General Practical Cryptanalysis of the Sum of Round-Reduced Block Ciphers and ZIP-AES

Antonio Flórez-Gutiérrez[1][0000-0001-7749-8925],
Lorenzo Grassi[2][0000-0003-1140-0520], Gregor Leander[2][0000-0002-2579-8587],
Ferdinand Sibleyras[1], and Yosuke Todo[1][0000-0002-6839-4777]

[1] NTT Social Informatics Laboratories, Tokyo, Japan
{antonio.florez,yosuke.todo}@ntt.com
[2] Ruhr University Bochum, Bochum, Germany
{lorenzo.grassi,gregor.leander}@rub.de

Abstract. We introduce a new approach between classical security proofs of modes of operation and dedicated security analysis for known cryptanalysis families: General Practical Cryptanalysis. This allows us to analyze generically the security of the sum of two keyed permutations against known attacks. In many cases (of course, not all), we show that the security of the sum is strongly linked to that of the composition of the two permutations. This enables the construction of beyond-birthday bound secure low-latency PRFs by cutting a known-to-be-secure block cipher into two equal parts. As a side result, our general analysis shows an inevitable difficulty for the key recovery based on differential-type attacks against the sum, which leads to a correction of previously published attacks on the dedicated design Orthros.

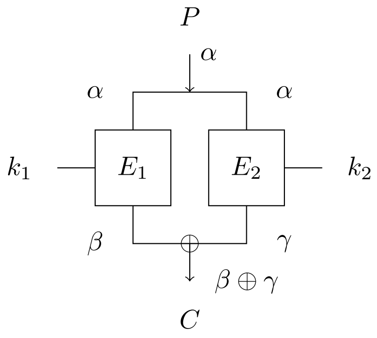
1 Introduction

Symmetric primitives are used to encrypt most of our sensitive data in virtually all applications. Block ciphers are arguably the most studied primitives.

Overhead of Modes. In order to encrypt actual data, primitives have to be used in a mode-of-operation. As a consequence of block ciphers being the most studied primitives, the majority of symmetric-key cryptographic schemes are built as block cipher modes. The advantage of using primitives in a mode-of-operation instead of directly designing an (authenticated) encryption is obvious: a well-designed mode comes with a proof that reduces its security to the security of the primitive. Using such a mode with a well-understood (block) cipher results in a secure scheme. One example is the counter-mode, where a pseudo-random function (PRF) is constructed by encrypting a counter. Indeed, AES-CRT is a frequently used scheme for encryption. In this paper, we instead focus on the sum of two block ciphers. Given two pseudo-random permutations (PRPs) (or independent block ciphers) E_k and E'_k, the sum $E_k(x) \oplus E'_k(x)$ is a secure PRF.

However, modes have a significant overhead. For example, AES-CRT is only secure only up to the birthday bound. For better security, modes with two (or more) calls to the block cipher are required. Turning our focus to the sum-of-PRP construction, we wonder whether it is necessary that both parts are secure PRPs. This question was already posed by the dedicated PRF Orthros [BIL+21], which consists of the sum of two specific keyed permutations that would not be secure block ciphers individually. A similar approach was taken in [MN17b], where AES-PRF is proposed as a round-reduced instance of the EDMD construction presented in [MN17a]. The security of AES-PRF required dedicated cryptanalysis to explain why known attacks do not apply. Interestingly, the authors of [MN17b] state that the sum construction seems more risky than the EDMD construction, an opinion we clearly object to as explained below. The main difference with AES-PRF and Orthros is that we are interested in a more general approach.

Link to Composition. As an example, consider a differential attack on the sum construction. One would typically consider an input difference α that would be input to both parts and try to find the most probable output differences β and γ for the individual parts, leading finally to an output difference of $\beta \oplus \gamma$.



The starting point for our work is the observation that the probability for this event, assuming the independence of the parts, is the same as the probability of the following differential trail on the composition of E_1^{-1} and E_2.

$$\begin{tikzcd} A \arrow[r, "\beta"] & \boxed{E_1^{-1}} \arrow[r, "\alpha"] & \boxed{E_2} \arrow[r, "\gamma"] & B \end{tikzcd}$$

That is, at least intuitively, *the sum construction is as secure as the composition with respect to differential distinguishers*. Ideally, we might hope for a result stating that if $E_2 \circ E_1^{-1}$ is a secure (strong) PRP, then $E_1 \oplus E_2$ is a secure PRF. Before discussing why this is not actually true, let us elaborate on how useful such a statement would be. Such a statement would allow us to *take any secure block cipher, split it into two parts, and obtain a secure PRF*. This would (i) remove the overhead of having two calls to a secure cipher (ii) remove the need for dedicated cryptanalysis as done in Orthros and (iii) result in a PRF with roughly half the latency of the corresponding block cipher.

The problem is, as mentioned, the result is wrong. The easiest example is to take E_1^{-1} to be identity. Then, the resulting scheme is the classical feed-forward construction for which distinguishing attacks exist with square-root complexity. So the main question was if and how this statement might be corrected without losing the great advantages it would provide.

Latency. Latency is an especially important fundamental criterion for the design of symmetric primitives. Indeed, compared to other performance criteria, low latency is much harder to achieve. In a nutshell, asking for a minimal latency cipher is asking about the minimal amount of computation necessary to obtain a secure cipher - a question as fundamental as it is open. Besides being a fundamental property, low latency ciphers have important applications, with memory encryption being one of the most prominent. There are a few dedicated low-latency designs, e.g. PRINCE [BCG+12], PRINCEv2 [BEK+20], MANTIS [BJK+16], QARMA [Ava17], QARMAv2 [ABD+23], and SPEEDY [LMMR21]. While all these designs use different ideas, their latency seems to converge. Differences in latency are mainly due to different security margins. Substantially improving latency with another block cipher design seems hard if not impossible, which means the possibility of essentially halving the latency with the sum of permutations construction is very enticing.

Our Contribution. It turns out it is possible to show that a practically identical statement holds to an extent. For this, we introduce a new approach which lies between general security reduction on modes of operation and dedicated security analysis of a specific primitive. Specifically, we compare, without analyzing the inside of each component, the security of the sum of two components with their composition. We name this approach General Practical Cryptanalysis.

We show that for many attack families, distinguishers on the sum construction are related to distinguishers on the composition. In the case of the two main attack families, differential and linear distinguishers, as well as their variants, their behaviors are very similar. In particular, (i) differential and linear trails have the same probability/correlation in $E_1 \oplus E_2$ as in $E_2 \circ E_1^{-1}$ or $E_2^{-1} \circ E_1$ and (ii) differential-linear and boomerang distinguishers on $E_2 \circ E_1^{-1}$ are equivalent to differential-and-linear and second order differential distinguishers on $E_1 \oplus E_2$. Of course, there are exceptions; for example, the sum construction is only as strong as the strongest part against the integral attack.

An attack on a symmetric primitive is, in most cases, built from a distinguisher and a key-recovery part. Equally interesting as the results on distinguishers is, therefore, to understand how one can add key-recovery rounds to the different distinguishers on the sum construction. Returning to the example of differential cryptanalysis, it is intuitively clear that adding key recovery at the end is unpromising. Adding key recovery at the top is also more difficult than for the composition, as one has to control both branches simultaneously. We argue that this is only possible under strict conditions. As an interesting side result, our general findings imply that the previous differential attack on Orthros published in [LSW22] must be reviewed.

This novel practical general approach leads to our main result: with respect to the most important attack vectors (with the exceptions mentioned above), the sum $E_1 \oplus E_2$ is as secure as the composition $E_2 \circ E_1^{-1}$. Taking a secure block cipher and splitting it into equal parts, with some additional analysis to cover the exceptions, leads to a PRF that is secure against all known attacks. Of course, this does not rule out the existence of new attacks, but this is the case for all new symmetric primitives.

Instances. To showcase the power and flexibility of our approach, we give a concrete instance in Sect. 4: ZIP-AES, a variant built as the sum of two 5-round AES. This results in a secure PRF with half the latency of AES-CTR and twice the security in terms of data complexity. When implemented with AES-NI, as inverse rounds are more costly, it does not achieve half the latency, but still provides slightly better running times, as detailed in Sect. 4.3.

We finally mention that a ZIP cipher based on a 64-bit lightweight block cipher is promising, e.g., ZIP-GIFT in Sect. 5. The resulting PRF is secure up to the entire 2^{64} blocks, which is enough for all practical cases, while the counter mode of such a 64-bit block cipher can be broken with only 2^{32} blocks of data complexity. Again, not only would security double, but the latency would also be halved, and therefore, it would be very competitive with the dedicated low-latency designs mentioned above.

2 Preliminaries

2.1 Known Attacks on Symmetric Primitives

We work a lot with linear and differential attacks and their variants. We expect the reader to be familiar with them and use this section to fix our notation.

Differential Cryptanalysis [BS90]**.** Differential attacks use pairs of plaintexts with a well-chosen difference. For a function $\mathsf{F} : \mathbb{F}_2^n \to \mathbb{F}_2^m$, a given input difference $\alpha \in \mathbb{F}_2^n$, and an output difference $\beta \in \mathbb{F}_2^m$, we denote by

$$\mathrm{Prob}(\alpha \xrightarrow{\mathsf{F}} \beta) = \frac{|\{x \in \mathbb{F}_2^n \mid \mathsf{F}(x) \oplus \mathsf{F}(x \oplus \alpha) = \beta\}|}{2^n}$$

the probability that the difference α results in the difference β. Given two (or more) functions $\mathsf{F} : \mathbb{F}_2^n \to \mathbb{F}_2^m$ and $\mathsf{G} : \mathbb{F}_2^m \to \mathbb{F}_2^\ell$, a differential trail or characteristic for $\mathsf{G} \circ \mathsf{F}$ also includes an intermediate difference γ. Its probability is usually estimated by multiplying the probabilities

$$\mathrm{Prob}(\alpha \xrightarrow{\mathsf{F}} \gamma \xrightarrow{\mathsf{G}} \beta) \simeq \mathrm{Prob}(\alpha \xrightarrow{\mathsf{F}} \gamma) \cdot \mathrm{Prob}(\gamma \xrightarrow{\mathsf{G}} \beta),$$

which can be justified if F and G are key-alternating ciphers with independent round keys and considering the average probability over all keys. From now on, we adopt this independence assumption. Without assumptions, it holds that

$$\mathrm{Prob}(\alpha \xrightarrow{\mathsf{G} \circ \mathsf{F}} \beta) = \sum_{\gamma} \mathrm{Prob}(\alpha \xrightarrow{\mathsf{F}} \gamma \xrightarrow{\mathsf{G}} \beta),$$

which is referred to as a differential in contrast to a differential trail.

Linear cryptanalysis [Mat93]. A linear approximation is a linear combination of input and output bits of the cipher. The main measure of its quality is its correlation. Given a function F, an input mask α, and output mask β, it's

$$\mathrm{cor}_{\mathsf{F}}(\alpha, \beta) = \mathrm{Prob}_x \left(\langle \beta, \mathsf{F}(x) \rangle = \langle \alpha, x \rangle \right) - \mathrm{Prob}_x \left(\langle \beta, \mathsf{F}(x) \rangle \neq \langle \alpha, x \rangle \right).$$

Again, given two functions, a linear trail for the composition is specified by an input mask α, an intermediate mask γ, and an output mask β, and its correlation contribution is formally defined as $\mathrm{cor}_{\mathsf{F}}(\alpha, \gamma)\, \mathrm{cor}_{\mathsf{G}}(\gamma, \beta)$. The set of all linear trails sharing the same input and output masks is often called linear hull. This definition is motivated by the fact that

$$\mathrm{cor}_{\mathsf{G} \circ \mathsf{F}}(\alpha, \beta) = \sum_{\gamma} \mathrm{cor}_{\mathsf{F}}(\alpha, \gamma)\, \mathrm{cor}_{\mathsf{G}}(\gamma, \beta).$$

Similarly, given a Boolean function $f : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2$, its correlation is

$$\mathrm{cor}(f) = \mathrm{Prob}_x \left(f(x) = 0 \right) - \mathrm{Prob}_x \left(f(x) = 1 \right).$$

Differential-linear cryptanalysis. The data complexity is given by the autocorrelation, which for an input difference δ and output mask α is defined as

$$\mathrm{Aut}_{\mathsf{F}}(\delta, \alpha) = \mathrm{Prob}_x \left(\langle \alpha, \mathsf{F}(x) \oplus \mathsf{F}(x \oplus \delta) \rangle = 0 \right) - \mathrm{Prob}_x \left(\langle \alpha, \mathsf{F}(x) \oplus \mathsf{F}(x \oplus \delta) \rangle = 1 \right).$$

In most cases, it is infeasible to obtain all trails in a linear hull or a differential. Hence, security arguments are often based on bounding the probability or correlation of trails. We mainly stick to this approach in this work.

2.2 The Sum-of-PRPs

Constructing PRFs from PRPs is a well-studied topic from a provable security perspective. The sum-of-PRPs construction is a well-known research topic. It was initially introduced by Bellare et al. at EUROCRYPT 1998 [BKR98]. The first proof of its security was given by Lucks at EUROCYPT 2000 [Luc00], where he proved a suboptimal security bound up to $2^{2n/3}$ queries. This was improved by Bellare and Impagliazzo [BI99] to $2^n/n$. Finally, with the introduction of the H-coefficient technique, Patarin [Pat10] proved the optimal full n-bit security, and Dutta et at. in [DNS22] filled some gaps in Patarin's proof. Very recently, Dinur [Din24], using Fourier-analysis, proved optimal bounds for the general case of the sum of permutations and the multi-user setting. A good survey of the state of the art of this and other constructions is given in the later paper as well as in [JN22].

Complementing this line of work, some recent work has focused on the question of constructing a public function from public (i.e., non-keyed) permutations.

This setting requires the notion of indifferentiability and is technically more involved. After several attempts that turned out to be flawed or non-optimal, the work of Gunsing et al. finally settled the result at CRYPTO 2023 [GBJ+23].

Despite the general usefulness of constructing a pseudo-random function, there was for a long time no practical cryptanalysis discussion against this construction, mainly because there were no practical instances that have been used or even proposed. The first concrete design was, to the best of our knowledge, Orthros [BIL+21]. Motivated by the fact that the output of each pseudo-random permutation is not visible to the attacker, the authors used the so-called proof-then-prune approach [HKR15] to realize an efficient pseudo-random function by reducing the rounds of the two parts. This significantly improved the latency of the resulting scheme but required dedicated cryptanalysis. As discussed below, getting this analysis right is more difficult than usual, in particular when considering differential-type attacks with key recovery.

To capture all designs derived by summing two not necessarily pseudo-random permutations, we give the following general definition.

Definition 1 **($P \oplus Q$).** *Let P, Q be two families of permutations, indexed by the keys k_p, k_q in the sets $\mathcal{P}$ and $\mathcal{Q}$, respectively:*

$$(x, k_P) \in \mathbb{F}_2^n \times \mathcal{P} \mapsto P_{k_p}(x) \in \mathbb{F}_2^n, \qquad (x, k_Q) \in \mathbb{F}_2^n \times \mathcal{Q} \mapsto Q_{k_q}(x) \in \mathbb{F}_2^n.$$

We define the $P \oplus Q$ construction as the following family of functions:

$$\begin{aligned} P \oplus Q : \mathbb{F}_2^n \times \mathcal{P} \times \mathcal{Q} &\to \mathbb{F}_2^n \\ (x, (k_P, k_Q)) &\mapsto P_{k_p}(x) \oplus Q_{k_q}(x). \end{aligned}$$

Unlike in provable security analysis, it is not assumed that P and Q are pseudo-random permutations. In other words, P and Q are not necessarily secure block ciphers with sound security claims on their own. Our objective is to reveal whether $P \oplus Q$ enhances the practical security in the context of cryptanalysis.

3 General Practical Cryptanalysis of $P \oplus Q$

This section discusses the resistance of the $P \oplus Q$ construction against well-known attack families, and compares it to compositions of P, Q and their inverses. As stated above, for our arguments, we make the usual assumption on the independence of rounds and therefore multiply probabilities over multiple rounds. While for attacks, this tends to lead to flawed complexity estimations, for security arguments there is currently no alternative technique avoiding this.

3.1 Differential Cryptanalysis

Differential Characteristic Equivalence. The differential trails of the parallel construction $P \oplus Q$ are tightly linked to those of the sequential construction $Q \circ P^{-1}$, as shown by the following result:

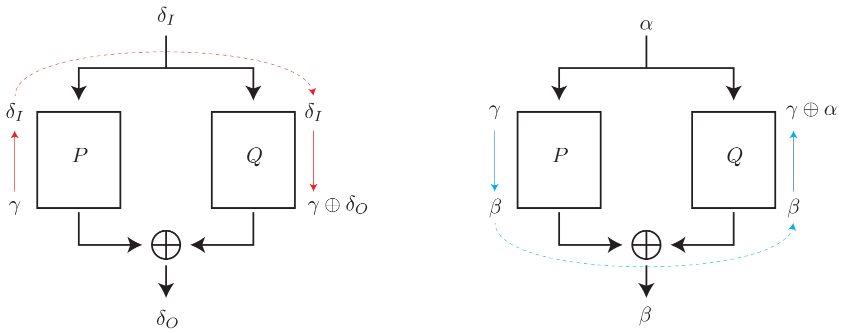


Fig. 1: Differential and linear trail equivalence

Proposition 1. *Let P, Q be two keyed permutations over $\mathbb{F}_2^n$, and let $\mathsf{F} := P \oplus Q$ and $\mathsf{S} := Q \circ P^{-1}$. For each differential trail with probability p traversing F, there is a trail traversing S with the same probability p.*

Proof. Given $\delta_I, \delta_O \in \mathbb{F}_2^n$, we consider the differential $\delta_I \xrightarrow{\mathsf{F}} \delta_O$. All its trails take the same form given by the choice of $\gamma \in \mathbb{F}_2^n$ and have probability

$$p = \mathrm{Prob}(\delta_I \xrightarrow{P} \gamma) \cdot \mathrm{Prob}(\delta_I \xrightarrow{Q} \gamma \oplus \delta_O).$$

Since $\mathrm{Prob}(\delta_I \xrightarrow{P} \gamma) = \mathrm{Prob}(\gamma \xrightarrow{P^{-1}} \delta_I)$, p is also the probability of the differential trail $\gamma \xrightarrow{P^{-1}} \delta_I \xrightarrow{Q} \gamma \oplus \delta_O$ traversing S. □

The left diagram in Fig. 1 shows the trail equivalence between $P \oplus Q$ and $Q \circ P^{-1}$.

Aggregating the Trails. While individual trails of $P \oplus Q$ and $Q \circ P^{-1}$ are equivalent (and thus both have the same maximum differential trail probability), it is hard to compare the resulting differential probabilities when adding up all the trail probabilities in a differential. We can try to compare the expected differential probability (EDP) of both constructions:

$$\mathrm{Prob}(\delta_I \xrightarrow{P \oplus Q} \delta_O) = \sum_{\gamma} \mathrm{Prob}(\gamma \xrightarrow{P^{-1}} \delta_I) \cdot \mathrm{Prob}(\delta_I \xrightarrow{Q} \gamma \oplus \delta_O),$$

$$\mathrm{Prob}(\delta_I \xrightarrow{Q \circ P^{-1}} \delta_O) = \sum_{\gamma} \mathrm{Prob}(\delta_I \xrightarrow{P^{-1}} \gamma) \cdot \mathrm{Prob}(\gamma \xrightarrow{Q} \delta_O).$$

However, we quickly realize that both sums cover sets of differential trails which are non-equivalent, which makes further analysis difficult. Indeed, in the case of $P \oplus Q$, the sum covers all trails $\gamma \xrightarrow{P^{-1}} \delta_I \xrightarrow{Q} \gamma \oplus \delta_O$ for all γ, and $\delta_I \xrightarrow{P^{-1}} \gamma \xrightarrow{Q} \delta_O$ in the case of $Q \circ P^{-1}$. Therefore, the maximum expected differential probability (MEDP) is not necessarily identical.

Taka et al. studied this effect on multiple-branch-based designs and investigated the differential clustering effect on Orthros [TISI23]. They focused on several γ, evaluated the clustering effect on each branch for each γ, and combined them. On the other hand, in general, we do not expect either $P \oplus Q$ or $Q \circ P^{-1}$ to have a stronger clustering effect because the number of terms in both sums is the same. More importantly, the clustering inside P and Q is exactly the same in both cases. We also note that if P and Q are almost the same structure, $\mathrm{Prob}(\delta_I \xrightarrow{P \oplus Q} 0)$ is expected to be high, but so will be $\mathrm{Prob}(\delta_I \xrightarrow{Q \circ P^{-1}} \delta_I)$.

On Key Recovery in Differential Cryptanalysis. Regarding the key recovery based on the differential attack, $P \oplus Q$ appears to be more resilient than $Q \circ P^{-1}$. More precisely, we find an inevitable difficulty in mounting an effective key-recovery attack on $P \oplus Q$.

The most common strategy for the key-recovery attack is to append key-recovery rounds to the differential distinguisher. We construct a differential distinguisher and append key-recovery rounds for attacking more rounds. The data complexity depends on the probability of the differential distinguisher, since the key-recovery rounds are deterministic under each key guess. We now consider two possible key-recovery strategies: it is added to the output or input.

Key Recovery on the Output Side. The output is $P(x) \oplus Q(x)$, where $P(x)$ and $Q(x)$ are unknown to the attacker. It is unlikely to add key recovery unless the attacker can compute at least part of (differences in) $P(x)$ or $Q(x)$. We suppose P and Q contain almost the same rounds. This implies that the key-recovery part can cover half of the total round when we attack the composition. As long as this is not the case, adding key-recovery at the output is not possible.

Key Recovery on the Input Side. Key recovery on the input side seems more natural because the attacker knows or even chooses the inputs to P and Q. We consider a differential key-recovery attack on $\mathsf{F} := (P_2 \circ P_1) \oplus (Q_2 \circ Q_1)$, where the input differences to P_2 and Q_2 are fixed to δ_P and δ_Q, respectively. Therefore, we exploit a high differential probability $p = \mathrm{Prob}((\delta_P, \delta_Q) \xrightarrow{P_2 \oplus Q_2} \delta_O)$ with key recovery on P_1 and Q_1. Conventionally, the data complexity can be p^{-1} in the optimal case, but we show such a strategy does not work.

Proposition 2. *Let $\mathsf{F} = (P_2 \circ P_1) \oplus (Q_2 \circ Q_1)$. We consider a differential key-recovery attack where the input differences of P_2 and Q_2 are fixed to δ_P and δ_Q, respectively, and the output difference is δ_O. The necessary key material from P_1 and Q_1 is guessed. Such an attack works only when*

$$\mathrm{Prob}((\delta_P, \delta_Q) \xrightarrow{P_2 \oplus Q_2} \delta_O) \cdot \mathrm{Prob}(\delta_P \xrightarrow{Q_1 \circ P_1^{-1}} \delta_Q) > 2^{-n} .$$

Proof. Let us count the number of input pairs X, X' to $P \oplus Q$ that produce a difference of δ_P after P_1 and δ_Q after Q_1 simultaneously.

$$\begin{aligned} T &= |\{(X, X') \mid P_1(X) \oplus P_1(X') = \delta_P \text{ and } Q_1(X) \oplus Q_1(X') = \delta_Q\}| \\ &= |\{(x, x \oplus \delta_P) \mid Q_1 \circ P_1^{-1}(x) \oplus Q_1 \circ P_1^{-1}(x \oplus \delta_P) = \delta_Q\}| \\ &= 2^n \cdot \mathrm{Prob}(\delta_P \xrightarrow{Q_1 \circ P_1^{-1}} \delta_Q) \end{aligned}$$

Observing that the expected data complexity for the distinguisher is at least the inverse of the probability of the differential and at most T, i.e.

$$\mathrm{Prob}((\delta_P, \delta_Q) \xrightarrow{P_2 \oplus Q_2} \delta_O)^{-1} < T$$

leads to the claimed result. □

In practice, the attacker would choose a differential trail given by $\delta_P \xrightarrow{P_2} \gamma$ and $\delta_Q \xrightarrow{Q_2} \gamma \oplus \delta_O$ and estimate the probability of the resulting distinguisher as

$$\mathrm{Prob}((\delta_P, \delta_Q) \xrightarrow{P_2 \oplus Q_2} \delta_O) \approx \mathrm{Prob}(\delta_P \xrightarrow{P_2} \gamma) \cdot \mathrm{Prob}(\delta_Q \xrightarrow{Q_2} \gamma \oplus \delta_O).$$

The usual condition $\mathrm{Prob}(\delta_P \xrightarrow{P_2} \gamma) \cdot \mathrm{Prob}(\delta_Q \xrightarrow{Q_2} \gamma \oplus \delta_O) > 2^{-n}$ is not sufficient for an attack to be possible. If

$$\mathrm{Prob}(\gamma \xrightarrow{P_2^{-1}} \delta_P) \cdot \mathrm{Prob}(\delta_P \xrightarrow{Q_1 \circ P_1^{-1}} \delta_Q) \cdot \mathrm{Prob}(\delta_Q \xrightarrow{Q_2} \gamma \oplus \delta_O) < 2^{-n},$$

there may be no pairs satisfying the differential characteristic.

Review of the Differential Key-Recovery Attack against Orthros in [LSW22]. Proposition 2 implies that the data complexity of a differential key-recovery attack must be estimated carefully. In a nice paper at Africacrypt 2022, Li, Sun, and Wang proposed differential cryptanalysis against round-reduced Orthros. Their attacks add a 1-round key recovery to the input side of both branches. Specifically, they prepared pairs of chosen plaintexts whose differences take the form

$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \delta_1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \delta_2, 0, 0, 0, \delta_3, 0).$$

Branch 1 requires three nibble difference transitions in the Sbox layer: $\delta_1 \xrightarrow{S}$ `0x2`, $\delta_2 \xrightarrow{S}$ `0x2`, and $\delta_3 \xrightarrow{S}$ `0x8`. Similarly, branch 2 requires $\delta_1 \xrightarrow{S}$ `0x8`, $\delta_2 \xrightarrow{S}$ `0x1`, and $\delta_3 \xrightarrow{S}$ `0x2`. Excluding these first S-box layers, the differential probability on each branch is estimated as 2^{-64} and 2^{-48}, so the total probability is $p = 2^{-112}$. They finally estimated the data complexity as 2^{115} based on their attack framework.

Proposition 2 implies that a key-recovery attack is possible only when

$$p \cdot \mathrm{Prob}(\texttt{0x2} \xrightarrow{S \circ S^{-1}} \texttt{0x8}) \cdot \mathrm{Prob}(\texttt{0x2} \xrightarrow{S \circ S^{-1}} \texttt{0x1}) \cdot \mathrm{Prob}(\texttt{0x8} \xrightarrow{S \circ S^{-1}} \texttt{0x2}) > 2^{-128}.$$

This probability highly depends on the key (difference) involved in $S \circ S^{-1}$. The detailed review is shown in Appendix A. We notice that the probability is zero for more than half of the keys in each Sbox. Therefore, it is a weak-key attack whose fraction of weak keys is $5/16 \times 7/16 \times 5/16 \approx 2^{-4.55}$.

We assume that one of the weak keys is used. Since the attacker does not know which (weak) key is used, the attacker must fully activate corresponding 12-bit inputs. Among 12-bit active inputs, we can construct about 2^{24} pairs. However, given a fixed key, the number of pairs satisfying input differences of both branches is limited. In some (weak) keys, the number is only 8 (see Appendix A for details). Therefore, to observe differential characteristics with $p = 2^{-112}$, we need at least 2^{109} texts in addition to the 12-bit active. As a result, the attacker must use at least $2^{109+12} = 2^{121}$ chosen plaintexts to lead a valid key-recovery attack for all keys belonging to the weak keys, which is more than 2^{115} by the analysis of [LSW22].

Remark 1. Assuming that the keys in three active S-boxes are identical in P and Q, the input differences of the two branches must be the same because $\mathrm{Prob}(\delta_P \xrightarrow{S \circ S^{-1}} \delta_Q) = 0$ for $\delta_P \neq \delta_Q$. In other words, to lead the key-recovery attack that is valid for all keys, it is necessary to construct differential characteristics whose input differences are equal in both branches.

3.2 Linear Cryptanalysis

Linear Characteristic Equivalence. Similarly to the differential cryptanalysis, the linear trails of $P \oplus Q$ are equivalent to those of the sequential construction $Q^{-1} \circ P$, as shown in the right diagram of Fig. 1, and by the following result:

Proposition 3. *Let P, Q be two keyed permutations over $\mathbb{F}_2^n$, and let $\mathsf{F} := P \oplus Q$ and $\mathsf{S}^* := Q^{-1} \circ P$. For each linear trail with correlation c traversing F, there is a linear trail with the same correlation c traversing S^*.*

Proof. Consider any masks $\alpha, \gamma, \beta \in \mathbb{F}_2^n$, let $c = \mathrm{cor}_P(\gamma, \beta)\,\mathrm{cor}_Q(\gamma \oplus \alpha, \beta)$ be the correlation of a linear trail through F. Again, notice that

$$\mathrm{cor}_P(\gamma, \beta)\,\mathrm{cor}_Q(\gamma \oplus \alpha, \beta) = \mathrm{cor}_P(\gamma, \beta)\,\mathrm{cor}_{Q^{-1}}(\beta, \gamma \oplus \alpha).$$

Thus, c is the correlation of the linear trail $\gamma \xrightarrow{P} \beta \xrightarrow{Q^{-1}} \gamma \oplus \alpha$ traversing S^*. □

Similar to differential cryptanalysis, while individual trails or characteristics are equivalent, it is hard to compare the resulting linear approximation correlation when adding up the trail correlation contributions:

$$\mathrm{cor}_{\mathsf{F}}(\alpha, \beta) = \sum_{\gamma} \mathrm{cor}_P(\gamma, \beta)\,\mathrm{cor}_Q(\gamma \oplus \alpha, \beta),$$
$$\mathrm{cor}_{\mathsf{S}^*}(\alpha, \beta) = \sum_{\gamma} \mathrm{cor}_P(\alpha, \gamma)\,\mathrm{cor}_Q(\beta, \gamma).$$

It is possible that the largest correlations of the linear approximations F and S^* are not the same, due to differences in the clustering effect for both constructions.

About Sequential Applications. One peculiar aspect of Propositions 1 and 3 is that the sequential function with equivalent trails or characteristics differs between the differential ($\mathsf{S} := Q \circ P^{-1}$) and linear ($\mathsf{S}^* := Q^{-1} \circ P$) cases. This occurs because differential trails traversing F must coincide in the input of the two branches (the output differentials can be added) while linear trails must coincide in the output of the two branches (the input masks can be added).

However, in the case in which $\mathcal{Q} = \{P^{-1} | P \in \mathcal{P}\}$, then the compositions $\mathsf{S} = \mathsf{S}^*$ conform the same set of permutations, and F has the same differential and linear characteristics as $P_1 \circ P_2$, where $P_1, P_2 \in \mathcal{P}$. This is the ZIP-design strategy we employ in Sections 4 and 5.

We note that the behavior of both constructions is not necessarily the same when it comes to trail clustering (so that the maximum differential probability or correlation may still differ). Again, all the clustering that happens within P and/or Q is equivalent in S, S^*, and $P \oplus Q$. Thus, even so our argument does not cover all possible clustering, it covers more than done in both attacks in the vast majority of cases.

On Key Recovery in Linear Cryptanalysis. Unlike with differential cryptanalysis, it is possible to mount linear key-recovery attacks on $P \oplus Q$. While it is not possible on the output side due to the irreversibility of the XOR operation, it is possible on the input side. Indeed, assume that the branches can be written as $P = P_2 \circ P_1$ and $Q = Q_2 \circ Q_1$. We are given a linear approximation of $P_2 \oplus Q_2$, and we want to perform key recovery over P_1 and Q_1. As long as the combined size of the necessary key guesses to determine the parity of the input masks to P_2 and Q_2 is small enough, it is possible to perform key recovery on both P_1 and Q_1 simultaneously without increasing the data complexity. Linear cryptanalysis is a known plaintext attack, so the cryptanalyst does not need to control internal values in either branch and, most notably, does not need to control both branches at the same time, which is the impediment to differential key-recovery attacks). In summary, linear key-recovery attacks over the first few rounds of P and Q can be carried out in the same manner as on an iterative block cipher. Thus, assuming that differential and linear distinguishers cover the same number of rounds, linear cryptanalysis may lead to stronger attacks.

3.3 Differential-Linear Cryptanalysis

We next look at differential-linear cryptanalysis. First, we investigate how the autocorrelation of $P \oplus Q$ is related to the properties of P and Q, and we find the following straightforward result:

Proposition 4. *Let P, Q be keyed permutations over $\mathbb{F}_2^n$ and let $\mathsf{F} = P \oplus Q$. Let $\delta \in \mathbb{F}_2^n$ be an input difference, and let $\alpha \in \mathbb{F}_2^n$ be an output linear mask. Then*

$$\mathrm{Aut}_{\mathsf{F}}(\delta, \alpha) = \mathrm{Aut}_P(\delta, \alpha) \cdot \mathrm{Aut}_Q(\delta, \alpha).$$

Proof. From the definition of the autocorrelation:

$$\begin{aligned}\mathrm{Aut}_{\mathsf{F}}(\delta,\alpha) &= \mathrm{cor}\left(\langle\alpha,\mathsf{F}(x)\rangle\oplus\langle\alpha,\mathsf{F}(x\oplus\delta)\rangle\right)\\ &= \mathrm{cor}\left(\langle\alpha,P(x)\rangle\oplus\langle\alpha,Q(x)\rangle\oplus\langle\alpha,P(x\oplus\delta)\rangle\oplus\langle\alpha,Q(x\oplus\delta)\rangle\right)\\ &= \mathrm{cor}\left(\langle\alpha,P(x)\rangle\oplus\langle\alpha,P(x\oplus\delta)\rangle\oplus\langle\alpha,Q(x)\rangle\oplus\langle\alpha,Q(x\oplus\delta)\rangle\right).\end{aligned}$$

Assuming the independence of both halves of the expression (or, alternatively, that $\mathrm{cor}\left(\langle\alpha,P(x)\rangle\oplus\langle\alpha,Q(x)\rangle\right)$ is negligible), we deduce:

$$\begin{aligned}\mathrm{Aut}_{\mathsf{F}}(\delta,\alpha) &= \mathrm{cor}\left(\langle\alpha,P(x)\rangle\oplus\langle\alpha,P(x\oplus\delta)\rangle\right)\cdot\mathrm{cor}\left(\langle\alpha,Q(x)\rangle\oplus\langle\alpha,Q(x\oplus\delta)\rangle\right)\\ &= \mathrm{Aut}_P(\delta,\alpha)\cdot\mathrm{Aut}_Q(\delta,\alpha)\end{aligned}$$

from the piling-up-lemma [Mat93]. □

We note two important differences between this result and the ones for differential and linear distinguishers. It describes the behavior of a whole differential-linear distinguisher without singling out an individual trail. However, the autocorrelation cannot generally be related to that on the composition of P, Q or their inverses, and relies just on the product of the autocorrelations for P and Q. This does not make a large difference for constructions in which the logarithm of the maximum autocorrelation decreases linearly with the number of rounds, but it may create a gap when this exponent decreases very quickly.

Practical Strategies for Finding DL Distinguishers. The autocorrelation of F is computed as the multiplication of the autocorrelations of P and Q having the same input difference and output mask. On the other hand, in practice a DL distinguisher is found by studying a trail perspective.

Traditionally, a cipher is separated into two parts, so that a differential trail is considered over the first part and a linear trail over the second. Let $P = P_l \circ P_d$ and $Q = Q_l \circ Q_d$, where differentials $\delta \xrightarrow{P_d} \delta_P$ and $\delta \xrightarrow{Q_d} \delta_Q$ and linear approximations on $\alpha_P \xrightarrow{P_l} \beta$ and $\alpha_Q \xrightarrow{Q_l} \beta$ are known. We consider the composition $\mathsf{S} := P_l^{-1} \circ Q_l \circ Q_d \circ P_d^{-1}$. Then, the differential-linear distinguishers $\delta \xrightarrow{\mathsf{F}} \beta$ and $\delta_P \xrightarrow{\mathsf{S}} \alpha_P$ are expected to have the same autocorrelation, assuming that these trails are dominant and independent. When P_d and P_l are iterations of the round function and Q_d and Q_l are iterations of the inverse round function, $P \oplus Q$ is equivalent to the composition.

On the other hand, we can consider truncated differentials, $(\delta_P, \delta_Q) \in U_P \times U_Q$, instead of a single differential trail. As mentioned later, the behavior of the truncated differential is different in $P \oplus Q$ and the composition. Moreover, the differential-linear hull aggregates multiple intermediate masks instead of a single intermediate mask. When we switch differential trails into linear trails, we also have the so-called independence assumption issue. In particular, the strategy above has two different switches for each side of P and Q. Considering such a complicated situation, it is preferable to analyze the autocorrelation of each branch rather than optimistically trusting the relationship to the composition.

On Key Recovery in Differential-Linear Cryptanalysis. Considering the differential-linear key recovery, a similar problem arises to the one shown in the differential key recovery: it is necessary to control input differences in both branches simultaneously, which puts a limitation on the usable distinguishers.

Proposition 5. *Let $\mathsf{F} = (P_2 \circ P_1) \oplus (Q_2 \circ Q_1)$. We consider a differential-linear key-recovery attack, where the input differences of P_2 and Q_2 are δ_P and δ_Q, respectively. The output linear mask is α. The necessary key material from P_1 and Q_1 is guessed. Such an attack works only when*

$$(\mathrm{Aut}_{P_2}(\delta_P, \alpha) \cdot \mathrm{Aut}_{Q_2}(\delta_Q, \alpha))^{-2} < 2^n \cdot \mathrm{Prob}(\delta_P \xrightarrow{Q_1 \circ P_1^{-1}} \delta_Q).$$

Proof. Let δ_P and δ_Q be fixed input differences of P_2 and Q_2, respectively. Let α be the output linear mask. Therefore, assuming the input pairs to P_2 and Q_2 already satisfy δ_P and δ_Q, the necessary number of pairs is estimated as $(\mathrm{Aut}_{P_2}(\delta_P, \alpha) \cdot \mathrm{Aut}_{Q_2}(\delta_Q, \alpha))^{-2}$. The number of available pairs satisfying δ_P and δ_Q at the same time is expected as $2^n \cdot \mathrm{Prob}(\delta_P \xrightarrow{Q_1 \circ P_1^{-1}} \delta_Q)$. Therefore, when this number is less than $(\mathrm{Aut}_{P_2}(\delta_P, \alpha) \cdot \mathrm{Aut}_{Q_2}(\delta_Q, \alpha))^{-2}$, the attacker cannot collect enough pairs to complete the attack. □

Review of the DL Key-Recovery Attack against Orthros in [LSW22]. We again review the existing attack against Orthros proposed at [LSW22]. It also presents differential-linear cryptanalysis. It uses a differential-linear distinguisher whose autocorrelation is 2^{-46}. They also estimated the data complexity to be 2^{95} chosen plaintexts.

This has the same problem as the key recovery in differential attacks, i.e., the attack is a weak-key attack and requires a higher data complexity than their estimation. The key-recovery structure is the same as the differential case. Therefore, the fraction of weak keys is $2^{-4.55}$. From 12-bit active inputs, there are weak keys, where the number of pairs satisfying input differences of both branches is only 8. Therefore, to recover any weak key, we need at least $2^{46 \times 2}/8 \times 2^{12} = 2^{101}$ chosen plaintexts, which is more than 2^{95} by the previous estimation.

3.4 Differential-and-Linear Key-Recovery Attack

In previous sections, we have noted that attacks which require the adversary to control an input difference in both branches are difficult to turn into key-recovery attacks. On the other hand, linear attacks lend themselves well to key recovery because of the known-plaintext nature. We next introduce a hybrid key-recovery attack which uses a differential-linear distinguisher on one of the branches and a linear distinguisher on the other. On the differential-linear branch, the key recovery can be performed because the attacker can control the input difference by choosing plaintexts as in a standard differential or differential-linear attack. On the linear branch, the attacker only needs to establish the parity of the input linear mask, so it does not interfere with the key recovery on the other branch.

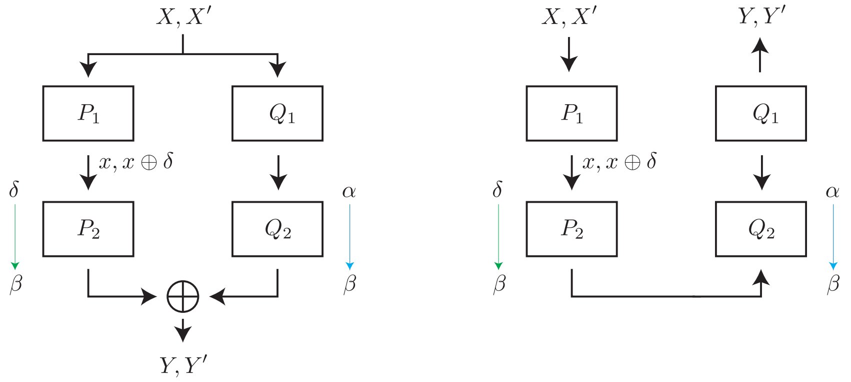


Fig. 2: The differential-and-linear key-recovery attack on $P \oplus Q$ (left) and differential-linear key-recovery attack on $Q^{-1} \circ P$.

Let us describe this situation in more detail (see Figure 2). P is divided into $P = P_2 \circ P_1$. Key recovery will be carried out on P_1 while a differential-linear distinguisher is considered on P_2 with input difference δ, output mask β, and autocorrelation c_1. Q is also divided into $Q = Q_2 \circ Q_1$ where Q_1 is reserved for key recovery, and a linear approximation with masks α and β and correlation c_2 is considered for Q_2. We note that the roles of P and Q can be exchanged.

By guessing parts of the key in P_1 and Q_1, the attacker can compute the following parity from arbitrary X.

$$\langle \beta, \mathsf{F}(X) \rangle \oplus \langle \beta, \mathsf{F}(P_1^{-1}(P_1(X) \oplus \delta)) \rangle \oplus \langle \alpha, Q_1(X) \rangle \oplus \langle \alpha, Q_1(P_1^{-1}(P_1(X) \oplus \delta)) \rangle.$$

Thus, by querying enough plaintexts, the attacker can obtain the experimental correlation.

We will first determine the correlation of this function, and then we will briefly describe the key-recovery attack algorithm. For the former, we note that we can, by expanding F, rearrange the formula as follows:

$$\begin{aligned} &\langle \beta, P_2(P_1(X)) \rangle \oplus \langle \beta, P_2(P_1(X) \oplus \delta) \rangle \oplus \\ &\langle \alpha, Q_1(X) \rangle \oplus \langle \beta, Q_2(Q_1(X)) \rangle \oplus \\ &\langle \alpha, Q_1(P_1^{-1}(P_1(X) \oplus \delta)) \rangle \oplus \langle \beta, Q_2(Q_1(P_1^{-1}(P_1(X) \oplus \delta))) \rangle \end{aligned}$$

From the assumptions on the distinguishers for P_2 and Q_2, the correlation of the first line is c_1, and the correlations of the second and third lines are c_2. As a result, and from the piling-up lemma, we deduce that the correlation for the whole expression is $c_1 \cdot c_2^2$, which means an attack can be mounted with data complexity $c_1^{-2} c_2^{-4}$.

We next sketch the key recovery algorithm for this attack. Using a key guess in P_1, the attacker can use structures to construct pairs (X, X') so that $P_1(X) \oplus P_1(X') = \delta$ in the same way they would for a differential or a differential-linear attack, and at the same cost. Once these pairs (X, X') are constructed, a guess of part of the key in Q_1 enables the attacker to determine the values of $\langle \alpha, Q_1(X) \rangle$ and $\langle \alpha, Q_1(X') \rangle$. With these, and for each key guess, the attacker can compute the experimental correlations of

$$\langle \beta, \mathsf{F}(X) \rangle \oplus \langle \beta, F(X') \rangle \oplus \langle \alpha, Q_1(X) \rangle \oplus \langle \alpha, Q_1(X') \rangle,$$

where X and X' are constructed so that $P_1(X) \oplus P_1(X') = \delta$. We verified our assumption and validity of our key-recovery attack by using ZIP-AES introduced in the next section. In detail, we discuss it in Appendix B.2.

Interestingly, again this kind of attack is related to a cryptanalysis on the composition of P and Q (see the right diagram of Fig. 2). Indeed, we notice that the differential-linear distinguisher on P_2 and the linear approximation of Q_2 can be combined into a differential-linear distinguisher on $Q_2^{-1} \circ P_2$. Furthermore, the whole key-recovery attack corresponds to a differential-linear key-recovery attack on $Q^{-1} \circ P$ guessing the same key material. However, we note that the autocorrelation of the differential-linear distinguisher on the composition may be larger, because the intermediate mask β is not fixed, while in the case of the attack on F the mask β has to be fixed by the attacker.

3.5 Truncated Differential Cryptanalysis

A variant of classical differential cryptanalysis is truncated differential cryptanalysis [Knu94], in which the attacker can predict only part of the difference between pairs of texts. When considering truncated differentials cryptanalysis, the parallel construction $\mathsf{F} := P \oplus Q$ seems to offer a security that is *hardly* comparable with any sequential construction and thus may require a dedicated analysis, which is also to be expected when compared to differential-linear attacks.

Firstly, the parallel and sequential constructions involving inverse permutations become hardly comparable as truncated differentials do not propagate backwards so that truncated differential characteristics in P generally differ from characteristics in P^{-1}.

Secondly, if we consider the sequential construction $\mathsf{S} := Q \circ P^{-1}$ then a truncated differential attack works as such for any linear subspaces $\mathcal{U}, \mathcal{V}, \mathcal{W}$:

$$\begin{aligned}
\text{Prob}\left(P^{-1}(x) \oplus P^{-1}(x \oplus \alpha) \in \mathcal{V} \mid x \in \mathbb{F}_2^n, \alpha \in \mathcal{U}\right) &= p \\
\text{Prob}\left(Q(x) \oplus Q(x \oplus \beta) \in \mathcal{W} \mid x \in \mathbb{F}_2^n, \beta \in \mathcal{V}\right) &= q \\
\Longrightarrow \text{Prob}\left(\mathsf{S}(x) \oplus \mathsf{S}(x \oplus \alpha) \in \mathcal{W} \mid x \in \mathbb{F}_2^n, \alpha \in \mathcal{U}\right) &\geq p \cdot q \,.
\end{aligned}$$

On the other hand, Proposition 6 shows how to mount a truncated differential attack on $P \oplus Q$:

Proposition 6. *Let P, Q be two keyed permutations over $\mathbb{F}_2^n$, and let $\mathsf{F} := P \oplus Q$. Let $\mathcal{U}_P, \mathcal{U}_Q, \mathcal{V}_P, \mathcal{V}_Q \subseteq \mathbb{F}_2^n$ be four non-trivial linear subspaces such that $\mathcal{U}_P \cap \mathcal{U}_Q$ is non-empty. Assume that the following truncated differentials hold with* *probabilities* $p, q \in (0, 1]$ *respectively:*

$$\begin{aligned}\text{Prob}\left(P(x) \oplus P(x \oplus \alpha) \in \mathcal{V}_P \mid x \in \mathbb{F}_2^n\,, \alpha \in \mathcal{U}_P\right) = p\,,\\ \text{Prob}\left(Q(x) \oplus Q(x \oplus \beta) \in \mathcal{V}_Q \mid x \in \mathbb{F}_2^n\,, \beta \in \mathcal{U}_Q\right) = q\,.\end{aligned}$$

Then:

$$\text{Prob}\left(\mathsf{F}(x) \oplus \mathsf{F}(x \oplus \gamma) \in \mathcal{V}_P \oplus \mathcal{V}_Q \mid x \in \mathbb{F}_2^n\,, \gamma \in \mathcal{U}_P \cap \mathcal{U}_Q\right) \geq p \cdot q\,.$$

We note that $\oplus$ denotes the sum of binary vector subspaces, which may not necessarily be a direct sum. Obviously, if $\mathcal{V}_P \oplus \mathcal{V}_Q = \mathbb{F}_2^n$ is the full space, the last probability is equal to 1, making the truncated differential to be meaningless. This is not the case for S.

Proof. Let $x \in \mathbb{F}_2^n$. We know that $P(x) \oplus P(x \oplus \gamma) \in \mathcal{V}_P$ with probability p over $\gamma \in \mathcal{U}_P$, and that $Q(x) \oplus Q(x \oplus \gamma) \in \mathcal{V}_Q$ with probability q over $\gamma \in \mathcal{U}_Q$. Assuming that both events are statistically independent of each other, over $\gamma \in \mathcal{U}_P \cap \mathcal{U}_Q$, the probability that they both occur at the same time is $p \cdot q$. Since $\mathcal{V}_P$ and $\mathcal{V}_Q$ are vector subspaces, we have

$$\mathsf{F}(x) \oplus \mathsf{F}(x \oplus \gamma) = P(x) \oplus Q(x) \oplus P(x \oplus \gamma) \oplus Q(x \oplus \gamma) \in \mathcal{V}_P \oplus \mathcal{V}_Q,$$

which concludes the proof. □

As shown in Proposition 6, an interesting constraint to find a truncated differential attack on $P \oplus Q$ is to find two linear subspaces $\mathcal{U}_P$ and $\mathcal{U}_Q$ such that both $\mathcal{U}_P \cap \mathcal{U}_Q$ is not empty and $\mathcal{V}_P \oplus \mathcal{V}_Q$ is not the full space $\mathbb{F}_2^n$. As a result, even if we find two truncated differentials, where p and q are high enough, it does not always guarantee a non-trivial truncated differential on F.

Based on this, we encourage to pay particular attention when arguing the security against truncated differentials.

On Key Recovery in Truncated Differential Attacks. Extending a truncated differential distinguisher into a key recovery presents the same problems discussed in Sect. 3.1 for the analogous case of differential cryptanalysis.

Proposition 7. *Let* $\mathsf{F} = (P_2 \circ P_1) \oplus (Q_2 \circ Q_1)$. *We consider a key-recovery attack, where the truncated input differences of* P_2 *and* Q_2 *are in the affine subspace* $\mathcal{U}_P$ *and* $\mathcal{U}_Q$ *respectively, and the key involved in* P_1 *and* Q_1 *is guessed. When* N *pairs are needed for the distinguishing attacks based on the truncated differential to succeed,* $(\mathcal{U}_P, \mathcal{U}_Q) \xrightarrow{P_2 \oplus Q_2} \mathcal{V}$, *such an attack works only when*

$$2^n \cdot |\mathcal{U}_P| \cdot \text{Prob}(\mathcal{U}_P \xrightarrow{Q_1 \circ P_1^{-1}} \mathcal{U}_Q) > N\,.$$

As the input of P_2, the number of pairs satisfying the truncated differential is $2^n \cdot |\mathcal{U}_P|$. To mount the key recovery, the attacker needs to find pairs that satisfy the truncated differential in the input of Q_2 simultaneously. Therefore, the number of pairs we can collect is $2^n \cdot |\mathcal{U}_P| \cdot \text{Prob}(\mathcal{U}_P \xrightarrow{Q_1 \circ P_1^{-1}} \mathcal{U}_Q)$. If this value is less than N, it is insufficient to execute the key-recovery attack.

Impossible (Truncated) Differentials. An impossible (truncated) differential [BBS99] is a (truncated) differential that holds with probability 0. In general, the existence of impossible differentials for the composition does not imply the existence of non-trivial[3] impossible differentials for $\mathsf{F} := P \oplus Q$.

Assuming $\mathrm{Prob}(\delta_I \xrightarrow{Q^{-1}\circ P} \delta_O) = 0$, let $\mathcal{V}_P$ and $\mathcal{V}_Q$ denote a subset satisfying $\mathrm{Prob}(\delta_I \xrightarrow{P} \mathcal{V}_P) = \mathrm{Prob}(\delta_O \xrightarrow{Q} \mathcal{V}_Q) = 1$, and $\mathcal{V}_P \cap \mathcal{V}_Q = \phi$. In contrast, assuming $\mathrm{Prob}(\delta_I \xrightarrow{\mathsf{F}} \delta_O) = 0$, it implies $\mathrm{Prob}(\delta_I \xrightarrow{P} \mathcal{V}_P) = \mathrm{Prob}(\delta_I \xrightarrow{Q} \mathcal{V}_Q) = 1$, and $\mathcal{V}_P \cap (\mathcal{V}_Q \oplus \delta_O) = \phi$. The former can choose both input differences for P and Q arbitrarily. The latter restricts them to be the same, but we can add arbitrary δ_O to $\mathcal{V}_Q$. While it finally depends on case by case, probably, the former is easier to find impossible differentials than the latter.

3.6 Algebraic and Integral Attacks

The security of $P \oplus Q$ against algebraic attacks does not seem much better than the most secure between P and Q against this family of cryptanalysis. In this section, we formulate the cipher as a polynomial on the key and input bits. More precisely, we interpret the cipher as a multivariate polynomial of the n input bits of x with coefficients that are functions of the key k,

$$\mathsf{F}(k,x) := \bigoplus_{u \in \mathbb{F}_2^n} f_u(k) x^u \ .$$

The degree of F is defined as the highest degree monomial with a non-zero coefficient, that is, $\deg(\mathsf{F}) := \max_u \{\mathrm{wt}(u) \mid f_u(k) \neq 0\}$, where $\mathrm{wt}(u)$ denotes the Hamming weight of u. Since the attacker usually exploits the weakest bit, or more generally component function, the *minimum degree* is more important than the degree: $\mathrm{minDeg}(\mathsf{F}) := \min_\beta \deg(\langle \beta, \mathsf{F}(k,x)\rangle)$. However, in terms of security, we rather look at non-constant coefficients only, as any monomial that is key-independent distinguishes the function from random. Therefore, we define $\widetilde{\deg}$ and $\widetilde{\mathrm{minDeg}}$ as follows:

$$\widetilde{\deg}(\mathsf{F}) := \max_u \{\mathrm{wt}(u) \mid f_u(k) \text{ is not constant}\},$$
$$\widetilde{\mathrm{minDeg}}(\mathsf{F}) := \min_\beta \widetilde{\deg}(\langle \beta, \mathsf{F}(k,x)\rangle).$$

Proposition 8. *Let P, Q be keyed permutations over $\mathbb{F}_2^n$ and $\mathsf{F} := P \oplus Q$, then:*

$$\widetilde{minDeg}(\mathsf{F}) = \min_\beta \max\{\widetilde{\deg}(\langle \beta, P\rangle), \widetilde{\deg}(\langle \beta, Q\rangle)\} \ .$$

Proof. Let k_P and k_Q in $\mathcal{K}_P$ and $\mathcal{K}_Q$, respectively, and let:

$$\langle \beta, P(k_P,x)\rangle := \bigoplus_{u \in \mathbb{F}_2^n} p_{\beta,u}(k_P) x^u \ , \qquad \langle \beta, Q(k_Q,x)\rangle := \bigoplus_{u \in \mathbb{F}_2^n} q_{\beta,u}(k_Q) x^u \ .$$

[3] If $\mathsf{F}(x)$ belongs to $\mathcal{U}$ with probability 1 for each $x \in \mathcal{V}$, then $\mathsf{F}(x) \in \mathcal{U}^c$ with probability 0, where $\cdot^c$ is the complimentary subspace.

Given $k := k_P \| k_Q \in \mathcal{K}_P \times \mathcal{K}_Q$, summing the polynomials for P and Q:

$$\langle \beta, \mathsf{F}(k,x) \rangle = \bigoplus_{u \in \mathbb{F}_2^n} f_{\beta,u}(k) x^u = \bigoplus_{u \in \mathbb{F}_2^n} \left(p_{\beta,u}(k_P) + q_{\beta,u}(k_Q) \right) x^u \,.$$

So we have $f_{\beta,u} = p_{\beta,u} + q_{\beta,u}$ defined on inputs $k \in K_P \times \mathcal{K}_Q$. Note that $f_{\beta,u}$ is constant if and only if $p_{\beta,u}$ **and** $q_{\beta,u}$ are constant. Therefore, we conclude by:

$$\begin{aligned} &\widetilde{\deg}(\langle \beta, \mathsf{F} \rangle) = \max_u \{ \mathrm{wt}(u) : p_{\beta,u} \text{ is not constant } \textbf{or } q_{\beta,u} \text{ is not constant} \} \\ &= \max \{ \max_u \{ \mathrm{wt}(u) : p_{\beta,u} \text{ is not constant} \}, \max_u \{ \mathrm{wt}(u) : q_{\beta,u} \text{ is not constant} \} \} \\ &= \max \{ \widetilde{\deg}(\langle \beta, P \rangle), \widetilde{\deg}(\langle \beta, Q \rangle) \} \,. \end{aligned}$$

□

To show that a cipher is secure against algebraic attacks often involves arguing that the cipher reaches a high degree. Proposition 8 shows that $P \oplus Q$ can only reach a high degree if either P or Q reaches it. Thus, integral attacks could be one of the most powerful attacks on $P \oplus Q$. Indeed, if a cipher has a degree d then the cipher is vulnerable to an integral attack for any linear subspace with dimension $d+1$. In particular, if P has degree d greater than Q, then any dimension $d+1$ linear subspace will allow an integral attack on both P and Q simultaneously, so on $P \oplus Q$ as well.

A similar statement holds for the stronger arguments against integral attacks as given in [HLLT21]. Again, to argue for full resistance against integral cryptanalysis either P or Q already has to be fully resistant.

On Key Recovery in Integral Attacks. On the other hand, we cannot expect a strong integral key-recovery attack. Usually, the integral key-recovery attack focuses on the ciphertext side, but it is impossible in $P \oplus Q$. In [FKL$^+$00], Ferguson et al. added one-round key recovery to the plaintext side, but it requires almost the full code book even for one-branch analysis. Besides, we must control the input of both branches in $P \oplus Q$. As discussed above, such a key recovery is difficult because the inputs of both branches are unlikely to take sets satisfying higher-order differences simultaneously after applying each key-recovery round from the common plaintext set.

The cube attack [DS09] is another possible key-recovery strategy. It is possible only when $f_{\beta,u}(k)$ is a very sparse polynomial. A common block cipher, where subkey is XORed every round, tends to have complicated polynomials, and the feature is used to guarantee the lower bound of the degree or the integral resistance property in [HLLT20, HLLT21]. Therefore, the cube attack is unlikely in such ciphers unless $\widetilde{\mathrm{minDeg}}(\mathsf{F})$ is insufficient.

Zero-Correlation Linear. Instead of considering the zero-correlation linear [BR14] explicitly, we first consider the link between the zero-correlation and integral [BLNW12,SLR+15]. When we have zero-correlation linear on F, we also have an integral distinguisher on F. Therefore, if F is secure enough against the integral, it should also be secure against the zero-correlation linear.

It is also possible to find the zero-correlation linear directly. However, because of the analogous argument of the impossible differential, we do not suppose that the sum is weaker than the composition against the zero-correlation linear.

3.7 Second-Order Differential Cryptanalysis

We look at attacks exploiting independent differential properties of P and Q. Interestingly, this distinguisher on $P \oplus Q$ is linked to the Boomerang distinguisher [Wag99] on $Q^{-1} \circ P$, as depicted in Figure 3.

Assuming we have two independent differential transitions that are $\mathrm{Prob}(\delta_P \xrightarrow{P} \delta'_P) = p$ and $\mathrm{Prob}(\delta_Q \xrightarrow{Q} \delta'_Q) = q$, then for some x:

$$\begin{cases} P(x) \oplus P(x \oplus \delta_P) = \delta'_P, \quad P(x \oplus \delta_Q) \oplus P(x \oplus \delta_Q \oplus \delta_P) = \delta'_P, \\ Q(x) \oplus Q(x \oplus \delta_Q) = \delta'_Q, \quad Q(x \oplus \delta_P) \oplus Q(x \oplus \delta_P \oplus \delta_Q) = \delta'_Q \end{cases}$$
$$\implies \mathsf{F}(x) \oplus \mathsf{F}(x \oplus \delta_P) \oplus \mathsf{F}(x \oplus \delta_Q) \oplus \mathsf{F}(x \oplus \delta_Q \oplus \delta_P) = 0\,.$$

With the usual independent assumptions, this happens with probability $(p \cdot q)^2$ for a random x when $\mathsf{F} = P \oplus Q$. Therefore, such a second-order differential requires about $4(p \cdot q)^{-2}$ queries to F.

We review the same differential transitions on $\mathsf{S} = Q^{-1} \circ P$ and perform the following boomerang attack. For some x,

$$\begin{cases} P(x) \oplus P(x \oplus \delta_P) = \delta'_P, \\ P(\mathsf{S}^{-1}(\mathsf{S}(x) \oplus \delta_Q)) \oplus P(\mathsf{S}^{-1}(\mathsf{S}(x) \oplus \delta_Q) \oplus \delta_P) = \delta'_P, \\ Q(\mathsf{S}(x)) \oplus Q(\mathsf{S}(x) \oplus \delta_Q) = \delta'_Q, \\ Q(\mathsf{S}(x \oplus \delta_P)) \oplus Q(\mathsf{S}(x \oplus \delta_P) \oplus \delta_Q) = \delta'_Q, \end{cases}$$
$$\implies \mathsf{S}^{-1}(\mathsf{S}(x) \oplus \delta_Q) \oplus \mathsf{S}^{-1}(\mathsf{S}(x \oplus \delta_P) \oplus \delta_Q) = \delta_P\,.$$

This well-known Boomerang holds with a probability of $(p \cdot q)^2$ with some independent assumptions. It requires about $4(p \cdot p^\star)^{-2}$ queries to S and S^{-1}.

Note that the relationship above ignores some independent issues when switching differential trails. For example, although $\delta_P = \delta_Q$ is a meaningful parameter for the Boomerang distinguisher on $Q^{-1} \circ P$, it is meaningless on $P \oplus Q$. Due to different independent issues, the resulting Boomerang probability on S and the 2nd order differential probability on $P \oplus Q$ differ. On the other hand, when p and q are reasonably high, that is a natural setting in real cryptanalysis, we would observe a similar feature in both cases.

On Key Recovery in 2nd-Order Differential Attacks. When considering key recovery, we observe a similar difficulty to that of differential key recovery.

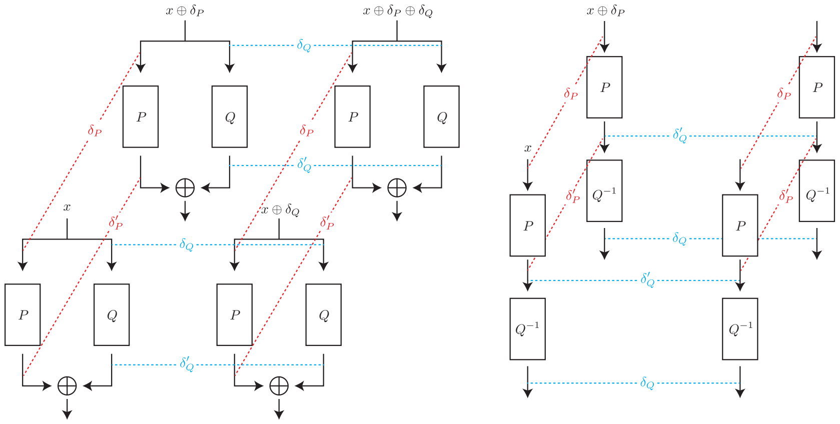


Fig. 3: 2nd-order differential on $P \oplus Q$ (left) and Boomerang on $P^{-1} \circ Q$ (right).

Let $P = P_2 \circ P_1$ and $Q = Q_2 \circ Q_1$. Assuming that there is a non-negligible 2nd-order differential distinguisher on $P_2 \oplus Q_2$. We apply the key recovery to P_1 and Q_1. Let (x_1, y_1), (x_2, y_2), (x_3, y_3), and (x_4, y_4) be the input of (P_2, Q_2). Then, a quartet satisfying $x_1 \oplus x_2 = x_3 \oplus x_4 = \delta_P$ and $y_1 \oplus y_3 = y_2 \oplus y_4 = \delta_Q$ is constructed by $y_1 = Q_1 \circ P_1^{-1}(x_1)$, $x_2 = x_1 \oplus \delta_P$, $y_2 = Q_1 \circ P_1^{-1}(x_2)$, $y_3 = y_1 \oplus \delta_Q$, $x_3 = P_1 \circ Q_1^{-1}(y_3)$, $x_4 = x_3 \oplus \delta_P$, and

$$y_4 = Q_1 \circ P_1^{-1}(x_4) = y_2 \oplus \delta_Q \,.$$

In general, $Q_1 \circ P_1^{-1}(x_4) = y_2 \oplus \delta_Q$ does not hold with a probability of 1.

3.8 Meet-in-the-middle Attacks

The meet-in-the-middle (MitM) attack [DH77] is another of the typical cryptanalysis of keyed symmetric primitives. In a traditional meet-in-the-middle attack, the adversary obtains a plaintext-ciphertext pair, and aims to extract the key faster than through an exhaustive search. The attacker guesses part of the key on the plaintext side and part of the key on the ciphertext side, and constructs two tables: one consists of all possible partial encryptions of the plaintext and the other of all possible partial decryptions of the ciphertext. When a collision between both tables is found, a candidate for both key guesses is obtained.

When applying this approach to the $P \oplus Q$ construction, we note that no information about the outputs of both branches can be obtained directly from the ciphertext. Thus, any MitM attack would require guessing part of one of the branches. However, by xoring the known ciphertext, this is equivalent to guessing part of an internal state of $Q^{-1} \circ P$, which is an ineffective guessing strategy in a MitM attack.

The DS-MitM attack [DS08] is an extension of the Meet-in-the-Middle attack and consists of the distinguisher and key recovery. When the distinguisher covers the initial few rounds in both branches, the key recovery requires the inverse query but there is no such query in the PRF. When the distinguisher covers the last few rounds in both branches, it involves the output of the PRF. Therefore, the parameter size of the distinguisher significantly increases. Consequently, using the distinguisher in either the inside of P or that of Q is promising, but then, such an attack is very similar to the attack against the composition, $Q^{-1} \circ P$ too.

3.9 Summary and Other Attacks

In this section, we analyzed differential, linear, differential-linear, differential-and-linear key recovery, (impossible) truncated differential, algebraic and integral, zero-correlation linear, the 2nd-order differential, and the MitM attacks. Some of them are strongly linked to the cryptanalysis against the composition.

When we mount a key recovery, where we need to control differences in two branches simultaneously, it is more difficult than the corresponding analysis against the composition. Notably, linear key recovery and differential-and-linear key recovery are promising attack strategies against the sum structure because they are friendly to key recovery, but they are strongly linked to linear key recovery and differential-linear key recovery against the composition.

Other well-known attacks exist. For example, Boomerang [Wag99] or Yo-Yo [BBD+98] attacks require adaptive chosen-plaintext-ciphertext attacks. However, the sum structure does not provide the decryption query, so applying these attacks is non-trivial. Note that an amplified Boomerang [KKS00] and Rectangle [BDK01] attacks are a chosen-plaintext variant of the Boomerang attack. However, it contains a probability of 2^{-2n} because the intermediate state size is $2n$ bits. Thus, it is unlikely that those attacks are applicable.

4 The ZIP Structure: Designing PRF in Light Work

Respecting the discussions in Sect. 3, we introduce the *ZIP structure*, which is defined as follows:

Definition 2 (ZIP structure). *Let $E = E_1 \circ E_0$ be a secure iterative block cipher. We define the ZIP construction of E as the following family of functions $E_0 \oplus E_1^{-1} : \mathbb{F}_2^n \to \mathbb{F}_2^n$. We suppose E_0 and E_1 contain almost the same rounds.*

The ZIP structure has three advantages:

- We can inherit many cryptanalysis results against E.
- Since the resulting primitive is a pseudo-random function, it derives beyond-birthday security in some modes of operation.
- On performance, the latency is about half of the original block cipher.

Of course, the discussion in Sect. 3 never shows that the ZIP structure has the same security as the original block cipher against all attack strategies. In particular, algebraic (integral), differential-linear, and truncated differential have to be carefully analyzed, but it is not as hard work as designing it from scratch.

In a practical application, the ZIP structure can achieve beyond-birthday security in some modes of operation while keeping the throughput in the case we use the original block cipher. It is useful in a wide situation. Moreover, its half latency is promising in several practical applications such as memory encryption or communication over the 5G and the beyond 5G as discussed in [ABC+24].

In this section, we focus on the ZIP-AES as an example.

4.1 ZIP-AES: A Concrete Instantiation via AES-128

AES-128. The Advanced Encryption Standard [DR20] is a SPN scheme designed by Daemen and Rijmen, and based on the Wide-Trail design strategy [DR01, DR02]. Focusing on AES-128, the key size is of 128 bits, and the number of rounds is 10. Each AES-round $R_{\text{AES}} : \mathbb{F}_{2^8}^{4\times 4} \to \mathbb{F}_{2^8}^{4\times 4}$ applies three operations besides the key-additon to the state x, that is, $x \mapsto R_{\text{AES}}(x) := MC \circ SR \circ SB(x)$. An additional AddRoundKey operation is applied at the input of the first round, and the last MixColumns operation is omitted (we denote a round without MC as $\hat{R}_{\text{AES}}$). We refer to [DR20] for the details of the key-schedule.

The ZIP-AES PRF. We define the ZIP-AES as

$$\forall x \in \mathbb{F}_{2^8}^{4\times 4} : \qquad \text{ZIP-AES}_5(x) := \text{AES}_5(x) \oplus \text{AES}_5^{-1}(x)\,,$$

where AES_5 denotes 5 encryption rounds of AES-128

$$\text{AES}_5(\cdot) = AK \circ \underbrace{MC \circ SR \circ SB}_{R_{\text{AES}}} \circ \cdots \circ AK \circ \underbrace{MC \circ SR \circ SB}_{R_{\text{AES}}} \circ AK(\cdot)$$

including the final MC in the last round as well, and where AES_5^{-1} denotes 5 decryption rounds of AES-128

$$\text{AES}_5^{-1}(\cdot) = AK^{-1} \circ \underbrace{(MC \circ SR \circ SB)^{-1}}_{R_{\text{AES}}^{-1}} \circ AK^{-1} \circ \ldots \circ \underbrace{(MC \circ SR \circ SB)^{-1}}_{R_{\text{AES}}^{-1}} \circ AK^{-1}(\cdot)$$

where $(MC \circ SR \circ SB)^{-1}(\cdot) := SB^{-1} \circ SR^{-1} \circ MC^{-1}(\cdot)$, and including the initial MC^{-1} in the first round as well.

Regarding the sub-keys, let $k_0 = \mathtt{k}, k_1, k_2, \ldots, k_{10} \in \mathbb{F}_{2^8}^{4\times 4}$ be the sub-keys generated by the key-schedule of AES-128, where $\mathtt{k} \in \mathbb{F}_{2^8}^{4\times 4}$ is the whitening key.

- AES_5 is instantiated with $k_0, k_1, k_2, k_3, k_4, k_5$;
- AES_5^{-1} is instantiated with $k_6, k_7, k_8, k_9, k_{10}, 0^{128}$.

We claim that ZIP-AES is a 128-bit secure pseudo-random function.

Design Rationale and Modified Versions of ZIP-AES. Before going on, we briefly discuss some technical choices regarding ZIP-AES, with particular attention both at the MixColumns operation at the end of AES_5, and at the inverse MixColumns operation at the beginning of AES_5^{-1}. As we pointed out, the final MC operation is omitted in AES. However, we decided to keep it for ZIP-AES.

This choice is necessary considering our motivation: ZIP-AES shares many cryptanalysis results to the original AES. As mentioned in Sect. 3, $P \oplus Q$ and $Q \circ P^{-1}$ shares the same differential characteristic, and $P \oplus Q$ and $Q^{-1} \circ P$ shares the same linear trail. If there is no inverse MixColumns in the beginning of AES_5^{-1}, the inverse MixColumns is missing between Q and P^{-1} in $Q \circ P^{-1}$. Similarly, if there is no MixColumns in the last of AES_5, the MixColumns is missing between Q^{-1} and P in $Q^{-1} \circ P$. In other words, such a construction corresponds to the variant of AES, where the MixColumns is omitted in the 5th round, which is clearly more insecure than the AES.

In practice, in order to prove this fact, in App. C, we consider these variants of ZIP-AES, in which the final MC operation for AES and/or the initial MC^{-1} operation for AES^{-1} are omitted. In there, we show that these modified versions are (much) weaker against attacks such as truncated differentials and mixture differentials with respect to the ZIP-AES defined here.

4.2 Security Analysis of ZIP-AES

In this section, we present our security analysis of ZIP-AES. Our results show that the strongest attack against it is the integral attack, which can distinguish up to $4+4$ rounds (namely, $\text{ZIP-AES}_{4,4}$) from a PRF. All other attacks (including classical linear and differential attacks, truncated differentials, mixture differentials, and so on) can only cover a smaller number of rounds. Moreover, in App. B.6, we also show that the attacks against $\text{AES-PRF}_{1,r}$ and $\text{AES-PRF}_{2,r}$ for any $r \geq 1$ proposed in [MN17b] work against $\text{ZIP-AES}_{1,r}$ and $\text{ZIP-AES}_{2,r}$ as well.

Unbalanced Variants. For the follow-up, we introduce "reduced-round variants" of ZIP-AES defined as $\text{ZIP-AES}_{r_0,r_1}(x) := \text{AES}_{r_0}(x) \oplus \text{AES}_{r_1}^{-1}(x)$. We encourage to analyze its security with particular attention to the case $r_0 = r_1 \geq 2$, in order to better evaluate ZIP-AES's resistance against attacks.

Differential and Linear Attacks. In the case of differential cryptanalysis, we have seen in Prop. 1 that, given two independent keyed permutations P, Q, then for each differential characteristic (trail) with probability p traversing $P \oplus Q$, there is a differential characteristic with the same probability p traversing $Q \circ P^{-1}$. Due to the wide-trail design strategy, it is well known that any differential characteristic over 4-round AES has a probability of at least 2^{-150}. This means that $\text{ZIP-AES}_{2,2}$ does *not* admit any differential characteristic with probability lower than 2^{-150}. Based on this, we claim that $\text{ZIP-AES}_{5,5}$ is secure against differential distinguishers and key-recovery attacks.

We have an analogous argument for linear cryptanalysis, differential-and-linear key recovery, and the 2nd order differential attacks.

Differential-Linear Attacks. The differential-linear distinguisher (autocorrelation) is estimated as the product of each branch's autocorrelation. In [HDE24a], the authors evaluated the autocorrelation of the AES. They are 1, $2^{-7.66}$, $2^{-31.66}$, and $2^{-55.66}$, for 2, 3, 4, and 5 rounds, respectively. Although there are no references in the AES inverse, we expect the autocorrelations to be similar, considering the well-aligned structure of the AES. Then, the autocorrelation of ZIP-AES$_{5,5}$ is expected as $2^{-55.66\times 2}$, which is unlikely to be observed with 2^{128}, full code-book, queries. In practice, the input difference and output mask must be the same in both branches. Such a restriction does not allow us to use the optimal autocorrelation for both branches simultaneously. We verified this observation by using ZIP-AES$_{3,3}$. When we used the 3-round differential-linear distinguisher shown in [HDE24a] in the left branch, we could not observe a significant autocorrelation in the right branch. Therefore, we expect that the autocorrelation is worse than the squared value of the best autocorrelation of each branch. In detail, see Appendix B.1.

Integral Attacks. Following [GRR16], we introduce the following subspaces of $\mathbb{F}_{2^8}^{4\times 4}$: the diagonal subspace $\mathcal{D}_i$, in which the i-th diagonal for $i \in \{0, 1, 2, 3\}$ is active and all the others are constant; the column subspace $\mathcal{C}_i := SR(\mathcal{D}_i)$, in which the i-th column for $i \in \{0, 1, 2, 3\}$ is active and all the others are constant; the anti-diagonal subspace $\mathcal{ID}_i := SR(\mathcal{C}_i)$, in which the i-th anti/inverse diagonal for $i \in \{0, 1, 2, 3\}$ is active and all the others are constant; the mixed subspace $\mathcal{M}_i := MC(\mathcal{ID}_i)$.

As it is well known [FKL$^+$00,KR07,Gil14], the following integral attacks hold

$$\bigoplus_{x\in\mathcal{D}_i\oplus\alpha} \text{AES}_4(x) = \bigoplus_{x\in\mathcal{M}_i\oplus\beta} \text{AES}_4^{-1}(x) = 0$$

for each $i \in \{0, 1, 2, 3\}$ and for any $\alpha, \beta \in \mathbb{F}_{2^8}^{4\times 4}$. It follows that for each $i, j \in \{0, 1, 2, 3\}$:

$$\bigoplus_{x\in(\mathcal{D}_i\oplus\mathcal{M}_j)\oplus\alpha} \text{ZIP-AES}_{4,4}(x) = 0$$

for each $\alpha \in \mathbb{F}_{2^8}^{4\times 4}$, where $\dim(\mathcal{D}_i \oplus \mathcal{M}_j) = 8$ – the dimension is considered at byte level. Therefore, we have the integral distinguisher by using 2^{64} chosen plaintexts.

Since no other integral distinguisher is known for 5 or more rounds of AES, and since appending a key recovery to the plaintext side is not easy (see Sect. 3 for more details), we claim that ZIP-AES$_{5,5}$ is secure against integral attacks.

Truncated Differential and Subspace Trail Attacks. With respect to the previous attacks and distinguishers, truncated differential requires a more ded-

Table 1: Practical tests on ZIP-AES over $\mathbb{F}_{2^8}^{4\times 4}$. In the table, we assume $|I| = |I'| = 3$ and $|J| = 2$ (P $\equiv$ Practical – Prob. $\equiv$ Probability).

# Rounds	Input Subspace	Output Subspace	ZIP-AES P-Prob.	PRF Prob.
$1+1$	$\mathcal{C}_i$	$\mathcal{D}_i \cap \mathcal{M}_i$	1	2^{-64}
$2+2$	$\mathcal{C}_i$	$\mathcal{C}_I$	$2^{-32} + 2^{-52.8}$	2^{-32}
$2+2$	$\mathcal{M}_J \cap \mathcal{D}_I$	$\mathcal{C}_{I'}$	$2^{-32} + 2^{-53.7}$	2^{-32}

icated analysis, since it is not possible to reduce the security of $\mathsf{F} := P \oplus Q$ to the one of any sequential construction (see Sect. 3.5 for more details).

We first re-call some results regarding the subspace trails presented in [GRR16]. Given $\mathcal{D}_I := \bigoplus_{i\in I} \mathcal{D}_i$, $\mathcal{C}_I := \bigoplus_{i\in I} \mathcal{C}_i$, $\mathcal{ID}_I := \bigoplus_{i\in I} \mathcal{ID}_i$, $\mathcal{M}_I := \bigoplus_{i\in I} \mathcal{M}_i$ for each $I \subseteq \{0,1,2,3\}$, we have that

- $\mathcal{D}_{i,i+2} = \mathcal{ID}_{i,i+2}$ for each $i \in \{0,1,2,3\}$,
- for each $I, J \subseteq \{0,1,2,3\}$: $\dim(\mathcal{C}_I \cap \mathcal{M}_J) = \dim(\mathcal{C}_I \cap \mathcal{D}_J) = |I| \cdot |J|$,
- for each $I, J \subseteq \{0,1,2,3\}$ with $|I| + |J| \leq 4$: $\mathcal{D}_I \cap \mathcal{M}_J = \mathcal{ID}_I \cap \mathcal{M}_J = \emptyset$,

where $|I|$ and $|J|$ represent the cardinality of I and J respectively.

Let $\text{AES}_r(\cdot)$ be r rounds of AES. For each $x \in \mathbb{F}_{2^8}^{4\times 4}$, and for each $I, J \subseteq \{0,1,2,3\}$, the following truncated differentials hold:

$$\begin{aligned}
\text{Prob}(\text{AES}_1(x) \oplus \text{AES}_1(x \oplus \delta) \in \mathcal{C}_I \mid \delta \in \mathcal{D}_I) &= 1\,,\\
\text{Prob}(\text{AES}_1(x) \oplus \text{AES}_1(x \oplus \delta) \in \mathcal{M}_I \mid \delta \in \mathcal{C}_I) &= 1\,,\\
\text{Prob}(\text{AES}_2(x) \oplus \text{AES}_2(x \oplus \delta) \in \mathcal{M}_I \mid \delta \in \mathcal{D}_I) &= 1\,,\\
\text{Prob}(\text{AES}_3(x) \oplus \text{AES}_3(x \oplus \delta) \in \mathcal{M}_J \mid \delta \in \mathcal{D}_I) &= 2^{8\cdot|I|\cdot(|J|-4)}\,.
\end{aligned}$$

We refer to [GR22, BR19] for truncated differentials up to 6-round AES.

Truncated Differentials for ZIP-AES$_{1,1}$. Since $\text{Prob}(\text{AES}_1(x) \oplus \text{AES}_1(x \oplus \delta) \in \mathcal{M}_i \mid \delta \in \mathcal{C}_i) = \text{Prob}(\text{AES}_1^{-1}(x) \oplus \text{AES}_1^{-1}(x \oplus \delta) \in \mathcal{D}_i \mid \delta \in \mathcal{C}_i) = 1$, the following truncated differentials on ZIP-AES$_{1,1}$ holds:

$$\text{Prob}(\text{ZIP-AES}_{1,1}(x) \oplus \text{ZIP-AES}_{1,1}(x \oplus \delta) \in \mathcal{D}_i \oplus \mathcal{M}_i \mid \delta \in \mathcal{C}_i) = 1\,.$$

For comparison, note that $\text{Prob}(\Pi(x) \oplus \Pi(x \oplus \delta) \in \mathcal{D}_i \oplus \mathcal{M}_i \mid \delta \in \mathcal{C}_i) = 2^{-64}$ for a PRF Π over $\mathbb{F}_{2^8}^{4\times 4}$.

Truncated Differentials for ZIP-AES$_{2,2}$: a Negative Result. Due to the existence of probability-1 truncated differentials for both 2-round AES and 2-round AES^{-1}, corresponding to $R^2(\mathcal{D}_I \oplus \alpha) = \mathcal{M}_I \oplus \beta$ and $R^{-2}(\mathcal{M}_J \oplus \alpha') = \mathcal{D}_J \oplus \beta'$, it could seem natural to combine them in order to set up a truncated differential for ZIP-AES$_{2,2}$, defined via an initial subspace $\mathcal{D}_I \cap \mathcal{M}_J$ and a final subspace $\mathcal{M}_I \oplus \mathcal{D}_J$. However, a problem arises, since

- $\mathcal{D}_I \cap \mathcal{M}_J$ contains only the zero-element for each I, J with $|I| + |J| \leq 4$, and

- $\mathcal{D}_J \oplus \mathcal{M}_I$ is the full space $\mathbb{F}_{2^8}^{4\times 4}$ for each I, J with $|I| + |J| \geq 4$,

due to the results listed before. For this reason, it seems impossible to set up a probability-1 truncated differential for ZIP-AES$_{2,2}$ via this strategy.

Truncated Differentials for ZIP-AES$_{r,r}$ with $r \geq 2$: Practical Results. At the same time, probabilistic truncated differential distinguishers for ZIP-AES$_{r,r}$ with $r \geq 2$ exist. Our practical results for ZIP-AES and for small-scale ZIP-AES (that is, AES over $\mathbb{F}_{2^4}^{4\times 4}$ as presented in [CMR05]) are summarized in Tables 1 and 4 in the appendix.[4] We refer to App. B.3 for more details about these practical results. As it is possible to observe, for all the considered cases, the probability that a truncated differential distinguisher holds for ZIP-AES$_{r,r}$ with $r \in \{2, 3\}$ is only slightly higher than the corresponding probability for a generic PRF.

Conclusion. Based on our practical tests, we conjecture that if a bias between the probability for ZIP-AES$_{r,r}$ for $r \geq 4$ and a generic PRF exists, it would be too small for being useful in practice. Together with the fact that extending a distinguisher that ends with $\mathcal{C}_I$ with $|I| \geq 2$ by 1 round is **not** possible, we claim that ZIP-AES$_{5,5}$ is secure against truncated differential distinguishers.

Mixture Differential Attacks (and More). A powerful attack on round-reduced AES is the mixture differential cryptanalysis [Gra18]. Given two plaintexts p_0, p_1 in the same column space $\mathcal{C}_I \oplus \gamma \subseteq \mathbb{F}_{2^8}^{4\times 4}$, let $p'_0, p'_1 \in \mathcal{C}_I \oplus \gamma$ be two new texts obtained by carefully swapping the *generating variables* of p_0, p_1. Independently of the values of the round-keys, the difference between p_0 and p_1 after 2-round AES is equal to the corresponding difference of p'_0 p'_1, that is,

$$\text{AES}_2(p_0) \oplus \text{AES}_2(p_1) = \text{AES}_2(p'_0) \oplus \text{AES}_2(p'_1)\,. \tag{1}$$

This is also known as the *integral mixture distinguisher* [GS20]. Moreover, p_0 and p_1 belong to the same coset of a mixed space $\mathcal{M}_J$ after 4-round AES if and only if p_0 and p_1 satisfy the same property, that is, $\forall J \subseteq \{0, 1, 2, 3\}$:

$$\text{AES}_4(p_0) \oplus \text{AES}_4(p_1) \in \mathcal{M}_J \quad \Longleftrightarrow \quad \text{AES}_4(p'_0) \oplus \text{AES}_4(p'_1) \in \mathcal{M}_J\,.$$

Similar distinguishers hold in the backward direction. (A variant of such distinguisher – the exchange attack [BR19] – is discussed in App. B.5).

(Deterministic) Mixture Integral Distinguishers for ZIP-AES$_{2,2}$: a Negative Result. At the current state, it does **not** seem possible to set up an integral mixture distinguisher for ZIP-AES$_{2,2}$, that is,

$$\text{ZIP-AES}_{2,2}(p_0) \oplus \text{ZIP-AES}_{2,2}(p_1) \neq \text{ZIP-AES}_{2,2}(p'_0) \oplus \text{ZIP-AES}_{2,2}(p'_1)$$

[4] Note that the truncated differentials are not affected by the details (as the degree) of the S-Box. Hence, we believe that the results on small-scale AES over $\mathbb{F}_{2^4}^{4\times 4}$ are a good representative of what happens for the "real" AES over $\mathbb{F}_{2^8}^{4\times 4}$.

Table 2: Performance comparison on the counter mode.

	cycle-per-byte						counter
	16B	32B	256B	2KB	16KB	128KB	
AES	3.56	1.84	0.51	0.36	0.34	0.34	integer
AES-PRF	3.63	1.94	0.55	0.39	0.37	0.37	integer
ZIP-AES	2.96	1.58	0.53	0.41	0.39	0.39	integer
AES	3.53	1.81	0.47	0.35	0.34	0.33	gray code
AES-PRF	3.57	1.88	0.51	0.36	0.34	0.34	gray code
ZIP-AES	2.90	1.61	0.47	0.34	0.33	0.33	gray code

in general, where $p_0, p_1, p'_0, p'_1 \in \mathcal{C}_I \oplus \gamma$ for $I \subseteq \{0,1,2,3\}$, and where p'_0 and p'_1 are constructed by carefully swapping the generating variables of p_0, p_1 in the same way described in [Gra18]. As discussed in details in App. B.4, the problem arises from the fact that generating variables of p_0, p_1 and the ones of $MC^{-1}(p_0), MC^{-1}(p_1)$ are different.

(Probabilistic) Mixture Differential Distinguishers for ZIP-AES$_{2,2}$. Having said that, it is possible to set up a *probabilistic* mixture differential distinguisher for ZIP-AES$_{2,2}$ by exploiting the following result.

Lemma 1. *Let $p_0, p_1 \in \mathcal{C}_i \oplus \alpha$. Let $p'_0, p'_1 \in \mathcal{C}_i \oplus \alpha$ be defined as the mixture couples generated by p_0 and p_1 such that Eq. (1) holds. For any $I \subseteq \{0,1,2,3\}$ with $|I| = 3$:*

$$\begin{aligned}\text{Prob}\big(&\textit{ZIP-AES}_{2,2}(p_0) \oplus \textit{ZIP-AES}_{2,2}(p_1) \\ &\oplus \textit{ZIP-AES}_{2,2}(p'_0) \oplus \textit{ZIP-AES}_{2,2}(p'_1) \in \mathcal{D}_I\big) \geq 2^{-16}\,.\end{aligned}$$

For comparison, $\text{Prob}\left(\Pi(p_0) \oplus \Pi(p_1) \oplus \Pi(p'_0) \oplus \Pi(p'_1) \in \mathcal{D}_I\right) = 2^{-32}$ *for a PRF Π over $\mathbb{F}_{2^8}^{4\times 4}$.*

See Appendix B.4 for the proof of Lemma 1.

At the current state, it does not seem possible to extend the previous distinguisher for more rounds of ZIP-AES. For this reason, we conjecture that ZIP-AES$_{5,5}$ is secure against such an attack.

4.3 Performance Evaluation

We implemented the counter mode of ZIP-AES to measure the performance. For the comparison, we also implemented the counter modes of AES-128 and AES-PRF-128 [MN17b]. All measurements were taken on a single core of Intel Core i7-1185G7 (Tiger Lake) with Turbo Boost and Hyperthreading disabled, and averaged over $100000 \times \frac{4096}{byte}$ repetitions, where *byte* denotes the processing data size in bytes. All subkeys are pre-computed, and the process is measured when the IV and plaintext are given in a byte array. The counter mode uses the 64-bit IV and 64-bit counter for the top and bottom halves of the input, respectively.

Table 2 (top 3 rows) summarizes the cycle-per-byte of each cipher for each size of processing message. As expected, ZIP-AES performs better than AES and AES-PRF for small data because the latency for one block processing is lower. On the other hand, when we encrypt more than 2KB, ZIP-AES performs worse than AES and AES-PRF. The reason is that `AESDEC` performs $AK^{-1} \circ MC^{-1} \circ SR^{-1} \circ SB^{-1}$ and is not the straightforward AES inverse round function. AES-NI consists of six instructions:

- `AESENC` performs $AK \circ MC \circ SR \circ SB$.
- `AESENCLAST` performs $AK \circ SR \circ SB$.
- `AESDEC` performs $AK^{-1} \circ MC^{-1} \circ SR^{-1} \circ SB^{-1}$.
- `AESDECLAST` performs $AK^{-1} \circ SR^{-1} \circ SB^{-1}$.
- `AESIMC` performs MC^{-1}. It is prepared to prepare subkeys for decryption.
- `AESKEYGENASSIST` assists to create round keys.

To perform AES_5^{-1}, we first use `AESIMC` and then use `AESDEC`. Unfortunately, `AESIMC` of the AES-NI is worse than the other main instructions. For example, on Tiger Lake CPU, the latency and throughput of the main four instructions are 3 and 0.5, respectively, but the latency and throughput of `AESIMC` are 6 and 1, respectively. The overhead by `AESIMC` is not negligible for long data.

To solve the overhead issue, we replace an integer counter with a gray code counter. In the gray code, the counting up is implemented by one XOR with a counter-dependent value. Notably, the counting up and MC^{-1} (and the whitening key XORing) is commutative. Given the IV, we first prepare the counter for AES_5 and prepare the counter for AES_5^{-1} by applying MC^{-1}. Then, we perform each counting up independently by one XOR. Then, we can avoid `AESIMC` for every block. Modern CPUs can perform XOR instructions in 3 ports, and the XOR instruction is executed with the AES instruction in parallel. Therefore, the overhead can be negligible. Table 2 (bottom 3 rows) summarizes each cycle-per-byte, where the counter is implemented by the gray code. We notice that the overhead of ZIP-AES for the long data can be resolved, and the performance is competitive with the case of AES and AES-PRF.

5 Future Work: Other ZIP Ciphers and Modes

In addition to ZIP-AES, one can consider several ZIP ciphers. Although we did not discuss it in this paper, we are interested in ZIP-AES-256; does it successfully derive the 256-bit secure PRF? Another interesting instance is the ZIP cipher using the 64-bit block cipher, e.g., ZIP-GIFT, instantiated by GIFT-64 [BPP+17].

GIFT-64 consists of 28 rounds. So, ZIP-GIFT consists of 14-round GIFT-64 and 14-round inverse GIFT-64. Unlike ZIP-AES, we do not provide a detailed analysis, and it is left as an open problem. As a reference, the following is a related analysis for GIFT-64. For the integral attack, in [HLLT21], the integral resistance property is guaranteed in 12-round GIFT-64, and the best integral distinguisher is up to 10 rounds. Therefore, ZIP-GIFT also guarantees integral resistance property. In [WLHL24], the autocorrelation is evaluated in GIFT-64, where the squared autocorrelation is $2^{-57.22}$ in 12 rounds. Therefore, the autocorrelation of ZIP-GIFT would be low enough.

Besides looking into more ZIP ciphers, it is promising to apply the general practical cryptanalysis to other structures. In particular, the feed-forward EDMD structure used in [MN17b, MN17a] to construct AES-PRF is a natural candidate to check which attack vectors link to AES and which do not. Another example is the generalization of the sum of two permutations, i.e., a sum of several permutations. There is already a concrete instance that has been designed, i.e., Gleeok [ABC+24] named after the multiple head dragon.

Finally, it is worth investigating if the new differential-and-linear attack that we introduced and liked to a differential-linear attack on the composition, is applicable to Orthros.

Acknowledgements. The authors would like to thank anonymous reviewers for their useful feedback in preparing the final version of this paper and Jan Vorloeper for performing the practical tests regarding the truncated differential on ZIP-AES.

This work has been funded in parts by the German Research foundation (DFG) within the framework of the Excellence Strategy of the Federal Government and the States – EXC 2092 CaSa – 39078197 and by the ERC project 101097056 (SYMTRUST).

References

ABC+24. Ravi Anand, Subhadeep Banik, Andrea Caforio, Tatsuya Ishikawa, Takanori Isobe, Fukang Liu, Kazuhiko Minematsu, Mostafizar Rahman, and Kosei Sakamoto. Gleeok: A family of low-latency prfs and its applications to authenticated encryption. *IACR TCHES*, 2024(2):545–587, 2024.

ABD+23. Roberto Avanzi, Subhadeep Banik, Orr Dunkelman, Maria Eichlseder, Shibam Ghosh, Marcel Nageler, and Francesco Regazzoni. The qarmav2 family of tweakable block ciphers. *IACR ToSC*, 2023(3):25–73, 2023.

Ava17. Roberto Avanzi. The QARMA block cipher family. almost MDS matrices over rings with zero divisors, nearly symmetric even-mansour constructions with non-involutory central rounds, and search heuristics for low-latency s-boxes. *IACR ToSC*, 2017(1):4–44, 2017.

BBD+98. Eli Biham, Alex Biryukov, Orr Dunkelman, Eran Richardson, and Adi Shamir. Initial observations on skipjack: Cryptanalysis of skipjack-3xor. In Stafford E. Tavares and Henk Meijer, editors, *SAC'98*, volume 1556 of *LNCS*, pages 362–376. Springer, 1998.

BBS99. Eli Biham, Alex Biryukov, and Adi Shamir. Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials. In Jacques Stern, editor, *EUROCRYPT '99*, volume 1592 of *LNCS*, pages 12–23. Springer, 1999.

BCG+12. Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof

Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. PRINCE - A low-latency block cipher for pervasive computing applications (full version). *IACR Cryptol. ePrint Arch.*, page 529, 2012.

BDK01. Eli Biham, Orr Dunkelman, and Nathan Keller. The rectangle attack - rectangling the serpent. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 340–357. Springer, 2001.

BEK$^+$20. Dusan Bozilov, Maria Eichlseder, Miroslav Knezevic, Baptiste Lambin, Gregor Leander, Thorben Moos, Ventzislav Nikov, Shahram Rasoolzadeh, Yosuke Todo, and Friedrich Wiemer. Princev2 - more security for (almost) no overhead. In Orr Dunkelman, Michael J. Jacobson Jr., and Colin O'Flynn, editors, *SAC 2020*, volume 12804 of *LNCS*, pages 483–511. Springer, 2020.

BI99. Mihir Bellare and Russell Impagliazzo. A tool for obtaining tighter security analyses of pseudorandom function based constructions, with applications to PRP to PRF conversion. *IACR Cryptol. ePrint Arch.*, page 24, 1999.

BIL$^+$21. Subhadeep Banik, Takanori Isobe, Fukang Liu, Kazuhiko Minematsu, and Kosei Sakamoto. Orthros: A low-latency PRF. *IACR ToSC*, 2021(1):37–77, 2021.

BJK$^+$16. Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 123–153. Springer, 2016.

BKR98. Mihir Bellare, Ted Krovetz, and Phillip Rogaway. Luby-rackoff backwards: Increasing security by making block ciphers non-invertible. In Kaisa Nyberg, editor, *EUROCRYPT '98*, volume 1403 of *LNCS*, pages 266–280. Springer, 1998.

BLNW12. Andrey Bogdanov, Gregor Leander, Kaisa Nyberg, and Meiqin Wang. Integral and multidimensional linear distinguishers with correlation zero. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 244–261. Springer, 2012.

BPP$^+$17. Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. GIFT: A Small Present - Towards Reaching the Limit of Lightweight Encryption. In Wieland Fischer and Naofumi Homma, editors, *CHES 2017*, volume 10529 of *LNCS*, pages 321–345. Springer, 2017.

BR14. Andrey Bogdanov and Vincent Rijmen. Linear hulls with correlation zero and linear cryptanalysis of block ciphers. *Des. Codes Cryptogr.*, 70(3):369–383, 2014.

BR19. Navid Ghaedi Bardeh and Sondre Rønjom. The Exchange Attack: How to Distinguish Six Rounds of AES with $2^{88.2}$ Chosen Plaintexts. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 347–370. Springer, 2019.

BS90. Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. In Alfred Menezes and Scott A. Vanstone, editors, *CRYPTO '90*, volume 537 of *LNCS*, pages 2–21. Springer, 1990.

CMR05. Carlos Cid, Sean Murphy, and Matthew J. B. Robshaw. Small Scale Variants of the AES. In Henri Gilbert and Helena Handschuh, editors, *FSE 2005*, volume 3557 of *LNCS*, pages 145–162. Springer, 2005.

DH77. Whitfield Diffie and Martin E. Hellman. Exhaustive cryptanalysis of the NBS data encryption standard. *Computer*, 10(6):74–84, 1977.

Din24. Itai Dinur. Tight indistinguishability bounds for the XOR of independent random permutations by fourier analysis. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part I*, volume 14651 of *LNCS*, pages 33–62. Springer, 2024.

DNS22. Avijit Dutta, Mridul Nandi, and Abishanka Saha. Proof of mirror theory for $\xi_{\max} = 2$. *IEEE Trans. Inf. Theory*, 68(9):6218–6232, 2022.

DR01. Joan Daemen and Vincent Rijmen. The Wide Trail Design Strategy. In Bahram Honary, editor, *8th IMA*, volume 2260 of *LNCS*, pages 222–238. Springer, 2001.

DR02. Joan Daemen and Vincent Rijmen. Security of a Wide Trail Design. In Alfred Menezes and Palash Sarkar, editors, *INDOCRYPT 2002*, volume 2551 of *LNCS*, pages 1–11. Springer, 2002.

DR20. Joan Daemen and Vincent Rijmen. *The Design of Rijndael - The Advanced Encryption Standard (AES), Second Edition.* Information Security and Cryptography. Springer, 2020.

DS08. Hüseyin Demirci and Ali Aydin Selçuk. A meet-in-the-middle attack on 8-round AES. In Kaisa Nyberg, editor, *FSE 2008*, volume 5086 of *LNCS*, pages 116–126. Springer, 2008.

DS09. Itai Dinur and Adi Shamir. Cube attacks on tweakable black box polynomials. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 278–299. Springer, 2009.

FKL$^+$00. Niels Ferguson, John Kelsey, Stefan Lucks, Bruce Schneier, Michael Stay, David A. Wagner, and Doug Whiting. Improved Cryptanalysis of Rijndael. In Bruce Schneier, editor, *FSE 2000*, volume 1978 of *LNCS*, pages 213–230. Springer, 2000.

GBJ$^+$23. Aldo Gunsing, Ritam Bhaumik, Ashwin Jha, Bart Mennink, and Yaobin Shen. Revisiting the indifferentiability of the sum of permutations. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part III*, volume 14083 of *LNCS*, pages 628–660. Springer, 2023.

Gil14. Henri Gilbert. A Simplified Representation of AES. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 200–222. Springer, 2014.

GR22. Lorenzo Grassi and Christian Rechberger. Truncated Differential Properties of the Diagonal Set of Inputs for 5-Round AES. In Khoa Nguyen, Guomin Yang, Fuchun Guo, and Willy Susilo, editors, *ACISP 2022*, volume 13494 of *LNCS*, pages 24–45. Springer, 2022.

Gra18. Lorenzo Grassi. Mixture Differential Cryptanalysis: a New Approach to Distinguishers and Attacks on round-reduced AES. *IACR ToSC*, 2018(2):133–160, 2018.

GRR16. Lorenzo Grassi, Christian Rechberger, and Sondre Rønjom. Subspace Trail Cryptanalysis and its Applications to AES. *IACR ToSC*, 2016(2):192–225, 2016.

GS20. Lorenzo Grassi and Markus Schofnegger. Mixture Integral Attacks on Reduced-Round AES with a Known/Secret S-Box. In Karthikeyan Bhargavan, Elisabeth Oswald, and Manoj Prabhakaran, editors, *INDOCRYPT 2020*, volume 12578 of *LNCS*, pages 312–331. Springer, 2020.

HDE24a. Hosein Hadipour, Patrick Derbez, and Maria Eichlseder. Revisiting differential-linear attacks via a boomerang perspective with application to aes, ascon, clefia, skinny, present, knot, twine, warp, lblock, simeck, and SERPENT. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part IV*, volume 14923 of *LNCS*, pages 38–72. Springer, 2024.

HDE24b. Hosein Hadipour, Patrick Derbez, and Maria Eichlseder. Revisiting differential-linear attacks via a boomerang perspective with application to aes, ascon, clefia, skinny, present, knot, twine, warp, lblock, simeck, and SERPENT. *IACR Cryptol. ePrint Arch.*, page 255, 2024.

HKR15. Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway. Robust authenticated-encryption AEZ and the problem that it solves. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 15–44. Springer, 2015.

HLLT20. Phil Hebborn, Baptiste Lambin, Gregor Leander, and Yosuke Todo. Lower bounds on the degree of block ciphers. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part I*, volume 12491 of *LNCS*, pages 537–566. Springer, 2020.

HLLT21. Phil Hebborn, Baptiste Lambin, Gregor Leander, and Yosuke Todo. Strong and tight security guarantees against integral distinguishers. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part I*, volume 13090 of *LNCS*, pages 362–391. Springer, 2021.

JN22. Ashwin Jha and Mridul Nandi. A survey on applications of h-technique: Revisiting security analysis of PRP and PRF. *Entropy*, 24(4):462, 2022.

KKS00. John Kelsey, Tadayoshi Kohno, and Bruce Schneier. Amplified boomerang attacks against reduced-round MARS and serpent. In Bruce Schneier, editor, *FSE 2000*, volume 1978 of *LNCS*, pages 75–93. Springer, 2000.

Knu94. Lars R. Knudsen. Truncated and higher order differentials. In Bart Preneel, editor, *FSE 2nd*, volume 1008 of *LNCS*, pages 196–211. Springer, 1994.

KR07. Lars R. Knudsen and Vincent Rijmen. Known-Key Distinguishers for Some Block Ciphers. In Kaoru Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 315–324. Springer, 2007.

LMMR21. Gregor Leander, Thorben Moos, Amir Moradi, and Shahram Rasoolzadeh. The SPEEDY family of block ciphers engineering an ultra low-latency cipher from gate level for secure processor architectures. *IACR TCHES*, 2021(4):510–545, 2021.

LSW22. Muzhou Li, Ling Sun, and Meiqin Wang. Automated key recovery attacks on round-reduced orthros. In Lejla Batina and Joan Daemen, editors, *AFRICACRYPT 2022*, volume 13503 of *LNCS*, pages 189–213. Springer Nature Switzerland, 2022.

Luc00. Stefan Lucks. The sum of prps is a secure PRF. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 470–484. Springer, 2000.

Mat93. Mitsuru Matsui. Linear Cryptanalysis Method for DES Cipher. In Tor Helleseth, editor, *EUROCRYPT '93*, volume 765 of *LNCS*, pages 386–397. Springer, 1993.

MN17a. Bart Mennink and Samuel Neves. Encrypted davies-meyer and its dual: Towards optimal security using mirror theory. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 556–583. Springer, 2017.

MN17b. Bart Mennink and Samuel Neves. Optimal PRFs from Blockcipher Designs. *IACR ToSC*, 2017(3):228–252, 2017.

Pat10. Jacques Patarin. Introduction to mirror theory: Analysis of systems of linear equalities and linear non equalities for cryptography. *IACR Cryptol. ePrint Arch.*, page 287, 2010.

Rø19. Sondre Rønjom. A Short Note on a Weight Probability Distribution Related to SPNs. *IACR Cryptol. ePrint Arc.*, page 750, 2019.

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 Rosario Gennaro and Matthew Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 95–115. Springer, 2015.

TISI23. Kazuma Taka, Tatsuya Ishikawa, Kosei Sakamoto, and Takanori Isobe. An efficient strategy to construct a better differential on multiple-branch-based designs: Application to orthros. In Mike Rosulek, editor, *CT-RSA 2023*, volume 13871 of *LNCS*, pages 277–304. Springer, 2023.

Wag99. David A. Wagner. The boomerang attack. In Lars R. Knudsen, editor, *FSE '99*, volume 1636 of *LNCS*, pages 156–170. Springer, 1999.

WLHL24. Shichang Wang, Meicheng Liu, Shiqi Hou, and Dongdai Lin. Differential-linear cryptanalysis of GIFT family and GIFT-based ciphers. *IACR Communications in Cryptology*, 1(1), 2024.

SUPPLEMENTARY MATERIAL

A Detail of the Review of Key Recovery Attacks in [LSW22]

Table 3: Summary of possible differential transition by $S(S^{-1}(x) \oplus k)$.

	$\texttt{0x2} \xrightarrow{S\circ S^{-1}} \texttt{0x8}$	$\texttt{0x2} \xrightarrow{S\circ S^{-1}} \texttt{0x1}$	$\texttt{0x8} \xrightarrow{S\circ S^{-1}} \texttt{0x2}$
$k = \texttt{0x0}$	-	-	-
$k = \texttt{0x1}$	(0x1, 0x3) → (0x0, 0x8)	-	(0x0, 0x8) → (0x1, 0x3)
$k = \texttt{0x2}$	-	-	-
$k = \texttt{0x3}$	-	-	-
$k = \texttt{0x4}$	(0x9, 0xB) → (0xF, 0x7)	-	(0xF, 0x7) → (0x9, 0xB)
$k = \texttt{0x5}$	(0x1, 0x3) → (0x8, 0x0)	-	(0x8, 0x0) → (0x1, 0x3)
	(0xD, 0xF) → (0x2, 0xA)	-	(0x2, 0xA) → (0xD, 0xF)
$k = \texttt{0x6}$	(0x9, 0xB) → (0x7, 0xF)	(0x8, 0xA) → (0x4, 0x5)	(0x7, 0xF) → (0x9, 0xB)
$k = \texttt{0x7}$	-	-	-
$k = \texttt{0x8}$	-	(0x0, 0x2) → (0xA, 0xB)	-
$k = \texttt{0x9}$	-	(0xC, 0xE) → (0x3, 0x2)	-
$k = \texttt{0xA}$	-	(0x8, 0xA) → (0x5, 0x4)	-
$k = \texttt{0xB}$	-	(0x0, 0x2) → (0xB, 0xA)	-
$k = \texttt{0xC}$	-	-	-
$k = \texttt{0xD}$	-	-	-
$k = \texttt{0xE}$	(0xD, 0xF) → (0xA, 0x2)	(0x5, 0x7) → (0x1, 0x0)	(0xA, 0x2) → (0xD, 0xF)
$k = \texttt{0xF}$	-	(0x5, 0x7) → (0x1, 0x0)	-
	-	(0xC, 0xE) → (0x2, 0x3)	-

Table 3 summarizes the input/output pairs satisfying the differential transition for each key. For example, when $k = \texttt{0x1}$, only two pairs

$$(\texttt{0x1}, \texttt{0x3}) \to (\texttt{0x0}, \texttt{0x8}) \qquad\qquad (\texttt{0x3}, \texttt{0x1}) \to (\texttt{0x8}, \texttt{0x0})$$

satisfy input differences 0x2 and 0x8 at the same time. When $k = \texttt{0x5}$,

$$\begin{aligned} &(\texttt{0x1}, \texttt{0x3}) \to (\texttt{0x8}, \texttt{0x0}), && (\texttt{0x3}, \texttt{0x1}) \to (\texttt{0x0}, \texttt{0x8}),\\ &(\texttt{0xD}, \texttt{0xF}) \to (\texttt{0x2}, \texttt{0xA}), && (\texttt{0xF}, \texttt{0xD}) \to (\texttt{0xA}, \texttt{0x2}), \end{aligned}$$

satisfy input differences 0x2 and 0x8 at the same time.

Note that the key k is the corresponding 4 bits of $RK_0^1 \oplus RK_0^2$, where RK_0^1 and RK_0^2 denote the first round key for the 1st and 2nd branches, respectively. As mentioned in [LSW22], the key-recovery attack involves 12 bits of RK_0^1 and RK_0^2, which are obtained from the 22 bits of the master key. Involved $RK_0^1 \oplus RK_0^2$ is represented by using the master key as follows:

$$\begin{aligned} (RK_0^1 \oplus RK_0^2)[56, 57, 58, 59] &= (K_{85}, K_{82}, K_{37}, K_{69}) \oplus (K_{107}, K_{22}, K_{85}, K_{113})\\ (RK_0^1 \oplus RK_0^2)[104, 105, 106, 107] &= (K_{52}, K_{96}, K_{43}, K_{61}) \oplus (K_{46}, K_{30}, K_{102}, K_{59})\\ (RK_0^1 \oplus RK_0^2)[120, 121, 122, 123] &= (K_{98}, K_{46}, K_{23}, K_{32}) \oplus (K_{110}, K_{65}, K_{100}, K_{73}) \end{aligned}$$

For the whole of the secret key, these 12-bit values take any value with uniform probability. The key-recovery attack works only when three active S-boxes use a key that allows differential transitions. Therefore, the fraction of weak keys is $5/16 \times 7/16 \times 5/16 \approx 2^{-4.55}$.

Let us consider keys 0x1, 0x6, and 0x1 are used for $\texttt{0x2} \xrightarrow{S \circ S^{-1}} \texttt{0x8}$, $\texttt{0x2} \xrightarrow{S \circ S^{-1}} \texttt{0x1}$, and $\texttt{0x8} \xrightarrow{S \circ S^{-1}} \texttt{0x2}$, respectively. Then, each active S-box contains only two pairs. Therefore, among about 2^{24} pairs constructed by activating 12-bit inputs, the number of pairs satisfying the input differences of both branches is only 8. In other words, assuming such keys are used, we have only 8 pairs from the 12-bit active. Note that the attacker must activate the 12-bit value to have these 8 pairs because the attacker does not know which RK_0^1 is used. Therefore, to observe the differential distinguisher with a probability of 2^{-112}, we need at least $2^{12} \times 2^{112}/8 = 2^{121}$ chosen plaintexts.

On Time Complexity for Differential Key-Recovery Attack. Let us consider the following attack procedure. We first fix inactive $128 - 12 = 116$ bits and query 2^{12} chosen inputs. Then, we pick pairs satisfying the output difference. After repeating $c \times 2^{110}$ times, where c is a small constant, we have about $c \times 2^{110} \times 8 \times 2^{-128} \approx c \times 2^4$ pairs. We analyze these pairs by guessing (weak) keys, where the cost is negligible. We have c pairs satisfying input differences simultaneously when the guess is correct. Otherwise, the number is a few.

The procedure above works on the time in about $c \times 2^{122}$, equivalent to the data complexity. Note that since it is a weak-key attack, the exhaustive search is $2^{128-4.55}$. Then, $c \leq 2^{2.45}$. It is a tough condition, but we can expect a non-negligible advantage.

On Time Complexity for Differential-Linear Key-Recovery Attack. We next discuss a considerable attack procedure for the key recovery of the differential-linear attack.

We first prepare about 2^{24} pairs, where the 3 active nibbles have non-zero differences. For each pair, we activate other $128 - 12 = 116$ bits and compute an empirical correlation by using $c \times 2^{96-3} = c \times 2^{93}$ pairs. The complexity is $2^{24} \times c \times 2^{93} = c \times 2^{117}$, and we have about 2^{24} empirical correlations depending on the pair. We finally guess the key, pick pairs that satisfy the input difference of the differential-linear distinguisher, and combine them. When the correct key is guessed, it computes the empirical correlation, where at least $c \times 2^{93} \times 8 = c \times 2^{96}$ pairs are used. Again, it is a weak-key attack, and the exhaustive search is $2^{128-4.55}$. Then $c \leq 2^{6.45}$. We expect that this c is enough to achieve an advantage in recovering the secret key.

B Distinguishers and Attacks on round-reduced ZIP-AES: Details

B.1 Autocorrelation

We used the 3-round differential-linear distinguisher shown in [HDE24b]. The input difference and output mask are

$$\delta = \texttt{0x00000000B4000000000000000000000000},$$
$$\alpha = \texttt{0x00000000000000009866AB3200000000},$$

respectively. Note that this hexadecimal representation differs from [HDE24b], but they are the same value. We experimentally evaluated the autocorrelation for the left branch by using 2^{30} pairs. As a result, it was $2^{-7.66}$, which is the same as the estimation in [HDE24b]. Next, we observed the autocorrelation for the right branch by using 2^{30} Paris. Then, it was $2^{-16.077}$, which was not significant. Thus, we expect that the autocorrelation for the right branch is much worse, and the autocorrelation of ZIP-AES3,3 is significantly worse than $2^{-7.66\times 2}$.

B.2 Differential-and-Linear Key Recovery Attack

To verify the independent assumption and the validity of our key recovery attack, we implemented reduced-round ZIP-AES and mounted the attack. We used ZIP-AES3,3, where we used a 2-round differential-linear distinguisher for the left branch and a 1-round linear distinguisher for the right branch. In detail, the input difference of the left branch δ, the input mask of the right branch α, and the output mask β are

$$\delta = \texttt{0x00000000B40000000000000000000000},$$
$$\alpha = \texttt{0x00000000000000000000007200000000},$$
$$\beta = \texttt{0x0000000000000000000000CC00000000},$$

Table 4: Practical tests on small-scale ZIP-AES over $\mathbb{F}_{2^4}^{4\times 4}$. In the table, we assume $|I| = |I'| = 3$ and $|J| = 2$ (P $\equiv$ Practical – Prob. $\equiv$ Probability).

# Rounds	Input Subspace	Output Subspace	ZIP-AES P-Prob.	PRF Prob.
$1+1$	$\mathcal{C}_i$	$\mathcal{D}_i \cap \mathcal{M}_i$	1	2^{-32}
$2+2$	$\mathcal{M}_J \cap \mathcal{D}_I$	$\mathcal{C}_{I'}$	$2^{-16} + 2^{-19.7}$	2^{-16}
$2+2$	$\mathcal{C}_i$	$\mathcal{C}_I$	$2^{-16} + 2^{-24.5}$	2^{-16}
$2+2$	$\mathcal{M}_I \cap \mathcal{D}_I$	$\mathcal{C}_{I'}$	$2^{-16} + 2^{-31}$	2^{-16}
$2+2$	$\mathcal{C}_J$	$\mathcal{C}_I$	$2^{-16} + 2^{-39}$	2^{-16}
$3+3$	$\mathcal{C}_i$	$\mathcal{C}_I$	$2^{-16} + 2^{-33}$	2^{-16}

respectively. Note that α is the linear mask after applying the inverse MixColumns. Therefore, there is only SubBytes (and ShiftRows) from α to β. We experimentally evaluated the autocorrelation from δ to β by the 2-round left branch and it was about $2^{-6.33}$. The correlation from α to β is 2^{-3}. Therefore, in total, the expected correlation is $2^{-12.33}$.

We experimentally evaluated the correlation when the right keys k_0, k_6, and k_7 are guessed. We use 2^{30} pairs and repeat it by 10 keys. As a result, the correlation was $2^{-12.41}$. We also experimentally evaluated the correlation when the wrong keys are guessed. Then, the correlation was $2^{-17.61}$, i.e., we do not observe significant correlation. Thus, this experiment justified our estimation.

B.3 Truncated Differential Distinguishers

Here, we provide more details regarding our practical results presented in the Tables 1 and 4.

Before going on, we limit ourselves to mention that the subspace in output is always fixed (e.g., $\mathcal{C}_I$ for a fixed I). By allowing for any I with a fixed cardinality $|I|$, the probabilities are increased by a factor $\binom{4}{|I|}$. This could play a crucial role in reducing the overall data and/or computational complexity of the distinguishers/attacks. Moreover, we emphasize that we are not able to detect the bias for truncated differentials that ends with $\mathcal{C}_i$, that is, zero-difference in three columns (note that the probability of such event is 2^{-96}, and the bias – if it exists – would be even smaller).

ZIP-AES$_{2,2}$: $\mathcal{C}_i \to \mathcal{C}_J$ with $|J| = 3$. Let's start by considering the case of plaintexts in $\mathcal{C}_i$. Due to the impossible differentials given before, for $|J| = 3$, we know that (i) $\text{Prob}(\text{AES}_2(x) \oplus \text{AES}_2(x \oplus \delta) \in \mathcal{C}_J \mid \delta \in \mathcal{C}_i) = 0$, and (ii) $\text{Prob}(\text{AES}_2^{-1}(x) \oplus \text{AES}_2^{-1}(x \oplus \delta) \in \mathcal{C}_J \mid \delta \in \mathcal{C}_i) = 0$. Given $\text{AES}_2(x) \oplus \text{AES}_2(x \oplus \delta) \notin \mathcal{C}_J$ and $\text{AES}_2^{-1}(x) \oplus \text{AES}_2^{-1}(x \oplus \delta) \notin \mathcal{C}_J$, our practical tests on AES show that $\text{ZIP-AES}_{2,2}(x) \oplus \text{ZIP-AES}_{2,2}(x \oplus \delta)$ belongs into $\mathcal{C}_J$ with probability slightly higher than 2^{32} (similar results hold on small-scale AES). We leave the *open problem* to explain this fact for future work.

ZIP-AES$_{3,3}$: $\mathcal{C}_i \to \mathcal{C}_J$ with $|J| = 3$. Next, we examine the previous truncated differential but for ZIP-AES$_{3,3}$. Due to the results proposed in [GR22], we know that

$$\text{Prob}(\text{AES}_3(x) \oplus \text{AES}_3(x \oplus \delta) \in \mathcal{C}_J \mid \delta \in \mathcal{C}_i) = 2^{-32} + 2^{-53}\,,$$
$$\text{Prob}(\text{AES}_3^{-1}(x) \oplus \text{AES}_3^{-1}(x \oplus \delta) \in \mathcal{C}_J \mid \delta \in \mathcal{C}_i) = 2^{-32} + 2^{-53}\,,$$

(respectively, approximately $2^{-16} + 2^{-24.7}$ for the case of small-scale AES). However, the biases of these two cases are too small for theoretically deriving some useful information about Prob(ZIP-AES$_{3,3}(x) \oplus$ZIP-AES$_{3,3}(x \oplus \delta) \in \mathcal{C}_I \mid \delta \in \mathcal{C}_i$). Our practical tests on small-scale AES show that this event occurs with probability slightly higher than 2^{-16}. We expect the same happening for real AES – we leave the *open problem* to explain this fact for future work.

ZIP-AES$_{2,2}$: $\mathcal{M}_I \cap \mathcal{D}_{I'} \to \mathcal{C}_J$ with $|I| + |I'| \geq 5$ and $|J| = 3$. Finally, let's consider the case in which the plaintexts are in $\mathcal{M}_I \cap \mathcal{D}_{I'}$ for $I = I'$ with $|I| = 3$. As we have seen before, since $|I| = |J| = 3$, we have that Prob(AES$_2(x) \oplus$AES$_2(x \oplus \delta) \in \mathcal{C}_J \mid \delta \in \mathcal{D}_I) = 2^{-24}$, and Prob(AES$_2^{-1}(x) \oplus$ AES$_2^{-1}(x \oplus \delta) \in \mathcal{C}_J \mid \delta \in \mathcal{M}_I) = 2^{-24}$, where $\mathcal{M}_I \cap \mathcal{D}_I \subseteq \mathcal{D}_I$ and $\mathcal{M}_I \cap \mathcal{D}_I \subseteq \mathcal{M}_I$. Moreover, if AES$_2(x) \oplus$AES$_2(x \oplus \delta) \notin \mathcal{C}_J$ (which occurs with probability $1 - 2^{-24}$) and if AES$_2^{-1}(x) \oplus$ AES$_2^{-1}(x \oplus \delta) \notin \mathcal{C}_J$ (which again occurs with probability $1 - 2^{-24}$),[5] then ZIP-AES$_{2,2}(x) \oplus$ ZIP-AES$_{2,2}(x \oplus \delta)$ belongs into $\mathcal{C}_J$ with probability approximately of $2^{-32} \pm \varepsilon$ for a very small $0 \leq \varepsilon \ll 1$. As a result, if $|\pm\varepsilon| <$ Prob(AES$_2(x) \oplus$AES$_2(x \oplus \delta) \in \mathcal{C}_J \mid \delta \in \mathcal{M}_I \cap \mathcal{D}_I) \cdot$ Prob(AES$_2^{-1}(x) \oplus$ AES$_2^{-1}(x \oplus \delta) \in \mathcal{C}_J \mid \delta \in \mathcal{M}_I \cap \mathcal{D}_I$), we expect that ZIP-AES$_{2,2}(x) \oplus$ZIP-AES$_{2,2}(x \oplus \delta)$ belongs into $\mathcal{C}_J$ with probability slightly higher than 2^{-32} (assuming $\delta \in \mathcal{M}_I \cap \mathcal{D}_I$). Our practical tests on small-scale AES confirm such result. We leave the *open problem* to explain this fact for future work. We also emphasize that an analogous conclusion holds also for $\mathcal{M}_I \cap \mathcal{D}_{I'} \to \mathcal{C}_J$ with (i) $|I| = |J| = 3$ and $|I'| = 2$, and (ii) $|I'| = |J| = 3$ and $|I| = 2$.

B.4 Mixture Differential Distinguisher/Attack

As mentioned in Sect. 4.2, it is currently impossible to set up an integral mixture distinguisher for ZIP-AES$_{2,2}$, that is,

$$\text{ZIP-AES}_{2,2}(p_0) \oplus \text{ZIP-AES}_{2,2}(p_1) \neq \text{ZIP-AES}_{2,2}(p_0') \oplus \text{ZIP-AES}_{2,2}(p_1')$$

in general, where $p_0, p_1, p_0', p_1' \in \mathcal{C}_I \oplus \gamma$ for $I \subseteq \{0,1,2,3\}$, and where p_0' and p_1' are constructed by carefully swapping the generating variables of p_0, p_1 (in the same way described in [Gra18]).

We limit ourselves to present the details for the specific case

$$p_0 = \begin{bmatrix} x_0 & 0 & 0 & 0 \\ x_1 & 0 & 0 & 0 \\ x_2 & 0 & 0 & 0 \\ x_3 & 0 & 0 & 0 \end{bmatrix} \oplus \gamma \equiv \langle x_0, x_1, x_2, x_3 \rangle \oplus \gamma \qquad \text{and} \qquad p_1 = \langle y_0, y_1, y_2, y_3 \rangle \oplus \gamma\,,$$

[5] Given a subspace $\mathfrak{X}$, note that $x \oplus y \notin \mathfrak{X}$ cannot occur if $x \in \mathfrak{X}$ and $y \notin \mathfrak{X}$.

where $\langle\cdot\rangle$ denotes the generating variables. Examples of p'_0 and p'_1 are given by $p'_0 = \langle y_0, x_1, x_2, x_3\rangle \oplus \gamma$ and $p'_1 = \langle x_0, y_1, y_2, y_3\rangle \oplus \gamma$. (We recall that if two generating variables are equal, e.g., $x_0 = y_0$, then it is possible to replace them with the same random value in $\mathbb{F}_{2^8}$.)

Let SSB be the super-Sbox operation defined as $SSB(\cdot) := SB \circ ARK \circ MC \circ SB(\cdot)$. Hence:

$$\begin{aligned}&\text{AES}_2(p_0) \oplus \text{AES}_2(p_1) = MC \circ SR \circ (SSB \circ SR(p_0) \oplus SSB \circ SR(p_1))\\ =&MC \circ SR \circ (SSB \circ SR(p'_0) \oplus SSB \circ SR(p'_1)) = \text{AES}_2(p'_0) \oplus \text{AES}_2(p'_1)\end{aligned}$$

since each column of $SR(p_0)$ and of $SR(p_1)$ depends on independent variables, and since SSB works independently on each column of the input text.

The problem arises from the fact that AES_2^{-1} starts with a MC^{-1} operation, which implies that the generating variables of $MC^{-1}(p_0)$ (similar for $MC^{-1}(p_1)$) are

$$\begin{aligned}\hat{x}_0 &= e \cdot x_0 \oplus b \cdot x_1 \oplus d \cdot x_2 \oplus 9 \cdot x_3, \qquad & \hat{x}_1 &= 9 \cdot x_0 \oplus e \cdot x_1 \oplus b \cdot x_2 \oplus d \cdot x_3\,,\\ \hat{x}_2 &= d \cdot x_0 \oplus 9 \cdot x_1 \oplus e \cdot x_2 \oplus b \cdot x_3, \qquad & \hat{x}_3 &= b \cdot x_0 \oplus d \cdot x_1 \oplus 9 \cdot x_2 \oplus e \cdot x_3\,,\end{aligned}$$

and not x_0, x_1, x_2, x_3. It follows that swapping some variables among x_0, x_1, x_2, x_3 and y_0, y_1, y_2, y_3 does not correspond to swapping the variables $\hat{x}_0, \hat{x}_1, \hat{x}_2, \hat{x}_3$ and $\hat{y}_0, \hat{y}_1, \hat{y}_2, \hat{y}_3$. Equivalently, the equivalence

$$\text{AES}_2^{-1}(p_0) \oplus \text{AES}_2^{-1}(p_1) = \text{AES}_2^{-1}(\hat{p}'_0) \oplus \text{AES}_2^{-1}(\hat{p}'_1)$$

holds for some *unknown*[6] texts $\hat{p}'_0$ and $\hat{p}'_1$ that are *different* from p'_0 and p'_1, since they are defined with respect to different generating variables.

Proof of Lemma 1. Due to the previous results, we have that

$$\begin{aligned}&\text{ZIP-AES}_{2,2}(p_0) \oplus \text{ZIP-AES}_{2,2}(p_1) \oplus \text{ZIP-AES}_{2,2}(p'_0) \oplus \text{ZIP-AES}_{2,2}(p'_1)\\ =&\text{AES}_2^{-1}(p_0) \oplus \text{AES}_2^{-1}(p_1) \oplus \text{AES}_2^{-1}(p'_0) \oplus \text{AES}_2^{-1}(p'_1)\end{aligned}$$

due to the fact that $\text{AES}_2(p_0) \oplus \text{AES}_2(p_1) = \text{AES}_2(p'_0) \oplus \text{AES}_2(p'_1)$.

Since $p_0 \oplus p_1 \in \mathcal{C}_i$, then the corresponding difference belongs to $\mathcal{D}_i$ after R_{AES}^{-1}. Moreover, it also belongs to $\mathcal{C}_I$ with $|I| = 3$ with probability 2^{-8} (if one byte of such difference is equal to zero). It follows that $\text{Prob}(AES_2^{-1}(p_0) \oplus AES_2^{-1}(p_1) \in \mathcal{D}_I \mid p_0 \oplus p_1 \in \mathcal{C}_i) = 2^{-8}$. A similar result holds for p'_0 and p'_1 as well, that is, $\text{Prob}(AES_2^{-1}(p'_0) \oplus AES_2^{-1}(p'_1) \in \mathcal{D}_I \mid p'_0 \oplus p'_1 \in \mathcal{C}_i) = 2^{-8}$, which implies probability (at least)[7] equal to $(2^{-8})^2 = 2^{-16}$ for the considered event.

[6] They are unknown due to the presence of the secret key.

[7] Note that $\text{AES}_2^{-1}(p_0) \oplus \text{AES}_2^{-1}(p_1) \oplus \text{AES}_2^{-1}(p'_0) \oplus \text{AES}_2^{-1}(p'_1)$ can belong into $\mathcal{D}_I$ even if $AES_2^{-1}(p_0) \oplus AES_2^{-1}(p_1) \notin \mathcal{D}_I$ and $AES_2^{-1}(p'_0) \oplus AES_2^{-1}(p'_1) \notin \mathcal{D}_I$.

B.5 Exchange Distinguisher/Attack

The exchange attack [BR19] introduced by Bardeh and Rønjom is a variant of the mixture differential cryptanalysis on 4-round AES. Given two pairs of plaintexts p_0, p_1 in a diagonal subspace $\mathcal{D}_I \oplus \alpha$ with $I \subseteq \{0,1,2,3\}$ and $|I| = 3$. The attacker generates two new pairs of plaintexts p'_0, p'_1 in the same diagonal subspace $\mathcal{D}_I \oplus \alpha$ by exchanging two full diagonals. The following distinguisher works

$$\text{AES}_4(p_0) \oplus \text{AES}_4(p_1) \in \mathcal{M}_J \qquad \Longleftrightarrow \qquad \text{AES}_4(p'_0) \oplus \text{AES}_4(p'_1) \in \mathcal{M}_J$$

for each $J \subseteq \{0,1,2,3\}$ with $|J| \geq 2$.

The reason why such distinguisher works is the following. By working as before, it is not hard to check that

$$\text{AES}_2(p_0) \oplus \text{AES}_2(p_1) = \text{AES}_2(p'_0) \oplus \text{AES}_2(p'_1)\,,$$

since $p'_0, p'_1 \in \mathcal{D}_I \oplus \alpha$ are obtained by swapping diagonals of p_0, p_1. Due to the probability-1 2-round truncated differential $\text{AES}_2(\mathcal{D}_I \oplus \alpha) = \mathcal{M}_I \oplus \beta$, we have that $x \in \mathcal{M}_I$ can belong to $\mathcal{D}_J$ if and only if $|I| + |J| \geq 5$. Hence, by choosing $|I| = 3$ and $|J| \geq 2$, we have that $\text{AES}_2(p_0) \oplus \text{AES}_2(p_1) \in \mathcal{D}_J$ implies $\text{AES}_2(p'_0) \oplus \text{AES}_2(p'_1) \in \mathcal{D}_J$ and so the results.

Since the exchange attack is a variant of the mixture differential attacks, it does not work on ZIP-AES due to the same reasons given before. In particular, the generating variables corresponding to the diagonals of p_0, p_1 are different than the generating variables of $MC^{-1}(p_0), MC^{-1}(p_1)$.

B.6 Attacks on ZIP-AES$_{1,r}$ and ZIP-AES$_{2,r}$

Finally, we show that the attacks proposed in [MN17b, Sect. 3.3] for the AES-PRF construction apply to our constructions ZIP-AES$_{1,r}$ and ZIP-AES$_{2,r}$ as well (analogous for ZIP-AES$_{r,1}$ and ZIP-AES$_{r,2}$). Here, we limit ourselves to adapt the attacks from [MN17b, Sect. 3.3] to our construction.

Let's first consider the case ZIP-AES$_{1,r}$, which is defined as

$$\text{ZIP-AES}_{1,r}(x) := \text{AES}_r^{-1}(x) \oplus R_{\text{AES}}(x \oplus k_0) \oplus k_1\,.$$

Consider inputs in a diagonal subspace $\mathcal{D}_i \oplus \alpha$. As we have seen, $R_{\text{AES}}(\mathcal{D}_i \oplus \alpha) = \mathcal{C}_i \oplus \beta$ for a certain β. The attack strategy consists in guessing the i-th diagonal of the key k_0 (for a total of 4 bytes). If the guessed value is correct, the attacker can predict the value of the i-th column of $R_{\text{AES}}(x \oplus k_0)$. By removing its contribution from ZIP-AES$_{1,r}(x)$, the resulting function is the permutation $\text{AES}_r^{-1}(x) \oplus \gamma$ for a certain secret constant $\gamma \in \mathbb{F}_{2^8}^{4\times 4}$. Hence, if the guessed key is correct, we expect that no collision occurs on the image of $\mathcal{D}_i \oplus \alpha$ via ZIP-AES$_{1,r}(x)$ once we removed the i-th column of $R_{\text{AES}}(x \oplus k_0)$. Equivalently, if a collision occurs, the guessed key is not the correct one. We refer to [MN17b, Sect. 3.3] for more details. In there, Mennink and Neves show that the attack requires approximately 2^{67} queries, 2^{101} computations, and 2^{67} memory.

A similar attack can potentially work for ZIP-AES$_{2,r}$ as well, by noting that $R^2_{\text{AES}}(\mathcal{D}_i \oplus \alpha) = MC(\mathcal{I}\mathcal{D}_i) \oplus \beta \equiv \mathcal{M}_i \oplus \beta$. In such a case, the attacker needs to guess at least one diagonal of the first round-key and one column of the second round-key, for a total of (at least) 8 bytes of the key. We leave the open problem to estimate the cost of such attack for future work.

C Security Analysis of Modified ZIP-AES

In this section, we show that a modified version of ZIP-AES in which the initial MC^{-1} operation of AES^{-1} and/or the final MC operation of AES are omitted is *much less secure* than the ZIP-AES scheme analyzed before. In order to avoid confusion, we denote this modified version as ZIP-AES′.

We summarize the results in the following, before presenting the details.

About Truncated Differentials: Effect of Omitting MC^{-1}. Omitting MC^{-1} at the beginning of AES^{-1} allows the attacker to break more rounds of ZIP-AES′ via truncated differential distinguishers. In more details:

- the attacker can break $2+2$ rounds of ZIP-AES via a probabilistic truncated differential distinguisher. Even if it is not clear if analogous truncated differential distinguishers exist for more rounds, our results suggest that, if a bias between the probability for ZIP-AES$_{r,r}$ for $r \geq 4$ and a generic PRF exists, it would be too small for being useful in practice. No distinguisher/attack is known for more rounds;
- the attacker can break $2+2$ rounds of ZIP-AES′ via a deterministic (i.e., probability 1) truncated differential distinguisher;
- the attacker can break $4+4$ rounds of ZIP-AES′ via probabilistic truncated differential distinguishers. We also do not exclude that an attacker can break more rounds.

About Mixture Differentials: Effect of Omitting MC^{-1}. As before, omitting MC^{-1} at the beginning of AES^{-1} allows the attacker to break more rounds of ZIP-AES′ via mixture differential distinguishers. In more details:

- the attacker can break $2+2$ rounds of ZIP-AES via a probabilistic mixture differential distinguisher. No distinguisher/attack is known for more rounds;
- the attacker can break $2+2$ rounds of ZIP-AES′ via a deterministic (i.e., probability 1) mixture differential distinguisher;
- the attacker can break $4+4$ rounds of ZIP-AES′ via a probabilistic mixture differential distinguisher.

C.1 Omitting the Final MC Operation in AES

Let's start with some observations about the consequence of omitting the final MC operation in AES. As we have seen before, several truncated and mixture

Table 5: Practical tests on *small-scale* ZIP-AES′ over $\mathbb{F}_{2^4}^{4\times 4}$. The symbol $\star$ denotes the fact that the final MixColumns operation is omitted in AES$_r$. In the table, we assume $|I| = 3$.

# Rounds	Input Subspace	Output Subspace	ZIP-AES T-Prob.	ZIP-AES P-Prob.	PRF Prob.
$1+1$	$\mathcal{C}_i$	$\mathcal{D}_i \oplus \mathcal{M}_i$	1	1	2^{-32}
$1+1$	$\mathcal{D}_i \cap \mathcal{ID}_i$	$\mathcal{C}_i$	1	1	2^{-48}
$2+2$	$\mathcal{D}_i \cap \mathcal{ID}_i$	$\mathcal{D}_i \oplus \mathcal{M}_i$	1	1	2^{-32}
$3^\star+3$	$\mathcal{D}_i \cap \mathcal{ID}_i$	$\mathcal{D}_I \oplus \mathcal{ID}_I$	2^{-7}	$2^{-6.985}$	2^{-8}
$3+3$	$\mathcal{D}_i \cap \mathcal{ID}_i$	$\mathcal{C}_I$	–	$2^{-16}+2^{-19.75}$	2^{-16}
$3+3$	$\mathcal{C}_i$	$\mathcal{C}_I$	–	$2^{-16}+2^{-27.7}$	2^{-16}
$4^\star+4$	$\mathcal{C}_i$	$\mathcal{D}_I \oplus \mathcal{ID}_I$	–	$2^{-8}+2^{-25.1}$	2^{-8}
$4^\star+4$	$\mathcal{D}_i \cap \mathcal{ID}_i$	$\mathcal{D}_I \oplus \mathcal{ID}_I$	–	$2^{-8}-2^{-18.5}$	2^{-8}

T ≡ Theoretical – P ≡ Practical – Prob. ≡ Probability

differentials for ZIP-AES end with the column space $\mathcal{C}_I$. Since $\mathcal{C}_I$ is mapped into $\mathcal{M}_I$ after 1-round AES, and into $\mathcal{D}_I$ after 1-round AES^{-1}, one may think that the simplest strategy to extend by 1 round a truncated/mixture differential that ends with $\mathcal{C}_I$ is by replacing $\mathcal{C}_I$ with $\mathcal{D}_J \oplus \mathcal{M}_L$. However, $\mathcal{D}_J \oplus \mathcal{M}_L$ corresponds to the full space for each $J, L \subseteq \{0, 1, 2, 3\}$ such that $|J| + |L| \geq 4$. Hence, this strategy could work only if we special cases.

If the final MC operation is omitted AES, then the result can be different. Indeed, in such a case, one would replace $\mathcal{C}_I$ with $\mathcal{D}_J \oplus \mathcal{ID}_L$, where $\mathcal{D}_J \oplus \mathcal{ID}_L$ is not necessarily the full space even if $|J| + |L| \geq 4$. As a concrete example, if $L = J = \{h, h+2\}$ for $h \in \{0, 1, 2, 3\}$, then $\mathcal{D}_{h,h+2} = \mathcal{ID}_{h,h+2}$, which implies that $\mathcal{D}_{h,h+2} \oplus \mathcal{ID}_{h,h+2} = \mathcal{D}_{h,h+2}$ which has dimension 8 out of 16. As a result, some truncated and mixture differential distinguishers can be easily extended by 1 round if the final MC operation is omitted, as we are going to show concretely in the following.

C.2 Truncated Differential and Subspace Trail Attacks

We refer to Sect. 4.2 for the details about truncated differential distinguishers on ZIP-AES. Here we focus on the case of ZIP-AES′, which corresponds to ZIP-AES with the initial MC^{-1} for AES^{-1} omitted. If the final MC operation is *also* omitted, we denote this case as ZIP-AES*.

In the following, we discuss the truncated differentials for round-reduced ZIP-AES. Our practical tests on *small-scale* ZIP-AES (that is, AES over $\mathbb{F}_{2^4}^{4\times 4}$ as presented in [CMR05]) are also summarized in Table 5.[8] Before going on, we recall that $\dim(\mathcal{D}_I \oplus \mathcal{ID}_J) = 14$ if $|I| = |J| = 3$ with $|I \cap J| \geq 2$ (e.g., if $I = J$).

Truncated Differentials for ZIP-AES$'_{1,1}$. Based on the previous results, it is possible to set up truncated differentials on ZIP-AES$_{1,1}$ by combining the facts that

[8] Note that the truncated differentials are not affected by the details (as the degree) of the S-Box. Hence, we believe that the results on small-scale AES are a good representative of what happens for the "real" AES.

(i) $\text{Prob}(\text{AES}_1(x) \oplus \text{AES}_1(x \oplus \delta) \in \mathcal{M}_I \mid \delta \in \mathcal{C}_I) = 1$ and (ii) $\text{Prob}(\text{AES}_1^{-1}(x) \oplus \text{AES}_1^{-1}(x \oplus \delta) \in \mathcal{D}_I \mid \delta \in \mathcal{C}_I) = 1$:

$$\text{Prob}(\text{ZIP-AES}'_{1,1}(x) \oplus \text{ZIP-AES}'_{1,1}(x \oplus \delta) \in \mathcal{C}_i \mid \delta \in \mathcal{D}_i \cap \mathcal{ID}_i) = 1\,,$$
$$\text{Prob}(\text{ZIP-AES}'_{1,1}(x) \oplus \text{ZIP-AES}'_{1,1}(x \oplus \delta) \in \mathcal{D}_i \oplus \mathcal{M}_i \mid \delta \in \mathcal{C}_i) = 1\,.$$

In this second case, we limit ourselves to mention that if the final MC operation is omitted in $\text{AES}_1(\cdot)$, then it is sufficient to replace $\mathcal{M}_i$ with $\mathcal{ID}_i$. Moreover, if AES_1^{-1} finishes with the MC^{-1} operation, then it is sufficient to replace $\mathcal{D}_i$ with $MC^{-1}(\mathcal{D}_i)$.

For comparison, note that $\text{Prob}(\Pi(x) \oplus \Pi(x \oplus \delta) \in \mathcal{C}_i \mid \delta \in \mathcal{D}_i \cap \mathcal{ID}_i) = 2^{-96}$ and $\text{Prob}(\Pi(x) \oplus \Pi(x \oplus \delta) \in \mathcal{D}_i \oplus \mathcal{M}_i \mid \delta \in \mathcal{C}_i) = 2^{-64}$ for a PRF Π over $\mathbb{F}_{2^8}^{4\times 4}$.

Truncated Differentials for ZIP-AES$'_{2,2}$. By combining the two previous probabilities, it is possible to set up a truncated differential for $\text{ZIP-AES}'_{2,2}$. Indeed, we have that

$$\text{Prob}(\text{ZIP-AES}'_{2,2}(x) \oplus \text{ZIP-AES}'_{2,2}(x \oplus \delta) \in \mathcal{D}_i \oplus \mathcal{M}_i \mid \delta \in \mathcal{D}_i \cap \mathcal{ID}_i) = 1\,.$$

As before, the same event occurs with probability 2^{-64} in the case of a PRF Π. Similar considerations as before hold if the final MixColumn operation is omitted or/and the final inverse MixColumns operation is included.

Truncated Differentials for ZIP-AES$'_{3,3}$. Besides assuming that the initial MC^{-1} is omitted, let's first consider the case in which the final MixColumns operation in $\text{AES}_3(\cdot)$ is omitted as well – denoted by $\text{AES}^\star_3(\cdot)$. By combining the previous probability, it is possible to set up a truncated differential for $\text{ZIP-AES}^\star_{3,3}$ as follows:

$$\text{Prob}(\text{ZIP-AES}^\star_{3,3}(x) \oplus \text{ZIP-AES}^\star_{3,3}(x \oplus \delta) \in \mathcal{D}_J \oplus \mathcal{ID}_J \mid \delta \in \mathcal{D}_i \cap \mathcal{ID}_i) \approx 2^{-15}\,,$$

where $i \in \{0, 1, 2, 3\}$ and $J \subseteq \{0, 1, 2, 3\}$ with $|J| = 3$. Indeed:

- the event

  $$\text{AES}_1(x) \oplus \text{AES}_1(x \oplus \delta) \in \mathcal{D}_J \qquad \text{and} \qquad \text{AES}_1^{-1}(x) \oplus \text{AES}_1^{-1}(x \oplus \delta) \in \mathcal{M}_J$$

  occurs with probability 2^{-16} (where $\text{AES}_1^{-1}(\cdot) = MC^{-1} \circ SR^{-1} \circ SB^{-1}(\cdot)$). If this happens, then $\text{ZIP-AES}^\star_{3,3}(x) \oplus \text{ZIP-AES}^\star_{3,3}(x \oplus \delta) \in \mathcal{D}_J \oplus \mathcal{ID}_J$ occurs with probability 1, due to the previous probability-1 2-round truncated differentials;
- instead, the events

  $$\text{AES}_1(x) \oplus \text{AES}_1(x \oplus \delta) \notin \mathcal{D}_J \qquad \text{and/or} \qquad \text{AES}_1^{-1}(x) \oplus \text{AES}_1^{-1}(x \oplus \delta) \notin \mathcal{M}_J$$

  occur with probability $(1 - 2^{-16})$. If this is the case, then $\text{ZIP-AES}^\star_{3,3}(x) \oplus \text{ZIP-AES}^\star_{3,3}(x \oplus \delta)$ can still belong to $\mathcal{D}_J \oplus \mathcal{ID}_J$ with probability approximately $2^{-16} \pm \varepsilon$ for a (very) small $0 \le \varepsilon \ll 1$.

Hence, the overall probability is well approximated by

$$2^{-16} \cdot 1 + (1 - 2^{-16}) \cdot (2^{-16} \pm \varepsilon) \approx 2^{-15} \,.$$

Note that the same event has probability 2^{-16} for the case of a PRF. We practically verified this case on small-scale AES, and we obtained a factor (slightly bigger than) 2 between the ZIP-AES^{s,*} case and a generic PRF (as expected).

Remark 2. Before going on, we limit ourselves to point out that our theoretical result from Sect. 3.5 just suggests us that the probability of such event is greater than 2^{-16}. However, this result by itself is not sufficient to set up a distinguisher.

A similar distinguisher holds when the final MC is not omitted. In particular, due to the same argument just given, we expect that an analogous truncated differential for ZIP-AES$'_{3,3}$ holds:

$$\begin{aligned}&\text{Prob}(\text{ZIP-AES}'_{3,3}(x) \oplus \text{ZIP-AES}'_{3,3}(x \oplus \delta) \in \mathcal{D}_j \oplus \mathcal{M}_l \mid \delta \in \mathcal{D}_{i,i+2} \equiv \mathcal{ID}_{i,i+2})\\ =&(2^{-64} \pm \varepsilon) \cdot (1 - 2^{-96}) + (2^{-48})^2 \approx 2^{-64} + 2^{-96} \,,\end{aligned}$$

assuming a (very) small $0 \leq \varepsilon \ll 1$, and where $j, l \in \{0, 1, 2, 3\}$. For comparison, the same event occurs with probability 2^{-64} for the case of random function. With respect to the case in which the final MC operation is omitted, we highlight that we cannot work with $\mathcal{D}_I \oplus \mathcal{M}_I$ for $|I| = 3$, since this is equivalent to the full space, and the probability would be trivially 1. For this reason, we chose $\mathcal{D}_j \oplus \mathcal{M}_l$ as the final subspace. Moreover, due to the MDS property of the MC matrix, we are forced to work with $\mathcal{D}_{i,i+2} \equiv \mathcal{ID}_{i,i+2}$ instead of $\mathcal{D}_{i,i+2} \cap \mathcal{ID}_{i,i+2}$.

Truncated Differentials for ZIP-AES$'_{r,r}$ with $r \geq 3$: Practical Results. As for the case of ZIP-AES, we practically tested the probability of several truncated differentials for *small-scale* ZIP-AES$'_{r,r}$ and ZIP-AES$^{\star}_{r,r}$ for $r \in \{3, 4\}$. Our practical results are listed in Table 5. As it is possible to observe, truncated differentials for up to ZIP-AES$'_{4,4}$ and ZIP-AES$^{\star}_{4,4}$ exist (we also do not exclude the possibility that truncated differentials exist for $r \geq 5$). We leave the *open problems* to explain them, and to set up analogous truncated differentials for "real" ZIP-AES$'_{r,r}$ and ZIP-AES$^{\star}_{r,r}$ as future work.

C.3 Mixture Differential Attacks (and More)

We refer to Sect. 4.2 for the details about mixture differential and the exchange distinguishers on ZIP-AES. Here we focus on the case of ZIP-AES$'$, which corresponds to ZIP-AES with the initial MC^{-1} for AES^{-1} omitted. If the final MC operation is *also* omitted, we denote this case as ZIP-AES*.

Mixture Integral Distinguisher for ZIP-AES$'_{2,2}$ As we are going to show, it is possible to set up an *integral mixture distinguisher* (using the same name proposed in [GS20]) for ZIP-AES$'_{2,2}$, that is,

$$\text{ZIP-AES}'_{2,2}(p_0) \oplus \text{ZIP-AES}'_{2,2}(p_1) = \text{ZIP-AES}'_{2,2}(p'_0) \oplus \text{ZIP-AES}'_{2,2}(p'_1) \quad (2)$$

where $p_0, p_1, p'_0, p'_1 \in \mathcal{C}_I \oplus \gamma$ for $I \subseteq \{0,1,2,3\}$, and where p'_0 and p'_1 are constructed by carefully swapping the generating variables of p_0, p_1 (in the same way described in [Gra18]).

We limit ourselves to present the details for the specific case

$$p_0 = \langle x_0, x_1, x_2, x_3 \rangle \oplus \gamma \qquad \text{and} \qquad p_1 = \langle y_0, y_1, y_2, y_3 \rangle \oplus \gamma$$

as defined before, where $\langle \cdot \rangle$ denotes the generating variables. We recall that examples of p'_0 and p'_1 are given by $p'_0 = \langle y_0, x_1, x_2, x_3 \rangle \oplus \gamma$ and $p'_1 = \langle x_0, y_1, y_2, y_3 \rangle \oplus \gamma$. Let SSB be the super-Sbox operation defined as $SSB(\cdot) := SB \circ ARK \circ MC \circ SB(\cdot)$. Hence:

$$\begin{aligned} \text{AES}_2(p_0) \oplus \text{AES}_2(p_1) &= MC \circ SR \circ (SSB \circ SR(p_0) \oplus SSB \circ SR(p_1)) \\ =& MC \circ SR \circ (SSB \circ SR(p'_0) \oplus SSB \circ SR(p'_1)) = \text{AES}_2(p'_0) \oplus \text{AES}_2(p'_1) \end{aligned}$$

since each column of $SR(p_0)$ and of $SR(p_1)$ depends on independent variables, and since SSB works independently on each column of the input text. In an analogous way, we have

$$\begin{aligned} \text{AES}_2^{-1}(p_0) \oplus \text{AES}_2^{-1}(p_1) &= SR^{-1} \circ \big(SSB^{-1} \circ SR^{-1}(p_0) \oplus SSB^{-1} \circ SR^{-1}(p_1)\big) \\ =& SR^{-1} \circ \big(SSB^{-1} \circ SR^{-1}(p'_0) \oplus SSB^{-1} \circ SR^{-1}(p'_1)\big) = \text{AES}_2^{-1}(p'_0) \oplus \text{AES}_2^{-1}(p'_1)\,. \end{aligned}$$

It is crucial to note that the generating variables remain independent (i.e., belong to different columns) after both SR and SR^{-1}:

$$\begin{bmatrix} x_0 & 0 & 0 & 0 \\ 0 & x_1 & 0 & 0 \\ 0 & 0 & x_2 & 0 \\ 0 & 0 & 0 & x_3 \end{bmatrix} \xleftarrow{SR^{-1}(\cdot)} \begin{bmatrix} x_0 & 0 & 0 & 0 \\ x_1 & 0 & 0 & 0 \\ x_2 & 0 & 0 & 0 \\ x_3 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{SR(\cdot)} \begin{bmatrix} x_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 \\ 0 & 0 & x_2 & 0 \\ 0 & x_3 & 0 & 0 \end{bmatrix}.$$

Hence, it is possible to construct mixing pairs that are *simultaneously* suitable for both the encryption and the decryption.

By combining all these facts, that is,

$$\begin{aligned} \text{AES}_2(p_0) \oplus \text{AES}_2(p_1) =& \text{AES}_2(p'_0) \oplus \text{AES}_2(p'_1)\,, \\ \text{AES}_2^{-1}(p_0) \oplus \text{AES}_2^{-1}(p_1) =& \text{AES}_2^{-1}(p'_0) \oplus \text{AES}_2^{-1}(p'_1)\,, \end{aligned} \tag{3}$$

the equality (2) follows immediately.

Global Mixture Differential on ZIP-AES$'_{3,3}$. In order to extend such distinguisher by 2 extra rounds as in [Gra18], it would be necessary to consider the case in which (i) $\text{AES}_2(p_0) \oplus \text{AES}_2(p_1) \in \mathcal{D}_I$ and (ii) $\text{AES}_2^{-1}(p_0) \oplus \text{AES}_2^{-1}(p_1) \in \mathcal{M}_J$ for certain $I, J \subseteq \{0,1,2,3\}$ in order to exploit the 1-/2-round probability-1 truncated differentials described before. However, the attacker can only observe the fact that

$$\begin{aligned} &\text{ZIP-AES}'_{3,3}(p_0) \oplus \text{ZIP-AES}'_{3,3}(p_1) \in \mathcal{C}_I \qquad \text{or} \\ &\text{ZIP-AES}'_{4,4}(p_0) \oplus \text{ZIP-AES}'_{4,4}(p_1) \in \mathcal{D}_i \oplus \mathcal{M}_j \end{aligned}$$

depending on the number of attacked rounds. By themselves, these conditions do **not** imply $\text{AES}_3(p_0) \oplus \text{AES}_3(p_1) \in \mathcal{C}_I$ (respectively, $\text{AES}_4(p_0) \oplus \text{AES}_4(p_1) \in \mathcal{M}_i$) and $\text{AES}_3^{-1}(p_0) \oplus \text{AES}_3^{-1}(p_1) \in \mathcal{C}_I$ (respectively, $\text{AES}_4^{-1}(p_0) \oplus \text{AES}_4^{-1}(p_1) \in \mathcal{D}_j$). Indeed, it is possible that $x \oplus y \in \mathcal{X}$ even if $x, y \notin \mathcal{X}$.

Here, we solve this by problem by introducing the "*global* mixture differential" distinguishers.

Lemma 2. *Let $p_0, p_1 \in \mathcal{C}_J \oplus \alpha$. Let $\mathfrak{X}_{p_0,p_1} \subseteq (\mathcal{C}_J \oplus \alpha) \times (\mathcal{C}_J \oplus \alpha)$ be the set defined by the mixture couples generated by p_0 and p_1. (We recall that $|\mathfrak{X}_{p_0,p_1}| \in \{0, 8, 2^{10}, 2^{17}\}$.)*

The probability that there exists a set $\mathfrak{X}_{p_0,p_1}$ for which all the corresponding pairs after ZIP-AES$'_{3,3}$ differ in the I-th column(s) after is higher for ZIP-AES$'_{3,3}$ then for a PRF. In particular, for $|I| = 3$, we have that

$$\begin{aligned} \mathit{Prob}\big(\forall (p,p') \in \mathfrak{X}_{p_0,p_1} : \mathit{ZIP\text{-}AES}'_{3,3}(p) \oplus \mathit{ZIP\text{-}AES}'_{3,3}(p') \in \mathcal{C}_I\big) \\ = 2^{-64} + (1 - 2^{-32})^2 \cdot 2^{-32 \cdot |\mathfrak{X}_{p_0,p_1}|} \approx 2^{-64} \end{aligned}$$

versus $\mathit{Prob}\,(\forall (p,p') \in \mathfrak{X}_{p_0,p_1} : \varPi(p) \oplus \varPi(p') \in \mathcal{C}_I) = 2^{-32 \cdot |\mathfrak{X}_{p_0,p_1}|} \le 2^{-256}$.

We called this distinguisher "global" since it is based on a property that must holds for all the pairs in the set $\mathfrak{X}_{p_0,p_1}$.

Proof. Let's start with the random case. In such a case, all the output pairs are independently. Since $\text{Prob}(x \in \mathcal{C}_I) = 2^{-32 \cdot (4 - |I|)}$, that is, 2^{-32} for $|I| = 3$, the result follows immediately.

Next, let's consider the case of ZIP-AES$_{3,3}$. Here, two cases can occur:

1. $\text{AES}_2(p_0) \oplus \text{AES}_2(p_1) \in \mathcal{D}_I$ and $\text{AES}_2^{-1}(p_0) \oplus \text{AES}_2^{-1}(p_1) \in \mathcal{M}_I$;
2. $\text{AES}_2(p_0) \oplus \text{AES}_2(p_1) \notin \mathcal{D}_I$ or/and $\text{AES}_2^{-1}(p_0) \oplus \text{AES}_2^{-1}(p_1) \notin \mathcal{M}_I$.

In the first case, we have that

$$\text{AES}_2(p_0) \oplus \text{AES}_2(p_1) \in \mathcal{D}_I \qquad \longrightarrow \qquad \text{AES}_2(p) \oplus \text{AES}_2(p') \in \mathcal{D}_I$$

for each $\{p, p'\} \in \mathfrak{X}_{p_0,p_1}$. This is due to the fact that $\{p, p'\}$ is a mixture of $\{p_0, p_1\}$, which implies

$$\text{AES}_2(p_0) \oplus \text{AES}_2(p_1) = \text{AES}_2(p) \oplus \text{AES}_2(p')$$

as proved before – see (3). Obviously, it follows that

$$\text{AES}_2(p) \oplus \text{AES}_2(p') \in \mathcal{D}_I \qquad \Longrightarrow \qquad \text{AES}_3(p) \oplus \text{AES}_2(p') \in \mathcal{C}_I.$$

In an analogous way, we have that

$$\begin{aligned} \text{AES}_2^{-1}(p_0) \oplus \text{AES}_2^{-1}(p_1) \in \mathcal{M}_I \qquad &\Longrightarrow \qquad \text{AES}_2^{-1}(p) \oplus \text{AES}_2^{-1}(p') \in \mathcal{M}_I \\ &\Longrightarrow \qquad \text{AES}_3^{-1}(p) \oplus \text{AES}_3^{-1}(p') \in \mathcal{C}_I\,. \end{aligned}$$

The probability of this first event is well approximated by $(2^{-32})^2 = 2^{-64}$, since

$$\begin{aligned}
&\text{Prob}\left(\text{AES}_2(p_0) \oplus \text{AES}_2(p_1) \in \mathcal{D}_I \quad \text{and} \quad \text{AES}_2^{-1}(p_0) \oplus \text{AES}_2^{-1}(p_1) \in \mathcal{M}_I\right)\\
=&\text{Prob}\left(\text{AES}_4(\hat{p}_0) \oplus \text{AES}_4(\hat{p}_1) \in \mathcal{D}_I \quad \text{and} \quad \hat{p}_0 \oplus \hat{p}_1 \in \mathcal{M}_I\right)\\
=&\text{Prob}\left(\text{AES}_4(\hat{p}_0) \oplus \text{AES}_4(\hat{p}_1) \in \mathcal{D}_I \mid \hat{p}_0 \oplus \hat{p}_1 \in \mathcal{M}_I\right) \cdot \text{Prob}\left(\hat{p}_0 \oplus \hat{p}_1 \in \mathcal{M}_I\right)\\
=&2^{-32} \cdot \text{Prob}\left(\text{AES}_8(\hat{p}_0) \oplus \text{AES}_8(\hat{p}_1) \in \mathcal{M}_I \mid \hat{p}_0 \oplus \hat{p}_1 \in \mathcal{D}_I\right) \approx (2^{-32})^2\,,
\end{aligned}$$

where no truncated differential is known for 8-round AES – see [Rø19, GR22, BR19]. (Analogous conclusion holds when including the condition $p_0 \oplus p_1 \in \mathcal{C}_J$ as well.)

Regarding the second case, the event $\text{AES}_2(p_0) \oplus \text{AES}_2(p_1) \notin \mathcal{D}_I$ or/and $\text{AES}_2^{-1}(p_0) \oplus \text{AES}_2^{-1}(p_1) \notin \mathcal{M}_I$ occurs with probability $1 - 2^{-64}$. However, we are only interested to the case $\text{AES}_2(p_0) \oplus \text{AES}_2(p_1) \notin \mathcal{D}_I$ **and** $\text{AES}_2^{-1}(p_0) \oplus \text{AES}_2^{-1}(p_1) \notin \mathcal{M}_I$, which occurs with probability $(1 - 2^{-32})^2$. Indeed, remember that $x \oplus y \in \mathcal{X}$ cannot occur if $x \in \mathcal{X}$ and $y \notin \mathcal{X}$. Hence, working as before, we know that for each $\{p, p'\} \in \mathfrak{X}_{p_0,p_1}$:

$$\text{AES}_2(p) \oplus \text{AES}_2(p') \notin \mathcal{D}_I \qquad \text{and} \qquad \text{AES}_2^{-1}(p) \oplus \text{AES}_2^{-1}(p') \notin \mathcal{M}_I\,,$$

which implies

$$\text{AES}_3(p) \oplus \text{AES}_3(p') \notin \mathcal{C}_I \qquad \text{and} \qquad \text{AES}_3^{-1}(p) \oplus \text{AES}_3^{-1}(p') \notin \mathcal{C}_I\,.$$

It follows that the probability that the event $\text{AES}_3(p) \oplus \text{AES}_3(p') \oplus \text{AES}_3^{-1}(p) \oplus \text{AES}_3^{-1}(p') \in \mathcal{C}_I$ occurs is 2^{-32} for each entry of $\mathfrak{X}_{p_0,p_1}$. The result follows immediately. □

Next, we analyze the cost of such distinguisher. As showed e.g. in [GR22, Theorem 4], we recall that each coset of $\mathcal{C}_i$ for $i \in \{0, 1, 2, 3\}$ contains:

- $2^{28} \cdot (2^8 - 1)^4$ sets $\mathfrak{X}_{p_0,p_1}$ of pairs p_0, p_1 with no equal generating variables,
- $2^{23} \cdot (2^8 - 1)^3$ sets $\mathfrak{X}_{p_0,p_1}$ of pairs p_0, p_1 with *exactly* one equal generating variable,
- $3 \cdot 2^{15} \cdot (2^8 - 1)^2$ sets $\mathfrak{X}_{p_0,p_1}$ of pairs p_0, p_1 with *exactly* two equal generating variables,

for a total of

$$2^{28} \cdot (2^8 - 1)^4 + 2^{23} \cdot (2^8 - 1)^3 + 3 \cdot 2^{15} \cdot (2^8 - 1)^2 \approx 2^{59.98}$$

different sets. Since each set satisfies the required event with probability 2^{-64}, in order to have a probability of success higher than 95%, we need N sets, where

$$1 - (1 - 2^{-64})^N \approx 1 - e^{-N \cdot 2^{-64}} \geq 0.95 \quad \Longrightarrow \quad N \geq -2^{64} \cdot \ln(0.05) \approx 2^{65.6}\,.$$

Hence, we need $2^{65.6}/2^{59.98} = 2^{5.7}$ initial cosets $\mathcal{C}_i$, for a data complexity of $2^{5.7} \cdot 2^{32} = 2^{37.7}$ chosen plaintexts. This number can be reduced by a factor 4 by considering all possible values of I with $|I| = 3$.

Regarding the computational complexity, a possible approach could be the following. For each coset $\mathcal{C}_i$:

- re-order the ciphertexts (and the corresponding plaintexts) with respect to the byte in column $\{0, 1, 2, 3\} \setminus I$ (e.g., using an algorithm such as Heapsort or Merge sort or others);
- working only on consecutive pair of sets, identify such pairs that belong to the same coset of $\mathcal{C}_I$;
- construct the corresponding set $\mathfrak{X}_{p_0,p_1}$ and check if the required property is satisfied or not.

An estimation of the total is hence given by

$$2^{5.7} \cdot \left(\underbrace{\mathcal{O}\left(2^{32} \cdot \log_2(2^{32})\right)}_{\text{sort}} + \underbrace{2^{-32} \cdot \binom{2^{32}}{2}}_{\text{pairs in } \mathcal{C}_I} \cdot \underbrace{\mathcal{O}\left(1 + 2^{-32} + \ldots + 2^{-32\cdot|\mathfrak{X}|}\right)}_{\text{check set } \mathfrak{X}} \right) \approx 2^{43}$$

steps. The memory cost is also practical.

Global Mixture Differential on ZIP-AES$^\star_{4,4}$ and ZIP-AES$'_{4,4}$. Next, we analyze the possibility to set up a similar distinguisher for ZIP-AES$'_{4,4}$. We start by considering the case the final MixColumns operation is omitted in which AES$_4(\cdot)$ – denoted by ZIP-AES$^\star_{4,4}$. In such a case, it is trivial to extend the previous distinguisher by considering $\mathcal{D}_I \oplus \mathcal{ID}_I$ instead of $\mathcal{C}_I$ (remember that $\dim(\mathcal{D}_I \oplus \mathcal{ID}_I) = 14$ for $|I| = 3$). Hence, due to an argument analogous to the one given before,[9] we have that

$$\begin{aligned}\text{Prob}\big(\forall (p, p') \in \mathfrak{X}_{p_0,p_1} : &\text{ZIP-AES}^\star_{4,4}(p) \oplus \text{ZIP-AES}^\star_{4,4}(p') \in \mathcal{D}_I \oplus \mathcal{ID}_I\big) \\ &= 2^{-64} + (1 - 2^{-64}) \cdot 2^{-16\cdot|\mathfrak{X}_{p_0,p_1}|} \approx 2^{-64}\end{aligned}$$

versus $\text{Prob}\left(\forall (p, p') \in \mathfrak{X}_{p_0,p_1} : \mathit{\Pi}(p) \oplus \mathit{\Pi}(p') \in \mathcal{D}_I \oplus \mathcal{ID}_I\right) = 2^{-16\cdot|\mathfrak{X}_{p_0,p_1}|} \le 2^{-128}$. It follows that the distinguisher works exactly as before.

In the case in which the final MixColumns operation is not omitted, we are forced to work with $\mathcal{D}_I \oplus \mathcal{M}_J$ for $|I| + |J| \le 3$ in order to guarantee that $\mathcal{D}_I \oplus \mathcal{M}_J$ is not the full space $\mathbb{F}_{2^8}^{4\times 4}$. By repeating an analogous computation given before, and assuming $|I| + |J| = 3$ (so that $\dim(\mathcal{D}_I \oplus \mathcal{M}_J) = 12$), we derive

$$\begin{aligned}\text{Prob}\big(\forall (p, p') \in \mathfrak{X}_{p_0,p_1} : &\text{ZIP-AES}'_{4,4}(p) \oplus \text{ZIP-AES}'_{4,4}(p') \in \mathcal{D}_I \oplus \mathcal{M}_J\big) \\ &= 2^{-64} \cdot 2^{-96} + (1 - 2^{-160}) \cdot 2^{-32\cdot|\mathfrak{X}_{p_0,p_1}|} \approx 2^{-160}\end{aligned}$$

versus $\text{Prob}\left(\forall (p, p') \in \mathfrak{X}_{p_0,p_1} : \mathit{\Pi}(p) \oplus \mathit{\Pi}(p') \in \mathcal{D}_I \oplus \mathcal{M}_J\right) = 2^{-32\cdot|\mathfrak{X}_{p_0,p_1}|} \le 2^{-256}$.

Let's analyze the cost of such distinguisher, by starting with considerations about the way in which the mixture pairs are constructed. By taking texts in a

[9] Note that $x \in \mathcal{X} \oplus \mathcal{Y}$ even if $x \notin \mathcal{X}$. Hence, we replaced the term $(1 - 2^{-32})^2$ with $1 - 2^{-64}$. However, this does not change the overall conclusion.

coset of $\mathcal{C}_{l,l+2}$ for $l \in \{0,1,2,3\}$, we have the following:

$$\underbrace{\begin{bmatrix} x_{0,0} & 0 & x_{0,2} & 0 \\ 0 & x_{1,0} & 0 & x_{1,2} \\ x_{2,2} & 0 & x_{2,0} & 0 \\ 0 & x_{3,2} & 0 & x_{3,0} \end{bmatrix}}_{\in \mathcal{D}_{l,l+2} \equiv \mathcal{ID}_{l,l+2}} \xleftarrow{SR^{-1}(\cdot)} \underbrace{\begin{bmatrix} x_{0,0} & 0 & x_{0,2} & 0 \\ x_{1,0} & 0 & x_{1,2} & 0 \\ x_{2,0} & 0 & x_{2,2} & 0 \\ x_{3,0} & 0 & x_{3,2} & 0 \end{bmatrix}}_{\in \mathcal{C}_{l,l+2}} \xrightarrow{SR(\cdot)} \underbrace{\begin{bmatrix} x_{0,0} & 0 & x_{0,2} & 0 \\ 0 & x_{1,2} & 0 & x_{1,0} \\ x_{2,2} & 0 & x_{2,0} & 0 \\ 0 & x_{3,0} & 0 & x_{3,2} \end{bmatrix}}_{\in \mathcal{D}_{l,l+2} \equiv \mathcal{ID}_{l,l+2}} .$$

Hence, it is possible to construct mixture pairs of texts that fit simultaneously both encryption and decryption. Having said that, working as in [GR22], each coset of $\mathcal{C}_L$ for $L \subseteq \{0,1,2,3\}$ with $|L| = 2$ as before contains

- $\frac{\left(2^{16}\cdot(2^{16}-1)\right)^4}{2\cdot 2^3} \approx 2^{124}$ sets $\mathfrak{X}_{p_0,p_1}$ of pairs p_0, p_1 with no equal generating variables,
- $\frac{4\cdot 2^{16}\cdot\left(2^{16}\cdot(2^{16}-1)\right)^3}{2\cdot 2^{18}} \approx 2^{95}$ sets $\mathfrak{X}_{p_0,p_1}$ of pairs p_0, p_1 with exactly one equal generating variable,
- $\frac{6\cdot 2^{32}\cdot\left(2^{16}\cdot(2^{16}-1)\right)^2}{2\cdot 2^{33}} \approx 2^{64.6}$ sets $\mathfrak{X}_{p_0,p_1}$ of pairs p_0, p_1 with exactly two equal generating variables,

for a total of $2^{124} + 2^{95} + 2^{64.6} \approx 2^{124}$ mixture sets. Hence, we need approximately $3 \cdot 2^{160} \cdot 2^{-124} \approx 2^{37.6}$ different cosets $\mathcal{C}_I$ for setting up the distiguisher with probability higher than 95%, for a total cost of $2^{37.6} \cdot 2^{64} = 2^{101.6}$ chosen plaintexts.

An Open Problem for Future Work. By making use of the same strategy proposed for ZIP-AES$'_{3,3}$ the estimated cost complexity of the distinguisher is slightly higher than 2^{128} (our rough estimation suggests a cost of $2^{132.6}$). We leave the *open problem* to optimize it in order to achieve a complexity smaller than 2^{128} for future work.

The Exchange Attack. Finally, we briefly consider the exchange attack on ZIP-AES$'$. As we are going to show, it does not outperfom the mixture differential distinguishers just discussed.

Global Exchange Attack on ZIP-AES$'_{3,3}$. Consider plaintexts in the same coset of $\mathcal{D}_{i,i+2} \cap \mathcal{C}_{i,i+2} \equiv \mathcal{ID}_{i,i+2} \cap \mathcal{C}_{i,i+2}$, that is,

$$p_0 = \begin{bmatrix} x_0 & 0 & x_1 & 0 \\ 0 & 0 & 0 & 0 \\ x_2 & 0 & x_3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \oplus \gamma \equiv \langle x_0, x_1, x_2, x_3 \rangle \oplus \gamma \qquad \text{and} \qquad p_1 = \langle y_0, y_1, y_2, y_3 \rangle \oplus \gamma .$$

Construct $p'_0, p'_1 \in \mathcal{D}_{i,i+2} \cap \mathcal{C}_{i,i+2} \oplus \gamma \equiv \mathcal{ID}_{i,i+2} \cap \mathcal{C}_{i,i+2} \oplus \gamma$ by exchanging the two diagonal/anti-diagonals, noting that they are independent:

$$\begin{bmatrix} x_0 & 0 & x_1 & 0 \\ 0 & 0 & 0 & 0 \\ x_2 & 0 & x_3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{SR \equiv SR^{-1}} \begin{bmatrix} x_0 & 0 & x_1 & 0 \\ 0 & 0 & 0 & 0 \\ x_3 & 0 & x_2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} .$$

Moreover, new pairs can be generated by changing the values of the bytes in the other two diagonals/anti-diagonals with other equal values, with the conditions that they are equal for the two pairs.[10] Let $\mathfrak{E}_{p_0,p_1}$ be the set containing the exchanged differential pairs just generated (note that $|\mathfrak{E}_{p_0,p_1}| = 2^9$). Hence, given $\{p_0, p_1\}, \{p'_0, p'_1\} \in \mathfrak{E}_{p_0,p_1} \subseteq \cup_{\gamma \in \Gamma} \mathcal{D}_{i,i+2} \cap \mathcal{C}_{i,i+2} \oplus \gamma$ for a particular Γ, we have that

$$\begin{aligned} \text{AES}_2(p_0) \oplus \text{AES}_2(p_1) = \text{AES}_2(p'_0) \oplus \text{AES}_2(p'_1) \in \mathcal{M}_{i,i+2} \\ \text{AES}_2^{-1}(p_0) \oplus \text{AES}_2^{-1}(p_1) = \text{AES}_2^{-1}(p'_0) \oplus \text{AES}_2^{-1}(p'_1) \in \mathcal{D}_{i,i+2}. \end{aligned}$$

This would allow us to set up a distinguisher on 2+2 rounds. A similar probabilistic distinguisher (in which the previous event happens with probability strictly less than 1) can be set up for 5- and 6-round AES.

As before, it is not possible to set up an exchange differential attack directly, since $\text{ZIP-AES}'_{3,3}(p_0) \oplus \text{ZIP-AES}'_{3,3}(p_1) \in \mathcal{C}_I$ does not imply $\text{ZIP-AES}'_{3,3}(p'_0) \oplus \text{ZIP-AES}'_{3,3}(p'_1) \in \mathcal{C}_I$. However, we can set up a global exchange differential attack on $\text{ZIP-AES}'_{3,3}$ as following

$$\begin{aligned} \text{Prob}\big(\forall (p,p') \in \mathfrak{E}_{p_0,p_1} : & \text{ZIP-AES}'_{3,3}(p) \oplus \text{ZIP-AES}'_{3,3}(p') \in \mathcal{C}_I\big) \\ & = 2^{-64} \cdot 1 + (1 - 2^{-64}) \cdot 2^{-32 \cdot |\mathfrak{E}_{p_0,p_1}|} \approx 2^{-64} \end{aligned}$$

for $|I| = 3$ and where $|\mathfrak{E}_{p_0,p_1}|$, since

- the probability that $\text{AES}_2(p_0) \oplus \text{AES}_2(p_1) = \text{AES}_2(p'_0) \oplus \text{AES}_2(p'_1) \in \mathcal{M}_{i,i+2}$ belongs into $\mathcal{D}_I$ for $|I| = 3$ is 2^{-32} (note that the probability is zero for $|I| \leq 2$);
- the probability that $\text{AES}_2^{-1}(p_0) \oplus \text{AES}_2^{-1}(p_1) = \text{AES}_2^{-1}(p'_0) \oplus \text{AES}_2^{-1}(p'_1) \in \mathcal{D}_{i,i+2}$ belongs into $\mathcal{M}_I$ for $|I| = 3$ is 2^{-32} (note that the probability is zero for $|I| \leq 2$);
- if the two previous events are satisfied, then $\text{ZIP-AES}_{3,3}(p) \oplus \text{ZIP-AES}_{3,3}(p') \in \mathcal{C}_I$ with probability 1;
- if $\text{AES}_2(p_0) \oplus \text{AES}_2(p_1) \notin \mathcal{D}_I$ and $\text{AES}_2^{-1}(p_0) \oplus \text{AES}_2^{-1}(p_1) \notin \mathcal{M}_I$, then the event $\text{ZIP-AES}_{3,3}(p) \oplus \text{ZIP-AES}_{3,3}(p') \in \mathcal{C}_I$ is satisfied with probability approximately 2^{-32} for each pair in $\mathfrak{E}_{p_0,p_1}$.

Hence, it is possible to set up a distinguisher similar to the previous global mixture differential on $3 + 3$ rounds.

About Global Exchange Attack on ZIP-AES$'_{4,4}$: an Open Problem. At the current state, it does *not* seem possible to set up a similar distinguisher on $4+4$ rounds. Indeed, as we already pointed out, the probability that $\text{AES}_2(p_0) \oplus \text{AES}_2(p_1) = \text{AES}_2(p'_0) \oplus \text{AES}_2(p'_1) \in \mathcal{M}_{i,i+2}$ belongs into $\mathcal{D}_I$ for $|I| \leq 2$ is zero. Still, we are forced to consider $|I| \leq 2$ if we aim to finish in a non-trivial subspace of the form $\mathcal{D}_I \oplus \mathcal{M}_J$ after $4 + 4$ rounds. Moreover, note that it is not possible to

[10] We limit ourselves to point out that this step can be further generalize and improve. However, we decided to omit the details since they are not useful for our goals.

exchange diagonals and anti-diagonals in a compatible way when working with the full space $\mathcal{D}_{i,i+2} \equiv \mathcal{ID}_{i,i+2}$. Indeed, the resulting exchanged pairs are **not** compatible:

$$\begin{tikzcd}
\begin{bmatrix} x_{0,0} & 0 & x_{0,2} & 0 \\ 0 & x_{1,1} & 0 & x_{1,3} \\ x_{2,0} & 0 & x_{2,2} & 0 \\ 0 & x_{3,1} & 0 & x_{3,3} \end{bmatrix} \arrow[r, "SR"] & \begin{bmatrix} x_{0,0} & 0 & x_{0,2} & 0 \\ x_{1,1} & 0 & x_{1,3} & 0 \\ x_{2,2} & 0 & x_{2,0} & 0 \\ x_{3,3} & 0 & x_{3,1} & 0 \end{bmatrix} & \textit{versus} \\
& \begin{bmatrix} x_{0,0} & 0 & x_{0,2} & 0 \\ x_{1,3} & 0 & x_{1,1} & 0 \\ x_{2,2} & 0 & x_{2,0} & 0 \\ x_{3,1} & 0 & x_{3,3} & 0 \end{bmatrix} & \begin{bmatrix} x_{0,0} & 0 & x_{0,2} & 0 \\ 0 & x_{1,1} & 0 & x_{1,3} \\ x_{2,0} & 0 & x_{2,2} & 0 \\ 0 & x_{3,1} & 0 & x_{3,3} \end{bmatrix} . \arrow[l, "SR^{-1}"']
\end{tikzcd}$$