IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2016, No. 2, pp. 192–225. DOI:10.13154/tosc.v2016.i2.192-225

Subspace Trail Cryptanalysis and its Applications to AES

Lorenzo Grassi[1], Christian Rechberger[1,3] and Sondre Rønjom[2,4]

[1] Institute of Applied Information Processing and Communications (IAIK), Graz University of Technology, Graz, Austria
[2] Norwegian National Security Authority (NSM), Sandvika, Norway
[3] Department of Applied Mathematics and Computer Science (DTU Compute), Technical University of Denmark, Kongens Lyngby, Denmark
[4] Department of Informatics, University of Bergen, Bergen, Norway
firstname.lastname@iaik.tugraz.at, Sondre.Ronjom@ii.uib.no

Abstract. We introduce subspace trail cryptanalysis, a generalization of invariant subspace cryptanalysis. With this more generic treatment of subspaces we do no longer rely on specific choices of round constants or subkeys, and the resulting method is as such a potentially more powerful attack vector. Interestingly, subspace trail cryptanalysis in fact includes techniques based on impossible or truncated differentials and integrals as special cases.
Choosing AES-128 as the perhaps most studied cipher, we describe distinguishers up to 5-round AES with a single unknown key. We report (and practically verify) competitive key-recovery attacks with very low data-complexity on 2, 3 and 4 rounds of AES. Additionally, we consider AES with a secret S-Box and we present a (generic) technique that allows to directly recover the secret key without finding any information about the secret S-Box. This approach allows to use e.g. truncated differential, impossible differential and integral attacks to find the secret key. Moreover, this technique works also for other AES-like constructions, if some very common conditions on the S-Box and on the MixColumns matrix (or its inverse) hold. As a consequence, such attacks allow to better highlight the security impact of linear mappings inside an AES-like block cipher.
Finally, we show that our impossible differential attack on 5 rounds of AES with secret S-Box can be turned into a distinguisher for AES in the same setting as the one recently proposed by Sun, Liu, Guo, Qu and Rijmen at CRYPTO 2016.

Keywords: AES · Invariant Subspace · Subspace Trail · Secret-Key Distinguisher · Key-Recovery Attack · Truncated Differential · Impossible Differential · Integral · Secret S-Box

1 Introduction

If a cryptographic primitive succumbs to a particular non-random behavior, it might be possible to distinguish it from what one would expect from sufficiently generic behavior. Invariant subspace cryptanalysis is a cryptanalytic technique that is powerful for certain block ciphers. If there exists an invariant subspace for the round function and for the key schedule, then this technique can be used to mount fast distinguishers and key recovery. This technique was introduced in [LAAZ11] at CRYPTO 2011 for the cryptanalysis of PRINTcipher. Its efficiency has also been demonstrated on the CAESAR candidate iSCREAM, on the LS-design Robin and on the lightweight cipher Zorro in [LMR15], and

The *extended version* of this paper can be found in [GRR16].

Received: 2016-06-01, Revised: 2016-09-01, Accepted: 2016-11-01, Published: 2017-02-03

on the block cipher Midori64 [GJN+15]. However, if such symmetries do not exist or are not found, invariant subspace cryptanalysis is not applicable. This leads to the natural question: *Can subspace properties still be used, even if no special symmetries or constants allow for invariant subspaces?* This paper will answer this question in the affirmative.

1.1 High-Level Overview of Subspace-Trail Technique

Our main contribution is the analysis of subspaces in SPNs (Substitution-Permutation Networks) constructions with a technique that can be seen as a generalization of invariant subspace attacks [LAAZ11, LMR15]. While invariant subspace cryptanalysis relies on iterative subspace structures, our analysis is concerned with *trails* of different subspaces[1]. To use an analogy, if invariant subspaces would correspond to iterative differential characteristics, then our method would be "subspace-counterpart" of differential characteristics.

In particular, we study the propagation of subspaces trough various building blocks like S-Boxes and linear layers. In that sense it has similarities with SASAS cryptanalysis [BS10], but also with Evertse's linear structures [Eve87], while another way to generalize invariant subspaces - called "nonlinear invariant attack" - has recently been introduced in [TLS16].

In this paper we investigate the behavior of subspaces in keyed permutations. At a high level, we fix subspaces of the plaintext that maintain predictable properties after repeated applications of a key-dependent round function. First we identify what we call *subspace trails* which is essentially a coset of a plaintext subspace that encrypts to proper subspaces of the state space over several rounds. The trails are formed by the affine hulls of the intermediate ciphertexts. Subspace trails typically consist of subspaces that increase in dimension for each round, meaning that if the plaintext subspace has low dimension in comparison to the block length, the subsequent subspaces dimension increases for each round. For byte-based ciphers (like AES), a quick and dirty test for subspaces is to compute the affine hulls of a n-round encryption (for a certain $n \geq 1$) of all values for each byte and then identify these subspaces. For bit-based ciphers, it is more important to determine what was coined a *nucleon* in [LMR15], that is candidate plaintext subspaces that seem to fit symmetries in the round function. Trails of affine hulls of the intermediate ciphertexts that grow slowly in dimension for each round, typically reflect slow diffusion in the round function. This is often the case for ciphers that iterate simple round functions many times. In this paper we will focus on what we call *constant dimensional subspace trails*, which are trails of cosets that preserve dimension over several rounds. We show how to connect two or more trails and form longer trails that preserve predictable structure. In particular, when we connect two trails we typically seek to describe an output coset of a first trail in terms of cosets of the input coset for the second trail.

To make the presentation more concrete, we focus on AES-128. The Rijndael block cipher [DR02] has been designed by Daemen and Rijmen in 1997 and accepted as the AES (Advanced Encryption Standard) since 2000 by NIST. Nowadays, it is probably the most used block cipher.

1.2 Contributions

There are four types of contributions in this paper. *Firstly* the **definition and description of the subspace trails technique**. As first examples, we describe it's application to secret-key distinguishers for up to 4 rounds of AES. In more details, the approach to the generalization of invariant subspace cryptanalysis to subspace trails is outlined in Sect. 2. In Sect. 3 we give technical preliminaries with respect to AES-like permutations, and in Sect. 3.3 we state central theorems related to subspace trails and their intersections. When

[1] Note that since we don't look for subspaces trails that are restricted to be invariant, the algorithm provided in [LMR15] is not suitable for finding subspace trails.

concretely applying it to AES, we describe in Sect. 4 distinguishers of round-reduced AES with a single unknown key up to 4 rounds. They correspond to known truncated differential, impossible differential, and integral distinguishers. From this it will become clear that these properties can be seen as special cases of subspace trails.

Secondly, in App. D of [GRR16] we describe *new low data-complexity key-recovery attacks on AES up to 4 rounds*, based on a combination of a truncated differential property (i.e. a relation among pairs of texts) and of properties of individual texts, which follows naturally from the proposed subspace trail approach.The relationship between these attacks and truncated differential cryptanalysis is discussed in App. D.3 of [GRR16].

Thirdly, in Sect. 5 and 7 we describe a **new and generic technique that can be used to attack AES with a *secret* S-Box**. Even if we do not improve the current best results in this model by Tiesen *et al.* [TKKL15], our technique allows (for the first time) to discover the secret key directly, without necessarily finding any equivalent representation or any other information about the S-Box. We show how not only integral attacks, but also truncated differential attacks and impossible differential attacks can exploit this technique. This technique can also be used to attack other AES-like block ciphers, if some very common conditions on the S-Box and on the MixColumns matrix (or its inverse) are guaranteed.

Finally, starting from the impossible differential attack on 5 rounds of AES with a secret S-Box, in Sect. 8 we describe a **new 5-round secret-key distinguisher for AES** in the same setting as the one presented in [SLG$^+$16]. A critically discussion of that particular distinguisher setting is proposed in Sect. 8.2.

Before starting with these detailed sections, we survey our concrete results: the distinguishers in the unknown (secret)-key model, and the key-recovery attacks in the cases of known and secret S-Box, and in both cases we compare them with earlier work.

1.3 Secret-Key Distinguishers for AES

The aim of a distinguishing attack is to find some properties of a cipher that random permutations don't have such that it is possible to distinguish a cipher from random permutations. In the usual security model, the adversary is given a *black box* (oracle) access to an instance of the encryption function associated with a random secret key and its inverse. The goal is to find the key or more generally to efficiently distinguish the encryption function from a random permutation.

In Table 1 we summarize the secret-key distinguishers for 1 up to 5 rounds. Such results often serve as a basis for key recovery attacks in the most relevant single-key setting. The subspace trail cryptanalysis includes as special cases of differential cryptanalysis techniques (like truncated or impossible differentials) and integral cryptanalysis, hence the complexities for distinguishers up to 4 rounds is the same.

The first distinguisher for five rounds of AES-128 has been proposed recently in CRYPTO 2016 [SLG$^+$16]. However, this distinguisher requires the *whole* input-output space to work, or less than the full codebooks if some knowledge of subkey bytes is assumed. In the same setting of this distinguisher, in Sect. 8 we propose our secret key distinguisher for five rounds of AES, which requires (much) less than the whole input-output space without any knowledge about subkeys. Since we derive this distinguisher in a natural way from the impossible differential attack on five rounds of AES with a secret S-Box, we introduce it in Sect. 1.6 together with the mentioned attack, and we focus for the moment only on the distinguishers up to four rounds.

Relation to other Distinguishers. The 1-, 2- and 3-round distinguishers presented in Sect. 4.1 and 4.3 exploit the same well-known structural properties that also truncated differentials exhibit. Using a different notation (namely the AES "Super S-Box"), 2-round subspace trails were already discovered and investigated in [DR06a] and [DR06b], with

Table 1: *AES secret-key distinguishers, working independent of key schedule.* Data complexity is measured in minimum number of chosen plaintexts CP or/and chosen ciphertexts CC which are needed to distinguish the AES permutation from a random permutation with probability higher than 95%. 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.

Rounds	Data	CP	CC	Property	Reference
1 - 1.5 - 2	2	×	×	Truncated Differential	[DR06a] - Sect. 4.1
2.5 - 3	$20 \simeq 2^{4.3}$	×	×	Truncated Differential	[BK07] - Sect. 4.3
2.5 - 3	2^8	×	×	Integral	[DKR97] - Sect. 4.3
3.5 - 4	$2^{16.25}$	×	×	Impossible Differential	[BK01] - Sect. 4.4
3.5 - 4	2^{32}	×	×	Integral	[DKR97] - Sect. 4.4
4.5 - 5	$2^{98.2}$	×		Impossible Differential	Sect. 8
5	2^{128}		×	Integral	[SLG+16]

the objective to understand how the components of the AES interact. In these papers, authors study the probability of differentials and characteristics over 2 rounds of AES, giving bound on the maximum differential probability.

In [DKR97], Daemen *et al.* proposed a new method that can break more rounds of SQUARE than differential and linear cryptanalysis, which is named the SQUARE attack consequently. In [KW02], Knudsen and Wagner proposed the integral cryptanalysis as a generalized case of such attacks. The first key-recovery attacks on round-reduced AES were obtained by exploiting a 3-round integral distinguisher to attack up to 6 rounds. A re-interpretation of this integral distinguisher (also commonly labeled as square distinguisher) using the subspace trail notation is proposed in Sect. 4.4.

Knudsen [Knu98] and Biham *et al.* [BBS99] independently proposed the impossible-differential cryptanalysis. This distinguisher exploits differential with probability zero, and it is re-proposed using the subspace trail notation in Sect. 4.4.

The subspace trail approach is mostly providing an alternative description of known properties under the umbrella of a single framework. However, there are other recent techniques that this approach does *not* seem to include. Recently integral distinguishers have been generalized by Todo [Tod15b] and in there also applied to AES-like primitives. Distinguishers for AES itself were not improved, but clear progress e.g. with MISTY cryptanalysis was demonstrated [Tod15a]. Todo's generalization can take S-Box properties into account, on the other hand the property exploited is still a type of zero-sum. Thus it complements our approach which is independent of the S-Box, but exploits properties more subtle than zero-sums. Subspace trails do not seem to capture other types of distinguisher. Among them are Polytopic distinguishers [Tie16], DS-type distinguishers [DS08a], or non-linear invariants [TLS16].

1.4 Low Data-Complexity Key-Recovery Attacks on AES

Since practical attacks on block ciphers became extremely rare in the last two decades, the approaches of the cryptanalysis community have been concentrating on attacking reduced-round variants of block ciphers and/or to allow the adversary more degrees of freedom in its control. In the first approach, the usual goal of the adversary is to maximize the number of rounds that can be broken, using less data than the entire codebook and less time than exhaustive key search. Attacks following such an approach are of importance, since they ensure that the block ciphers are strong enough and because they help to establish the security margins offered by the cipher. However, aiming for the highest number of rounds often leads cryptanalyst to attacks very close to brute force ones, or requiring completely impractical amounts of chosen/known inputs up to the full codebook. Practical attacks, especially those focusing on low data complexity, rightfully gained more attention recently, and this is also the focus of the key-recovery part in this paper.

State of the Art. AES with its wide-trail strategy was designed to withstand *differential* and *linear cryptanalysis* [DR02], so pure versions of these techniques have limited applications in attacks. Hence, it is widely believed that no regular differential attack can be mounted on more than 5 rounds of AES (see [PSC+02] for details). For achieving the highest number of rounds, the most effective single-key recovery methods are *impossible differential cryptanalysis* (which yielded the first attack on the 7-round AES-128 [ZWF07] with non-marginal data complexity) and *integral attacks* [DKR97]. Another attack that initially has obtained less attention than the previous ones is the Meet-in-the-Middle attack [DS08b], which has potential if enhanced by other techniques/attacks, as the differential attack [DKS10, DFJ13, DF13] or as the *bicliques technique* [BKR11].

In works like [BDD+12] authors consider *Low-Data Complexity* attacks on reduced-rounds of AES, that is they apply attacks assuming the attacker has limited resources, e.g. few plaintext/ciphertext pairs, which is often much more relevant in practice than attacks only aiming at the highest number of rounds. The results of this work have then been improved in [BDF11]. In that paper, authors set up tools which try to find attacks automatically by searching some classes of Guess-and-Determine and Meet-in-the-Middle attacks. These tools take as input a system of equations that describes the cryptographic primitive and some constraints on the plaintext and ciphertext variables. Then, they first run a search for an “ad hoc” solver for the equations to solve, build it, and then run it to obtain the actual solutions.

Another work in the low-data complexity scenario is the *Polytopic Cryptanalysis* presented in [Tie16], which is a generalization of differential cryptanalysis. In particular, the impossible polytopic cryptanalysis variant (that is, polytopic cryptanalysis that makes use of differentials with probability zero) was shown to allow competitive low-data attacks on round-reduced AES.

Our Key-Recovery Results

In this paper, we present key-recovery truncated differential attacks on reduced-round variants of AES-128 based on subspace trail cryptanalysis. A comparison of all known and relevant attacks on AES and our attacks presented in this paper is given in Table 2. To better understand this table, we highlight some aspects. Without going into the details here, AES is a key-iterated block cipher that consists of the repeated application of a round transformation on the state (called intermediate result). Each round transformation is a sequence of four steps. All the rounds are equal, except for the last one which is a slightly different. One of the steps that compose each round (the MixColumns operation) is omitted in the last round. The effect of the omission of the last round MixColumns has been studied in detail e.g. in [DK10], and often doesn’t affect the security of AES.

On the other hand, MitM-style attacks can sometimes work better when all rounds are the same. Since the attacks presented in [BDD+12] and found by the tool described in [BDF11] mainly exploit the MitM technique, they are sometimes affected by the presence of the final MixColumns, that is the data and the computational complexities are not equal if the final MixColumns is omitted[2]. In contrast, note that our attack (based on the truncated differential technique) is independent of the presence of the last MixColumns.

As the data complexity and number of rounds attacked is not always directly comparable, we re-ran the tool from [BDF11] in our settings. As a result, we are able to provide the

[2] As an example, the attack on 3 rounds with 2 chosen plaintexts has lower computational complexity and memory requirements when the final MixColumns is not omitted (2^{16} encryption and 2^8 of memory) rather than omitted (2^{24} encryption and 2^{16} memory).

Table 2: *Comparison of low-data attacks on round-reduced AES.* Data complexity is measured in number of required known/chosen plaintexts (KP/CP). Time complexity is measured in round-reduced AES encryption equivalents (E) and in memory accesses (M). Memory complexity is measured in plaintexts (16 bytes). The case in which the MixColumns operation is omitted in the last round is denoted by "r.5 rounds", that is r full rounds and the final round. New attacks are in bold.

Attack	Rounds	Data	Computation (E)	Memory	Reference
G&D-MitM	2.5	2 KP	2^{80}	2^{80}	[BDF11]
D-MitM	3	2 CP	2^{32}	2^{1}	[BDD+12]
TrD	**2.5 - 3**	**2 CP**	**2^{32} M +$2^{31.55}$ E ≈ $2^{31.6}$**	**2^{8}**	[GRR16] - App. D
G&D-MitM	2.5	2 CP	2^{24}	2^{16}	[BDF11]
G&D-MitM	3	2 CP	2^{16}	2^{8}	[BDF11]
TrD	**2.5 - 3**	**3 CP**	**$2^{11.2}$**	**1**	[GRR16] - App. D
G&D-MitM	3	3 CP	2^{8}	2^{8}	[BDF11]
TrD	**2.5 - 3**	**3 CP**	**2^{10} M +$2^{5.1}$ E ≈ $2^{5.7}$**	**2^{12}**	[GRR16] - App. D
D-MitM	4	2 CP	2^{104}	1	[BDD+12]
TrD (EE)	**3.5 - 4**	**2 CP**	**2^{96}**	**1**	[GRR16] - App. D.4
G&D-MitM	4	2 CP	2^{88}	2^{8}	[BDF11]
G&D-MitM	4	2 CP	2^{80}	2^{80}	[BDF11]
G&D-MitM	3.5	2 CP	2^{72}	2^{72}	[BDF11]
TrD (EE)	**3.5 - 4**	**3 CP**	**$2^{74.7}$**	**1**	[GRR16] - App. D.4
G&D-MitM	4	3 CP	2^{72}	2^{8}	[BDF11]
TrD (EE)	**3.5 - 4**	**3 CP**	**2^{76} M +2^{64} E ≈ $2^{69.7}$**	**2^{12}**	[GRR16] - App. D.4
G&D-MitM	4	4 CP	2^{32}	2^{24}	[BDF11]
D-MitM	4	5 CP	2^{64}	2^{68}	[BDD+12]
I-Pol	3.5 - 4	8 CP	2^{38}	2^{15}	[Tie16]
D-MitM	4	10 CP	2^{40}	2^{43}	[BDD+12]
TrD (EB)	**3.5 - 4**	**24 CP**	**$2^{40.6}$ M +$2^{33.9}$ E ≈ $2^{35.1}$**	**2^{17}**	[GRR16] - App. D.5
I	3.5 - 4	2^{9} CP	2^{14}	small	[DKR97]

G&D: Guess & Det., D: Diff., MitM: Meet-in-the-Middle, TrD: Truncated Differential, I: Integral, I-Pol: Imp. polytopic, EE: Extension at End, EB: Extension at Beginning.

computational cost of the best attack found by the tool for the case of 3.5 rounds (that is, 4 rounds of AES without the final MixColumns operation) using 2 chosen plaintexts.

Our attack on 3 rounds as described in App. D of [GRR16] is based on the property that a coset of a particular subspace $\mathcal{D}$ of the plaintexts space is always mapped into a coset of another particular subspace $\mathcal{M}$ after two rounds. Exploiting the particular shape of the subspace $\mathcal{M}$ and given two ciphertexts (which plaintexts belong to the same coset of $\mathcal{D}$), the right key is one of those such that these two ciphertexts belong to the same coset of $\mathcal{M}$ one round before. We show how to extend this approach in order to attack 4 rounds in App. D.4 of [GRR16] by extend our attack at the end, while in App. D.5 of [GRR16] we show how to extend it at the beginning.

1.5 Attack on AES with a Single Secret S-Box

The subspace trail framework also allows to consider attacks on 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. If the choice of the S-Box is made uniformly at random from all 8-bits S-Boxes, the size of the secret information increases from 128-256 bits (i.e. the key size in AES) to 1812-1940. Thus, this could be a good attempt to strengthen the cipher or all to reduce the number of rounds. Note that AES was designed in order to achieve good resistance against differential and linear cryptanalysis, and this includes the choice of the S-Box. However, a randomly chosen S-Box is very highly resistant against these attacks as well.

In [TKKL15], authors are able to attack up to 6-round of AES with identical and secret S-Box using techniques from integral cryptanalysis. Authors demonstrate that despite the increased size of the secret information in the cipher, they are able to recover both the secret key and the S-Box for the 4-round, 5-round and 6-round versions of AES-128. More precisely, authors are able to find the whitening key up to 256 variants, that is $(k_0, k_0 \oplus k_1, ..., k_0 \oplus k_{15})$ (where k_i is the i-th byte of the whitening key) for unknown k_0. We emphasize that to obtain this result, authors must determine the secret S-Box (up to an additive constant before and after the S-Box, i.e. S-Box$(x) \sim a\oplus$ S-Box$(b \oplus x)$) in order to find the key. In other words, using their technique it is not possible to find the key independent of the S-Box. To the best of our knowledge, this is the only work in the literature regarding attacks on AES with a secret S-Box.

However, several other results in literature consider (other) encryption schemes with secret part. PRESENT with a secret S-Box has for example been considered in [BKLT11, LJQ14]. One of the first work in this context has been presented by Biryukov and Shamir [BS01], who applied integral cryptanalysis to a generalized SPN structure denoted SASAS, which consists of three substitution layers separated by two affine layers. In their paper, the attacker is assumed not to have any knowledge about the linear layer or the S-boxes which are all allowed to be chosen independently at random. The SASAS attack recovers an equivalent representation of this SPN and thus allows decryption of any ciphertext. The attack allows to break the equivalent of three rounds of AES. A follow up work is [BBK14], where authors considered the ASASA scheme in order to design public key or white-box constructions using symmetric cipher components.

In all the previous works, an attacker must work both on the secret S-Box and on the secret key, that is she has to first find information on the secret S-Box in order to discover the secret key. Thus, a natural questions arise: *Is it also possible to directly find the secret key without having to discover any information about the secret S-Box?* In this paper, we show that it's possible if some (very common) assumptions are guaranteed. Using the subspace-trail framework, we present a *generic* technique to discover *directly* (i.e. without working on the S-Box) the secret key of AES up to some variants, and we show how it is exploited by a truncated differential attack in Sect. 6 (in particular, we consider 3 rounds of AES in App. F.1 of [GRR16] and 4 rounds in App. G of [GRR16]), by an impossible differential attack in Sect. 7 and by an integral attack in App. F.3 of [GRR16].

The assumptions required are that the S-Boxes are identical, that each row of the MixColumns matrix has two identical elements and that each row has these two identical element in different positions. An example is the MixColumns matrix of AES, or any cyclic matrix with two identical elements.

A comparison between this technique and the one presented in [TKKL15] is shown in Table 3. Even if the assumptions are the same (i.e. the assumption of secret and identical S-Box), our goals are different from the one of [TKKL15]. Similar to [TKKL15], using our attack it is only possible to find the whitening key up to $(256)^4 = 2^{32}$ variants, if no information about the S-Box are discovered and used. Anyway, these 2^{32} variants can be reduced up to 256, working also on the secret S-Box and using a strategy similar to the one of [TKKL15], as shown in detail in App. F.2.1 of [GRR16].

Finally, we recall the advice given in [SLG$^+$16] "*when design an AES-like cipher, it is better to choose those MDS matrices M_{MC} such that both M_{MC} and M_{MC}^{-1} do not have*

Table 3: *Comparison of attacks on round-reduced AES with secret S-Box.* Data complexity is measured in number of required chosen plaintexts/ciphertexts (CP/CC). Time complexity is measured in round-reduced AES encryption equivalents (E), in memory accesses (M) or XOR operations (XOR). 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. The symbol * denotes an attack that can *not* work independently of the S-Box and the key.

Attack	Rounds	Data	Computation	Memory	Reference
SASAS	2.5	2^{16} CP	2^{21} E	2^{16}	[BS01]
TrD	**2.5 - 3**	$\mathbf{2^{13.6}}$ **CP**	$\mathbf{2^{13.2}}$ **XOR**	**small**	**Sect. 6**
I	**2.5 - 3**	$\mathbf{2^{19.6}}$ **CP**	$\mathbf{2^{19.6}}$ **XOR**	**small**	[GRR16] - App. F.3
TrD	**3.5 - 4**	$\mathbf{2^{30}}$ **CP**	$\mathbf{2^{36}}$ **M** $\approx$ $\mathbf{2^{29.7}}$ **E**	$\mathbf{2^{30}}$	**Sect. 6**
I*	3.5 - 4	2^{16} CC	$2^{17.7}$ E	2^{16}	[TKKL15]
I*	3.5 - 4	2^{16} CP	$2^{28.7}$ E	2^{16}	[TKKL15] (see Sect. 3.5)
ImD	**4.5 - 5**	$\mathbf{2^{102}}$ **CP**	$\mathbf{2^{107}}$ **M** $\approx$ $\mathbf{2^{100.4}}$ **E**	$\mathbf{2^{8}}$	**Sect. 7**
I*	4.5 - 5	2^{40} CC	$2^{38.7}$ E	2^{40}	[TKKL15]
I*	4.5 - 5	2^{40} CP	$2^{54.7}$ E	2^{40}	[TKKL15] (see Sect. 3.5)
I	5	2^{128} CC	$2^{129.6}$ XOR	small	[SLG+16] - Sect. 8.1

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

identical elements in the same columns", which allows to protect the cipher against our attacks presented in this paper and in [SLG+16].

1.6 The 5-round Secret Key Distinguisher for AES-128

In [SLG+16], authors presented the first 5-round secret key distinguisher for AES-128, based on the balanced property.

This distinguisher is constructed in two steps. At first step, authors assume that some of the subkey bits are known. Using this knowledge, they show how to choose the ciphertexts such that the balanced property holds on 5 rounds of AES. This distinguisher requires 2^{120} texts if the difference of two bytes (i.e. 8 bits) of the subkey is known, or 2^{96} texts if the differences of four pairs of bytes (i.e. 32 bits) are known. In the second step, authors assume that no secret key material is known. The idea is basically to repeat the first step of the distinguisher for each possible values of the subkey bits used to choose the ciphertexts. For the AES case, when this guess is correct (i.e. when these guessed bits are equal to that of the secret key) the balanced property holds for 5 rounds, which surely occurs in an exhaustive search.

Note that this distinguisher requires all the input-output space to work, that is the data complexity is 2^{128} texts when no subkey byte is known[3]. Moreover, the distinguisher presented in [SLG+16] doesn't exploit the details of the S-Box (which can be considered secret), that is the ciphertexts are chosen independently of the definition of the S-Box, but it requires some assumptions on the MixColumns matrix (which are the same ones we described for the key recovery attacks on AES with secret S-Box).

As we show in Sect. 8, our impossible differential attack on 5 rounds of AES with a secret S-Box can be turned into a distinguisher *in the same setting* of the one proposed by [SLG+16]. In our case, we consider an impossible differential trail instead of the balance property. As in the CRYPTO paper, the idea is to check the existence of a key for which the impossible differential trail is satisfied. Note that with respect to a key recovery attack,

[3] This was also confirmed with Bing Sun via personal communication.

both our distinguisher and the one presented in the CRYPTO paper have the advantage that it is not necessary to find the entire key to distinguish the two cases, since a limited number of bytes (e.g. the XOR difference of two bytes) is sufficient. Moreover, as in [SLG+16], our distinguisher is independent by the details of the S-Box operation, but requires the same assumption on the MixColumn matrix M_{MC} (i.e. at least one column must have two identical elements). A critical discussion of these distinguishers is provided in Sect. 8.2, arguing that in some sense the quest for the first 5-round distinguisher is still open despite the recent results.

As maybe the most interesting aspect, this distinguisher provides a counter-example to the conjectures made in [SLG+16], besides the fact that it doesn't need the entire input-output space but only $2^{98.2}$ chosen plaintexts. Indeed, the distinguisher presented in [SLG+16] is constructed in the chosen-ciphertext mode, and only in the case in which MixColumns in the last round is not omitted. For this reason, authors claim that "*since the 5-round distinguisher for AES can only be constructed in the chosen-ciphertexts mode, the security margin for the round-reduced AES under the chosen-plaintext attack may be different from that under the chosen-ciphertext attack*". However, our distinguisher is constructed in the chosen-plaintexts setting, and it works independent of the last MixColumns operation. Hence it seems *there is no clear evidence that chosen-ciphertext security is less than chosen-plaintext security in AES*.

In Sect. 8.1 we show that also the distinguisher of [SLG+16] can be turned into a key recovery attack, while in Sect. 8.2 we critically discuss the model in which these distinguishers work.

1.7 Practical Verification

All results in the paper have been verified using a C/C++ implementation:

Secret-Key Distinguishers. We practically verified the secret-key distinguishers for up to 5 rounds, and we have found that the practical results are consistent with our theory. The source codes of the secret-key distinguishers can be found in [git16c].

Low-Data Complexity Key-Recovery Attacks on AES. We practically verified the low-data complexity attacks on 1, 2, 3 and 4 rounds. For the 3 rounds attack, one or two pairs of plaintexts (that is two or three different plaintexts) are sufficient to discover the key of the final round, as predicted. Since the attack on 4 rounds described in App. D.4 of [GRR16] has a very high computational cost, we have tested it in a different way, which is explained in detail with the presentation of the attack. The source codes of the low-data complexity attacks can be found in [git16b].

Key-Recovery Attacks on AES with a Secret S-Box. We practically verified the truncated differential attacks on AES with a secret S-Box on 3 and 4 rounds, and the integral attack on AES with a secret S-Box on 3 rounds. The experimental results are in according to our theory. In particular, considerations about the practical computational costs of these attacks (in comparison with the theoretical ones) are reported in Sect. 6. The source codes of the key-recovery attacks on AES with a single secret S-Box can be found in [git16a].

2 Subspace Trails

In this section, we recall the invariant subspace cryptanalysis of [LAAZ11, LMR15] (depicted in Fig. 1), and then we introduce the concept of subspace trails (Fig. 2).

Invariant subspace cryptanalysis can be a powerful cryptanalytic tool. Let F denote a round function in an iterative key-alternating block cipher $E_K(\cdot)$:

$$E_K(m) = k_n \oplus F(\ldots k_2 \oplus F(k_1 \oplus F(k_0 \oplus m)))),$$

where the round keys $k_0, ..., k_n$ are derived from the master key K using some key schedule f: $(k_0, ..., k_n) = f(K)$. Assume there exists a coset[4] $V \oplus a$ such that $F(V \oplus a) = V \oplus a'$. Then if the round key K resides in $V \oplus (a \oplus a')$, it follows that $F(V \oplus a) \oplus K = V \oplus a$ and we get an iterative invariant subspace.

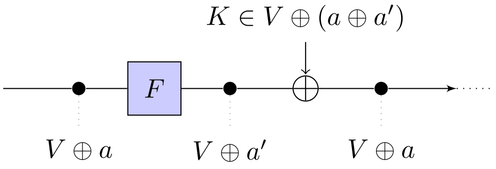

Figure 1: Invariant subspaces.

A slightly more powerful property can occur if for each a, there exists unique b such that $F_K(V \oplus a) := F(V \oplus a) \oplus K = V \oplus b$ meaning that the subspace property is invariant, but not the initial coset. That is, for each initial coset $V \oplus a$, its image under the application of F_K is another coset of V, in general different from the initial one. Equivalently, the initial coset $V \oplus a$ is mapped into another coset $V \oplus b$, where b depends on a and on the round key. In this paper, we generalize this concept and search for trails of subspaces. In the simplest case we look for pairs of subspaces V_1 and V_2 such that

$$F(V_1 \oplus a) \oplus K = V_2 \oplus b$$

holds for any constant a, that is for each a there exists unique b for which the previous equivalence is satisfied.

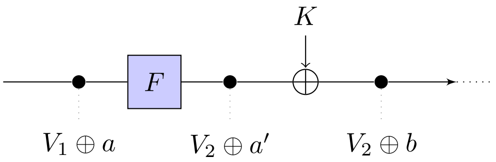

Figure 2: Trail of subspaces.

A *subspace trail* of length r is then simply a set of $r + 1$ subspaces $(V_1, V_2, \ldots, V_{r+1})$ that satisfy

$$F(V_i \oplus a_i) \oplus K \subseteq V_{i+1} \oplus a_{i+1}.$$

When the relation holds with equality, the trail is called a *constant-dimensional* subspace trail. In this case, if F_K^t denotes the application of t rounds with fixed keys, it follows that

$$F_K^t(V_1 \oplus a_1) = V_{t+1} \oplus a_{t+1}.$$

Definition 1. Let $(V_1, V_2, ..., V_{r+1})$ denote a set of $r + 1$ subspaces with $\dim(V_i) \leq \dim(V_{i+1})$. If for each $i = 1, ..., r$ and for each $a_i \in V_i^{\perp}$, there exist (unique) $a_{i+1} \in V_{i+1}^{\perp}$ such that

$$F_K(V_i \oplus a_i) \subseteq V_{i+1} \oplus a_{i+1},$$

then $(V_1, V_2, ..., V_{r+1})$ is *subspace trail* of length r for the function F_K. If all the previous relations hold with equality, the trail is called a *constant-dimensional subspace trail.*

[4] We recall the definition of coset, often used in the paper. Let W a vector space and V a subspace of W. If a is an element of W, a *coset* $V \oplus a$ of V in W is a subset of the form $V \oplus a = \{v \oplus a \,|\, \forall v \in V\}$.

Note that a_{i+1} depends on a_i and on the secret round key, but to simplify notation we use a_{i+1} instead of $a_{i+1}(a_i, k)$. With subspace structures at hand, we might ask questions about the probability that ciphertexts or sums of ciphertexts reside in certain subspaces, given that the plaintexts obey certain subspace structure (e.g. their sum is also in a fixed subspace). If the sum is over two texts this approaches resembles (truncated) differential cryptanalysis, if the sum is over more it can resemble integral cryptanalysis.

For AES-type block ciphers, we are typically not able to construct very long trails. In this case we can connect trails together and depending on the intersection properties of the endpoints of the trails, get predictable subspace properties for longer trails. However, in general these are not necessarily simple constant dimensional trails. In the following we describe subspace trail cryptanalysis and later-on distinguishers based on it. For sake of concreteness and better exposition, we focus on the case of AES. We'd like to emphasize that the properties described here extend almost immediately to any AES-like cipher with little modifications.

3 Preliminaries

3.1 Description of AES

The Advanced Encryption Standard [DR02] is a *Substitution-Permutation network* that supports key size 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 fields $\mathbb{F}_{256}$, defined using the irreducible polynomial $x^8 + x^4 + x^3 + x + 1$. Depending on the version of AES, N_r round are applied to the state: $N_r = 10$ for AES-128, $N_r = 12$ for AES-192 and $N_r = 14$ for AES-256. An AES round applies four operations to the state matrix:

- *SubBytes* (S-Box) - applying the same 8-bit to 8-bit invertible S-Box 16 times in parallel on each byte of the state (it provides the non-linearity in the cipher);
- *ShiftRows* (SR) - cyclic shift of each row (i-th row is shifted by i bytes to the left);
- *MixColumns* (MC) - multiplication of each column by a constant 4×4 invertible matrix over the field $GF(2^8)$ (it and ShiftRows provide diffusion in the cipher[5]);
- *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.

As we consider only AES with 128-bit key, we shall describe only its key schedule algorithm. The key schedule of AES-128 takes the user key and transforms it into 11 subkeys of 128 bits each. The subkey array is denoted by $W[0, ..., 43]$, where each word of $W[\cdot]$ consists of 32 bits and where the first 4 words of $W[\cdot]$ are loaded with the user secret key. The remaining words of $W[\cdot]$ are updated according to the following rule:

- if $i \equiv 0 \mod 4$, then $W[i] = W[i-4] \oplus RotByte(\text{S-Box}(W[i-1])) \oplus RCON[i/4]$,
- otherwise, $W[i] = W[i-1] \oplus W[i-4]$,

where $i = 4, ..., 43$, $RotByte$ rotates the word by 8 bits to the left and $RCON[\cdot]$ is an array of predetermined constant.

The Notation Used in the Paper. Let x denote a plaintext, a ciphertext, an intermediate state or a key. Then $x_{i,j}$ with $i, j \in \{0, ..., 3\}$ denotes the byte in the row i

[5] ShiftRows makes sure column values are spread and MixColumns makes sure each column is mixed.

and in the column j. We denote by k^r the key of the r-th round, where k^0 is the secret key. If only the key of the final round is used, then we denote it by k to simplify the notation. Finally, we denote by R one round of AES[6], while we denote i rounds of AES by $R^{(i)}$. If the MixColumns operation is omitted in the last round, then we denote it by R_f. As last thing, in the paper we often use the term "partial collision" (or, more simply, *collision*) when two texts belong to the same coset of a given subspace X.

3.2 Subspaces through One Round of AES

For a vector space V and a function F on $\mathbb{F}_{2^8}^{4\times 4}$, let $F(V) = \{F(v) \,|\, v \in V\}$ (as usual). For a subset $I \subseteq \{1, 2, \ldots, n\}$ and a subset of vector spaces $\{G_1, G_2, \ldots, G_n\}$, we define G_I as $G_I := \bigoplus_{i\in I} G_i$.

In the following we define four families of subspaces essential to AES: the diagonal spaces $\mathcal{D}_I$, the inverse-diagonal spaces $\mathcal{ID}_I$, the column spaces $\mathcal{C}_I$ and the mixed spaces $\mathcal{M}_I$. Since AES operates on 4×4 matrices over $\mathbb{F}_{2^8}$, then we work with vectors and vector spaces over $\mathbb{F}_{2^8}^{4\times 4}$ (that is, all the subspaces considered in the paper are subspace over $\mathbb{F}_{2^8}^{4\times 4}$). Moreover, we denote with $E = \{e_{0,0}, ..., e_{3,3}\}$ 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. (**Column spaces**) The *column spaces* $\mathcal{C}_i$ are defined as

$$\mathcal{C}_i = \langle e_{0,i}, e_{1,i}, e_{2,i}, e_{3,i} \rangle.$$

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

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

Definition 3. (**Diagonal spaces**) The *diagonal spaces* $\mathcal{D}_i$ are defined as

$$\mathcal{D}_i = SR^{-1}(\mathcal{C}_i) = \langle e_{0,i}, e_{1,i+1}, e_{2,i+2}, e_{3,i+3} \rangle$$

where the index $i + j$ is computed modulo 4. For instance, the diagonal space $\mathcal{D}_0$ corresponds to the symbolic matrix

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

Definition 4. (**Inverse-Diagonal spaces**) The *inverse-diagonal spaces* $\mathcal{ID}_i$ are defined as

$$\mathcal{ID}_i = SR(\mathcal{C}_i) = \langle e_{0,i}, e_{1,i-1}, e_{2,i-2}, e_{3,i-3} \rangle.$$

where the index $i - j$ is computed modulo 4. For instance, $\mathcal{ID}_0 = SR(\mathcal{C}_0)$ corresponds to the symbolic matrix

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

The last type of subspaces we define are called mixed subspaces.

[6] Sometimes we use the notation R_K instead of R to highlight that the round key is K.

Definition 5. (**Mixed spaces**) The i-th *mixed subspace* $\mathcal{M}_i$ is defined as

$$\mathcal{M}_i = MC(\mathcal{ID}_i).$$

These subspaces are formed by applying ShiftRows and then MixColumns to a column space. For instance, $\mathcal{M}_0$ corresponds to symbolic matrix

$$\mathcal{M}_0 = \left\{ \begin{bmatrix} \alpha \cdot x_1 & x_4 & x_3 & (\alpha+1)\cdot x_2 \\ x_1 & x_4 & (\alpha+1)\cdot x_3 & \alpha \cdot x_2 \\ x_1 & (\alpha+1)\cdot x_4 & \alpha \cdot x_3 & x_2 \\ (\alpha+1)\cdot x_1 & \alpha \cdot x_4 & x_3 & x_2 \end{bmatrix} \middle| \forall x_1, x_2, x_3, x_4 \in \mathbb{F}_{2^8} \right\}.$$

where 0x02 $\equiv \alpha$ and 0x03 $\equiv \alpha + 1$.

Definition 6. Given $I \subseteq \{0,1,2,3\}$ where $0 < |I| \leq 3$, we define:

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

The dimension[7] of any of the spaces $\mathcal{D}_I$,$\mathcal{ID}_I$,$\mathcal{C}_I$ and $\mathcal{M}_I$ is $4 \cdot |I|$. The essential subspaces in AES are built from diagonal spaces $\mathcal{D}_i$, inverse-diagonal spaces $\mathcal{ID}_i$, column spaces $\mathcal{C}_j$ and mixed spaces $\mathcal{M}_k$. There are four of each of these spaces, and direct sums of them result in higher-dimensional diagonal, inverse-diagonal, column and mixed spaces.

It is easy to see that SubBytes maps cosets of diagonal and column spaces to cosets of diagonal and column spaces. Since SubBytes operates on each byte individually and it is bijective, and since the bytes of column and diagonal spaces are independent, its only effect is to change the coset. It is also easy to see that ShiftRows maps a coset of a diagonal space to a coset of a column space, since diagonals are mapped to columns, and it maps a coset of a column space to a coset of an inverse-diagonal space. The effect of MixColumns to a columns space $\mathcal{C}_I \oplus a$ is simply to change the coset, since applying the MixColumns matrix to a column space $\mathcal{C}_i$ has no effect.

Lemma 1. *Let $I \subseteq \{0,1,2,3\}$ where $0 < |I| \leq 3$ and $a \in \mathcal{D}_I^{\perp}$. There exists unique $b \in \mathcal{C}_I^{\perp}$ such that*

$$R_K(\mathcal{D}_I \oplus a) = \mathcal{C}_I \oplus b.$$

Proof. As we have just seen, since SubBytes is bijective and operates on each byte independently, it simply changes the coset $\mathcal{D}_I \oplus a$ to $\mathcal{D}_I \oplus a'$, where $a'_{i,j} = \text{S-Box}(a_{i,j})$ for each $i, j = 0, ..., 3$. ShiftRows simply moves the bytes of $\mathcal{D}_I \oplus a'$ to a column space $\mathcal{C}_I \oplus b^{'}$, where $b' = SR(a')$. MixColumns affects only the constant columns, thus $MC(\mathcal{C}_I \oplus b^{'}) = \mathcal{C}_I \oplus MC(b') = \mathcal{C}_I \oplus b^{''}$. Key addition then changes the coset to $\mathcal{C}_I \oplus b$. □

This simply states that a coset of a sum of diagonal spaces $\mathcal{D}_I$ encrypt to a coset of a corresponding sum of column spaces $\mathcal{C}_I$ through one round. We recall that two different cosets $V \oplus a$ and $V \oplus b$ (i.e. $a \neq b$) of the same generic subspace V are *equivalent* (i.e. $V \oplus a \sim V \oplus b$) if and only if $a \oplus b \in V$. Thus, in the previous lemma (similarly in the following), b is unique with respect to this equivalence relationship.

Lemma 2. *Let $I \subseteq \{0,1,2,3\}$ where $0 < |I| \leq 3$ and $a \in \mathcal{C}_I^{\perp}$. There exists unique $b \in \mathcal{M}_I^{\perp}$ such that*

$$R_K(\mathcal{C}_I \oplus a) = \mathcal{M}_I \oplus b.$$

[7] Since AES is a byte-oriented encryption scheme, we consider the dimension of the subspace as the number of active and independent bytes. This implies for example that the dimension of the subspaces is constant through SubBytes and MixColumns operations.

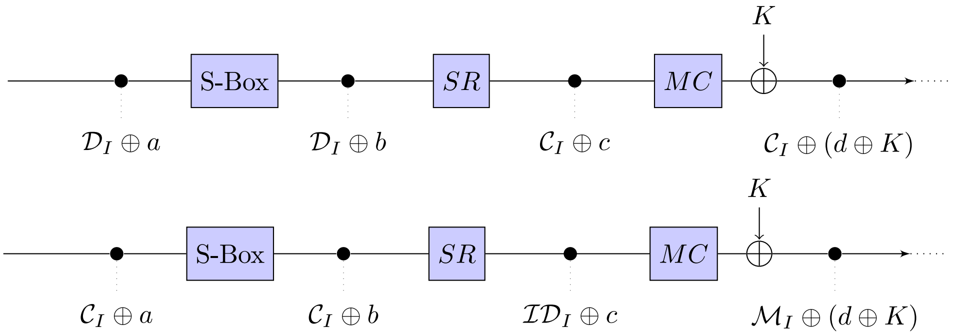

Figure 3: The essential subspaces in the AES round.

Proof. By definition 5, the mixed spaces $\mathcal{M}_I$ are defined as the application of the MixColumns operation to inverse-diagonal space $\mathcal{ID}_I$. Since a ShiftRows operation maps a column space to an inverse-diagonal space, a mixed space $\mathcal{M}_I$ is equivalently defined as the application of the linear layer in AES to column spaces $\mathcal{C}_I$. Since the SubBytes layer only moves a coset $\mathcal{C}_I \oplus a$ to a coset $\mathcal{C}_I \oplus a'$, it follows that for any fixed coset $\mathcal{C}_I \oplus a$, there exists $b \in \mathcal{M}_I^{\perp}$ such that $MC \circ SR \circ \text{S-Box}(\mathcal{C}_I \oplus a) \oplus K = \mathcal{M}_I \oplus b$, where $b = MC \circ SR(a') \oplus K$ and $a'_{i,j} = \text{S-Box}(a_{i,j})$ for each $i, j = 0, ..., 3$. □

Similarly to before, this simply states that a coset of a sum of column spaces $\mathcal{C}_I$ encrypts to a coset of the corresponding sum of mixed spaces $\mathcal{M}_I$ over one round.

3.3 Intersecting AES Subspaces

We continue with useful properties of AES subspaces. In this section we show the following: diagonal spaces and column spaces have non-trivial intersection, column spaces and mixed spaces have non-trivial intersection, but diagonal spaces and mixed spaces have only trivial intersection. This will be useful for creating subspace trails covering a higher number of rounds. For the following, let $I, J \subseteq \{0, 1, 2, 3\}$ and we assume that all the indexes are taken modulo 4. All the proofs are given in App. A of [GRR16].

Lemma 3. *$\mathcal{D}_i \cap \mathcal{C}_j = \langle e_{i+j,j} \rangle$ and $\mathcal{ID}_i \cap \mathcal{C}_j = \langle e_{i-j,j} \rangle$.*

It follows that $\mathcal{D}_I \cap \mathcal{C}_J = \langle e_{j+i,j} \,|\, i \in I, j \in J \rangle$ and $\mathcal{ID}_I \cap \mathcal{C}_J = \langle e_{i-j,j} \,|\, i \in I, j \in J \rangle$ ($j + i$ and $i - j$ are taken modulo 4), where the intersections have dimension $|I| \cdot |J|$.

Lemma 4. *$\mathcal{C}_i \cap \mathcal{M}_j = \langle MC(e_{j+i,i}) \rangle$.*

It follows that $\mathcal{C}_I \cap \mathcal{M}_J = \langle MC(e_{j+i,i}) \,|\, i \in I, j \in J \rangle$ ($i + j$ is taken modulo 4), which has dimension $|I| \cdot |J|$.

While the spaces $\mathcal{D}_I$ and $\mathcal{C}_J$, $\mathcal{ID}_I$ and $\mathcal{C}_J$, and $\mathcal{C}_I$ and $\mathcal{M}_J$ intersect non-trivially, the spaces $\mathcal{D}_I$ and $\mathcal{M}_J$ and the spaces $\mathcal{ID}_I$ and $\mathcal{M}_J$ intersect trivially. In particular:

Lemma 5. *$\mathcal{D}_I \cap \mathcal{M}_J = \mathcal{ID}_I \cap \mathcal{M}_J = \{0\}$ for all I and J such that $|I| + |J| \leq 4$.*

4 Distinguishers for 1, 2, 3 and 4 Rounds of AES with Secret Round-Keys

In this section we describe a series of subspace trails for AES. Additionally we also describe how these trails can be used to formulate ways to detect non-randomness, often colloquially referred to a distinguishers. All distinguishers in this section, ranging from two up to four

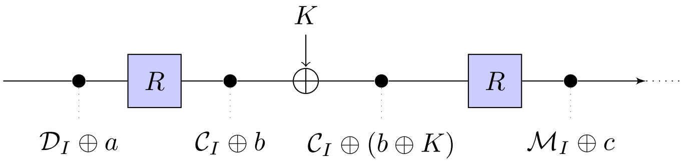

Figure 4: Subspaces over 2 rounds of AES.

rounds, are independent of the round keys and are formulated without the knowledge of the key. From now on, we assume that any subspaces $\mathcal{D}_I$, $\mathcal{C}_I$ or $\mathcal{M}_I$ has nonzero dimension (that is, $I \subseteq \{0, 1, 2, 3\}$ is not empty). Moreover, when we intersect two subspaces $\mathcal{D}_I$ and $\mathcal{M}_J$, where both I and J are assumed non-empty, we always assume that the sum of their dimensions is not larger than 16. Typically, the sum of their dimensions will be exactly 16.

4.1 2-Round Subspace Trail for AES

It follows directly from Section 3.2 that plaintexts from diagonal spaces are encrypted over two rounds to ciphertexts in mixed subspaces. Let $R^{(2)}$ denote two AES rounds with fixed random round keys $K = K_1, K_2$. Let $I \subseteq \{1, 2, 3, 4\}$ nonzero and fixed. By Lemma 1, a coset $\mathcal{D}_I \oplus a$ of dimension $4 \cdot |I|$ encrypts to a coset $R_{K_1}(\mathcal{D}_I \oplus a) = \mathcal{C}_I \oplus a'$ over one round. By Lemma 2, there exists unique b (relative to the round keys and the constant a') such that $R_{K_2}(\mathcal{C}_I \oplus a') = \mathcal{M}_I \oplus b$. By combining the two rounds, we get that for each $a \in \mathcal{D}_I^{\perp}$, there exists unique $b \in \mathcal{M}_I^{\perp}$ such that $R^{(2)}(\mathcal{D}_I \oplus a) = \mathcal{M}_I \oplus b$.

Consequently, we get the following properties. If two plaintexts belong to the same coset of a diagonal space $\mathcal{D}_I$, then their encryption belongs to the same coset of a mixed space $\mathcal{M}_I$. In particular, for a two round encryption R^2 with fixed keys, we have that

$$Pr(R^{(2)}(u) \oplus R^{(2)}(v) \in \mathcal{M}_I \,|\, u \oplus v \in \mathcal{D}_I) = 1 \qquad (1)$$

for nonzero set I of $\{0, 1, 2, 3\}$ (i.e. $|I| \neq 0$). The opposite follows directly: if two plaintexts belong to different cosets of a diagonal space $\mathcal{D}_I$, then their encryption belongs to different cosets of a mixed space $\mathcal{M}_I$. In other words

$$Pr(R^{(2)}(u) \oplus R^{(2)}(v) \in \mathcal{M}_I \,|\, u \oplus v \notin \mathcal{D}_I) = 0.$$

These properties are used to set up the distinguisher for two rounds. However, other interesting properties hold when one considers two rounds of encryption. In particular, by Lemma 5, the intersection between a mixed space $\mathcal{M}_I$ space and a diagonal space $\mathcal{D}_J$ space contains only zero, if $|I| + |J|$ is less than 4. Thus, if two plaintexts are in the same coset of $\mathcal{M}_I$, they must belong to different cosets of $\mathcal{D}_J$. In other words, for $\mathcal{D}_I$ and $\mathcal{D}_J$ such that $\dim(\mathcal{D}_I) + \dim(\mathcal{D}_J) \leq 16$ (and $|I|, |J| \neq 0$)

$$Pr(R^{(2)}(u) \oplus R^{(2)}(v) \in \mathcal{D}_J \,|\, u \oplus v \in \mathcal{D}_I) = 0 \qquad (2)$$

where $u \neq v$, since $R^{(2)}(u)$ and $R^{(2)}(v)$ are both in the same coset of $\mathcal{M}_I$ and thus are always in different cosets of $\mathcal{D}_J$. We can get similar results for the mixed spaces $\mathcal{M}_I$. In particular, if two plaintexts belong to the same coset of a mixed space $\mathcal{M}_I$, then their two round encryptions belong to different cosets of any mixed space $\mathcal{M}_J$. Indeed, two (different) elements of $\mathcal{M}_I$ belong to different cosets of $\mathcal{D}_J$ (since $\mathcal{M}_I \cap \mathcal{D}_J = \{0\}$). Since $R^{(2)}(u) \oplus R^{(2)}(v) \in \mathcal{M}_J$ if and only if $u \oplus v \in \mathcal{D}_J$, we obtain the desired result. Thus, for $\mathcal{M}_I$ and $\mathcal{M}_J$ such that $0 < \dim(\mathcal{M}_I) + \dim(\mathcal{M}_J) \leq 16$, we have that

$$Pr(R^{(2)}(u) \oplus R^{(2)}(v) \in \mathcal{M}_J \,|\, u \oplus v \in \mathcal{M}_I) = 0 \qquad (3)$$

```
Data: Pair of texts $c^1$ and $c^2$.
Result: $i$ such that $c^1 \oplus c^2 \in \mathcal{M}_i$, $-1$ otherwise.
$c \leftarrow MC^{-1}(c^1 \oplus c^2)$;
for $i$ from 0 to 3 do
    if $c_{(i+1)\%4,0} = 0$ AND $c_{(i+2)\%4,0} = 0$ AND $c_{(i+3)\%4,0} = 0$
     AND $c_{i,1} = 0$ AND $c_{(i+1)\%4,1} = 0$ AND $c_{(i+2)\%4,1} = 0$
     AND $c_{i,2} = 0$ AND $c_{(i+1)\%4,2} = 0$ AND $c_{(i+3)\%4,2} = 0$
     AND $c_{i,3} = 0$ AND $c_{(i+2)\%4,3} = 0$ AND $c_{(i+3)\%4,3} = 0$ then
        return $i$;
    end
end
return $-1$.
```

Algorithm 1: Pseudo-code for distinguisher of 2 rounds of AES.

if $u \neq v$. We'll use these probabilities to set up an efficient 4 rounds distinguisher.

A Concrete Distinguisher for 2 Rounds. As we have seen, if two plaintexts belong to the same coset of $\mathcal{D}_I$, then they belong to the same coset of $\mathcal{M}_I$ with probability 1 after two rounds - for each I. Consider instead two random texts. By simple computation, the probability that there exists I such that they belong to the same cosets of $\mathcal{M}_I$ is $\binom{4}{|I|} \cdot (2^8)^{-16+4\cdot|I|}$ (note that there are $\binom{4}{|I|}$ different subspaces $\mathcal{M}_I$). Setting $|I| = 1$, this probability is equal to 2^{-94}.

Thus, one pair of plaintexts (that is two texts) is sufficient to distinguish the random case from the other one. Indeed, on average in the random case we expect $2^{-94} \cdot 2 = 2^{-93} \simeq 0$ *collisions* (a "collision" occurs when two elements belong to the same coset of $\mathcal{M}_I$), while this number is always equal to 1 in the other case. The cost of this distinguisher is hence two texts. An equivalent distinguisher over 2 rounds was already introduced in [DR06b], where authors investigated how the components of the AES interact over 2 rounds.

Finally, note that a similar distinguisher can be used for the 1 round case. Indeed, note that if two plaintexts belong to the same coset of $\mathcal{D}_I$ (equivalently $\mathcal{C}_I$), then they belong to the same coset of $\mathcal{C}_I$ (equivalently $\mathcal{M}_I$) with probability 1 for each I after 1 round. Moreover, observe that it also is possible to set up a 2 rounds distinguisher using the impossible differential properties defined in (2) or (3).

4.2 Truncated Differential Key-Recovery Attacks for 3- and 4-round of AES

Before to go on, we highlight that in App. D of [GRR16] we present new key-recovery attacks for 3- and 4-round of AES that exploit the 2-round subspace trail of AES just presented.

For 3 rounds, the idea is simply to exploit the fact that two elements in the same coset of a diagonal space $\mathcal{D}_I$ belong to the same coset of a mixed space $\mathcal{M}_I$ after 2 rounds - see Prob. (1). Thus, given two plaintexts p^1 and p^2 in the same coset of $\mathcal{D}_I$ (that is $p^1 \oplus p^2 \in \mathcal{D}_I$) and the corresponding ciphertexts c^1 and c^2 after 3 rounds, the final key k must satisfy the following relationship:

$$R_k^{-1}(c^1) \oplus R_k^{-1}(c^2) \in \mathcal{M}_I.$$

In order to find the secret key and to minimize the data and the computational costs, these attacks exploit the shape of the mixed space $\mathcal{M}_I$, that is the facts that the columns of a coset of a mixed space $\mathcal{M}_I$ depend on different and independent variables and the relationships that hold among the bytes of the same column of $\mathcal{M}_I$. The attacks on 4 rounds are obtained extending at the end or at the beginning this attack on 3 rounds.

As highlight in App. D.3 of [GRR16], these attacks are truncated differential in nature, and are competitive with the other low-data complexity attacks present in literature, as the ones proposed in [BDF11] and [BDD$^+$12]. We refer to App. D of [GRR16] for a detailed analysis.

4.3 3-Round Subspace Trail for AES

There are several techniques that can be used to set up a 3-round distinguisher for AES, as for example (1) truncated differential, (2) balance property and (3) impossible differential. In this section, we only describe the truncated differential distinguisher using the subspace trail, which we'll be used to set up the attack on 4-round of AES with secret S-Box. The other two distinguishers based on the balance property and on the impossible differential are presented in details using the subspace trail in the next section. Note that the arguments in next section used for 4 rounds of AES holds also for the 3-round case.

The most competitive distinguisher on 3-round of AES is based on truncated differential trails, and an example of it is depicted in Fig. 5. In the following, we re-interpret it using the subspace trail.

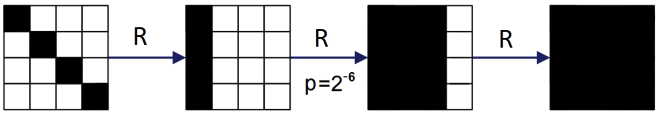

Figure 5: Truncated differential characteristic over 3-round AES. White box denotes a byte with a zero difference, while black box denotes a byte with a non-zero difference.

Consider a coset of $\mathcal{D}_I$ as starting point. After two rounds, this coset is mapped into a coset of $\mathcal{M}_I$ with probability 1. Indeed, as we have seen in Lemma 1, a coset of $\mathcal{D}_I$ is mapped into a coset of $\mathcal{C}_I$ with probability 1 after one round, and, as we have seen in Lemma 2, a coset of $\mathcal{C}_I$ is mapped into a coset of $\mathcal{M}_I$ with probability 1 after one round. Thus, if we consider two elements that belong to the same cosets of $\mathcal{D}_I$, after two rounds they belong in the same coset of $\mathcal{M}_I$ for sure. However, at the same time and with a certain probability, it is possible that these two elements belong to the same coset of $\mathcal{C}_J \cap \mathcal{M}_I \subseteq \mathcal{C}_J$ for a certain J after two rounds. In particular, the following proposition holds:

Proposition 1. *For any $\mathcal{M}_I$ and $\mathcal{C}_J$, we have that $Pr(x \in \mathcal{C}_J \,|\, x \in \mathcal{M}_I) = (2^8)^{-4|I|+|I|\cdot|J|}$.*

The proof can be found in App. A of [GRR16]. That is, if two elements belong to the same coset of $\mathcal{M}_I$, then they belong to the same coset of $\mathcal{C}_J$ with probability $(2^8)^{-4|I|+|I|\cdot|J|}$. More precisely, given two texts in the same coset of $\mathcal{D}_I$, after two rounds they belong to the same coset of $\mathcal{M}_I \cap \mathcal{C}_J$ with probability $(2^8)^{-4|I|+|I|\cdot|J|}$ (where $\mathcal{M}_I \cap \mathcal{C}_J \subseteq \mathcal{C}_J$). As we have just seen, a coset of $\mathcal{C}_J$ is mapped into a coset of $\mathcal{M}_J$ after one round. It follows that if two elements belong to the same coset of $\mathcal{D}_I$, the probability that they belong to the same coset of $\mathcal{M}_J$ after three rounds is equal to $(2^8)^{-4|I|+|I|\cdot|J|}$. The case $|I| = 1$ and $|J| = 3$ is depicted in Fig. 5.

For a more detailed explanation using subspace trail, consider the following argument. Given a coset of $\mathcal{M}_I$, it can be seen as a union of coset of $\mathcal{C}_J$, that is:

$$\mathcal{M}_I \oplus a = \bigcup_{x \in \mathcal{M}_I \oplus a \setminus \mathcal{C}_J} \mathcal{C}_J \oplus x,$$

as depicted in Fig. 6. In particular, note that the number of $x \in \mathcal{M}_I \oplus a \setminus \mathcal{C}_J$ is exactly $(2^8)^{4\cdot|I|-|I|\cdot|J|}$. Given two elements in the same coset of $\mathcal{D}_I$, then after two rounds they

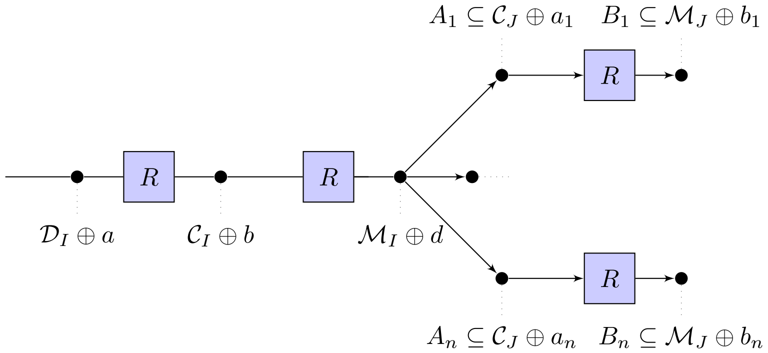
$A_1 \subseteq \mathcal{C}_J \oplus a_1$
$B_1 \subseteq \mathcal{M}_J \oplus b_1$
R
R
R
R
$\mathcal{D}_I \oplus a$
$\mathcal{C}_I \oplus b$
$\mathcal{M}_I \oplus d$
$A_n \subseteq \mathcal{C}_J \oplus a_n$
$B_n \subseteq \mathcal{M}_J \oplus b_n$

Figure 6: 3-round distinguishers for AES (the index n is defined as $n := (2^8)^{4\cdot|I|-|I|\cdot|J|}$).

belong to the same coset of $\mathcal{M}_I$. Since a coset of $\mathcal{M}_I$ can be seen as the union of $(2^8)^{4\cdot|I|-|I|\cdot|J|}$ cosets of $\mathcal{C}_J$, the probability that these two elements belong to the same coset of $\mathcal{C}_J$ after two rounds is exactly $(2^8)^{-4\cdot|I|+|I|\cdot|J|}$. Also in this way, one obtains the previous result.

Moreover, note the a similar result can be obtained in the decryption direction. That is, if two elements belong to the same coset of $\mathcal{M}_I$, then they belong to the same coset of $\mathcal{D}_J$ three rounds before with probability $(2^8)^{4\cdot|I|-|I|\cdot|J|}$. Finally and only for completeness, it is possible to obtain the same result considering the intersection of $\mathcal{C}_I$ and $\mathcal{D}_J$ after one round, instead of the intersection of $\mathcal{M}_I$ and $\mathcal{C}_J$ after two rounds. All the details of this (analogous) case are given in App. B.1 of [GRR16].

A Concrete Distinguisher for 3 Rounds. In order to set up the distinguisher, we exploit the difference of probability to have a collision in the ciphertexts set between the case in which two plaintexts are taken in a random way and the case in which two plaintexts belong to the same coset of $\mathcal{D}_I$.

The probabilities that two elements drawn randomly from $\mathbb{F}_{2^8}^{4\times 4}$ (denoted by p_1) and that two plaintexts drawn from a coset of $\mathcal{D}_I$ (denoted by p_2) belong to the same coset of $\mathcal{M}_J$ are respectively:

$$p_1 = \binom{4}{|J|} \cdot (2^8)^{-16+4|J|}, \qquad p_2 = \binom{4}{|J|} \cdot (2^8)^{-4|I|+|I||J|}.$$

It is very easy to observe that the probability to have a collision in the second case is higher than in the random case. In particular, for $|J| = 3$ and $|I| = 1$, we obtain that $p_2 = 2^{-6}$ while $p_1 = 2^{-30}$. Thus, the idea is to look for the minimum number of texts m in order to guarantee at least one collision in the "subspace case" and zero in the random case (with high probability).

To do this, we recall the *birthday paradox*. Given d (equally likely) values and n variables, the probability that at least two of them have the same value is given by:

$$p = 1 - \frac{n!}{(n-d)! \cdot n^d} = 1 - \frac{(d)!}{n^d} \cdot \binom{n}{d} \simeq 1 - e^{\frac{-d(d-1)}{2n}}, \tag{4}$$

where the last one is an useful approximation.

Since if we encrypt two plaintexts from a coset of $\mathcal{D}_I$, each of them can only belong to one of the 2^8 cosets of $\mathcal{M}_J$ defined as before, the probability that there is at least one collision in a coset is equal to the probability that two elements belong to the same cosets of $\mathcal{M}_J$, that is $p = 1 - e^{-m(m-1)/(2\cdot 2^8)}$. However, this property holds if we choose any of the four 12-dimensional space $\mathcal{M}_J$ as a target distinguisher space, each yielding

```
Data: 20 texts $c^i$ (for $i = 1, ..., 20$).
Result: number of collisions.
$n \leftarrow 0$;
for each pair $(c^i, c^j)$ with $i \neq j$ do
    $c \leftarrow MC^{-1}(c^i \oplus c^j)$;
    for $k$ from 0 to 3 do
        if $c_{k,0} = 0$ AND $c_{(3+k)\%4,1} = 0$ AND $c_{(2+k)\%4,2} = 0$ AND $c_{(1+k)\%4,3} = 0$
         then
            $n \leftarrow n + 1$;
            next pair
        end
    end
end
return $n$.
```

Algorithm 2: *Distinguisher for 3-round of AES - Pseudo-code.*

an independent experiment. Since this experiments are independent, we have that the probability to have at least one collision in the subspace case given m texts is:

$$p = 1 - \left(\frac{2^8!}{(2^8 - m)! \cdot (2^8)^d} \right)^4 \simeq 1 - \left(e^{\frac{-m(m-1)}{2 \cdot 2^8}} \right)^4 = 1 - e^{\frac{-m(m-1)}{2 \cdot 2^6}}.$$

Thus, if we set $m = 20$, the probability to have at least one collision in one of the four different $\mathcal{M}_J$ spaces (with $|J| = 1$) is 95.25% (14 texts are sufficient to have at least one collision with probability greater than 75%). In order to distinguish the two sets (that is, the random one and the "subspace" one), the verifier has to construct all the possible pairs of texts and to count the number of collisions, for each of them. In particular, given 20 texts (that is, 190 different pairs), we expect $190 \cdot 2^{-6} \simeq 3$ collisions in the subspace case and $190 \cdot 2^{-30} = 2^{-22.4} \simeq 0$ in the random case. Finally, observe that the distinguisher works in similar way in the decryption direction, with the same complexity.

4.4 4-Round Subspace Trail for AES

As for 3-round of AES, there are several techniques that can be used to set up a 4-round distinguisher for AES, as (1) impossible differential and (2) balance property. In the following, we present the 4-round impossible differential distinguisher in details, while the description of the distinguisher based on the balance property is provided in App. B.2 of [GRR16]. In both cases, the same analysis holds also for 3-round of AES.

From now on, we assume that I and J satisfy the condition $0 < |I| + |J| \leq 4$ (in order to use Lemma 5). To set up the 4-round impossible differential distinguisher, we start from the 2-round differential ones. Fix $\mathcal{D}_I$ and $\mathcal{D}_J$ such that $0 < \dim(\mathcal{D}_I) + \dim(\mathcal{D}_J) \leq 16$. We can construct a four round trail by simply combining two-round subspaces properties. Indeed, we have seen that

$$Pr(R^{(2)}(u) \oplus R^{(2)}(v) \in \mathcal{M}_I \,|\, u \oplus v \in \mathcal{D}_I) = 1, \; Pr(R^{(2)}(u) \oplus R^{(2)}(v) \in \mathcal{M}_J \,|\, u \oplus v \in \mathcal{M}_I) = 0$$

if $u \neq v$. Combining these two probabilities for 2-round yields a 4-round probability

$$Pr(R^{(4)}(u) \oplus R^{(4)}(v) \in \mathcal{M}_J \,|\, u \oplus v \in \mathcal{D}_I) = 0 \qquad (5)$$

where $u \neq v$. This means that the adversary can pick any coset of a non-zero plaintext space $\mathcal{D}_I$ and a non-zero ciphertext space $\mathcal{M}_J$, as long as $0 < \dim(\mathcal{D}_I) + \dim(\mathcal{M}_J) \leq 16$, and distinguish on the fact that the probability that two plaintexts encrypt to the same

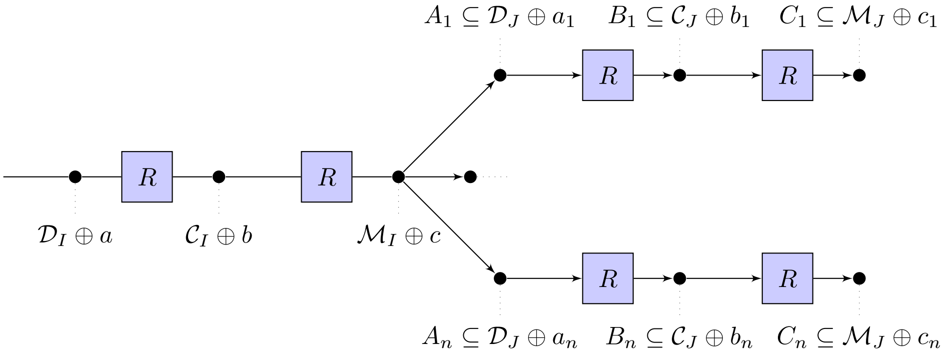
$A_1 \subseteq \mathcal{D}_J \oplus a_1$
$B_1 \subseteq \mathcal{C}_J \oplus b_1$
$C_1 \subseteq \mathcal{M}_J \oplus c_1$
R
R
R
R
R
R
R
R
$\mathcal{D}_I \oplus a$
$\mathcal{C}_I \oplus b$
$\mathcal{M}_I \oplus c$
$A_n \subseteq \mathcal{D}_J \oplus a_n$
$B_n \subseteq \mathcal{C}_J \oplus b_n$
$C_n \subseteq \mathcal{M}_J \oplus c_n$

Figure 7: 4-round distinguishers for AES (where the index n is defined as $n := (2^8)^{4|I|}$ and the indexes I and J satisfy the condition $0 < |I| + |J| \leq 4$).

coset of the ciphertext space is zero over four rounds.

A Concrete Distinguisher for 4 Rounds. The idea is pick parameters that maximize probability in the random case. The best minimal data complexity is found if we choose $|J| = 3$. This implies that $|I| = 1$, since we have the condition that $|I| + |J| \leq 4$. In this case, the probability that two random elements belong to the same coset of $\mathcal{M}_J$ for a certain J with $|J| = 3$ is 2^{-30} (as we have already seen). Instead, the probability that two elements, that belong to the same coset of $\mathcal{D}_I$, belong to the same coset of $\mathcal{M}_J$ after four rounds is 0.

Exactly as before, the idea is to look for the minimum number of texts m in order to guarantee at least one collision in the random case with high probability. Since there are four 12-dimensional space $\mathcal{M}_J$ and using the birthday paradox, the probability to have at least one collision in the random case given m texts is well approximated by $p = 1 - e^{-m(m-1)/(2\cdot 2^{30})}$. Thus, $m \simeq 2^{16.25}$ texts are sufficient to set up a 4-round distinguisher (in this case, the probability to have a collision in the random case is approximately 95% - note that $2^{15.75}$ texts are sufficient to have at least one collision with probability of 75%). Indeed, given $2^{16.25}$ texts (that is about $2^{31.5}$ pairs), the number of collision in the random case is on average $2^{31.5} \cdot 2^{-30} = 2^{1.5} \approx 3$, while the number of collision in the other case is $2^{31.5} \cdot 0 = 0$. That is, $2^{16.25}$ chosen plaintexts are sufficient for this distinguisher.

Note that this distinguisher exploits the Impossible Differential property presented in [BK01]. Thus, it is not a surprise that the computational complexity of these two distinguishers is the same. Only for completeness, note that it is possible to set up a 0-probability distinguishers also for the 3-round case:

$$Pr(R^{(3)}(x) \oplus R^{(3)}(y) \in \mathcal{M}_I \,|\, x \oplus y \in \mathcal{C}_J) = Pr(R^{(3)}(x) \oplus R^{(3)}(y) \in \mathcal{C}_I \,|\, x \oplus y \in \mathcal{D}_J) = 0$$

where $0 < |I| + |J| \leq 4$. Since in the random case, the probability that two elements belong to the same coset of $\mathcal{C}_I$ or $\mathcal{M}_I$ is upper bounded by 2^{-30} for each I and J, one needs at least $2^{15.75}$ chosen plaintexts to set up this distinguisher. That is, in the case of 3-round AES, the 0-probability distinguisher is worse than the one described in the previous section[8].

Moreover, note that this 4-round distinguisher (as also the 3-round one) works also in the decryption direction. In this case, using the same argument as before, if we two texts belong to the same coset of $\mathcal{M}_I$, then they belong to two different cosets of $\mathcal{D}_J$ four rounds before for $|I| + |J| \leq 4$.

[8] Only for completeness, a similar result can also be obtained for the 2-round case, exploiting the probability $Pr(R^{(2)}(x) \oplus R^{(2)}(y) \in \mathcal{C}_I \,|\, x \oplus y \in \mathcal{C}_J) = 0$ where $0 < |I| + |J| \leq 4$.

```
Data: $2^{16.25}$ texts $c^i$ (for $i = 1, ..., 2^{16.25}$).
Result: 1 if there is at least one collision, 0 otherwise.
for each pair $(c^i, c^j)$ with $i \neq j$ do
    $c \leftarrow MC^{-1}(c^i \oplus c^j)$;
    for $k$ from $0$ to $3$ do
        if $c_{k,0} = 0$ AND $c_{(3+k)\%4,1} = 0$ AND $c_{(2+k)\%4,2} = 0$ AND $c_{(1+k)\%4,3} = 0$
         then
            return 1;
        end
    end
end
return 0.
```

Algorithm 3: Pseudocode for Distinguisher for 4-round AES.

Finally, starting from this 4-round impossible subspace trail, it is possible to re-define the impossible differential attack in a very natural way. We highlight this relationship in App. C of [GRR16], giving all the details.

5 Key-recovery Attacks on AES with a Secret S-Box

From now on, we focus on AES with a single secret S-Box, and we show how to exploit subspace trails in order to set up key-recovery attacks. More precisely, assume to consider *AES with secret and identical* (bijective) *S-Box*. Here we present a generic strategy related to the presented subspace trail that can be used to recover directly the secret key (that is, without finding any information or equivalent representation of the secret S-Box). In particular, in the following we show how truncated differential, impossible differential, and square attacks can exploit this strategy to attack 3- up to 5-round of AES.

The main idea of our attack on AES with a secret S-Box is the following. As we have seen, a coset of $\mathcal{D}_i$ is mapped into a coset of $\mathcal{C}_i$ after one round. Using some particular (but very common) properties of the MixColumns matrix, it is possible to choose a subset of a coset of $\mathcal{D}_i$ which depends on the secret key, such that it is mapped after one round into a subset of a coset of $\mathcal{D}_J \cap \mathcal{C}_i \subseteq \mathcal{D}_J$ with probability 1. That is, consider a subset of a coset of $\mathcal{D}_i$ which depends on the guessed values of some bytes of the secret key. If these guessed values are wrong, then after one round this subset of $\mathcal{D}_i$ is mapped into a subset of a coset of $\mathcal{C}_i$. Instead, if these guessed values are correct, then after one round this subset of $\mathcal{D}_i$ is mapped into a subset of a coset of $\mathcal{D}_J$ with probability 1. Note that also when the guessed values are wrong it is possible that the initial subset is mapped into a subset of a coset of $\mathcal{D}_J$ after one round, but this happens with probability strictly less than 1. Using this property together with other considerations, the attacker can identify the right key.

This attack exploits some particular (but very common) properties of the MixColumns matrix M_{MC}. However, before to list these properties of M_{MC} used for the attack, we define the concepts of (two) *consecutive-row bytes* and of (two) *consecutive-diagonal bytes.*

Definition 7. Let $t \in \mathbb{F}_{2^8}^{4\times 4}$ a text. Given two different bytes $t_{i,j}$ and $t_{l,k}$ (where the indexes are taken modulo 4):

- if they lie in the same row, they are "*consecutive-row bytes*" if $i = l$, and if $j + 1 = k$ for $j < k \leq 3$ or $k + 1 = j$ otherwise;
- if they lie in the same diagonal, they are "*consecutive-diagonal bytes*" if $i + 1 = l$ for $i < l \leq 3$ or $l + 1 = i$ otherwise, and if $j + 1 = k$ for $j < k \leq 3$ or $k + 1 = j$ otherwise.

Examples of two consecutive-row bytes are $(t_{0,0}, t_{0,1})$ or $(t_{0,0}, t_{0,3})$, while examples of two consecutive-diagonal bytes are $(t_{0,0}, t_{1,1})$ or $(t_{0,0}, t_{3,3})$. Using this definition, the two properties of the MixColumns matrix M_{MC} that we are going to use are:

- each row of M_{MC} has two *identical* consecutive-row bytes;
- each row of M_{MC} has these two identical consecutive-row bytes in different positions, that is two different rows can not have the two identical consecutive-row bytes in the same columns.

Note that a cyclic matrix[9] with two identical elements for each row satisfies these conditions. Moreover, these conditions can be a little generalized, since for example it is not necessary that the two identical byte are consecutive.

Using this properties of M_{MC}, our attack is based on the following proposition.

Proposition 2. *Let p^1 and p^2 two texts such that $p^1_{i,j} = p^2_{i,j}$ for each $(i, j) \neq \{(0, 0), (1, 1)\}$ and $p^1_{0,0} \oplus p^1_{1,1} = p^2_{0,0} \oplus p^2_{1,1}$. If $p^1_{0,0} \oplus p^1_{1,1} = p^2_{0,0} \oplus p^2_{1,1} = k_{0,0} \oplus k_{1,1}$ (where k is the secret key of the first round), then after one round they belong to the same coset of $\mathcal{C}_0 \cap \mathcal{D}_{0,1,3} \subseteq \mathcal{D}_{0,1,3}$, that is $R(p^1) \oplus R(p^2) \in \mathcal{C}_0 \cap \mathcal{D}_{0,1,3} \subseteq \mathcal{D}_{0,1,3}$.*

Proof. First of all, note that these two texts p^1 and p^2 belong in the same coset of $\mathcal{D}_0 \cap \mathcal{C}_{0,1} \subseteq \mathcal{D}_0$ (by definition of $\mathcal{D}_0$). As we have already seen, if two elements belong to the same coset of $\mathcal{D}_0$, then after one round they belong to the same coset of $\mathcal{C}_0$. Thus, it is sufficient to prove that $R(p^1) \oplus R(p^2) \in \mathcal{D}_{0,1,3}$.

Since $R(p^1) \oplus R(p^2) \in \mathcal{C}_0$, in order to prove that $R(p^1) \oplus R(p^2) \in \mathcal{D}_{0,1,3}$ it is sufficient to prove that $R(p^1)_{2,0} \oplus R(p^2)_{2,0} = 0$. By simple computation:

$$\begin{aligned} R(p^1)_{2,0} = {} & \text{S-Box}(p^1_{0,0} \oplus k^0_{0,0}) \oplus \text{ S-Box}(p^1_{1,1} \oplus k^0_{1,1}) \oplus \\ & \oplus \alpha \cdot \text{ S-Box}(p^1_{2,2} \oplus k_{2,2}) \oplus (\alpha + 1) \cdot \text{ S-Box}(p^1_{3,3} \oplus k_{3,3}). \end{aligned}$$

First of all observe that S-Box$(p^1_{0,0} \oplus k^0_{0,0}) \oplus$ S-Box$(p^1_{1,1} \oplus k^0_{1,1}) = 0$. Indeed, since $p^1_{0,0} \oplus p^1_{1,1} = k_{0,0} \oplus k_{1,1}$ by definition, then $p^1_{0,0} \oplus k^0_{0,0} = p^1_{1,1} \oplus k^0_{1,1}$, that is S-Box$(p^1_{0,0} \oplus k^0_{0,0}) =$ S-Box$(p^1_{1,1} \oplus k^0_{1,1})$, or equivalently S-Box$(p^1_{0,0} \oplus k^0_{0,0}) \oplus$ S-Box$(p^1_{1,1} \oplus k^0_{1,1}) = 0$. Thus:

$$R(p^1)_{2,0} = \alpha \cdot \text{ S-Box}(p^1_{2,2} \oplus k_{2,2}) \oplus (\alpha + 1) \cdot \text{ S-Box}(p^1_{3,3} \oplus k_{3,3})$$

and in a similar way:

$$R(p^2)_{2,0} = \alpha \cdot \text{ S-Box}(p^2_{2,2} \oplus k_{2,2}) \oplus (\alpha + 1) \cdot \text{ S-Box}(p^2_{3,3} \oplus k_{3,3}).$$

Since $p^1_{2,2} = p^2_{2,2}$ and $p^1_{3,3} = p^2_{3,3}$ by definition, it follows that $R(p^1)_{2,0} = R(p^2)_{2,0}$, and so the thesis. □

Note that no information on the S-Box is used, and, as shown in the following, this fact allows to discover directly the secret key. This proposition can be easily generalized for each possible combination of consecutive-diagonal bytes.

[9] A circulant or cyclic matrix is a matrix where each row vector is rotated one element to the right relative to the preceding row vector, that is:

$$circ(c_0, c_1, ..., c_{n-1}) = \begin{bmatrix} c_0 & c_1 & \cdots & c_{n-1} \\ c_{n-1} & c_0 & \cdots & c_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ c_1 & c_2 & \cdots & c_0 \end{bmatrix}.$$

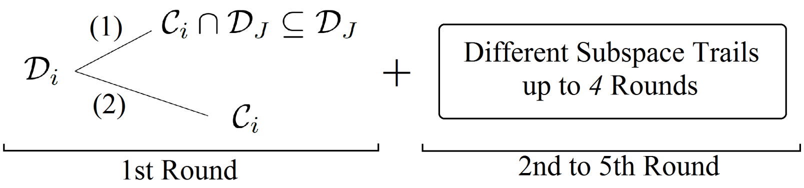
(1)
$\mathcal{C}_i \cap \mathcal{D}_J \subseteq \mathcal{D}_J$
$\mathcal{D}_i$
(2)
$\mathcal{C}_i$
+
Different Subspace Trails up to 4 Rounds
1st Round
2nd to 5th Round

Figure 8: *Strategy of the attacks on AES with a secret S-Box.* Starting with a subset of a coset of $\mathcal{D}_i$ which depends on the guessed values of the secret key, it is mapped after one round into a subset of a coset of $\mathcal{D}_J$ if the guessed values is correct - case (1), or into a subset of a coset of $\mathcal{C}_i$ if the guessed values is wrong - case (2). As a consequence, the subspace trails up to the 5-*th* round are different for the two cases, and this allows to set up various key-recovery attacks.

Proposition 3. *Let p^1 and p^2 two texts such that*

$$p^1_{i,j} = p^2_{i,j} \qquad \forall (i,j) \neq \{(n,m),(k,l)\}$$

and

$$p^1_{k,l} \oplus p^1_{n,m} = p^2_{k,l} \oplus p^2_{n,m},$$

where $p_{k,l}$ and $p_{n,m}$ are two consecutive-diagonal bytes. If $p^1_{k,l} \oplus p^1_{n,m} = p^2_{k,l} \oplus p^2_{n,m} = k_{k,l} \oplus k_{n,m}$ (where k is the secret key of the first round), then after one round they belong to the same coset of $\mathcal{C}_{l-k} \cap \mathcal{D}_{\{0,1,2,3\}\setminus r} \subseteq \mathcal{D}_{\{0,1,2,3\}\setminus r}$ (the indexes are taken modulo 4), where r is defined as the row of the MixColumn matrix M_{MC} such that $MC_{r,n} = MC_{r,k}$. Equivalently, $R(p^1) \oplus R(p^2) \in \mathcal{C}_{k-l} \cap \mathcal{D}_{\{0,1,2,3\}\setminus r}$.

Note that $l - k \equiv_4 m - n$ since they are two consecutive-diagonal bytes. We refer to Fig. 9 for an example of application of this Proposition.

The idea is to exploit this property in order to set up attacks on AES. Indeed, consider a subset of a coset of $\mathcal{D}_i$ related to the guess secret key as plaintexts. If the guess value is correct - case (1) of Fig. 8 (that is, if the difference of two consecutive-diagonal bytes of the plaintexts is equal to the difference of the same bytes of the secret key), then this set is mapped into a subset of a coset of $\mathcal{C}_i \cap \mathcal{D}_J \subseteq \mathcal{D}_J$ for a certain J with $|J| = 3$. If the guess value is wrong - case (2) of Fig. 8, then this set is mapped into a subset of a coset of $\mathcal{C}_i$. Using the subspace trails of Sect. 4, this implies for example that:

- after 3 rounds, the previous subset of $\mathcal{D}_i$ is mapped into a subset of a coset of $\mathcal{M}_J$ with probability 1 in case (1), while this happens only with probability 2^{-8} - i.e. strictly less than 1 - in case (2);
- after 4 rounds, the probability that two texts in the previous subset of $\mathcal{D}_i$ are mapped into the same coset of $\mathcal{M}_J$ is higher in case (1) - approximately 2^{-22} - than in case (2) - approximately 2^{-30};
- after 5 rounds, the probability that two texts in the previous subset of $\mathcal{D}_i$ are mapped into the same coset of $\mathcal{M}_j$ is equal to zero in case (1), while is strictly different from zero in case (2) - approximately 2^{-94}.

These different subspace trails allow to recover information about the secret key. In particular, in the following we show how to exploits it to set up a truncated differential attack on 3- and 4- rounds, an impossible differential attack on 5-round and a square attack on 3-round of AES with a secret S-Box.

Finally, observe that a similar strategy can be used to set up attacks on AES-like block ciphers, with identical (secret) S-Box and with a MixColumns matrix that satisfies the

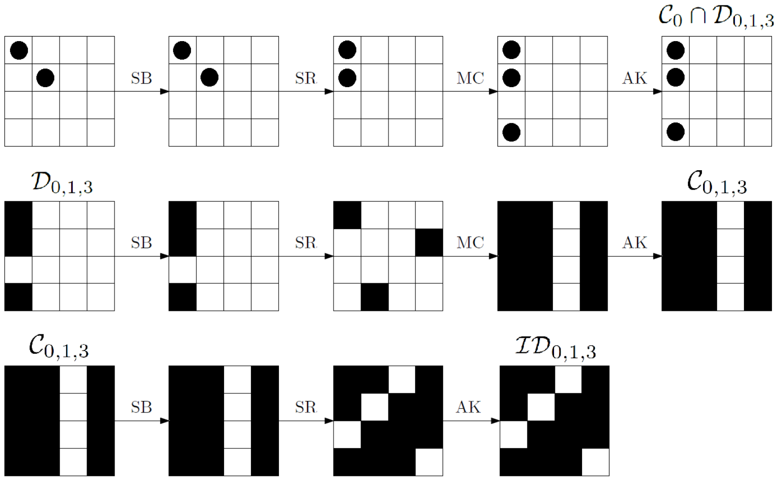

Figure 9: *3-rounds Truncated Differential Attack on AES with a single secret S-Box.* The choice of the plaintexts (i.e. $p_{0,0} \oplus p_{1,1} = k_{0,0} \oplus k_{1,1}$) guarantees that after one round there are only three bytes with non-zero difference instead of four, that is the plaintexts belong to the same coset of $\mathcal{C}_0 \cap \mathcal{D}_{0,1,3}$. White box denotes denotes a byte with a zero-difference, while a black box denotes a byte with non-zero difference.

previous requirement. Moreover, we stress that, with respect to other attacks present in literature in the same setting (i.e. AES with a secret S-Box), for the first time we show that it is possible to discover the secret key directly, that is without discovering any information (e.g. an equivalent class) about the secret S-Box.

6 Truncated Differential Attacks on 3 and 4 Rounds of AES with a Secret S-Box

In this section, we briefly show how to exploit the previous strategy to set up truncated differential attacks on 3- and 4-round of AES with a secret S-Box. We limit here to give the idea of these attacks, and we refer to App. F of [GRR16] and App. G of [GRR16] for all the details together with the presentation of the square attack on AES with a secret S-Box (see App. F.3 of [GRR16]).

Truncated Diff. Attack on 3 rounds of AES with Secret S-Box The *attack on 3-rounds* - illustrated in Fig. 9 - works as follows. Consider a pair of plaintexts p^1 and p^2 with the condition $p^1_{i,j} = p^2_{i,j}$ for each $(i, j) \neq \{(0, 0), (1, 1)\}$ and $p^1_{0,0} \oplus p^1_{1,1} = p^2_{0,0} \oplus p^2_{1,1}$. As we have seen, if $p^1_{0,0} \oplus p^1_{1,1} = p^2_{0,0} \oplus p^2_{1,1} = k_{0,0} \oplus k_{1,1}$, then p^1 and p^2 belong to the same coset of $\mathcal{D}_{0,1,3}$ after one round with probability 1. Consequently, after three rounds they belong to the same coset of $\mathcal{M}_{0,1,3}$ with probability 1 (or of $\mathcal{ID}_{0,1,3}$ if the final MixColumns is omitted), since a coset of $\mathcal{D}_{0,1,3}$ is mapped into a coset of $\mathcal{M}_{0,1,3}$ with probability 1. Instead, if $p^1_{0,0} \oplus p^1_{1,1} = p^2_{0,0} \oplus p^2_{1,1} \neq k_{0,0} \oplus k_{1,1}$, then p^1 and p^2 belong to the same coset of $\mathcal{D}_{0,1,3}$ after one round only with probability 2^{-8} (that is, only if $R(p^1)_{2,0} \oplus R(p^2)_{2,0} = 0$). Thus, after three rounds they belong to the same coset of $\mathcal{M}_{0,1,3}$ only with probability 2^{-8}. Our attack exploits these different probabilities in order to find $k_{0,0} \oplus k_{1,1}$. More details are given in App. F.1 of [GRR16].

Truncated Diff. Attack on 4 rounds of AES with Secret S-Box. The truncated differential *attack on 4 rounds* of AES works in a similar way, and it exploits the subspace trail described in Sect. 4.3. In particular, if two texts belong to the same coset of $\mathcal{D}_J$ for $|J| = 3$ fixed, then after three rounds they belong to the same coset of $\mathcal{M}_I$ for $|I| = 3$ with probability $4 \cdot 2^{-24} = 2^{-22}$ in the AES case and with probability $4 \cdot 2^{-32} = 2^{-30}$ in the random case. Exploiting these different probabilities and the fact that a coset of a subset of $\mathcal{D}_i$ (which depends on the guessed values of the key) is mapped into a subset of a coset of $\mathcal{D}_{0,1,3}$ only for the correct guessed values of the key, it is possible to discover the whitening key of 4-rounds of AES with a secret S-Box up to 2^{32} variants. More details are given in App. G of [GRR16].

7 Impossible Differential Attack on 5-round of AES with a single Secret S-Box

Using the strategy presented in the previous section, it is possible to set up an impossible differential attack on 5 rounds of AES with a secret S-Box. As before, the goal is to find the secret key without needing to discover any information about the S-Box.

Starting from this attack, we show how to turn it into a secret key distinguisher for AES, and we compare it in details with the distinguisher presented in [SLG+16] at CRYPTO 2016. As we have already said, also the key recovery attack can be used as distinguisher. However, we show that in order to distinguish a random permutation from an AES one, it is not necessary to find the entire key.

7.1 Key-Recovery Attack using Impossible Differential - General Idea

For the following, we define the set of plaintexts-ciphertexts V_δ with $|V_\delta| = 2^8$:

$$\begin{aligned} V_\delta =&\{(p^i, c^i) \text{ for } i = 0, ..., 2^8 - 1 \,|\, p^i_{0,0} \oplus p^i_{1,1} = \delta \quad \forall i \quad \text{and} \\ &\text{and} \quad p^i_{k,l} = p^j_{k,l} \quad \forall (k,l) \neq \{(0,0),(1,1)\} \text{ and } i \neq j\}, \end{aligned} \tag{6}$$

i.e. plaintexts with 14 constants bytes and with the difference on the other two bytes fixed.

Consider two different pairs (p^1, c^1) and (p^2, c^2) that belong to the same V_δ. By Prop. 3, we know that if $\delta = k_{0,0} \oplus k_{1,1}$, then p^1 and p^2 belong to the same coset of $\mathcal{D}_{0,1,3} \cap \mathcal{C}_0 \subseteq \mathcal{D}_{0,1,3}$ after one round (that is, $R(p^1) \oplus R(p^2) \in \mathcal{D}_{0,1,3} \cap \mathcal{C}_0 \subseteq \mathcal{D}_{0,1,3}$) with probability 1. If $\delta \neq k_{0,0} \oplus k_{1,1}$, they belong to the same coset of $\mathcal{C}_0$ after one round with probability 1, and to the same coset of $\mathcal{D}_{0,1,3} \cap \mathcal{C}_0 \subseteq \mathcal{D}_{0,1,3}$ with probability 2^{-8} (or to the same coset of $\mathcal{D}_J$ for $|J| = 3$ after one round with probability $4 \cdot 2^{-8} = 2^{-6}$).

Consider first the case $\delta = k_{0,0} \oplus k_{1,1}$. Since $R(p^1) \oplus R(p^2) \in \mathcal{D}_{0,1,3}$ for each pair of plaintexts p^1 and p^2 in V_δ, then $R^{(4)} \circ R(p^1) \oplus R^{(4)} \circ R(p^2) = R^{(5)}(p^1) \oplus R^{(5)}(p^2) \notin \mathcal{M}_J$ for $|I| + |J| \leq 4$ with probability 1 due to the 4-round impossible differential distinguisher of Sect. 4.4. That is, for each $(p^1, c^1) \neq (p^2, c^2)$

$$Pr(R^{(5)}(p^1) \oplus R^{(5)}(p^2) \in \mathcal{M}_J \,|\, (p^1, c^1), (p^2, c^2) \in V_\delta) = 0,$$

for each J with $|J| = 1$ and where $\delta := k_{0,0} \oplus k_{1,1}$ is known. As usual, a similar result holds also in the case in which the final MixColumns operation is omitted (in this case, $\mathcal{M}_J$ is replaced by $\mathcal{ID}_J$).

Instead, if $\delta \neq k_{0,0} \oplus k_{1,1}$, note that it's possible that two elements of V_δ belong to the same coset of $\mathcal{M}_J$ for $|J| = 1$ after 5-round. In particular, the probability that two elements p and q in V_δ belong to the same coset of $\mathcal{M}_J$ after 5-round for a certain J with $|J| = 1$ is approximately[10] $4 \cdot 2^{-96} = 2^{-94}$.

[10] The exact probability for a *wrong* $\delta \neq k_{0,0} \oplus k_{1,1}$ is given by $Pr(R^{(5)}(p^1) \oplus R^{(5)}(p^2) \in \mathcal{M}_J \,|\, p^1 \oplus p^2 \in V_\delta) = 2^{-6} \cdot 0 + (1 - 2^{-6}) \cdot 4 \cdot 2^{-96} = 2^{-94} - 2^{-100} \simeq 2^{-94}$, which is derived considering the two cases $R(p^1) \oplus R(p^2) \in \mathcal{D}_J$ and $R(p^1) \oplus R(p^2) \notin \mathcal{D}_J$ for $|J| = 3$.

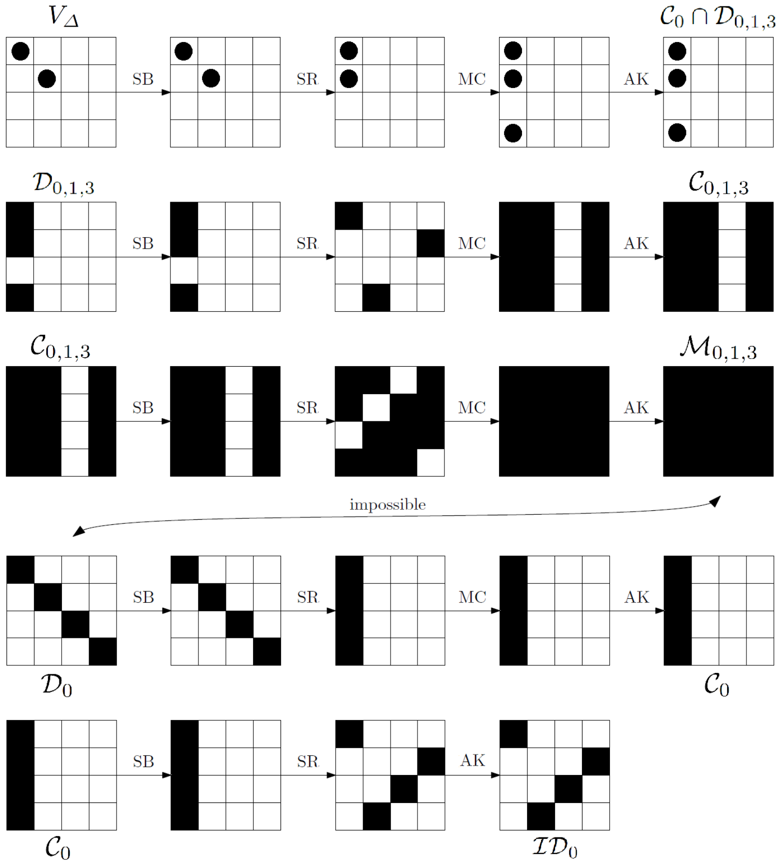

Figure 10: *5-Round Secret Key Distinguisher for AES with a single secret S-Box* with data complexity $2^{98.2}$ based on the Impossible Subspace Trail on 4-Round (from Sect. 4.4). The choice of the plaintexts (i.e. $p_{0,0} \oplus p_{1,1} = k_{0,0} \oplus k_{1,1}$) guarantees that after one round there are only three bytes with non-zero difference instead of four, that is the plaintexts belong to the same coset of $\mathcal{C}_0 \cap \mathcal{D}_{0,1,3}$. The probability the two ciphertexts belong to the same coset of $\mathcal{M}_k$ for $|k| = 1$ is zero. White box denotes denotes a byte with a zero-difference, while a black box denotes a byte with non-zero difference.

The idea is to exploit these different probabilities in order to find the key. In particular, a key candidate δ can be declared wrong if there is at least one collision, i.e. two different pairs of texts (p^1, c^1) and (p^2, c^2) such that $p^1 \oplus p^2 \in V_\delta$ and $c^1 \oplus c^2 \in \mathcal{M}_J$ for $|J| = 1$. Thus, in the following we look for the minimum number of texts necessary to have at least one collision *for each* $\delta \neq k_{0,0} \oplus k_{1,1}$ with high probability.

Before to proceed, note that a similar impossible differential attack can be set up for 4-round AES with secret S-Box, exploiting the fact that two elements in the same coset of $\mathcal{D}_J$ can not belong to the same coset of $\mathcal{C}_I$ after three rounds for $|I| + |J| \leq 4$.

7.2 Data Complexity and Computational Cost

The attack is constructed in two steps. First we focus on a single difference among two bytes of the secret key, and then we show how to find the entire key. In this section, we limit to report the results for the data and the computational complexity of the attack, and we refer to App. H.1 of [GRR16] for a complete discussion.

```
Data: $2^{98.5}$ collections ($2^{90.2}$ one for each possible value of $\delta$, or equivalently $2^{82.2}$
      different sets $V_\delta$ as defined in (6).
Result: $k_{0,0} \oplus k_{1,1}$.
for $\Delta$ from 0 to $2^8 - 1$ do
    $flag \leftarrow 0$;
    divide the $2^{90.2}$ ciphertexts in the corresponding $2^{82.2}$ different sets $V_\delta$;
    for each one of the $2^{82.2}$ different sets $V_\Delta$ do
        let $(p^i, c^i)$ for $i = 0, ..., 2^8 - 1$ the $2^8$ (plaintexts, ciphertexts) of a single set
         $V_\delta$;
        re-order this set of elements as described in App. ??;
        for $i$ from 0 to $2^8 - 2$ do
            if $c^i \oplus c^{i+1} \in \mathcal{M}_k$ for $|k| = 1$ then        // e.g.  see Algorithm 1
                $flag \leftarrow 1$;
                next collection (i.e. next $\delta$);
            end
        end
        if $flag = 0$ then
            identify $\delta$ as candidates of $k_{0,0} \oplus k_{1,1}$;
        end
    end
end
return Candidates of $k_{0,0} \oplus k_{1,1}$.
```

Algorithm 4: *Attack for 5-round of AES using Impossible Differential - Pseudo Code.* The same attack can be use to find the remaining part of the key.

Data Complexity. First of all, we consider the case in which the goal is to find only *one byte of the secret key* (i.e. the difference of two fixed bytes of the key). As we have seen, given two texts in the same set V_δ defined as in Prop. 6, after 5 rounds they can not belong to the same coset of $\mathcal{M}_J$ for $|J| = 1$ if $\delta = k_{0,0} \oplus k_{1,1}$. Instead, if $\delta \neq k_{0,0} \oplus k_{1,1}$ then they belong to the same coset of $\mathcal{M}_J$ for $|J| = 1$ with probability 2^{-94}. Thus, in order to find the difference of these two bytes of the key, one needs at least one collision in the same coset of $\mathcal{M}_J$ for $|J| = 1$ *for each* one of the $2^8 - 1 \simeq 2^8$ $\delta \neq k_{0,0} \oplus k_{1,1}$. As shown in details in App. H.1 of [GRR16], to find this byte with probability higher than 95%, for each possible δ one needs approximately $2^{82.2}$ different sets V_δ, for a total cost of $2^{98.2}$ plaintextexts/ciphertexts.

In order to find *the entire key* (up to 2^{32} variants), the idea is simply to repeat the previous attack 12 times (i.e. three times for each possible diagonal). As shown in details in App. H.1 of [GRR16], to find this byte with probability higher than 95%, for each possible δ one needs approximately $2^{82.4}$ different sets V_δ (not $2^{82.2}$ as before - see App. H.1 of [GRR16] for details), for a total cost of approximately, 2^{102} plaintextexts/ciphertexts, which is lower than the entire input-output space.

Computational Complexity. Using a re-ordering algorithm, the computational cost of the attack to find one byte of the key is well approximated by $2^{103.2}$ table-look ups, or $2^{96.56}$ five rounds AES encryption, while the cost to find the entire key is approximately 2^{107} table-look ups, or $2^{100.35}$ five rounds AES encryption.

8 The 5-Round Secret Key Distinguisher for AES

Next we show how to turn the previous key recovery attack into a distinguisher for AES, *in the same setting* of the distinguisher presented in [SLG+16]. The idea is simply to consider only the first part of the attack, i.e. it is sufficient to recover one byte of the key as $k_{0,0} \oplus k_{1,1}$.

Consider the previous key recovery attack, and let the set V_δ defined as before. For each one of the 2^8 possible values of δ, the idea is to consider $2^{82.2}$ different sets V_δ, for a total of $2^{98.2}$ chosen plaintexts. As we have just seen, for the AES permutation, there exists one δ (which is equal to $k_{0,0} \oplus k_{1,1}$) for which there are no collisions. That is, for the AES permutation and for $\delta = k_{0,0} \oplus k_{1,1}$, no pairs (p^1, c^1) and (p^2, c^2) can satisfy $p^1 \oplus p^2 \in V_\delta$ and $c^1 \oplus c^2 \in \mathcal{M}_J$ for $|J| = 1$. Instead, for the random permutation and with probability 95%, for each δ there is at least one pair with the previous property. Thus, it is possible to distinguish the random permutation from an AES one.

To summarize, suppose to have 2^8 collections (one for each δ), each one with $2^{82.2}$ different sets V_δ, where each of this set contains 2^8 texts, for a total of $2^{98.2}$ texts. In the random case and with probability 95%, we expect that in each one of these 2^8 collections there is at least one collision. Note that the average number of collisions for each collection (i.e. for each δ) is about $2^{-94} \cdot 2^{97.2} = 2^{3.2} \simeq 9$. For the AES permutation, we expect that there exists one δ for which there is no collision with probability 1 in the corresponding collection of sets. For all the other collections, we expect to have at least one collision with probability 95%. We highlight that given the $2^{98.2}$ texts defined as before, it is always possible to divide them in 2^8 collections (one for each δ), and that each collection can be divided in a very simple way in $2^{82.2}$ different sets V_δ (simply using the definition of V_δ). Finally, using the argument of our impossible differential key recovery attack, the computational cost of this distinguisher (i.e. the cost to check if there exists at least one pair of ciphertexts that belong to the same coset of $\mathcal{M}_J$ for a certain J with $|J| = 1$ for all possible values of δ) is $2^{103.2}$ table look-ups, using the ordering algorithm.

8.1 Comparison with 5-Round Distinguisher proposed by Sun, Liu, Guo, Qu and Rijmen, and Possible Generalizations

In [SLG+16], authors presented a similar secret key distinguisher to the one just presented, using the balance property instead of our impossible differential trail.

In order to construct the secret key distinguisher presented in [SLG+16], authors simply consider all the input-output space, and divide it in the 2^8 subsets $\tilde{V}_\Delta$ defined as $\tilde{V}_\Delta = \{(p, c) \,|\, c_{0,0} \oplus c_{1,3} = \Delta\}$ for each possible $\Delta \in \mathbb{F}_{2^8}$, and without any other assumptions on the other bytes. Note that $|\tilde{V}_\Delta| = 2^{120}$. Then, using the link between zero-correlation linear hulls and the integral/balance property, they are able to prove that for an AES permutation and for $\Delta = k_{0,0} \oplus k_{1,3}$ the sum of the plaintexts of the corresponding set $\tilde{V}_\Delta$ is equal to zero, that is the balance property holds[11]. Instead, for a random permutation, the probability that there exists one Δ with the previous property is only 2^{-120}. This distinguisher works only in the decryption direction (i.e. using chosen ciphertexts) and only if the final MixColumns operation is not omitted. Moreover, there is no evidence that this distinguisher can work with less than the entire input-output space[12]. We refer to [SLG+16] for more details. To summarize, this distinguisher requires the full codebook (i.e. 2^{128} texts), and the verification cost is well approximated by 2^{128} XOR operations.

For comparison, our distinguisher requires only $2^{98.2}$ different (plaintexts, ciphertexts), works in the encryption direction (i.e. using chosen plaintexts) independently by the presence of the final MixColumns operation. Thus, as we said in the introduction, this

[11] In [SLG+16], authors presented also a similar distinguisher always based on balance property. In this case, the idea is to divide the entire input-output space in 2^{32} subsets $\tilde{W}_\Delta$ defined as $\tilde{W}_\Delta = \{(p, c) \,|\, c_{0,0} \oplus c_{1,3} = \delta_0, c_{0,1} \oplus c_{3,2} = \delta_1, c_{1,2} \oplus c_{2,1} = \delta_2, c_{2,0} \oplus c_{3,3} = \delta_3\}$, where $\Delta = (\delta_0, \ldots, \delta_3)$. Also in this case, for an AES permutation there exists one Δ for which the balance property holds among the plaintexts, while for a random permutation this happens only with probability 2^{-96}

[12] It may be possibile to use the recently proposed statistical integral distinguisher [WCC+16] to modify the distinguisher presented in [SLG+16] into a statistical integral one, with the goal to reduce the data complexity at the cost of success probability.

provides a counter-example to the claims made in [SLG+16]. That is, as we have already discussed in details in the introduction, this distinguisher provides a counter-example to the conjecture made by these authors that the security margin for round-reduced AES under the chosen plaintext attack is different from that under the chosen-ciphertexts attack.

Only for completeness, we prove that if our distinguisher uses all the input-output space, the probability of success is $1-2^{-2^{25.5}}$. If all the texts are used, then for each δ there are 2^{112} different sets V_δ. Thus, it is possible to construct approximately $2^{15} \cdot 2^{112} = 2^{127}$ different pairs. The probability that for a (wrong) δ no one of these pairs satisfy the required property is approximately $1-(1-2^{-94})^{2^{127}} \simeq 1-e^{-2^{33}} \simeq 1-2^{-2^{33.5}}$. Thus, the probability of success is approximately $(1-2^{-2^{33.5}})^{2^8} \simeq 1-2^{-2^{25.5}}$ if all the input-output space is used, which is much higher than for the integral distinguisher (which is approximately $1-2^{-120}$). We stress that our distinguisher works even using a less data complexity that the entire input-output space, and that $2^{110.5}$ different (plaintexts, ciphertexts) (or equivalently $2^{94.5}$ different sets V_δ for each δ) are sufficient to have approximately the same probability of success of [SLG+16].

Finally, in [SLG+16] authors exploit the link between zero-correlation linear hulls and the integral property to set up our distinguisher, while our distinguisher presented in this paper exploits the impossible differential trails. For completeness, we recall that Impossible Differential, Integral and Zero-Linear Correlation are not independent, as shown in details in [SLR+15]. In particular, the presence of a zero correlation linear hull distinguisher (very likely) implies the existence of an Impossible Differential distinguisher and of an Integral one.

Turn the CRYPTO Distinguisher into a Key-Recovery Attack. As we have turned our key recovery attack into a distinguisher, it is also possible to turn the distinguisher of [SLG+16] into an attack, as also the authors observed in their paper. The idea is to repeat the distinguisher three times (using the version presented in Corollary 5 of [SLG+16] and reported in the footnote, it is possible to recover four bytes of the key), in order to recover the secret key up to 2^{32} variants. Note that also in this case as for our attacks, it is not possible to eliminate more variants of the key without using any information about the secret S-Box. This attack requires the entire input-output space, and it has a cost of $3 \cdot 2^{128} = 2^{129.6}$ XOR operations.

Final Observations. Finally, it is very easy to generalize our distinguisher and the one proposed in [SLG+16] to any AES-like block cipher with the following properties: (1) the encryption scheme adopts *identical S-Boxes* and (2) *at least one row of the MixColumns matrix* M_{MC} (or its inverse) *contains* (at least) *two identical elements*. If one of these two assumptions is missing, the above distinguishers don't work. As a consequence, note that the distinguisher described in this section can not work in the decryption mode (that is, with chosen ciphertexts instead of chosen plaintexts), since no one of the columns of the inverse MixColumns M_{MC}^{-1} has two equal elements. Actually, the first requirement can be relaxed. Indeed, it is sufficient that only the two S-Boxes that are in the positions in which the MixColumns matrix has identical elements are equal.

Note that these assumptions are similar but not equal to the ones required for a key recovery attacks. Indeed, for our key recovery attacks on an AES-like block cipher with secret S-Box, all the S-Boxes must be identical and each row of the MixColumns matrix M_{MC} must contain (at least) two identical elements in different positions.

We emphasize that these assumptions are quite common for the construction of AES-like ciphers (or more in general, for SPNs ciphers). Indeed, symmetric encryption schemes are usually a trade-off between the security and computational efficiency. Thus, to enhance the performance of an encryption scheme (especially for lightweight cryptography), designers usually use identical S-Box and a diffusion layer which maximize the number of 1's (or elements with relatively low hamming weights).

8.2 Critical Discussion of the Distinguisher Model and Open Problems

In this section, we have shown how to interpret the 5-round secret S-Box attack from the section before as a distinguisher for 5-round AES which corresponds to the model used in [SLG+16], with our main point being to give a counter-example to the conjecture motivated by the results therein. By doing that we also significantly improved the complexity of such a distinguisher.

Since any key recovery attack can be used as a distinguisher, the natural question that arises is if such a distinguisher is actually meaningful. Both ours and the distinguisher from [SLG+16] have two properties that set them apart from "any" key recovery-attack:

1. for both distinguishers it is sufficient to find *only part of the key* (e.g. one byte) to distinguish an AES permutation with a secret S-Box from a random one. In other words, it is not necessary finding the entire secret key but only part of it;

2. both distinguishers don't need any information/details about the S-Box (i.e. they don't find or/and exploit any information/details of the secret S-Box) in order to find part of the key.

In order to better highlight this sub-class of attacks/distinguishers, we denote them by *"weak" secret-key distinguishers.* In contrast, we refer to (pure) secret-key distinguisher when a property which is independently of the secret key is exploited, and to key-recovery attack when (at least) one of the two previous properties is not satisfied.

We emphasize that all the secret-key distinguishers currently present in literature - and presented in this paper in Sect. 4 - exploit a property which is independent of the key and of the details of the S-Box. In particular, it is not necessary to know the details of the S-Box to check the integral property or verify is two texts belong or not to the same coset of a mixed space $\mathcal{M}_I$ (that is, the property exploited by the truncated and the impossible differential distinguishers). This second property is in common with the "weak" secret-key distinguishers just defined. On the opposite, a key-recovery attack (e.g. an integral attack, a truncated differential one, ...) usually exploits such details to find the key. As an example, we highlight that also the attacks presented in [TKKL15] on AES with a single secret S-Box don't satisfy the second requirement (i.e. it exploits the details of the S-Box to find the secret key). Indeed, even if the S-Box is secret, such attacks necessarily need to find/know the details of the S-Box (up to an equivalent class) before to discover the secret key, or in other words they can not discover the secret key without exploiting the details of the secret S-Box. Thus, such attacks can not be considered as "weak" secret-key distinguishers with respect to categorization just defined, but falls in the generic category of the key-recovery attacks.

Even if there are key-recovery attacks on up to 7 rounds for AES-128 [MDRM10] with known S-Box, and up to 6 rounds for AES-128 with a secret S-Box [TKKL15], it seems for example not possible to find a distinguisher with properties (1) and (2) for even 6 rounds. We leave this as an open problem for future investigation.

9 Conclusion

We have generalized invariant subspace cryptanalysis to subspace trails and have seen that it includes truncated differential-, impossible differential- and integral attacks. For concrete applications we focused on AES-128, and this led to a method that can use *all* the aforementioned techniques to recovery the secret key for up to 5 rounds without needing to know the S-Box apart from assuming it being a permutation. When the S-Box is known we described new truncated-differential attacks with very low data complexity that are competitive with the best known attacks. It is conceivable that such attacks are also found without the subspace trail approach (truncated differential + ad-hoc optimizations of the key-recovery method that go beyond looking at the differences only), but the combination of properties of individual texts and sums of text follows more naturally from the subspace trail approach.

As one of the major results, we have proposed a new strategy to attack SPNs cipher with a single secret S-Box, if some very generic assumptions on the MixColumns matrix are satisfied. In particular, we showed how several techniques like truncated differential, impossible differential and integral attack can exploit it to recover directly (i.e. without discovering anything of the secret S-Box) the secret key for 1- up to 5- rounds of AES

We also used this approach to give a counter-example to the conjecture of Sun et al. [SLG+16] related to 5-round distiguishers. By doing that we also significantly improved the complexity of a distinguisher in their model, arguing however that the quest for a real 5-round distinguisher (that is, a 5-round secret key distinguisher for an AES permutation which is not derived from a key recovery attack but exploits a property which is independent of the secret key) is still open. Future work includes trying to exploit the subspace properties in other ways to get more efficient or longer distinguishers, perhaps by considering also S-Box properties, to use this approach to devise more key-recovery attacks and to apply the approach to other schemes.

Acknowledgments. The work in this paper has been partially supported by the Austrian Science Fund (project P26494-N15). We thank Bing Sun for helpful discussions on [SLG+16]. We also thank Charles Bouillaguet for helping us with the tool described in [BDF11] to find attacks that better match the settings we consider. We finally thank anonymous reviewers for their comments.

References

[BBK14] Alex Biryukov, Charles Bouillaguet, and Dmitry Khovratovich. *Cryptographic Schemes Based on the ASASA Structure: Black-Box, White-Box, and Public-Key*, pages 63–84. Springer Berlin Heidelberg, 2014.

[BBS99] Eli Biham, Alex Biryukov, and Adi Shamir. Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT 1999: International Conference on the Theory and Application of Cryptographic Techniques, Czech Republic. Proceedings*, pages 12–23, 1999.

[BDD+12] Charles Bouillaguet, Patrick Derbez, Orr Dunkelman, Pierre-Alain Fouque, Nathan Keller, and Vincent Rijmen. Low-Data Complexity Attacks on AES. *IEEE Trans. Information Theory*, 58(11):7002–7017, 2012.

[BDF11] Charles Bouillaguet, Patrick Derbez, and Pierre-Alain Fouque. Automatic search of attacks on round-reduced AES and applications. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA. Proceedings*, pages 169–187, 2011.

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

[BK07] Alex Biryukov and Dmitry Khovratovich. Two New Techniques of Side-Channel Cryptanalysis. In *Cryptographic Hardware and Embedded Systems -*

CHES 2007: 9th International Workshop, Austria. Proceedings, pages 195–208, 2007.

[BKLT11] Julia Borghoff, Lars R. Knudsen, Gregor Leander, and Søren S. Thomsen. Cryptanalysis of PRESENT-Like Ciphers with Secret S-Boxes. In *Fast Software Encryption - FSE 2011: 18th International Workshop, Denmark. Revised Selected Papers*, pages 270–289, 2011.

[BKR11] Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. Biclique cryptanalysis of the full AES. In *Advances in Cryptology – ASIACRYPT 2011: 17th International Conference on the Theory and Application of Cryptology and Information Security, South Korea. Proceedings*, pages 344–371, 2011.

[BS01] Alex Biryukov and Adi Shamir. Structural Cryptanalysis of SASAS. In *Advances in Cryptology - EUROCRYPT 2001: International Conference on the Theory and Application of Cryptographic Techniques, Austria. Proceeding*, pages 394–405, 2001.

[BS10] Alex Biryukov and Adi Shamir. Structural Cryptanalysis of SASAS. *Journal of Cryptology*, 23(4):505–518, 2010.

[DF13] Patrick Derbez and Pierre-Alain Fouque. Exhausting Demirci-Selçuk Meet-in-the-Middle Attacks Against Reduced-Round AES. In *Fast Software Encryption - FSE 2013: 20th International Workshop, Singapore. Revised Selected Papers*, pages 541–560, 2013.

[DFJ13] Patrick Derbez, Pierre-Alain Fouque, and Jérémy Jean. Improved key recovery attacks on reduced-round AES in the single-key setting. In *Advances in Cryptology - EUROCRYPT 2013: 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Greece. Proceedings*, pages 371–387. 2013.

[DK10] Orr Dunkelman and Nathan Keller. The Effects of the Omission of Last Round's MixColumns on AES. *Inf. Process. Lett.*, 110(8-9):304–308, 2010.

[DKR97] Joan Daemen, Lars R. Knudsen, and Vincent Rijmen. The block cipher square. In *Fast Software Encryption, 4th International Workshop, FSE '97, Haifa, Israel, January 20-22, 1997, Proceedings*, pages 149–165, 1997.

[DKS10] Orr Dunkelman, Nathan Keller, and Adi Shamir. Improved single-key attacks on 8-round AES-192 and AES-256. In *Advances in Cryptology - ASIACRYPT 2010: 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore. Proceedings*, pages 158–176, 2010.

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

[DR06a] Joan Daemen and Vincent Rijmen. Two-Round AES Differentials. Cryptology ePrint Archive, Report 2006/039, 2006. http://eprint.iacr.org/2006/039.

[DR06b] Joan Daemen and Vincent Rijmen. Understanding Two-Round Differentials in AES. In *Security and Cryptography for Networks 2006*, volume 4116, pages 78 – 94, 2006.

[DS08a] Hüseyin Demirci and Ali Aydın Selçuk. *A Meet-in-the-Middle Attack on 8-Round AES*, pages 116–126. 2008.

[DS08b] Hüseyin Demirci and Ali Aydin Selçuk. A meet-in-the-middle attack on 8-round AES. In *Fast Software Encryption - FSE 2008: 15th International Workshop, Switzerland. Revised Selected Papers*, pages 116–126. 2008.

[Eve87] Jan-Hendrik Evertse. Linear Structures in Blockciphers. In *Advances in Cryptology - EUROCRYPT 1987: Workshop on the Theory and Application of of Cryptographic Techniques, Netherlands. Proceedings*, pages 249–266, 1987.

[git16a] Key-recovery attacks on up to 4-round AES with a single secret S-Box, 2016. https://github.com/Krypto-iaik/AttacksAESSecretSBox.

[git16b] Low data-complexity attacks on up to 4-round AES, 2016. https://github.com/Krypto-iaik/LowDataAttacks_AES.

[git16c] Verification of all distinguishers up to 4-rounds AES, 2016. https://github.com/Krypto-iaik/Distinguishers_AES.

[GJN$^+$15] Jian Guo, Jérémy Jean, Ivica Nikolic, Kexin Qiao, Yu Sasaki, and Siang Meng Sim. Invariant Subspace Attack Against Full Midori64. Cryptology ePrint Archive, Report 2015/1189, 2015.

[GRR16] Lorenzo Grassi, Christian Rechberger, and Sondre Rønjom. Subspace Trail Cryptanalysis and its Applications to AES - Extended Version. Cryptology ePrint Archive, Report 2016/592, 2016. http://eprint.iacr.org/2016/592.

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

[KW02] Lars R. Knudsen and David Wagner. Integral Cryptanalysis. In *Fast Software Encryption - FSE 2002: 9th International Workshop, Belgium. Revised Papers*, pages 112–127, 2002.

[LAAZ11] Gregor Leander, Mohamed Ahmed Abdelraheem, Hoda AlKhzaimi, and Erik Zenner. A Cryptanalysis of PRINTcipher: The Invariant Subspace Attack. In *Advances in Cryptology – CRYPTO 2011: 31st Annual Cryptology Conference, Santa Barbara, CA, USA, 2011. Proceedings*, pages 206–221, 2011.

[LJQ14] Guo-qiang Liu, Chen-Hui Jin, and Chuan-Da Qi. Improved Slender-Set Linear Cryptanalysis. In *Fast Software Encryption - FSE 2014: 21st International Workshop, UK. Revised Selected Papers*, pages 431–450, 2014.

[LMR15] Gregor Leander, Brice Minaud, and Sondre Rønjom. A Generic Approach to Invariant Subspace Attacks: Cryptanalysis of Robin, iSCREAM and Zorro. In *Advances in Cryptology - EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Bulgaria. Proceedings, Part I*, pages 254–283, 2015.

[MDRM10] Hamid Mala, Mohammad Dakhilalian, Vincent Rijmen, and Mahmoud Modarres-Hashemi. Improved impossible differential cryptanalysis of 7-round AES-128. In *Progress in Cryptology - INDOCRYPT 2010: 11th International Conference on Cryptology in India, India. Proceedings*, pages 282–291. 2010.

[PSC$^+$02] Sangwoo Park, Soo Hak Sung, Seongtaek Chee, E-Joong Yoon, and Jongin Lim. On the Security of Rijndael-Like Structures against Differential and Linear Cryptanalysis. In *Advances in Cryptology - ASIACRYPT 2002: 8th International Conference on the Theory and Application of Cryptology and Information Security, New Zealand. Proceedings*, pages 176–191. 2002.

[SLG+16] Bing Sun, Meicheng Liu, Jian Guo, Longjiang Qu, and Vincent Rijmen. New Insights on AES-Like SPN Ciphers. In *Advances in Cryptology – CRYPTO 2016: 36th Annual International Cryptology Conference, Santa Barbara, CA, USA. Proceedings, Part I*, pages 605–624, 2016.

[SLR+15] Bing Sun, Zhiqiang Liu, Vincent Rijmen, Ruilin Li, Lei Cheng, Qingju Wang, Hoda Alkhzaimi, and Chao Li. Links Among Impossible Differential, Integral and Zero Correlation Linear Cryptanalysis. In *Advances in Cryptology – CRYPTO 2015: 35th Annual Cryptology Conference, Santa Barbara, CA, USA. Proceedings, Part I*, pages 95–115, 2015.

[Tie16] Tyge Tiessen. Polytopic Cryptanalysis. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Austria. Proceedings, Part I*, pages 214–239, 2016.

[TKKL15] Tyge Tiessen, Lars R. Knudsen, Stefan Kölbl, and Martin M. Lauridsen. Security of the AES with a Secret S-Box. In *Fast Software Encryption - FSE 2015: 22nd International Workshop, Turkey. Revised Selected Papers*, pages 175–189, 2015.

[TLS16] Yosuke Todo, Gregor Leander, and Yu Sasaki. Nonlinear Invariant Attack –Practical Attack on Full SCREAM, iSCREAM, and Midori64. Cryptology ePrint Archive, Report 2016/732, 2016. http://eprint.iacr.org/2016/732.

[Tod15a] Yosuke Todo. Integral cryptanalysis on full MISTY1. In *Advances in Cryptology - CRYPTO 2015: 35th Annual Cryptology Conference, Santa Barbara, CA, USA. Proceedings, Part I*, pages 413–432, 2015.

[Tod15b] Yosuke Todo. Structural evaluation by generalized integral property. In *Advances in Cryptology - EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Bulgaria. Proceedings, Part I*, pages 287–314, 2015.

[WCC+16] Meiqin Wang, Tingting Cui, Huaifeng Chen, Ling Sun, Long Wen, and Andrey Bogdanov. Integrals Go Statistical: Cryptanalysis of Full Skipjack Variants. In *Fast Software Encryption - FSE 2016: 23rd International Conference, Germany. Revised Selected Papers*, pages 399–415, 2016.

[ZWF07] Wentao Zhang, Wenling Wu, and Dengguo Feng. New Results on Impossible Differential Cryptanalysis of Reduced AES. In Kil-Hyun Nam and Gwangsoo Rhee, editors, *Information Security and Cryptology - ICISC 2007: 10th International Conference, Korea. Proceedings*, pages 239–250, 2007.