 Eq. (6). We refer to Algorithm 3 for a complete pseudo code. Since the technique used to extend the attack is well-known in the literature, here we only analyze its cost. We remark that *no details of the key schedule* are used to set up the attack. All details of the attack are given in Appendix B.

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 last key, i.e., $(k^4_{0,0}, k^4_{1,3}, k^4_{2,2}, k^4_{3,1})$ and 4 bytes of the second-to-last key, i.e., $(k^3_{0,0}, k^3_{1,0}, k^3_{2,0}, k^3_{3,0})$ (analogous for the other four cases). Using the subkey bytes, after 2-round decryptions they can verify the zero-sum property on (at most) four bytes of the texts. Moreover, using only 4 chosen plaintexts, the probability the a key passes

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

Algorithm 3: Mixture integral key-recovery attack on 4-round AES (repeat an analogous attack to find the full key).

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 and let $SB(\cdot) \equiv$ S-box$(\cdot)$.
for *each* (i, j, h, l) *from* $(0, 0, 0, 0)$ *to* $(0xFF, 0xFF, 0xFF, 0xFF)$ *such that* $i \neq j, i \neq h, i \neq l, j \neq h, j \neq l, h \neq l$ **do**
  for *each* κ *from* 0 *to* $0xFF$ **do**
    if $SB^{-1}(i \oplus \kappa) \oplus SB^{-1}(j \oplus \kappa) \oplus SB^{-1}(h \oplus \kappa) \oplus SB^{-1}(l \oplus \kappa) = 0$ **then**
      $A[i + 2^8 \times j + 2^{16} \times h + 2^{24} \times l] \leftarrow \kappa$.
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* $i \in \{1, 2, 3\}$ **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 $[\tilde{d}^i_{0,0}, \tilde{d}^i_{1,0}, \tilde{d}^i_{2,0}, \tilde{d}^i_{3,0}]^T$)
  Let $g(\cdot, \cdot, \cdot, \cdot) : \mathbb{N}^4 \to \mathbb{N}$ be 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 $A[g(\tilde{c}^1_{0,0}, \tilde{d}^1_{0,0}, \tilde{c}^2_{0,0}, \tilde{d}^2_{0,0})] = A[g(\tilde{c}^1_{0,0}, \tilde{d}^1_{0,0}, \tilde{c}^3_{0,0}, \tilde{d}^3_{0,0})]$ **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**
          $\forall i = 0, 1, 2, 3 : \quad \hat{k}^3_{i,0} \leftarrow A[g(\tilde{c}^1_{i,0}, \tilde{c}^2_{i,0}, \tilde{d}^1_{i,0}, \tilde{d}^2_{i,0})]$.
$[k^3_{0,0}, k^3_{1,0}, k^3_{2,0}, k^3_{3,0}]^T \leftarrow MC \cdot [\hat{k}^3_{0,0}, \hat{k}^3_{1,0}, \hat{k}^3_{2,0}, \hat{k}^3_{3,0}]^T$.
return $(k^3_{0,0}, k^3_{1,0}, k^3_{2,0}, k^3_{3,0})$ *and* $(k^4_{0,0}, k^4_{1,3}, k^4_{2,2}, k^4_{3,1})$

the test is 2^{-32}, which means that the number of remaining keys is 2^{32} (= values of k^4) $\cdot\, 2^{32}$ (= values of k^3) $\cdot\, 2^{-32} = 2^{32}$. Without exploiting the key schedule, the attacker thus needs at least another pair of texts (p^3, q^3) to detect wrongly guessed keys (without using exhaustive search), for a total of 6 chosen plaintexts.

Computational Cost. First of all, a 1-round decryption costs 2^{32} (anti-diagonal of k^4) $\cdot\, 4$ (number of S-boxes) $\cdot\, 6$ (number of texts) $= 3 \cdot 2^{35}$ S-box lookups. For each anti-diagonal of k^4, the cost of finding 4 bytes of k^3 is $2 \cdot (1 + 2^{-8} + 2^{-16} + 2^{-24}) = 2$ table lookups. Indeed, note that since the probability that the first condition holds ($A[g(\tilde{c}^1_{0,0}, \tilde{d}^1_{0,0}, \tilde{c}^2_{0,0}, \tilde{d}^2_{0,0})] = A[g(\tilde{c}^1_{0,0}, \tilde{d}^1_{0,0}, \tilde{c}^3_{0,0}, \tilde{d}^3_{0,0})]$ in Algorithm 3) is 2^{-8}, it is 2^{-16} for both the first and the second condition, and so on. If the first condition is not satisfied, the attacker does not need to evaluate the others (and similarly for the other cases). Hence, the cost of finding the full key is around $4 \cdot 3 \cdot 2^{35} \cdot 2 = 3 \cdot 2^{38}$ S-box lookups, i.e., $2^{33.3}$ 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).

4.3 Impossible Mixture Integral Attack on 4-Round AES

In Section 3.2, we presented a new 3-round AES secret-key distinguisher which is independent of the key (namely, the impossible mixture integral distinguisher, which is different from the one just used for the key-recovery attacks just presented). Using the techniques just described, it is possible to set up a key-recovery attack on 4-round AES (see **??** for details). However, since the corresponding attack is not competitive w.r.t. other attacks in the literature, we only present its details in the extended version of the paper [13].

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

The 3-round mixture integral attack proposed in Section 4.1 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 [18]. The strategy consists of two steps:

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

5.1 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 make use of Eq. (7), namely $\bigoplus_{i=1,2}[\ \text{S-box}^{-1}(c^i_{0,0} \oplus k_{0,0}) \oplus \text{S-box}^{-1}(\tilde{c}^i_{0,0} \oplus k_{0,0})] = 0$. This is similar to what is done in [18], 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 [18], 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 sufficiently many linear equations to be able to determine $L \circ \text{S-box}^{-1}(\cdot \oplus a) \oplus b$. However, we are only able to determine

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

Algorithm 4: Mixture integral key-recovery attack on 3-round AES with a single secret S-box.

Data: $2^{9.6}$ sets of plaintexts $\mathfrak{S}$ defined as in Eq. (6), for a total of $2^{11.6}$ chosen plaintexts/ciphertexts $(p^i, c^i = R^3(p^i))$.
Result: S-box (up to affine layer) and secret key (up to 2^8 values).
1st Step: Finding the S-box.
Let $z_i := \text{S-box}^{-1}(i)$ for each $i = 0\text{x}00, \ldots, 0\text{x}FF$.
Use the plaintexts to generate a set of equations in $z_0, \ldots, z_{255}$ with rank 247:

$$\forall \text{ sets of plaintexts } \mathfrak{S}: \qquad \bigoplus_{x \in \mathfrak{S}} \text{S-box}^{-1}(x) = 0.$$

Solve it in order to find the S-box up to an affine layer: $A \circ \text{S-box}^{-1}(\cdot \oplus a)$.
2nd Step: Finding the secret key (up to 2^8 values).
Once the S-box is found (up to an affine layer), exploit Eq. (9) in order to find the key (analogous to the previous key-recovery attacks).
return *S-box (up to affine layer) and secret key (up to 2^8 values)*

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. This is sufficient 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 the last condition holds since $\mathcal{X}$ has an even number of elements and since A is an affine operation. This also implies that for the attack it is sufficient to recover the linear part of the affine operation A, since the constant b does not change the result of the sum due to the fact that $\mathcal{X}$ has an even number of elements (for this reason we only focus on the linear part in the following).

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 Eq. (7). By practical experiments, we found that using $3 \cdot 256 = 768 \approx 2^{9.6}$ different sets $\mathfrak{S}$ of pairs of texts as defined in Eq. (6) 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).[8]

With this set of equations and working as in [18], we now start to assign 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 try a different one instead, in order to avoid variable assignments which result in a system with no solu-

[8] We highlight that it seems not possible to simply re-use sets of texts $\mathfrak{T}$ defined as in Eq. (5) in order to decrease the total data complexity, since this has an impact on the rank of the generated sets of equations. We leave the open problem to analyze alternative strategies that allow to reduce the data complexity.

tions (this countermeasure is sufficient due to the Rouché-Capelli theorem[9]). 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. Such a set of variable assignments can easily be found after a small number of trials.

Finding the Key Given the S-Box. Given $A \circ \text{S-box}^{-1}(\cdot \oplus k_{0,0})$ for some unknown A and $k_{0,0}$, it is possible to find $k_{0,0} \oplus k_{i,j}$ for $0 \leq i \leq 3, 0 \leq j \leq 3$ (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

$$\begin{aligned} &\left[S^{-1}\left([c^1_{i,j} \oplus (k_{i,j} \oplus k_{0,0})] \oplus k_{0,0}\right)\right] \oplus \left[S^{-1}\left([c^2_{i,j} \oplus (k_{i,j} \oplus k_{0,0})] \oplus k_{0,0}\right)\right] \\ \oplus &\left[S^{-1}\left([\tilde{c}^1_{i,j} \oplus (k_{i,j} \oplus k_{0,0})] \oplus k_{0,0}\right)\right] \oplus \left[S^{-1}\left([\tilde{c}^2_{i,j} \oplus (k_{i,j} \oplus k_{0,0})] \oplus k_{0,0}\right)\right] = 0, \end{aligned} \tag{9}$$

is satisfied (see Eq. (2)), where $S(\cdot) = \text{S-box}(\cdot)$. 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 find $k_{i,j} \oplus k_{0,0}$ by guessing $k_{i,j} \oplus k_{0,0}$ (2^8 values) and verifying that Eq. (2) is fulfilled.

Computational Cost. As we have just seen, $2^{11.6}$ chosen plaintexts are needed for our attack to work with high probability. 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 $m = 768$ (the value of m is obtained via practical tests – see before), therefore we need at most $768 \cdot 256^2 \approx 2^{25.6}$ 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 recalculate the rank for each assignment, note that we assign values to variables such that we can *(1)* reuse the previous matrix (which is in row echelon form) and *(2)* minimize the number of operations needed by choosing variables efficiently.[10] For example, if 10 trials are needed to find 9 suitable variables (which is sufficient in most cases according to our tests), we can choose them such that at most $10^4 \approx 2^{13.3}$ XOR operations are needed (note that these 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 evaluate 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.

[9] 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 increases when assigning a variable and that it reaches 256 is sufficient for our purposes.

[10] For example, we can choose assignments with low hamming weight.

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 (without 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.

Acknowledgment. The authors thank the reviewers for their valuable comments. Lorenzo Grassi is supported by the European Research Council under the ERC advanced grant agreement under grant ERC-2017-ADG Nr. 788980 ESCADA.

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: CRYPTO 2018. LNCS, vol. 10992, pp. 185–212 (2018)
2. Bardeh, N.G., Rønjom, S.: The Exchange Attack: How to Distinguish Six Rounds of AES with 2ˆ88.2 Chosen Plaintexts. In: ASIACRYPT 2019. LNCS, vol. 11923, pp. 347–370 (2019)
3. Bouillaguet, C., Derbez, P., Fouque, P.A.: Automatic Search of Attacks on Round-Reduced AES and Applications. In: 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 Trans. Symmetric Cryptol. **2019**(1), 170–191 (2019)
5. Daemen, J., Knudsen, L.R., Rijmen, V.: The Block Cipher Square. In: 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. Dunkelman, O., Keller, N., Ronen, E., Shamir, A.: The Retracing Boomerang Attack. In: Advances in Cryptology - EUROCRYPT 2020. LNCS, vol. 12105, pp. 280–309 (2020)
8. Grassi, L.: MixColumns Properties and Attacks on (Round-Reduced) AES with a Single Secret S-Box. In: 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.: Rigorous Analysis of Truncated Differentials for 5-round AES. IACR Cryptol. ePrint Arch. p. 182 (2018)
11. Grassi, L., Rechberger, C., Rønjom, S.: Subspace trail cryptanalysis and its applications to AES. IACR Trans. Symmetric Cryptol. **2016**(2), 192–225 (2016)

12. Grassi, L., Rechberger, C., Rønjom, S.: A new structural-differential property of 5-round AES. In: EUROCRYPT 2017. LNCS, vol. 10211, pp. 289–317 (2017)
13. Grassi, L., Schofnegger, M.: Mixture Integral Attacks on Reduced-Round AES with a Known/Secret S-Box. IACR Cryptol. ePrint Arch. p. 772 (2019)
14. Knudsen, L.R., Wagner, D.A.: Integral Cryptanalysis. In: FSE 2002. LNCS, vol. 2365, pp. 112–127 (2002)
15. Rønjom, S., Bardeh, N.G., Helleseth, T.: Yoyo Tricks with AES. In: ASIACRYPT 2017. LNCS, vol. 10624, pp. 217–243 (2017)
16. Sun, B., Liu, M., Guo, J., Qu, L., Rijmen, V.: New Insights on AES-Like SPN Ciphers. In: CRYPTO 2016. LNCS, vol. 9814, pp. 605–624 (2016)
17. Tiessen, T.: Polytopic Cryptanalysis. In: EUROCRYPT 2016. LNCS, vol. 9665, pp. 214–239 (2016)
18. Tiessen, T., Knudsen, L.R., Kölbl, S., Lauridsen, M.M.: Security of the AES with a Secret S-Box. In: FSE 2015. LNCS, vol. 9054, pp. 175–189 (2015)

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

In Section 3.2, 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, where:

- A denotes an active byte (namely, for each pair of texts, the value in the byte in such position is different);
- C denotes a constant byte (namely, the value of the byte in such position is constant/fixed);
- B denotes a balance byte (namely, the sum of all texts is equal to zero in such byte).

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 $i = 0, \ldots, 2^8 - 1$ 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*

$$\forall j, l \in \{0, 1, 2, 3\} : \qquad \bigoplus_{i=0}^{2^8-1} \text{S-box}^{-1}(c^i_{j,l} \oplus k_{j,l}) = 0. \tag{10}$$

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 Eq. (10) does not have any solution k. This result is given by the following theorem.

Proposition 1. *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, the event defined by the following 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. *there exist $\{\gamma^1, \delta^1\}, \ldots, \{\gamma^{127}, \delta^{127}\}$ where $\gamma^i, \delta^i \in \{0, 1, \ldots, 2^8 - 1\} \setminus \{\alpha, \beta\}$ s.t.*
   (a) for each $i = 1, \ldots, 127$: $c^\gamma \oplus c^\delta = 0$;
   (b) for each $i \neq j$: $\gamma^i \neq \gamma^j$ and $\delta^i \neq \delta^j$;
   (c) for each i, j: $\gamma^i \neq \delta^j$;
   (hence, $\bigcup_i \gamma^i \cup \bigcup_i \delta^i = \{0, 1, \ldots, 2^8 - 1\} \setminus \{\alpha, \beta\}$)

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 exist $\alpha, \beta \in \{0, 1, \ldots, 2^8 - 1\}$ 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 for each $t = 1, \ldots, 127$:

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

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 Eq. (10), 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.

Lemma 3. *Given $2N$ texts, there are*

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

ways to split them into a union of different independent pairs of texts.

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 $2(N + 1) = 2N + 2$. Given texts $\{t^{(1)}, t^{(2)}, \ldots, t^{(2N+1)}, t^{(2N+2)}\}$, it is possible to construct $2N + 1$ different pairs of texts that contain $t^{(1)}$, i.e., $(t^{(1)}, t^{(i)})$ for $i \in \{2, 3, \ldots, 2N+2\}$. For each one of that pair, consider the texts$\{t^{(2)}, t^{(3)}, \ldots, t^{(2N+1)}, t^{(2N+2)}\} \setminus \{t^{(i)}\}$ of $2N$ texts. Due to the assumption of induction, it is possible to construct

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

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

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

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 Proposition 1 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 Proposition 1. 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" Proposition 1, in order to consider more cases for which it is possible to distinguish 4-round AES from a random permutation without guessing the key.

B Impossible Mixture Integral Attack on 4-Round AES

In Section 3.2, 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 [11] 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

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

where $\mathfrak{S} = \{p^1, p^2, p^3, p^4\}$ is defined as in Eq. (6) and where $R_f^3(\cdot)$ 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 hypothesis*[12], 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 $i = 1, \ldots, 4$ satisfy the property of Theorem 3

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

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

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 3.2) 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, 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.

Pseudo-Code. A pseudo-code of the attack is given in Algorithm 5.

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. $1-(1-2^{-8})^{4\cdot 6} \approx 8.97\%$ (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 Eq. (6) 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 Eq. (5) 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.

Algorithm 5: Impossible mixture integral attack on 4-round AES.

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 Eq. (5), 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 Ψ, Φ 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: 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* $(\Psi, \Phi) \neq (\psi, \phi)$*) and each* $i \in \{0, 1, 2, 3\}$ **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.

$flag \leftarrow 1$.

if $flag = 0$ **then**

return *Possible key candidate* $(k^4_{0,0}, k^4_{1,3}, k^4_{2,2}, k^4_{3,1})$

Repeat the same procedure for the next 3 anti-diagonals of the final key.

if *more than a single key passed the test* **then**

Find the key by trying all remaining candidates.

return *Final secret key k*

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.