Cryptanalysis of AES-PRF and Its Dual

Patrick Derbez, Tetsu Iwata, Ling Sun, Siwei Sun, Yosuke Todo, Haoyang Wang, Meiqin Wang

Cryptanalysis of AES-PRF and Its Dual

Patrick Derbez[1], Tetsu Iwata[2], Ling Sun[3,4], Siwei Sun[5,6,7], Yosuke Todo[8], Haoyang Wang[4] and Meiqin Wang[3]

[1] Univ Rennes, CNRS, IRISA, France
[2] Nagoya University, Japan
[3] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, China
[4] School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore
[5] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, China
[6] Data Assurance and Communication Security Research Center, Chinese Academy of Sciences, China
[7] School of Cyber Security, University of Chinese Academy of Sciences, China
[8] NTT Secure Platform Laboratories, Japan

patrick.derbez@irisa.fr, tetsu.iwata@nagoya-u.jp, lingsun@mail.sdu.edu.cn, sunsiwei@iie.ac.cn, todo.yosuke@lab.ntt.co.jp, wang1153@e.ntu.edu.sg, mqwang@sdu.edu.cn

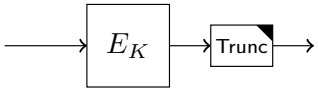
Abstract. A dedicated pseudorandom function (PRF) called AES-PRF was proposed by Mennink and Neves at FSE 2018 (ToSC 2017, Issue 3). AES-PRF is obtained from AES by using the output of the 5-th round as the feed-forward to the output state. This paper presents extensive security analysis of AES-PRF and its variants. Specifically, we consider unbalanced variants where the output of the s-th round is used as the feed-forward. We also analyze the security of "dual" constructions of the unbalanced variants, where the input state is used as the feed-forward to the output of the s-th round. We apply an impossible differential attack, zero-correlation linear attack, traditional differential attack, zero correlation linear distinguishing attack and a meet-in-the-middle attack on these PRFs and reduced round versions. We show that AES-PRF is broken whenever $s \leq 2$ or $s \geq 6$, or reduced to 7 rounds, and Dual-AES-PRF is broken whenever $s \leq 4$ or $s \geq 8$. Our results on AES-PRF improve the initial security evaluation by the designers in various ways, and our results on Dual-AES-PRF give the first insight to its security.

Keywords: AES-PRF · Dual-AES-PRF · Impossible differential · Zero-correlation linear · Meet-in-the-middle

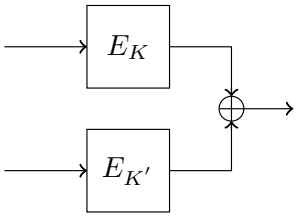
1 Introduction

A pseudorandom permutation (PRP) is one of the main primitives in symmetric-key cryptography to realize security functionalities such as encryption, authentication and authenticated encryption. A PRP is a keyed permutation, where for a randomly chosen key, it is indistinguishable from a truly random permutation [LR88], and this security notion is the main security goal in the design of block ciphers. Cryptanalysis of block ciphers is a long-standing topic in symmetric-key cryptography, and design approaches for an efficient block cipher resisting all known attacks are well studied. Many secure block ciphers are readily available, some of which are standardized and stand the test of extensive cryptanalysis. For instance there is a comfortable consensus in the community that AES [DR02] is indeed a PRP.

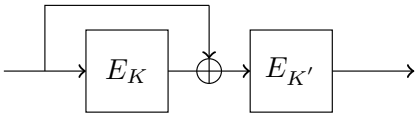
IACR Transactions on Symmetric Cryptology ISSN 2519-173X, Vol. 2018, No. 2, pp. 161–191
DOI:10.13154/tosc.v2018.i2.161-191



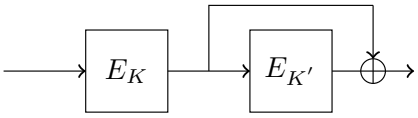
(a) Truncation [HWKS98, BI99, GGM18]



(b) XoP: XOR of Permutations [BKR98, Luc00, BI99, Pat08, Pat10, Pat13]



(c) EDM: Encrypted Davies-Meyer [CS16]



(d) EDMD: The Dual of EDM [MN17a]

Figure 1: Common PRP-to-PRF conversion schemes, where E_K is an n-bit PRP

The invertibility of a block cipher is necessary in some modes of operation. For instance the CBC encryption mode [Dwo01] needs the decryption of the block cipher for its decryption. The authenticated encryption mode OCB [KR11] also needs the block cipher decryption. However, there are various other examples where the invertibility is unnecessary, and the security actually increases if a PRP is replaced with a pseudorandom function (PRF), which is a keyed function that is indistinguishable from a truly random function [GGM86]. For instance the CTR encryption mode [Dwo01] remains secure only if the query complexity is sufficiently smaller than $2^{n/2}$ [BDJR97], where n is the block length of the underlying PRP, but this limitation becomes void if a PRF is used instead. The same argument holds for the authenticated encryption GCM [MV04, Dwo07]. This limitation of the query complexity is often referred to as the birthday bound, and the examples illustrate that highly secure symmetric-key schemes can be obtained once we have a highly secure PRF.

Given ample candidate block ciphers for PRPs, various techniques to convert a PRP into a PRF have been developed. This approach is called the PRP-to-PRF conversion, and the simplest way is to regard the PRP itself as a PRF, but it is well known that the security is limited to the birthday bound. There have been various developments to obtain a PRF with higher security. A scheme that remains secure even beyond the birthday query complexity is said to have beyond the birthday bound security, and we list some of such methods in Fig. 1. However, all these constructions have non-small efficiency costs. The truncation method decreases the rate at which randomness is generated, and each of the other three methods is twice as expensive as one block cipher call.

To maintain both efficiency and beyond the birthday bound security, based on the design called SURF by Bernstein [Ber97] and inspired by EDMD [MN17a], Mennink and Neves [MN17b] explored a dedicated design of a PRF. Specifically, given an r-round iterative block cipher E_K, let E_K^1 be the first $r/2$ rounds of E_K and E_K^2 be the last $r/2$ rounds. Their proposal called FastPRF turns it into a non-invertible function, a PRF, by

$$\mathrm{FastPRF}_K(X) = E_K(X) \oplus E_K^1(X).$$

We see that it runs as fast as the underlying block cipher, and incurs the cost of, besides the block cipher, one additional XOR and the management of one extra data block. The approach is generic in that it transforms any block cipher into a PRF, and [MN17b] proposes a concrete instantiation with AES, which is the main target of this paper.

Table 1: Summary of results. The mark * in column s is $10-t$ and that in column t is $10-s$, but they can take any value.

Target	s	t	Time	Data	Memory	Method	Ref
AES-PRF	1	*	2^{101}	2^{67} CP	2^{67}	ID	[MN17b]
	*	1	–	–	–	Statistics	
AES-PRF	1	*	2^{71}	2^{71} CP	2^{64}	ID	Sect. A.1
	1	*	$2^{122.49}$	$2^{103.34}$ KP	2^{96}	ZC	Sect. A.2
	2	*	2^{94}	2^{94} CP	2^{88}	ID	Sect. 4.1
	2	*	$2^{115.14}$	$2^{115.06}$ KP	2^{65}	ZC	Sect. 4.2
	*	3	$2^{84.96}$	$2^{84.96}$ KP	$2^{84.96}$	ZC distinguisher	Sect. 4.3
	*	4	$2^{96.95}$	$2^{96.95}$ KP	2^{64}	ZC distinguisher	Sect. 4.3
	s	$7-s$	2^{107}	2^{107} CP	2^{104}	MitM	Sect. 6.1
Dual-AES-PRF	*	1	2^{71}	2^{71} CP	2^{64}	ID	Sect. B.1
	*	1	$2^{122.49}$	$2^{103.34}$ KP	2^{96}	ZC	Sect. B.2
	*	2	2^{104}	2^{104} CP	2^{72}	ID	Sect. 5.1
	*	2	$2^{115.14}$	$2^{115.06}$ KP	2^{65}	ZC	Sect. 5.2
	3	*	2^{97}	2^{97} CP	2^{32}	Differential	Sect. B.4
	4	*	2^{121}	2^{121} CP	2^{8}	Differential	Sect. 5.3

The PRF, called AES-PRF, is as efficient as AES, and inherits the design principle of EDMD. The 128-bit key version of AES has 10 rounds, which is the focus of this paper[1], and we decompose it into the first s rounds and the last t rounds, where $s+t=10$. We then XOR the output state of the s-th round to the ciphertext, and this gives the output of AES-PRF, which we write AES-PRF$_{s,t}$ (See Fig 3). The primal proposal of [MN17b] is the case $(s,t)=(5,5)$, i.e., the balanced case, while [MN17b] also proposes unbalanced cases to evaluate the general security that AES-PRF offers.

The efficiency and cost-effectiveness of AES-PRF comes at the cost of provable security, i.e., the provable security result of EDMD no longer applies to AES-PRF, since the proof requires that the components are two independent PRPs. This implies that the security of AES-PRF relies on the evaluation by cryptanalysts, which we present in this paper.

In [MN17b], the initial security evaluation is presented, and it was shown that the cases $(s,t)=(1,9)$ and $(9,1)$ can be broken, while the security of the case $(s,t)=(2,8)$ is left as an open question. They also summarize generic attacks, where it can always break AES-PRF with 2^n query complexity, or if the query complexity is q, then the success probability of the distinguishing attack is $q^2/2^{2n}$ when $q<2^{n/2}$, and $O(q/2^{3n/2})$ if $q>2^{n/2}$ [MN17b]. They conjecture that AES-PRF cannot be distinguished from a random function significantly faster than by either bruteforcing the key or by the above mentioned generic attacks.

In this paper, we extensively analyze the security of AES-PRF$_{s,t}$. We also evaluate the security of the dual version of AES-PRF, which we write Dual-AES-PRF, that corresponds to the EDM counterpart of AES. From the provable security view point, EDM and EDMD have roughly the same security bound. More precisely, EDM is secure up to about $2^n/(67n)$ query complexity, and EDMD is secure up to about $2^n/67$ query complexity [MN17b]. The effect of the slight difference of the security bound is unknown, and it would be therefore interesting to see the security of both AES-PRF and Dual-AES-PRF from the cryptanalytic perspective. Dual-AES-PRF is depicted in Fig. 4.

We consider a set of rich cryptanalytic techniques that we find to be effective on these PRFs, and we apply an impossible differential attack [BBS99, Knu98], zero correlation

[1]There are also 192-bit and 256-bit key versions [MN17b].

linear attack [BLNW12, BR14], traditional differential attack [BS90], zero correlation linear distinguishing attack [BLNW12, BR14] and a meet-in-the-middle attack [DS08, DKS10, DFJ13]. See Table 1 for the summary of our results. These results improve the initial security evaluation by the designers in various ways, and significantly improve the insight of their security. From these results, our observations can be summarized as follows.

- Our results indicate that the security of AES-PRF is higher than Dual-AES-PRF from the applicability of differential attacks. This is consistent with the rationale discussed in [MN17b] for the preference of EDMD over EDM for the base scheme of AES-PRF.
- In terms of the number of rounds of the first part (s rounds), both AES-PRF and Dual-AES-PRF have only one round as the security margin.
- The balanced case $(s, t) = (5, 5)$ is certainly a natural choice of the design. However, our results indicate that $(s, t) = (4, 6)$ for AES-PRF is potential to be more secure, since the margin with respect to the attacked rounds becomes larger.

This paper is organized as follows. In Sect. 2, we describe AES-PRF and Dual-AES-PRF as well as the underlying block cipher. In Sect. 3, we present an overview of our results with the cryptanalysis techniques we use. Then, Sects. 4, 5 and 6 are the core of our paper where we detail our attacks against AES-PRF, Dual-AES-PRF and round-reduced AES-PRF, respectively. We conclude the paper in Sect. 7. Many attacks are also described in the Supplemental Material A and B.

2 AES-PRF and Its Dual

2.1 Description of AES

AES is the most common block cipher whose block length is 128 bits. AES accepts 128, 192 and 256-bit secret keys, and each is referred to as AES-128, AES-192 and AES-256, respectively. In this paper, we focus on the analysis of PRFs instantiated with AES-128.

0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15

Figure 2: AES byte order.

The internal state of AES-128 is represented as a 4×4 matrix whose elements take a 1-byte value, and we refer to a particular byte of the internal state x by $x[i]$, as depicted in Fig. 2. We write $x[i_1, i_2, \ldots, i_m]$ to denote m bytes with position $i_1, i_2, \ldots, i_m$, and $x[i : j]$ simply denotes consecutive bytes as positions between i and j.

The round function updates the state by applying four basic transformations: SubBytes (SB), ShiftRows (SR), MixColumns (MC) and AddRoundKey (AK). SubBytes is a nonlinear byte-wise substitution that applies an S-box to every byte of internal state. ShiftRows is a rotation of i-th row by i bytes to the left, where i starts from 0. MixColumns is a linear transformation that applies on each column by multiplying an invertible 4×4 matrix. AddRoundKey is an exclusive-or of internal state with round key. AES-128 iterates the round function 10 times, where an additional whitening key is XORed before the first round, and MixColumns is omitted in the last round.

In this paper, we use the following notation: x_i^I denotes the input of the round i, while x_i^S, x_i^R, x_i^M and x_i^O denote the intermediate values after the application of SubBytes, ShiftRows, MixColumns and AddRoundKey operations of round i, respectively. Then, we

have $x_{i-1}^O = x_i^I$ for $i \geq 2$. The i-th round key is denoted as K_i, and the initial whitening subkey is K_0. In some cases, we interchange the order of the MixColumns and AddRoundKey operations since these operations are linear. We denote the equivalent subkey by EK_i, that is $EK_i = MC^{-1}(K_i)$, and x_i^E represents the intermediate value after the application of AddRoundKey with equivalent subkey.

The key schedule works as follows. The 128-bit master key is divided into four 32-bit words $(W[0], W[1], W[2], W[3])$. Then, $W[i]$ for $i \geq 4$ is computed as

$$W[i] = \begin{cases} W[i-4] \oplus SB(RotByte(W[i-1])) \oplus Rcon[i/8] & i \equiv 0 \bmod 4, \\ W[i-4] \oplus W[i-1] & \text{otherwise,} \end{cases}$$

where $RotByte$ is a one byte rotation to the left, and $Rcon$ denotes the round-dependent constant. The i-th round key is $(W[4i], W[4i+1], W[4i+2], W[4i+1])$.

2.2 AES-PRF and Dual-AES-PRF

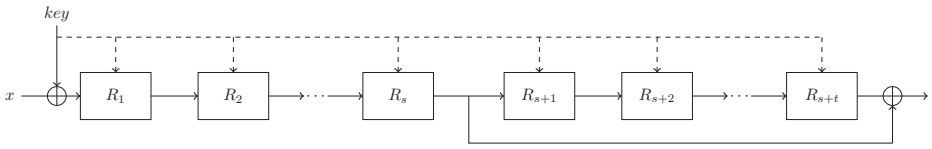


Figure 3: AES-PRF$_{s,t}$

The construction of AES-PRF is proposed by [MN17b]. The AES is decomposed into sub-blocks, and they are chained like the dual of encrypted Davies-Meyer construction. There are several settings for the decomposition, e.g., 10 rounds of AES-128 can be decomposed into the first s rounds and the last $t = 10 - s$ rounds, as depicted in Fig. 3.

Definition 1 (AES-PRF$_{s,t}$[2])**.** $(s+t)$-round AES is decomposed into the first s rounds and the last t rounds. The output of AES-PRF$_{s,t}$ is the XOR between the state encrypted by s-round AES and the state encrypted by $(s+t)$-round AES.

Unless otherwise stated, AES-PRF$_{s,t}$ adopts AES-128 with $s + t = 10$ rounds or its round-reduced variant when $s + t < 10$. The construction with $s = 0$ is equivalent to the Davies-Meyer construction, and that with $t = 0$ is clearly insecure.

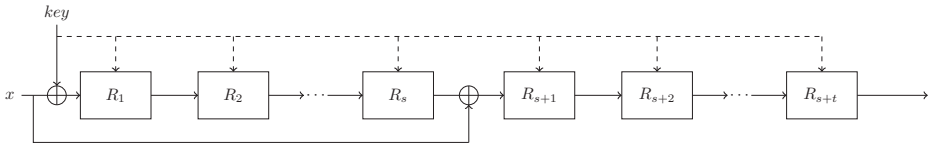
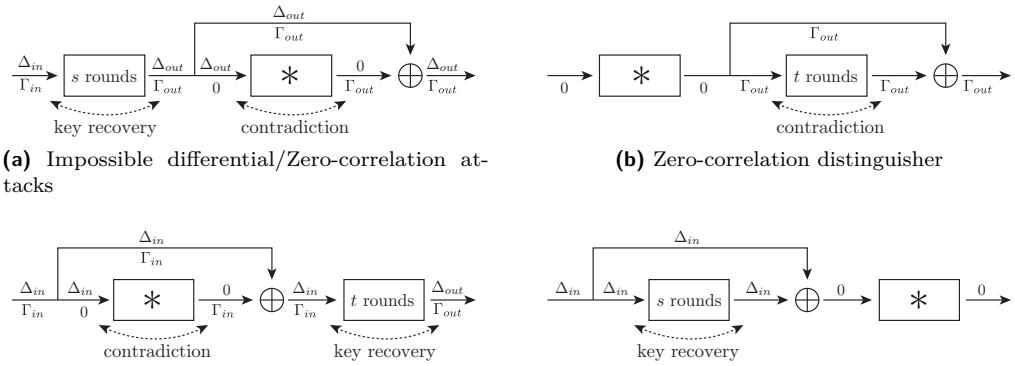


Figure 4: Dual-AES-PRF$_{s,t}$

It is natural to consider the dual of AES-PRF depicted in Fig. 4, and we call it Dual-AES-PRF. Unlike AES-PRF, in Dual-AES-PRF, the plaintext is used as the feed-forward instead of the intermediate state. Similarly to AES-PRF$_{s,t}$, when the first sub-block has s rounds and the last sub-block has t rounds, we call it Dual-AES-PRF$_{s,t}$. The construction with $s = 0$ is obviously insecure, and that with $t = 0$ is exactly the same as the Davies-Meyer construction.

Therefore, s can be chosen from 1 to 9 for both AES-PRF and Dual-AES-PRF. The designers also showed a few attacks on AES-PRF, where both $s = 1$ and $t = 1$ are broken.

[2]This notation is different from that in [MN17b], where AES-PRF$_s$ is used instead of AES-PRF$_{s,10-s}$.



(a) Impossible differential/Zero-correlation attacks

(b) Zero-correlation distinguisher

(c) Impossible differential/Zero-correlation attacks

(d) Differential attack

Figure 5: Summary of attacks, where either of the first or last sub-block can be regarded a random permutation.

We note that, as noted in [MN17b], Dual-AES-PRF is exposed to some risks that the adversary has control over the intermediate state.

3 Overview of Our Attacks

In this paper, we show various types of attack against AES-PRF and Dual-AES-PRF. Before we present the details of our attacks, we first summarize the overview to help readers to see the relationship among these attacks. Our attacks can be separated into two categories. The first one is that either of the first or the last sub-block can be regarded as a random permutation, and the other is that the number of rounds in both sub-blocks is restricted. We show five different types of attack: impossible differential attack, zero correlation linear attack, differential attack, zero correlation linear distinguisher and meet-in-the-middle attack. The first four attacks belong to the first category, and the meet-in-the-middle attack belongs to the second category. We present in Fig. 5 the summary of the attacks in the first category.

Impossible differential attack. An impossible differential attack was proposed independently by Biham et al. [BBS99] and Knudsen [Knu98]. Incorrect keys are discarded by using differentials that never occur in real ciphers. See Figs. 5a and 5c. When the sub-block with the feed-forward structure is ideal, we can construct a very simple impossible differential as the input has non-zero differences but the difference of the output before XOR is 0. In other words, if the output after XORing the feed-forward value has the same difference as the input, it is an impossible differential. Our goal is to recover the secret key in the other sub-block. As a result, we can attack both AES-PRF with $s \leq 2$ and Dual-AES-PRF with $t \leq 2$. The designers of AES-PRF also showed the attack against AES-PRF with $s = 1$, but the attack against AES-PRF with $s = 2$ was left as an open problem [MN17b]. Therefore, we solve this open problem. Details are presented in Sects. 4.1 and 5.1.

Zero-correlation linear attack. A zero-correlation linear attack was successfully used by Bogdanov and Rijmen [BLNW12, BR14]. The correct key is recovered by using linear approximations that hold for exactly 50% of the input values. The attack outline is similar to the impossible differential attack above (see Figs. 5a and 5c again). Namely, if the output of the sub-block with the feed-forward structure has the same linear mask as the input, it is zero-correlation linear. Our goal is to recover the secret key in the other sub-block.

Similarly to the impossible differential, we can attack both AES-PRF with $s \leq 2$ and Dual-AES-PRF with $t \leq 2$. Details are presented in Sects. 4.2 and 5.2.

Differential attack. The most simple differential attack exploits differentials that hold with a high probability, but our differential attack exploits the differential that holds with probability 1. As the designers of AES-PRF claimed, Dual-AES-PRF has vulnerability where the intermediate state can be controlled by observing the collision of the output. If the output of Dual-AES-PRF collides, we know the difference of the intermediate state always coincides with the difference of the input (see Fig. 5d). Therefore, we can recover the secret key by using such a differential that holds with probability 1. As a result, we can attack only Dual-AES-PRF when $s \leq 4$. Note that this attack cannot be applied to AES-PRF because we cannot control the difference of the intermediate state. Details are presented in Sect. 5.2.

Zero-correlation linear distinguisher. One of the reasons that the designers of AES-PRF chose the EDMD construction instead of the EDM construction is the vulnerability described above. Namely, we cannot control the difference of the intermediate state of the AES-PRF. However, we show that it is possible to control the linear mask very well thanks to the duality [Mat94]. See Fig. 5b. When the input linear mask is zero and the output linear mask is non-zero, the linear masks for the input and output of the last sub-block are the same as the output of the linear mask. Therefore, if chosen linear masks are zero correlation for the last sub-block, we obtain a zero-correlation linear distinguisher. We can construct such a linear mask up to $t \leq 4$. As a result, AES-PRF with $t \leq 4$ is also vulnerable similar to Dual-AES-PRF. On the other hand, this attack only brings the distinguisher, and it is left as an open problem to recover the secret key by exploiting this idea. Details are presented in Sect. 4.3.

Meet-in-the-Middle attack. AES-PRF relies on AES as the underlying block cipher. As the best known results against AES are based on meet-in-the-middle attacks, it makes sense to study how this cryptanalysis technique applies to AES-PRF. As a result, we can attack all variants of AES-PRF reduced to 7 rounds. On the other hand, Dual-AES-PRF seems to provide more resistance against such attacks and we can only break few variants. Surprisingly, unbalanced variants are the ones offering the best security. The details of our attacks are presented in Sect. 6.1.

4 Attacks on AES-PRF

4.1 Impossible Differential

We show impossible differential attacks against AES-PRF$_{s,t}$ with $s = 1$ or 2, where Fig. 5a illustrates the impossible differential attack for AES-PRF$_{s,t}$. In this subsection, we focus on AES-PRF$_{2,8}$ because AES-PRF$_{1,9}$ is clearly less secure than AES-PRF$_{2,8}$. Please refer to Supplemental Material A.1 for the attack against AES-PRF$_{1,9}$.

4.1.1 Data Requirement for Impossible Differential Attack

Before explaining the detail procedure to attack AES-PRF by using the impossible differential, we first introduce a formula to estimate the data requirement.

The impossible differential attack exploits differentials that never occur in real ciphers. Assume that each pair $((P, C), (P', C'))$ can reject 2^σ keys and 2^τ is the size of the targeted key space, the probability that an incorrect key is rejected by one pair is $2^{\sigma-\tau}$. The average

number of pairs N required to be left with at most 2^α key candidates is given by the following formula [BBS99, BNS14]:

$$2^\alpha \geq 2^\tau \left(1 - 2^{\sigma-\tau}\right)^N \approx 2^\tau e^{-N/2^{\tau-\sigma}}.$$

This inequality can be rewritten as:

$$N \geq 2^{\tau-\sigma} \times \frac{\tau - \alpha}{\log_2 e}. \tag{1}$$

4.1.2 Property of AES S-Box

We exploit the property of the differential distribution table (DDT) of AES S-box to recover the secret key.

Property 1. For a given input difference ΔX, let us consider the output difference ΔY. For 129/256 such pairs, the differential transition is impossible. For 126/256 such pairs, there are two ordered pairs, i.e., $S(X) \oplus S(X \oplus \Delta X) = \Delta Y$ and $S(X \oplus \Delta X) \oplus S(X) = \Delta Y$. And for the remaining 1/256 pair, there are four ordered pairs.

This property implies that pairs of input/output values of the AES S-box are immediately recovered once a pair of input/output differences is given. Moreover, the number of recovered values is 1 in average because $0 \times 129/256 + 2 \times 126/256 + 4 \times 1/256 = 1$. The key recovery attack based on this property has been applied to AES [BDD+12], and we also exploit this property to recover the secret key.

4.1.3 Impossible Differential Attack for AES-PRF$_{2,8}$

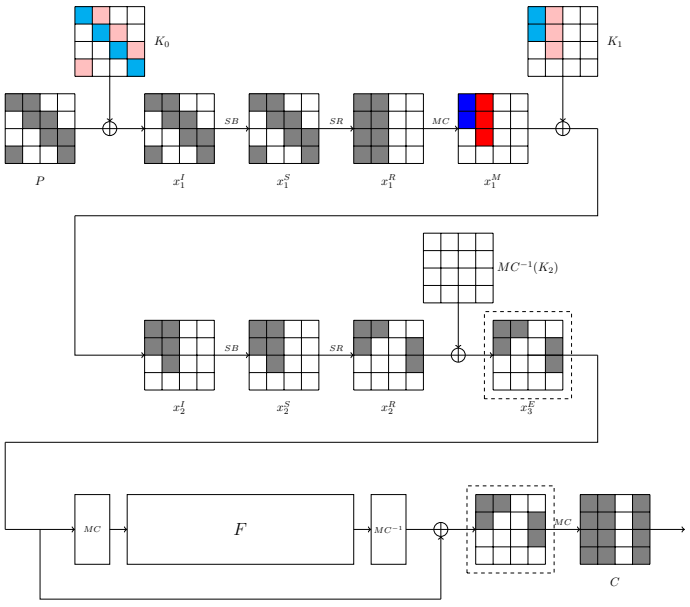


Figure 6: Analysis of AES-PRF$_{2,8}$.

It is easy to check that AES-PRF$_{2,8}$ is equivalent to the keyed function depicted in Fig. 6, that is, for every plaintext P, the function presented in Fig. 6 always outputs AES-PRF$_{2,8}(P)$. Before the analysis, we emphasize that the following contents should be read with the company of Fig. 6.

Our attack targets 11 subkey bytes: $K_0[0], K_0[3:5], K_0[9:10], K_0[14:15], K_1[0:1]$ and $K_1[6]$. Before the attack, we store the set $\mathbb{C}$ of all 2^{88} possible values for the 11 subkey bytes in memory, and incorrect keys will be removed from $\mathbb{C}$ based on impossible differential attack shown in Algorithm 1.

Assume that we have a pair (P, P') of plaintexts with difference ΔP. Let C and C' be the corresponding ciphertexts such that 11 bytes of $MC^{-1}(\Delta C)$ are 0 as depicted in Fig. 6. Then, any key guess under which the input difference of $MC^{-1} \circ F \circ MC(\cdot)$ is also $MC^{-1}(\Delta C)$ must be incorrect, since this leads to an impossible differential $MC^{-1}(\Delta C) \nrightarrow 0$ of the permutation $MC^{-1} \circ F \circ MC(\cdot)$.

Prepare 2^{64} plaintexts traversing all the 2^{64} possibilities of the 8 bytes marked with ■, while the remaining bytes are fixed to constants. These 2^{64} plaintexts forms a structure from which we obtain

$$\frac{2^{64} \times (2^{64} - 1)}{2} \times 2^{-88} \approx 2^{39}$$

pairs of plaintexts with 11 bytes of $MC^{-1}(\Delta C)$ being 0. In practice, such pairs can be identified with the following approach. We encrypt the 2^{64} plaintexts and insert them into a hash table $\mathbb{H}$ according to $MC^{-1}(\Delta C)$. Good pairs can be created in those slots of the hash table with more than one elements.

For each such pair (good pair), we guess the 2 bytes of Δx_1^M marked with ■. For any of the 2^{16} possible guesses, we can get corresponding $x_1^I[0, 5, 10, 15]$ and $x_1^S[0, 5, 10, 15]$ from Property 1. Since the plaintexts are known, we can derive 4 bytes of the subkey K_0 marked by ■. At this point, we can compute $x_1^M[0:1]$ and $\Delta x_2^I[0:1] (= \Delta x_1^M[0:1])$. Recall that keys satisfying $\Delta x_2^S = SR^{-1}(MC^{-1}(\Delta C))$ are incorrect, and we can get such $x_2^I[0:1]$ from Property 1. Since both $x_2^I[0:1]$ and $x_1^M[0:1]$ are known, we can derive the 2 bytes of K_1 marked by ■. We store the 2^{16} possible subkey guesses in a hash table $\mathbb{G}$ indexed by $(K_1[0], K_0[5] + K_1[1])$. Similarly, we guess the 3 red bytes ■ of Δx_1^M, from which the 7 pink bytes ■ of K_0 and K_1 can be derived.

Since the subkeys must satisfy the following equations extracted from the key schedule algorithm of AES,

$$\begin{cases} K_1[4] + K_0[4] = K_1[0] \\ K_1[5] = K_0[5] + K_1[1] \end{cases}$$

whose probability is 2^{-16}, for each of the 2^{24} guesses of $(K_0[3], K_0[4], K_0[9], K_0[14], K_1[4:6])$, we check the hash table $\mathbb{G}$ at the pin $(K_1[4] + K_0[4], K_1[5])$. If it is empty, we discard the guess. Otherwise, we remove the subkey guesses produced by combining the current guess and the guesses in the hash table $\mathbb{G}$ indexed by $(K_1[4] + K_0[4], K_1[5])$ from the set $\mathbb{C}$. Note that given one good pair approximately $2^{16} \times 2^{24} \times 2^{-16} = 2^{24}$ keys are rejected.

Complexity Analysis. In this attack, the targeted 11 key bytes are $K_0[0], K_0[3:5], K_0[9:10], K_0[14:15], K_1[0:1]$ and $K_1[6]$. Hence, we have $(\tau, \sigma) = (88, 24)$. We will collect enough data such that only 2^{50} candidates remain in $\mathbb{C}$, and therefore the whole key can be recovered with complexity $2^{50} \times 2^{128-11\times8} = 2^{90}$ after the analysis. According to equation (1), we need $2^{88-24} \times \frac{88-50}{\log_2 e} < 2^{69}$ good pairs, which demands $2^{69}/2^{39} = 2^{30}$ structures. In summary, the attack requires $2^{64+30} = 2^{94}$ chosen plaintexts, $2^{30} \times (2^{64} + 2^{39} \times (2^{16} + 2^{24})) \approx 2^{94}$ computations, and 2^{88} memory. This attack solves the open problem proposed by Mennink and Neves (see Sect. 3.3.3 of [MN17b]).

4.2 Zero-Correlation Linear

Figure 5a illustrates the zero-correlation linear attack for AES-PRF$_{s,t}$. Similarly to the impossible differential attack, we focus on AES-PRF$_{2,8}$, and please refer to Supplemental

Algorithm 1: Impossible differential attack on AES-PRF$_{2,8}$ with one structure

Ask for the encryption of one structure of 2^{64} plaintexts P of the form indicated by Fig. 6, and store them in a hash table $\mathbb{H}$ according to the value of $MC^{-1}(\Delta C)[2, 3, 5:12, 15]$.

Store the set of 2^{88} possible values for the 11 subkey bytes $K_0[0]$, $K_0[3:5]$, $K_0[9:10]$, $K_0[14:15]$, $K_1[0:1]$ and $K_1[6]$ in $\mathbb{C}$.

for each pair $((P, P'), (C, C'))$ in $\mathbb{H}$ with $MC^{-1}(\Delta C)[2, 3, 5:12, 15] = 0$ **do**
  for *every of the* 2^{16} *values of* Δx_1^M *marked with* ■ **do**
    Derive $K_0[0, 5, 10, 15]$ (marked by ■) from $\Delta P[0, 5, 10, 15]$ and $\Delta x_1^S[0, 5, 10, 15]$
    Derive $K_1[0, 1]$ (marked by ■) from $\Delta x_2^I[0, 1]$ and $\Delta x_2 = SR^{-1}(MC^{-1})(\Delta C)$
    Store the derived $K_0[0, 5, 10, 15]$ and $K_1[0, 1]$ in a hash table $\mathbb{G}$ indexed by $(K_1[0], K_0[5] + K_1[1])$
  end
  for *every of the* 2^{24} *values of* Δx_1^M *marked with* ■ **do**
    Derive the 7 bytes of K_0 and K_1 marked by ■
    if *The slot of* $\mathbb{G}$ *indexed by* $(K_1[4] + K_0[4], K_1[5])$ *is empty* **then**
      Discard the derived key guess
    else
      Remove the key values derived by combining the 7-byte value and the guesses in the hash table $\mathbb{G}$ indexed by $(K_1[4] + K_0[4], K_1[5])$ from $\mathbb{C}$
    end
  end
end

Material A.2 for the attack against AES-PRF$_{1,9}$.

4.2.1 Data Requirement for Zero-Correlation Linear Attack

In this section, we briefly introduce the data complexity of zero-correlation linear attack. For more information, please refer to [BLNW12, BN17, SCW17].

Let the adversary be given N plaintext-ciphertext pairs and ℓ non-trivial zero-correlation linear approximations for an n-bit block cipher. For each of the ℓ given approximations, the adversary computes the number $V[i]$ of times the linear approximations are fulfilled on N plaintext, $i \in \{1, 2, \ldots, \ell\}$. Each $V[i]$ suggests an empirical correlation value $\hat{c}_i = 2\frac{V[i]}{N} - 1$. Then, the adversary evaluates the statistic:

$$T = N \sum_{i=1}^{\ell} {\hat{c}_i}^2 = N \sum_{i=1}^{\ell} \left(2\frac{V[i]}{N} - 1\right)^2.$$

After setting the type-I error probability (the probability to miss the right key) to α_0, and the type-II error probability (the probability to accept a wrong key) to α_1, the number N

of known-plaintexts[3] in the attack is

$$N \approx \frac{2^n(\chi^{(\ell)}_{1-\alpha_0} - \chi^{(\ell)}_{\alpha_1})}{\chi^{(\ell)}_{\alpha_1}}, \tag{2}$$

where $\chi^{(\ell)}_{1-\alpha_0}$ and $\chi^{(\ell)}_{\alpha_1}$ are the respective quantiles of the χ^2-distribution with ℓ degrees of freedom evaluated on the points $1-\alpha_0$ and α_1, respectively.

4.2.2 Zero-Correlation Linear Attack for AES-PRF$_{2,8}$

The linear mask Γ that we use under this case is depicted in Fig. 7. The number of non-trivial zero-correlation linear approximations is $\ell = (2^8-1)^4$ under this setting.

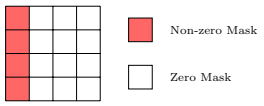


Figure 7: Linear mask Γ for AES-PRF$_{2,8}$.

Note that AES-PRF$_{2,8}$ is equivalent to the keyed function depicted in Fig. 8. The detailed key-recovery attack can be found in Algorithm 2.

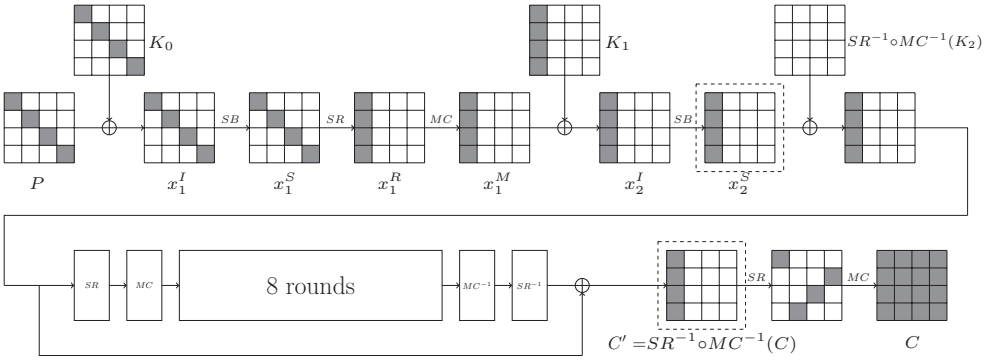


Figure 8: Zero-correlation attack for AES-PRF$_{2,8}$.

Complexity Analysis. In this attack, we use $\ell = (2^8-1)^4$ non-trivial zero-correlation linear approximations. We set the type-I error probability to $\alpha_0 = 2^{-4}$, and the type-II error probability to $\alpha_1 = 2^{-17}$. Thus, the data complexity is $2^{115.06}$ known-plaintexts[4]. Then, the key space is reduced to $2^{64-17} = 2^{47}$, and we exhaustively guess the remaining 64 bits. Therefore, the final exhaustive search requires the complexity of $2^{64+47} = 2^{111}$. Since V_1 and V_2 constitute the largest memory and the sizes of other counters are negligible, the memory complexity is roughly 2^{65}. The time complexity on Steps 5–21 of Algorithm 2 is $2^{106.00}$. Thus, the total computational time, which is dominated by the data collection part and the final exhaustive search phase, is about $2^{115.14}$.

[3] The data requirement will decrease if we use distinct known-plaintexts, which have been discussed in [BN17]. However, using distinct known-plaintext sampling method will increase the memory complexity. For almost all cases included in this paper, the data requirement computed by these two sampling methods only have slight difference. Thus, we only consider known-plaintext sampling method.

[4] All the computations involving the quantiles of the χ^2-distribution are conducted with Maplesoft (http://www.maplesoft.com/), and the computation is accurate to 500 decimal places.

Algorithm 2: Zero-correlation linear attack on AES-PRF$_{2,8}$

Allocate a global counter $V_1[z_1]$ for each of 2^{64} possible values of z_1.
for *each of N plaintext-ciphertext pairs* **do**
  Compute $z_1 = P[0,5,10,15] \| C'[0:3]$ and update V_1 as $V_1[z_1] \mathrel{+}= 1$.
end
Allocate counters V_i for each value of z_i, $2 \leqslant i \leqslant 6$.
for *each possible 32-bit subkey value* $K_0[0,5,10,15]$ **do**
  Reset V_2, compute $z_2 = x_1^M[0:3] \| C'[0:3]$ and update V_2 as $V_2[z_2] \mathrel{+}= V_1[z_1]$.
  for *each possible 8-bit subkey value* $K_1[0]$ **do**
    Reset V_3, compute $z_3 = x_1^M[1:3] \| C'[1:3] \| (C'[0] \oplus x_2^S[0])$ and update V_3 as $V_3[z_3] \mathrel{+}= V_2[z_2]$.
    for *each possible 8-bit subkey value* $K_1[1]$ **do**
      Reset V_4, compute $z_4 = x_1^M[2:3] \| C'[2:3] \| (C'[0:1] \oplus x_2^S[0:1])$ and update V_4 as $V_4[z_4] \mathrel{+}= V_3[z_3]$.
      for *each possible 8-bit subkey value* $K_1[2]$ **do**
        Reset V_5, compute $z_5 = x_1^M[3] \| C'[3] \| (C'[0:2] \oplus x_2^S[0:2])$ and update V_5 as $V_5[z_5] \mathrel{+}= V_4[z_4]$.
        for *each possible 8-bit subkey value* $K_1[3]$ **do**
          Reset V_6, compute $z_6 = C'[0:3] \oplus x_2^S[0:3]$ and update V_6 as $V_6[z_6] \mathrel{+}= V_5[z_5]$.
          Allocate a counter $V[z]$ for each of $(2^{8-1})^4$ zero-correlation linear approximations, and set it to zero.
          Update $V[z]$ by $V_6[z_6]$ and compute $T = N \sum_{i=1}^{\ell} \left(2\frac{V[i]}{N} - 1\right)^2$.
          if $T < \tau$ **then**
            The guessed key bytes constitute a possible subkey candidate.
            All master keys that are compatible with are tested exhaustively against a maximum of 2 plaintext-ciphertext pairs.
          end
        end
      end
    end
  end
end

4.3 Zero-Correlation Linear Distinguisher

For the distinguishing attack, we use the zero-correlation linear distinguisher illustrated in Fig. 5b, where we show distinguishing attacks against AES-PRF$_{7,3}$ and AES-PRF$_{6,4}$. The main observation is that $0 \xrightarrow{\text{AES-PRF}_{s,t}} \Gamma$ constitutes a zero-correlation linear approximation for AES-PRF$_{s,t}$, if $\Gamma \xrightarrow{\text{AES}_t} \Gamma$ is a zero-correlation linear approximation, where AES$_t$ denotes the last t rounds of AES. Since the input mask is zero, we only need to evaluate the zero-correlation property at the output. After determining Γ, we compute the χ^2-statistic with N plaintext-ciphertext pairs. Then, we can distinguish this construction with a random function by comparing the value of T with a predetermined threshold τ.

Let n be the block length and ℓ be the number of non-trivial zero-correlation linear approximations. Then, the data complexity of our distinguishing attack is roughly estimated as $\mathcal{O}(2^{n-\ell/2})$ [BW12, BLNW12]. In order to reduce the data complexity, we should increase the number of involved zero-correlation linear approximations. We only consider truncated

linear trails and exhaustively search all 2^{16} input/output patterns $\Gamma^p = (\gamma_0, \gamma_1, \ldots, \gamma_{15})$, $\gamma_i \in \{0, 1\}$.

4.3.1 Zero-Correlation Linear Distinguisher against AES-PRF$_{7,3}$

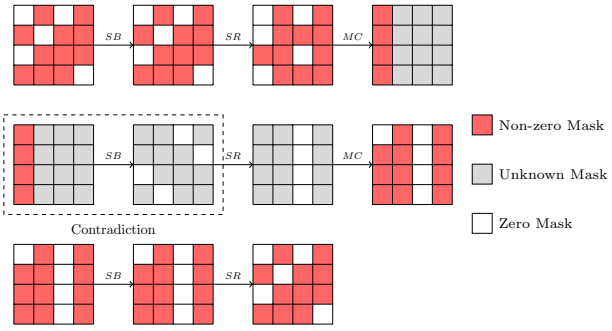


Figure 9: Zero-correlation distinguisher for AES$_3$.

The maximum Hamming weight of Γ^p such that $\Gamma^p \xrightarrow{\text{AES}_3} \Gamma^p$ constitutes a zero-correlation trail achieves 11. One family of zero-correlation linear approximations satisfying this case can be found in Fig. 9. The bytes marked with ■ denote bytes with non-zero linear masks. We omit AddRoundKey operation because it does not affect the propagation of linear mask.

Complexity Analysis. For the attack of AES-PRF$_{7,3}$, we use $\ell = (2^8 - 1)^{11}$ non-trivial zero-correlation linear approximations. We set the type-I error probability to $\alpha_0 = 2^{-2}$, and the type-II error probability to $\alpha_1 = 2^{-2}$. Thus, the data complexity is $2^{84.96}$ known-plaintexts. The time complexity is about $2^{84.96}$ because data collection phase dominates the time complexity. The memory complexity is roughly $2^{84.96}$.

4.3.2 Zero-Correlation Linear Distinguisher against AES-PRF$_{6,4}$

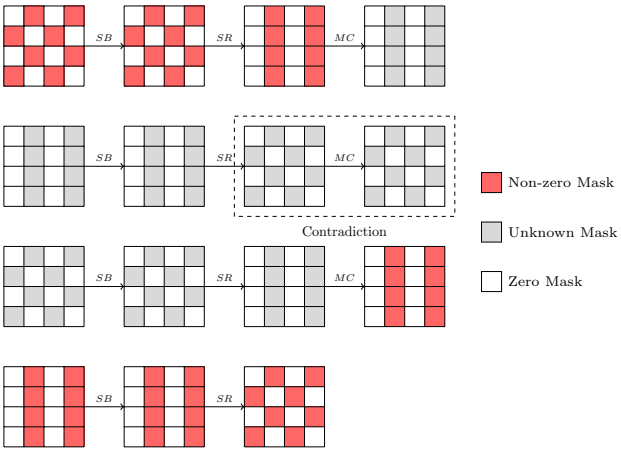


Figure 10: Zero-correlation distinguisher for AES$_4$.

The maximum Hamming weight of Γ^p such that $\Gamma^p \xrightarrow{\text{AES}_4} \Gamma^p$ constitutes a zero-correlation trail is 8. One family of zero-correlation linear approximations satisfying this

condition can be found in Fig. 10, where bytes marked with ■ denote bytes with non-zero linear masks.

Complexity Analysis. For the attack of AES-PRF$_{6,4}$, we use $\ell = (2^8-1)^8$ non-trivial zero-correlation linear approximations. We set the type-I error probability to $\alpha_0 = 2^{-2}$, and the type-II error probability to $\alpha_1 = 2^{-2}$. Thus, the data complexity is $2^{96.95}$ known-plaintexts. The time complexity is $2^{96.95}$, and the memory complexity is roughly 2^{64}.

5 Attacks on Dual-AES-PRF

5.1 Impossible Differential

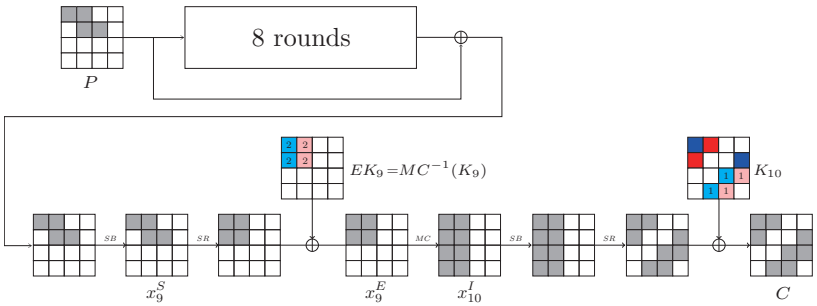


Figure 11: Overview of impossible differential against Dual-AES-PRF$_{8,2}$.

We show impossible differential attacks against Dual-AES-PRF$_{s,t}$ with $t = 1$ or 2. In this subsection, we focus on Dual-AES-PRF$_{8,2}$ because Dual-AES-PRF$_{9,1}$ is less secure than Dual-AES-PRF$_{8,2}$. Please refer to Supplemental Material B.1 for the attack against Dual-AES-PRF$_{1,9}$.

The overview of the impossible differential attack against Dual-AES-PRF$_{8,2}$ is shown in Fig. 11. We first prepare pairs (P, P') of plaintexts whose difference ΔP is active in 4 bytes $P[0, 4, 5, 9]$. Then, we pick pairs where the difference of corresponding ciphertexts is inactive in $C[2, 3, 5, 6, 8, 9, 12, 15]$. Any key guess under which the difference of the intermediate state is also ΔP must be incorrect, since this leads to an impossible differential $\Delta P \nrightarrow 0$ of the permutation.

Prepare 2^{32} plaintexts traversing all the 2^{32} possibilities of the 4 bytes marked with ■, while the remaining bytes are fixed to constants. From such 2^{32} plaintexts, the probability that the pair of ciphertexts has the difference above is $\frac{2^{32}\times(2^{32}-1)}{2} \times 2^{-64} \approx 2^{-1}$. For each such pair (good pair), we guess the 2 bytes of K_{10} marked with ■. For any of the 2^{16} possible guesses, we can derive $\Delta x_{10}^I[0, 1]$. Since $\Delta x_9^E[2] = \Delta x_9^E[3] = 0$, from the inverse MixColumns, we can solve the following equation system

$$\begin{cases} 13 \cdot \Delta x_{10}^I[0] + 09 \cdot \Delta x_{10}^I[1] + 14 \cdot \Delta x_{10}^I[2] + 11 \cdot \Delta x_{10}^I[3] = 0 \\ 11 \cdot \Delta x_{10}^I[0] + 13 \cdot \Delta x_{10}^I[1] + 09 \cdot \Delta x_{10}^I[2] + 14 \cdot \Delta x_{10}^I[3] = 0 \end{cases}$$

to get the values of $\Delta x_{10}^I[2]$ and $\Delta x_{10}^I[3]$. Combining the knowledge of $\Delta C = C \oplus C'$, $\Delta x_{10}^I[2, 3]$, we obtain $x_{10}^I[2, 3]$ from Property 1. Then, the key bytes marked with 1 can be derived. At this point, the main diagonal $K_{10}[0, 7, 10, 13]$ is known, and $x_9^E[0, 1]$ can be computed. Together with the knowledge of $\Delta P[0, 5]$ and $\Delta x_9^S[0, 5] (= \Delta x_9^E[0, 1])$, we obtain $x_9^S[0, 5]$ from Property 1. Then, the key bytes of EK_9 marked with 2 can be derived. Similarly, we continue the guess on the 2 bytes of K_{10} marked with ■, which leads to the determination of the bytes of K_{10} marked with 1 and the bytes of EK_9 marked with 2.

Therefore, for each good pair, approximately $2^{16} \times 2^{16} = 2^{32}$ keys are rejected. Moreover, the time complexity is 2^{32} for each good pair.

We use the same formula that is introduced in Sect. 4.1. Then, we have $(\tau, \sigma) = (96, 32)$. According to equation (1), we need $2^{96-32} \times \frac{96}{\log_2 e} < 2^{71}$ good pairs to remove all incorrect keys, which demands $2^{71}/2^{-1} = 2^{72}$ structures. Organising the attack with the *early abort technique* [LKKD08], we only need to store the 2^{71} pairs instead of 2^{96} keys. In summary, the attack requires $2^{32+72} = 2^{104}$ chosen plaintexts, $2^{32+72} = 2^{104}$ computations, and approximately 2^{72} 128-bit blocks in memory. The procedure of the key recovery attack is described in Algorithm 3.

Algorithm 3: Impossible differential attack on Dual-AES-PRF$_{8,2}$ with one structure

Allocate a memory $\mathbb{H}$ to store good pairs.
for *every of the* 2^{72} *distinct values of* $P[1:3, 6:8, 10:15]$ **do**
  Ask for 2^{32} plaintexts P of the form indicated by Fig. 11, and sort 2^{32} (P, C) according to the value of $C[0, 1, 4, 7, 10, 11, 13, 14]$.
  When we find pairs whose value of $C[0, 1, 4, 7, 10, 11, 13, 14]$ collides, the pair of the values $((P, P'), (C, C'))$ is stored in $\mathbb{H}$.
end
for *every of the* 2^{16} *values of* K_{10} *marked with* ■ **do**
  for *every of the* 2^{16} *values of* K_{10} *marked with* ■ **do**
    for each pair $((P, P'), (C, C'))$ in $\mathbb{H}$ **do**
      Derive $\Delta x_{10}^I[0, 1]$, and get $\Delta x_{10}^I[2, 3]$ from $\Delta x_9^E[2, 3] = 0$.
      Derive $K_{10}[7, 10]$ (marked by 1) from $\Delta x_{10}^I[2, 3]$ and $\Delta C[7, 10]$.
      Derive $EK_9[0, 1]$ (marked by 2) from $\Delta x_9^S[0, 5]$ and $\Delta x_9^E[0, 1]$.
      Similarly to the procedure above, derive $K_{10}[11, 14]$ (marked by 1) and $EK_9[4, 5]$ (marked by 2).
      Discard the derived impossible key bytes, $EK_9[0, 1, 4, 5]$ and $K_{10}[7, 10, 11, 14]$.
    end
    if *Key bytes that are not discarded remain* **then**
      Store the $(4 + 8)$-byte key as the candidate of the correct key.
    else
    end
  end
end

5.2 Zero-Correlation Linear

Figure 5c illustrates the zero-correlation linear attack for Dual-AES-PRF$_{s,t}$. The analysis in this subsection is restricted to $t = 2$, and please refer to Supplemental Material B.2 for the zero-correlation attack against Dual-AES-PRF$_{9,1}$.

The linear mask Γ that we use under this case is the same as the one given in Fig. 7. The number of non-trivial zero-correlation linear approximations is $\ell = (2^8 - 1)^4$ under this setting.

Note that Dual-AES-PRF$_{8,2}$ is equivalent to the keyed function depicted in Fig. 12. The key-recovery attack can be found in Algorithm 4.

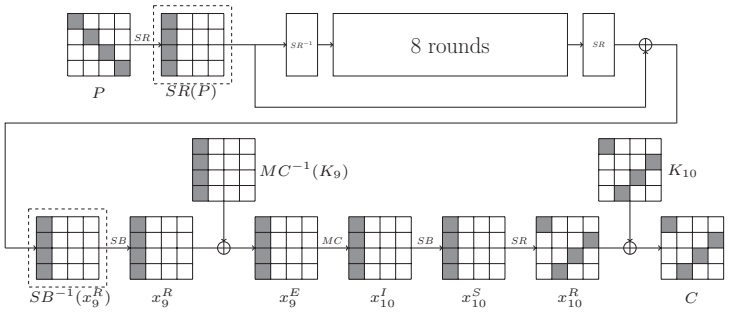


Figure 12: Zero-correlation attack for Dual-AES-PRF$_{8,2}$.

Algorithm 4: Zero-correlation linear attack on Dual-AES-PRF$_{8,2}$

Allocate a global counter $V_1[z_1]$ for each of 2^{64} possible values of z_1.
for *each of N plaintext-ciphertext pairs* **do**
  Compute $z_1 = SR(P)[0:3] \| C[0,7,10,13]$ and update V_1 as $V_1[z_1] \mathrel{+}= 1$.
end
Allocate counters V_i for each value of z_i, $2 \leqslant i \leqslant 6$.
for *each possible 32-bit subkey value* $K_{10}[0,7,10,13]$ **do**
  Reset V_2, compute $z_2 = SR(P)[0:3] \| x_9^E[0:3]$ and update V_2 as $V_2[z_2] \mathrel{+}= V_1[z_1]$.
  for *each possible 8-bit subkey value* $MC^{-1}(K_9)[0]$ **do**
    Reset V_3, compute $z_3 = SR(P)[1:3] \| x_9^E[1:3] \| \left(SR(P) \oplus SB^{-1}(x_9^R)\right)[0]$ and update V_3 as $V_3[z_3] \mathrel{+}= V_2[z_2]$.
    for *each possible 8-bit subkey value* $MC^{-1}(K_9)[1]$ **do**
      Reset V_4, compute $z_4 = SR(P)[2:3] \| x_9^E[2:3] \| \left(SR(P) \oplus SB^{-1}(x_9^R)\right)[0:1]$ and update V_4 as $V_4[z_4] \mathrel{+}= V_3[z_3]$.
      for *each possible 8-bit subkey value* $MC^{-1}(K_9)[2]$ **do**
        Reset V_5, compute $z_5 = SR(P)[3] \| x_9^E[3] \| \left(SR(P) \oplus SB^{-1}(x_9^R)\right)[0:2]$ and update V_5 as $V_5[z_5] \mathrel{+}= V_4[z_4]$.
        for *each possible 8-bit subkey value* $MC^{-1}(K_9)[3]$ **do**
          Reset V_6, compute $z_6 = SR(P)[0:3] \oplus SB^{-1}(x_9^R)[0:3]$ and update V_6 as $V_6[z_6] \mathrel{+}= V_5[z_5]$.
          Allocate a counter $V[z]$ for each of $(2^8-1)^4$ zero-correlation linear approximations, and set it to zero.
          Update $V[z]$ by $V_6[z_6]$ and compute $T = N \sum_{i=1}^{\ell} \left(2\frac{V[i]}{N} - 1\right)^2$.
          if $T < \tau$ **then**
            The guessed key bytes constitute a possible subkey candidate.
            All master keys that are compatible with are tested exhaustively against a maximum of 2 plaintext-ciphertext pairs.
          end
        end
      end
    end
  end
end

Complexity Analysis. In this attack, we use $\ell = (2^8 - 1)^4$ non-trivial zero-correlation linear approximations. We set the type-I error probability to $\alpha_0 = 2^{-4}$, and the type-II error probability to $\alpha_1 = 2^{-17}$. Thus, the data complexity is $2^{115.06}$ known-plaintexts. Since V_1 and V_2 constitute the largest memory and the sizes of other counters are negligible, the memory complexity is roughly 2^{65}. The time complexity on Steps 5–21 of Algorithm 4 is $2^{106.00}$. Thus, the total computational time, which is dominated by the data collection part and the final exhaustive search phase, is about $2^{115.14}$.

5.3 Differential

The differential attack that we utilize in this subsection is illustrated in Fig. 5d. We show the differential attack against Dual-AES-PRF$_{4,6}$ in this subsection. Please refer to Supplemental Material B.3 and B.4 for the differential attacks against Dual-AES-PRF$_{2,8}$ and Dual-AES-PRF$_{3,7}$, respectively.

5.3.1 Differential Attack for Dual-AES-PRF$_{4,6}$

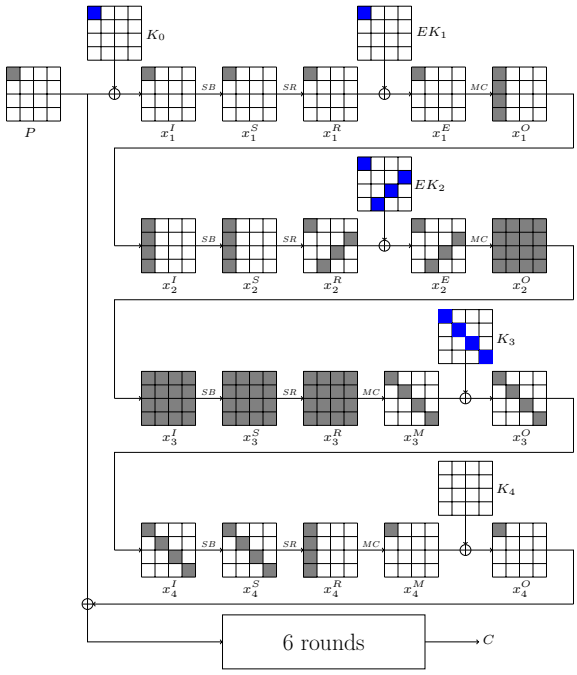


Figure 13: Key-recovery attack for Dual-AES-PRF$_{4,6}$.

An illustration for the key-recovery procedure can be found in Fig. 13. Independently of any probabilities, once the key is fixed, AES is a permutation, and then, Dual-AES-PRF$_{s,t}(x)$ = Dual-AES-PRF$_{s,t}(y)$ is equivalent to $\text{AES}_s(x) \oplus x = \text{AES}_s(y) \oplus y$, which can be rewritten as $x \oplus y = \text{AES}_s(x) \oplus \text{AES}_s(y)$. In our attack, we encrypt plaintexts with only one active byte. Hence, whenever a pair (x, y) collides, then $\text{AES}_4(x) \oplus \text{AES}_4(y)$ is equal to $x \oplus y$, so differences at input and output of 4-round AES have only one active byte (at the same position) and thus the trail depicted on Fig. 13 is followed with probability 1.

In order to obtain one collision pair at the output, we need to create 2^{128} pairs. Consider a structure of 2^8 plaintexts that the unique gray byte shown in Fig. 13 is active, while the remaining bytes in white are fixed to constants. From one structure of 2^8 plaintexts, we

are able to construct $2^8 \times (2^8 - 1)/2 \approx 2^{15}$ pairs. Thus, 2^{113} structures are required. If we find a collision at the output, we know the difference at x_4^O.

For the collision pair (P, P') satisfying the input difference, we enumerate all 2^{72} possible differences at x_1^S, x_2^S, and x_4^I. Then, all input and output differences of the SubBytes operations in the first four rounds are known. From Property 1, partial values of x_i^I and x_i^S $(1 \leq i \leq 4)$, whose positions correspond to all active bytes of the SubBytes operation, can be recovered. Then, 10-byte information of the subkey can be obtained, which are dyed in blue in Fig. 13. For each guess of the differences, we are able to retrieve 80-bit information of the subkeys, and all master keys that are compatible with are tested exhaustively against a maximum of two plaintext-ciphertext pairs. A detailed description can be found in Algorithm 5.

In total, the data complexity of this attack is 2^{121} chosen plaintexts. The time complexity is also 2^{121}. Since the input difference of the collision pair must follow the input difference of the distinguisher, we do not need to consider pairs constructed by choosing plaintexts from different structures. In other words, in the collision searching phase, each structure can be handled, independently. Thus, the memory complexity is 2^8.

Algorithm 5: Differential attack on Dual-AES-PRF$_{4,6}$ with one collision pair

Input: The collision pair, the input difference Δx_1^I, and the output difference Δx_4^O.

for *each of the* 2^8 *active values of* Δx_1^S **do**
  Deduce $x_1^I[0]$ and $x_1^S[0]$ from $\Delta x_1^I[0]$ and $\Delta x_1^S[0]$.
  Compute $K_0[0] = P[0] \oplus x_1^I[0]$.
  for *each of the* 2^{32} *active values of* Δx_2^S **do**
    Compute $\Delta x_2^I = MC \circ SR(\Delta x_1^S)$.
    Deduce $x_2^I[0:3]$ and $x_2^S[0:3]$ from $\Delta x_2^I[0:3]$ and $\Delta x_2^S[0:3]$.
    Compute $EK_1[0] = MC^{-1}(x_2^I[0:3])[0] \oplus x_1^S[0]$.
    for *each of the* 2^{32} *active values of* Δx_4^I **do**
      Compute $\Delta x_3^S = SR^{-1} \circ MC^{-1}(\Delta x_4^I)$ and $\Delta x_3^I = MC \circ SR(\Delta x_2^S)$.
      Deduce x_3^I and x_3^S from Δx_3^I and Δx_3^S.
      Compute $EK_2[0,7,10,13] = MC^{-1}(x_3^I)[0,7,10,13] \oplus x_2^S[0,3,2,1]$.
      Deduce $x_4^I[0,5,10,15]$ from $\Delta x_4^I[0,5,10,15]$ and $MC^{-1} \circ SR^{-1}(\Delta x_4^O)[0,5,10,15]$.
      Compute $K_3 = x_4^I[0,5,10,15] \oplus MC \circ SR(x_3^S)[0,5,10,15]$.
      All master keys that are compatible with the deduced bytes of K_0 - K_3 are tested exhaustively against a maximum of 2 plaintext-ciphertext pairs.
    end
  end
end

6 Attacks on Round-Reduced Versions of AES-PRF

In this section we describe key-recovery attacks against AES-PRF and Dual-AES-PRF when the number of rounds is reduced to 7. Indeed, the best known attacks against round-reduced versions of AES-128 are able to break up to 7 rounds and it is worth to show that using the feed-forward of an internal state does not increase the security.

6.1 Demirci-Selçuk attack against AES-PRF$_{3,4}$

In this section, we show how to apply a *Demirci-Selçuk* attack [DS08] against AES-PRF$_{3,4}$. Interested reader may refer to [DF16] for further details and improvements of this cryptanalysis technique.

We want to emphasize this is the first time that this cryptanalysis technique is applied to a primitive which is not a block cipher. Hence, beside its interest in understanding the security of AES-PRF, we believe this attack opens a new research line as future work may try to extend the application range of *Demirci-Selçuk* attacks.

6.1.1 4-round Distinguisher

Our attack against AES-PRF$_{3,4}$ relies on the exact same 4-round distinguisher than the original attack of Demerci-Selçuk. Denoting δ-set a collection of 256 plaintexts such that one byte is active and takes all the possible values while other ones are constant, we have the following property:

Lemma 1 (4-round distinguisher)**.** *Consider the encryption of a δ-set through four full AES rounds. For each of the 16 bytes of the state, the ordered sequence of 255 differences of that byte in the corresponding ciphertexts is fully determined by just 25 byte parameters. Consequently, for any fixed byte position, there are at most $(2^8)^{25} = 2^{200}$ possible sequences when we consider all the possible choices of keys and δ-sets (out of the $(2^8)^{255} = 2^{2040}$ theoretically possible 255-byte sequences).*

The proof of this lemma is straightforward and can be found in [DF13].

6.1.2 Differential Enumeration Technique

The 4-round distinguisher above cannot be used to attack AES-128 since there are too many sequences to store. However, in 2010 Dunkelman *et al.* proposed a powerful technique to reduce the memory requirement of the attack [DKS10]. This technique, later improved by Derbez *et al.* in [DFJ13], asks to store only the sequences built from a δ-set containing a message P that belongs to a pair (P, P') following a well-chosen truncated differential characteristic, depicted on Fig. 14.

The 4-round distinguisher is between rounds 2 and 6. Given a δ-set such that coloured byte of x_2^I takes all the possible values, the sequence of differences in coloured byte of x_6^R is fully determined by the 25 coloured bytes of x_3^I, x_4^I, x_5^I and x_6^I. Indeed, if one knows the value of those bytes for one message of the δ-set, he can propagate the differences from x_2^R to x_6^R and hence build the sequence. However, if the message belongs to a pair following the truncated differential characteristic of Fig. 14, the 25 coloured bytes can assume only $(2^8)^{11} = 2^{88}$ values. Indeed, it is enough to know the differences in coloured bytes of x_2^R, x_3^R, x_5^I, x_6^I and x_6^R to deduce the required bytes, since differences before and after the S-box is known for all of them.

6.1.3 Attack against AES-PRF$_{3,4}$

The attack is quite similar to the original attack against AES-128. As it, we start by computing and storing in a hash table all the 2^{88} sequences constructed by following the differential enumeration technique. This is the offline phase and this step has a time complexity of $2^{88} \times 2^8 = 2^{96}$ partial encryptions and $2^{88} \times 2^8 = 2^{96}$ bytes of memory are required.

Then, in the online phase, one has to find a pair following the truncated differential characteristic. Classically, we start by asking for a structure of 2^{32} messages with one

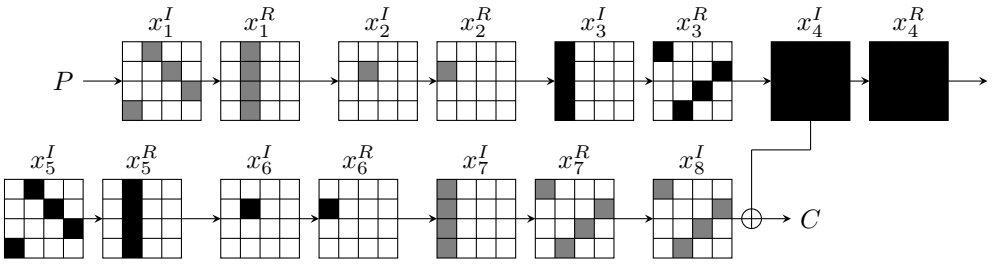


Figure 14: Truncated differential characteristic on AES-PRF$_{3,4}$. Differences in white squares are null.

diagonal active and other bytes constant. Then, if a pair follows the characteristic, difference of ciphertexts belongs to a subspace of dimension 8 because difference in state x_4^I belongs to a subspace of 4, as well as difference in state x_8^I. Hence, we sort the 2^{32} messages according to the 8 constant (linear combinations of) bytes to identify pairs that may follow the truncated differential. Hopefully, for each such pair, it is straightforward to recover Δx_8^I and Δx_4^I assuming the pair follows the characteristic.

Then one guesses Δx_2^I and Δx_2^R to recover the actual value of coloured bytes of x_1^I, x_1^R, x_2^I, x_2^R, x_3^I and x_3^R. Similarly, it is enough to guess Δx_6^R to recover the actual value of x_7^I and x_7^R. At this step, we have a message P_0 for which we know the actual value of grey bytes as well as black bytes of x_3^I and x_3^R. Hence, we now have to compute a δ-set from this message, compute the corresponding sequence of differences and check whether it is in the table or not. If the sequence does not belong to the table, we know with probability 1 that either the pair did not follow the characteristic or we made one wrong guess. In the opposite case, if the sequence is in the table, with very high probability the pair follows the truncated characteristic and the guesses are correct. To compute the sequence, we propagate difference on x_2^I to both x_4^I and the plaintext. Then, using the corresponding ciphertext and Δx_4^I we compute Δx_8^I and propagate it to x_6^R.

Complexities. Time and memory complexities both are around 2^{96}. In the online phase, to get a pair following the truncated differential characteristic, we need about 2^{81} structures of 2^{32} plaintexts. Indeed, each pair of such structure follows the characteristic with probability $2^{-24-120} = 2^{-144}$ and one structure contains $2^{32+31} = 2^{63}$ pairs. Hence, the data complexity of the attack is 2^{113} chosen plaintexts. In the first step, for each structure we sort the corresponding ciphertexts to identify pairs that may follow the characteristic. This step requires $2^{32} \times 16 = 2^{36}$ bytes of storage and its time complexity is around 2^{32}. Repeated for each structure, the time complexity of this step is around 2^{113}. Approximately $2^{81+63-64} = 2^{80}$ pairs pass this step. For each of them we then need to guess 2 differences and propagate 2^8 differences, leading to a complexity around $2^{80+16+8} = 2^{104}$ encryptions.

Key-recovery. At the end of the attack, we have a message for which we also know the right value of 4 bytes in state x_1^I. This directly leads to the knowledge of 4 key bytes. One can recover the remaining key bytes by exhaustive search.

Trade-off. It is possible to optimize the overall complexity of the attack by storing several tables. Indeed, with the same structure we have 4 choices for active byte of x_2^I as well as 16 choices for the actives byte of x_6^R. Hence, we can decrease the overall complexity to $2^{113-2-4} = 2^{107}$. It is worth noticing that the overall complexity of this attack is very close to 2^{100}, overall complexity of the original attack against 7-round AES-128.

Remarks. The main difference between attacking AES-PRF$_{3,4}$ and AES-128 is the dimension of the subspace in which ΔC belongs. Let denote by V_4 and V_8 the subspaces

in which Δx_4^I and Δx_8^I respectively belong. Note that whatever the position of the active byte on x_2^I as well as the position of the active byte on x_6^I, $V_4 \cap V_8 = \{0\}$ and thus we can directly deduce both Δx_4^I and Δx_8^I from ΔC. This is why the complexity of the attack is similar for both AES-PRF$_{3,4}$ and AES-128.

It is worth mentioning that if the last MixColumns operation is not omitted the result above does not hold. In that case, for some positions of the active bytes, $V_4 = V_8$ and thus one should make four extra guesses to identify Δx_4^I and Δx_8^I, increasing the time complexity by a factor 2^{32}.

6.1.4 Attack against Other Variants

This attack can be applied to AES-PRF$_{s,7-s}$ in straightforward way for all values of s. We only distinguish two cases: difference in state x_s^I will be computed from x_2^I for $s \leq 3$ and from x_6^I otherwise. For Dual-AES-PRF$_{s,7-s}$, surprisingly, it seems that we can only mount attacks with $s \in \{3, 4, 7\}$. Indeed, in other cases, the truncated differential characteristic used in the attack becomes impossible, making the differential enumeration technique not efficient enough to beat exhaustive search for 128-bit version.

7 Conclusions

In this paper, we performed an extensive security analysis of the pseudo-random function AES-PRF proposed by Mennink and Neves at FSE 2018. By applying several well-known cryptanalysis techniques to AES-PRF, we complemented the initial analysis provided by the designers with stronger attacks. Surprisingly, we found that the unbalanced version AES-PRF$_{4,6}$ seems to offer better security margins than the original design AES-PRF$_{5,5}$ according to the current results of our cryptanalysis. We also evaluated the dual version of AES-PRF and showed that this construction is a bit weaker, as expected by Mennink and Neves. Indeed, we were able to mount a key-recovery attack against Dual-AES-PRF$_{4,*}$, while we only found a distinguisher against AES-PRF$_{*,4}$. Finally, we also studied round-reduced versions of both AES-PRF and Dual-AES-PRF. The best known attacks against the underlying block cipher, AES-128, break up to 7 rounds and we wondered whether the PRF reduced to 7 rounds was secure or not. As a result, we found that all variants of AES-PRF reduced to 7 rounds can be broken with complexities similar to attacks against AES-128.

All in all, while our attacks only apply to the unbalanced or round-reduced versions of AES-PRF, and do not endanger the full design, they provide further insight into its security.

Finally, this paper focuses on the 128-bit key version of AES-PRF and Dual-AES-PRF, and we leave the analyses of 192-bit and 256-bit key versions as open questions.

Acknowledgements

The authors thank the anonymous FSE 2019 reviewers and Samuel Neves for careful reading and many helpful comments. We also thank the organizers of ASK 2017 (the 7th Asian Workshop on Symmetric Key Cryptography), where the collaboration was initiated. Patrick Derbez is supported by the French Agence Nationale de la Recherche through the CryptAudit project under Contract ANR-17-CE39-0003. Meiqin Wang and Ling Sun are partially supported by National Natural Science Foundation of China (Grant No. 61572293), National Basic Research Program of China (973 Program) (Grant No. 2013CB834205), Science and Technology on Communication Security Laboratory of China (Grant No. 9140c110207150c11050), Key Science Technology Project of Shandong Province (Grant No. 2015GGX101046), and Chinese Major Program of National Cryptography

Development Foundation (Grant No. MMJJ20170102). The work of Siwei Sun is supported by the National Natural Science Foundation of China (61772519), the Youth Innovation Promotion Association of Chinese Academy of Sciences, the Chinese Major Program of National Cryptography Development Foundation, and the Institute of Information Engineering, CAS (Grant No. Y7Z0341103).

References

[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.

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

[BDJR97] Mihir Bellare, Anand Desai, E. Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *FOCS*, pages 394–403. IEEE Computer Society, 1997.

[Ber97] Daniel J. Bernstein. SURF: simple unpredictable random function. https://cr.yp.to/papers.html#surf, 1997.

[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 Cryptology ePrint Archive*, 1999:24, 1999.

[BKR98] Mihir Bellare, Ted Krovetz, and Phillip Rogaway. Luby-Rackoff backwards: Increasing security by making block ciphers non-invertible. In *EUROCRYPT*, 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.

[BN17] Céline Blondeau and Kaisa Nyberg. Joint data and key distribution of simple, multiple, and multidimensional linear cryptanalysis test statistic and its impact to data complexity. *Des. Codes Cryptography*, 82(1-2):319–349, 2017.

[BNS14] Christina Boura, María Naya-Plasencia, and Valentin Suder. Scrutinizing and improving impossible differential attacks: Applications to CLEFIA, Camellia, LBlock and Simon. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 179–199. Springer, 2014.

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

[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.

[BW12] Andrey Bogdanov and Meiqin Wang. Zero correlation linear cryptanalysis with reduced data complexity. In *FSE 2012*, pages 29–48, 2012.

[CS16] Benoît Cogliati and Yannick Seurin. EWCDM: an efficient, beyond-birthday secure, nonce-misuse resistant MAC. In *CRYPTO (1)*, volume 9814 of *LNCS*, pages 121–149. Springer, 2016.

[DF13] Patrick Derbez and Pierre-Alain Fouque. Exhausting demirci-selçuk meet-in-the-middle attacks against reduced-round AES. In *FSE 2013*, pages 541–560, 2013.

[DF16] Patrick Derbez and Pierre-Alain Fouque. Automatic search of meet-in-the-middle and impossible differential attacks. In *CRYPTO 2016, Part II*, pages 157–184, 2016.

[DFJ13] Patrick Derbez, Pierre-Alain Fouque, and Jérémy Jean. Improved key recovery attacks on reduced-round AES in the single-key setting. In *EUROCRYPT 2013*, pages 371–387, 2013.

[DKS10] Orr Dunkelman, Nathan Keller, and Adi Shamir. Improved single-key attacks on 8-round AES-192 and AES-256. In *ASIACRYPT 2010*, pages 158–176, 2010.

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

[DS08] Hüseyin Demirci and Ali Aydin Selçuk. A meet-in-the-middle attack on 8-round AES. In *FSE 2008*, pages 116–126, 2008.

[Dwo01] Morris Dworkin. Recommendation for block cipher modes of operation, methods and techniques. NIST Special Publication 800-38A, 2001.

[Dwo07] Morris Dworkin. Recommendation for block cipher modes of operation: Galois/counter mode (GCM) and GMAC. NIST Special Publication 800-38D, 2007.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986.

[GGM18] Shoni Gilboa, Shay Gueron, and Ben Morris. How many queries are needed to distinguish a truncated random permutation from a random function? *J. Cryptology*, 31(1):162–171, 2018.

[HWKS98] Chris Hall, David A. Wagner, John Kelsey, and Bruce Schneier. Building PRFs from PRPs. In *CRYPTO*, volume 1462 of *LNCS*, pages 370–389. Springer, 1998.

[Knu98] Lars Ramkilde Knudsen. DEAL: A 128-bit block clipher. Technical report, Department of Informatics, University of Bergen, Norway, 1998.

[KR11] Ted Krovetz and Phillip Rogaway. The software performance of authenticated-encryption modes. In *FSE*, volume 6733 of *LNCS*, pages 306–327. Springer, 2011.

[LKKD08] Jiqiang Lu, Jongsung Kim, Nathan Keller, and Orr Dunkelman. Improving the efficiency of impossible differential cryptanalysis of reduced camellia and MISTY1. In *CT-RSA 2008*, pages 370–386, 2008.

[LR88] Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, 17(2):373–386, 1988.

[Luc00] Stefan Lucks. The sum of PRPs is a secure PRF. In *EUROCRYPT*, volume 1807 of *LNCS*, pages 470–484. Springer, 2000.

[Mat94] Mitsuru Matsui. On correlation between the order of S-boxes and the strength of DES. In Alfredo De Santis, editor, *EUROCRYPT '94*, volume 950 of *LNCS*, pages 366–375. Springer, 1994.

[MN17a] Bart Mennink and Samuel Neves. Encrypted davies-meyer and its dual: Towards optimal security using mirror theory. In *CRYPTO (3)*, volume 10403 of *LNCS*, pages 556–583. Springer, 2017.

[MN17b] Bart Mennink and Samuel Neves. Optimal PRFs from blockcipher designs. *IACR Trans. Symmetric Cryptol.*, 2017(3):228–252, 2017.

[MV04] David A. McGrew and John Viega. The security and performance of the galois/counter mode (GCM) of operation. In *INDOCRYPT*, volume 3348 of *LNCS*, pages 343–355. Springer, 2004.

[Pat08] Jacques Patarin. A proof of security in $O(2^n)$ for the xor of two random permutations. In *ICITS*, volume 5155 of *LNCS*, pages 232–248. Springer, 2008.

[Pat10] Jacques Patarin. Introduction to mirror theory: Analysis of systems of linear equalities and linear non equalities for cryptography. *IACR Cryptology ePrint Archive*, 2010:287, 2010.

[Pat13] Jacques Patarin. Security in $O(2^n)$ for the xor of two random permutations – proof with the standard H technique. *IACR Cryptology ePrint Archive*, 2013:368, 2013.

[SCW17] Ling Sun, Huaifeng Chen, and Meiqin Wang. Zero-correlation attacks: statistical models independent of the number of approximations. *Des. Codes Cryptography*, 10 2017.

A Attacks on AES-PRF

A.1 Impossible Differential Attack for $\text{AES-PRF}_{1,9}$

We hereby describe an attack on $\text{AES-PRF}_{1,9}$ based on impossible differential. Consider a structure of 2^{64} plaintexts such that the 8 gray bytes shown in Fig. 15 are active (taking all possible values), while the remaining bytes in white are fixed to constants. Wrong keys associated with this structure can be deleted from the set of key candidates with Algorithm 6.

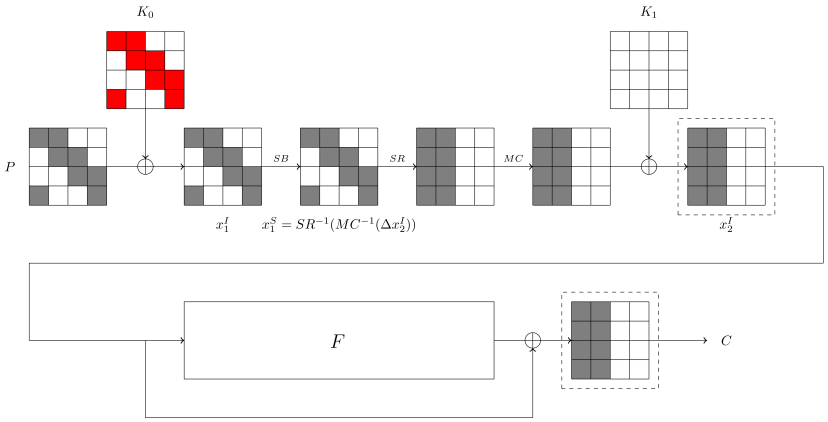


Figure 15: The attack on $\text{AES-PRF}_{1,9}$, where we only consider pairs with input and output difference patterns as depicted, and the red bytes in K_0 are the targeted partial subkey information.

Algorithm 6: Impossible differential attack on $\text{AES-PRF}_{1,9}$ with one structure

1 Ask for the encryption of one structure of 2^{64} plaintexts P of the form indicated by Fig. 15, and store them in a hash table according to the value of $C[8:15]$.

2 **for** each pair $((P, P'), (C, C'))$ in the hash table with $\Delta C[8:15] = 0$ **do**

3 Compute $t = SR^{-1}(MC^{-1}(\Delta C))$

4 Remove all key values from the set of key candidates for $K_0[0, 3, 4, 5, 9, 10, 14, 15]$ such that $SB(P \oplus K_0) \oplus SB(P' \oplus K_1) = t$

5 **end**

From one structure of 2^{64} plaintexts, we can obtain $2^{64} \times (2^{64} - 1)/2 \approx 2^{127}$ pairs, in which there are approximately $2^{127-64} = 2^{63}$ pairs whose ciphertext differences satisfy the output difference pattern shown in Fig. 15. Since only such pairs are used in Algorithm 6, we call them good pairs. For each good pair, we try to identify partial subkey values which lead the underlying differential characteristic to follow the pattern specified in Fig. 15 where $\Delta x_2^I = \Delta C$, and we remove the partial subkey values as such from the candidate key set, since $\Delta x_2^I = \Delta C$ impose an impossible differential over the last permutation (i.e., form nonzero input difference to zero output difference).

The input difference Δx_1^I and output difference Δx_1^S of the first S-box layer can be computed as $\Delta x_1^I = \Delta P$ and $\Delta x_1^S = SR^{-1}(MC^{-1}(\Delta x_2^I))$. Let S be the AES S-box. Then the expected number of solutions of the equation $S(a \oplus k) \oplus S(b \oplus k) = c$ in k for randomly chosen known values of a, b, and c is 1. Therefore, by setting $\Delta x_2^I = \Delta C$, we can identify one incorrect key guess in average from one good pair.

In our analysis, $\sigma = 0$ and $\tau = 8 \times 8 = 64$ (the 8 red bytes in K_0). According to equation (1), we need about $2^{64-0} \times 64/\log_2 e < 2^{70}$ good pairs to remove all incorrect candidate keys (only the right key survives). Thus, we have to repeat Algorithm 6 for $2^{70-63} = 2^7$ structures. Consequently, the attack requires $2^{64} \times 2^7 = 2^{71}$ chosen plaintexts, $2^{64} \times 2^7 = 2^{71}$ computations, and 2^{64} memory, while the attack of Mennink and Neves (see Sect. 3.3.2 of [MN17b]) requires 2^{67} chosen plaintexts, 2^{101} computations, and 2^{67} memory.

A.2 Zero-Correlation Attack for AES-PRF$_{1,9}$

The linear mask Γ we use in this case is depicted in Fig. 16. Thus, the number of non-trivial zero-correlation linear approximations is $\ell = (2^8 - 1)^7$ under this setting.

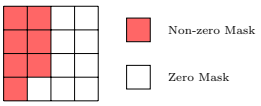


Figure 16: Linear mask Γ for AES-PRF$_{1,9}$.

Note that AES-PRF$_{1,9}$ is equivalent to the keyed function depicted in Fig. 17. The key-recovery attack can be found in Algorithm 7.

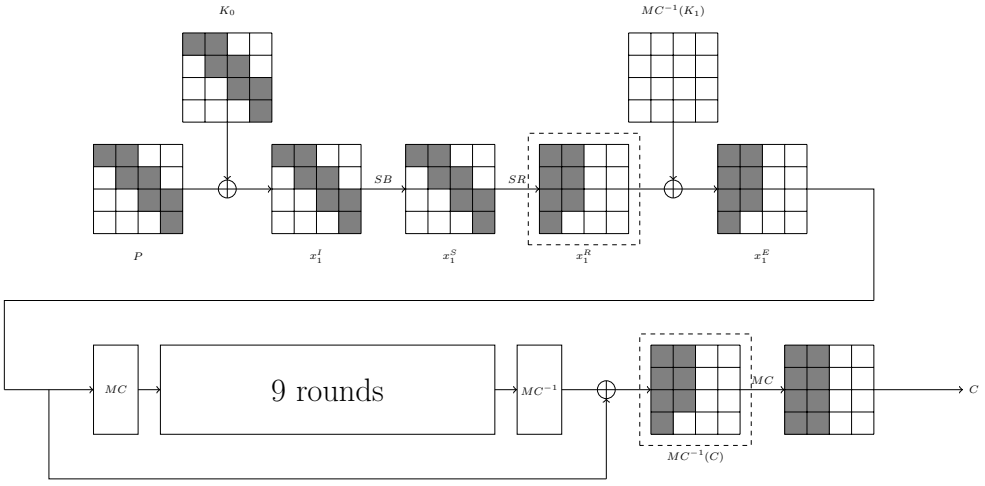


Figure 17: Zero-correlation attack for AES-PRF$_{1,9}$.

Complexity Analysis In this attack, we use $\ell = (2^8 - 1)^7$ non-trivial zero-correlation linear approximations. We set the type-I error probability to $\alpha_0 = 2^{-4}$, and the type-II error probability to $\alpha_1 = 2^{-26}$. Thus, the data complexity is $2^{103.34}$ known-plaintexts. Since V_1 is the largest memory and the sizes of the other counters are negligible, the memory complexity is roughly 2^{96}. The time complexity, which is dominated by the subkey guessing part, is about $2^{122.49}$.

Algorithm 7: Zero-correlation linear attack on AES-PRF$_{1,9}$

Allocate counters V_i for each value of z_i, $1 \leqslant i \leqslant 6$.
for *each possible 16-bit subkey value* $K_0[0,5]$ **do**
  Reset V_1, compute $z_1 = P[4,9,10,14,15] \| MC^{-1}(C)[2:6] \| \left(x_1^R \oplus MC^{-1}(C)\right)[0:1]$ and update V_1 with N known-plaintexts.
  for *each possible 8-bit subkey value* $K_0[10]$ **do**
    Reset V_2, compute $z_2 = P[4,9,14,15] \| MC^{-1}(C)[3:6] \| \left(x_1^R \oplus MC^{-1}(C)\right)[0:2]$ and update V_2 as $V_2[z_2] \mathrel{+}= V_1[z_1]$.
    for *each possible 8-bit subkey value* $K_0[15]$ **do**
      Reset V_3, compute $z_3 = P[4,9,14] \| MC^{-1}(C)[4:6] \| \left(x_1^R \oplus MC^{-1}(C)\right)[0:3]$ and update V_3 as $V_3[z_3] \mathrel{+}= V_2[z_2]$.
      for *each possible 8-bit subkey value* $K_0[4]$ **do**
        Reset V_4, compute $z_4 = P[9,14] \| MC^{-1}(C)[5:6] \| \left(x_1^R \oplus MC^{-1}(C)\right)[0:4]$ and update V_4 as $V_4[z_4] \mathrel{+}= V_3[z_3]$.
        for *each possible 8-bit subkey value* $K_0[9]$ **do**
          Reset V_5, compute $z_5 = P[14] \| MC^{-1}(C)[6] \| \left(x_1^R \oplus MC^{-1}(C)\right)[0:5]$ and update V_5 as $V_5[z_5] \mathrel{+}= V_4[z_4]$.
          for *each possible 8-bit subkey value* $K_0[14]$ **do**
            Reset V_6, compute $z_6 = \left(x_1^R \oplus MC^{-1}(C)\right)[0:6]$ and update V_6 as $V_6[z_6] \mathrel{+}= V_5[z_5]$.
            Allocate a counter $V[z]$ for each of $(2^8-1)^7$ zero-correlation linear approximations, and set it to zero.
            Update $V[z]$ by $V_6[z_6]$ and compute $T = N \sum_{i=1}^{\ell} \left(2\frac{V[i]}{N} - 1\right)^2$.
            if $T < \tau$ **then**
              The guessed key bytes constitute a possible subkey candidate.
              All master keys that are compatible with are tested exhaustively against a maximum of 2 plaintext-ciphertext pairs.
            end
          end
        end
      end
    end
  end
end

B Attacks on Dual-AES-PRF

B.1 Impossible Differential for Dual-AES-PRF$_{9,1}$

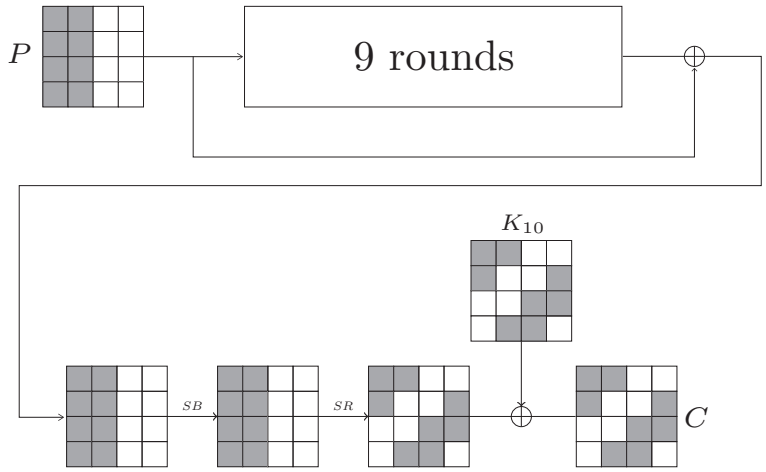


Figure 18: Overview of impossible differential against Dual-AES-PRF$_{9,1}$.

We first prepare pairs (P, P') of plaintexts whose difference ΔP is active in the left two columns as shown in Fig. 18. Then we pick pairs where the difference of the corresponding ciphertexts is inactive in $C[2, 3, 5, 6, 8, 9, 12, 15]$. Then any key guess under which the difference of the intermediate state before the last S-box layer is also ΔP must be incorrect, since this leads to an impossible differential $\Delta P \nrightarrow 0$ of the permutation.

Prepare 2^{64} plaintexts traversing all the 2^{64} possibilities of the 8 bytes marked with ■, while the remaining bytes are fixed to constants. From such 2^{64} plaintexts, we can get

$$\frac{2^{64} \times (2^{64} - 1)}{2} \times 2^{-64} \approx 2^{63}.$$

pairs of ciphertexts satisfying difference above.

Combining the knowledge of ΔC and ΔP, we can derive the impossible key bytes in $K_{10}[0, 1, 4, 7, 10, 11, 13, 14]$. Since every pair derives 1 impossible key in average, approximately 2^{63} keys are rejected.

We use the same formula that is introduced in Sect. 4.1. Then, we have $(\tau, \sigma) = (64, 0)$. According to equation (1), we need $2^{64-0} \times \frac{64}{\log_2 e} < 2^{70}$ good pairs to remove all incorrect keys, which demands $2^{70-63} = 2^7$ structures. In summary, the attack requires $2^{64+7} = 2^{71}$ chosen plaintexts, $2^{64+7} = 2^{71}$ computations, and 2^{64} memory.

B.2 Zero-Correlation Linear for Dual-AES-PRF$_{9,1}$

The linear mask Γ we use under this case is same as the one given in Fig. 16. The number of non-trivial zero-correlation linear approximation is $\ell = (2^8 - 1)^7$ under this setting.

Note that Dual-AES-PRF$_{9,1}$ is equivalent to the keyed function depicted in Fig. 19. The key-recovery attack can be found in Algorithm 8.

Complexity Analysis In this attack, we use $\ell = (2^8 - 1)^7$ non-trivial zero-correlation linear approximations. We set the type-I error probability to $\alpha_0 = 2^{-4}$, and the type-II error probability to $\alpha_1 = 2^{-26}$. Thus, the data complexity is $2^{103.34}$ known-plaintexts. Since V_1 is the largest memory and the sizes of the other counters are negligible, the

Algorithm 8: Zero-correlation linear attack on Dual-AES-PRF$_{9,1}$

Allocate counters V_i for each value of z_i, $1 \leqslant i \leqslant 6$.
for *each possible 16-bit subkey value* $K_{10}[0,13]$ **do**
  Reset V_1, compute $z_1 = P[2:6] \| C[1,4,7,10,14] \| \left(P \oplus x_{10}^I\right)[0:1]$ and update V_1 with N known-plaintexts.
  for *each possible 8-bit subkey value* $K_{10}[10]$ **do**
    Reset V_2, compute $z_2 = P[3:6] \| C[1,4,7,14] \| \left(P \oplus x_{10}^I\right)[0:2]$ and update V_2 as $V_2[z_2] \mathrel{+}= V_1[z_1]$.
    for *each possible 8-bit subkey* $K_{10}[7]$ **do**
      Reset V_3, compute $z_3 = P[4:6] \| C[1,4,14] \| \left(P \oplus x_{10}^I\right)[0:3]$ and update V_3 as $V_3[z_3] \mathrel{+}= V_2[z_2]$.
      for *each possible 8-bit subkey* $K_{10}[4]$ **do**
        Reset V_4, compute $z_4 = P[5:6] \| C[1,14] \| \left(P \oplus x_{10}^I\right)[0:4]$ and update V_4 as $V_4[z_4] \mathrel{+}= V_3[z_3]$.
        for *each possible 8-bit subkey* $K_{10}[1]$ **do**
          Reset V_5, compute $z_5 = P[6] \| C[14] \| \left(P \oplus x_{10}^I\right)[0:5]$ and update V_5 as $V_5[z_5] \mathrel{+}= V_4[z_4]$.
          for *each possible 8-bit subkey* $K_{10}[14]$ **do**
            Reset V_6, compute $z_6 = \left(P \oplus x_{10}^I\right)[0:6]$ and update V_6 as $V_6[z_6] \mathrel{+}= V_5[z_5]$.
            Allocate a counter $V[z]$ for each of $(2^8-1)^7$ zero-correlation linear approximations, and set it to zero.
            Update $V[z]$ by $V_6[z_6]$ and compute $T = N \sum_{i=1}^{\ell} \left(2\frac{V[i]}{N} - 1\right)^2$.
            if $T < \tau$ **then**
              The guessed key bytes constitute a possible subkey candidate.
              All master keys that are compatible with are tested exhaustively against a maximum of 2 plaintext-ciphertext pairs.
            end
          end
        end
      end
    end
  end
end

memory complexity is $\mathcal{O}(2^{96})$. The time complexity, which is dominated by the subkey guessing part, is about $2^{122.49}$.

B.3 Differential Attack for Dual-AES-PRF$_{2,8}$

An illustration for the key-recovery attack can be found in Fig. 20.

Birthday Paradox indicates that we are able to obtain a collision pair with 2^{64} plaintexts. Denote the collision pair as P_1 and P_2. Then, the differences of Δx_1^I and Δx_2^O at x_1^I and x_2^O are known as $\Delta x_1^I = \Delta x_2^O = P_1 \oplus P_2$.

We enumerate all 2^{32} differences of x_1^S at bytes $[0,5,10,15]$. Then, the values of x_1^I and x_1^S at bytes $[0,5,10,15]$ are known. The value of K_0 at bytes $[0,5,10,15]$ can be computed as $K_0 = P \oplus x_1^I$. We compute the difference Δx_2^I of x_2^I at bytes $[0:3]$ as

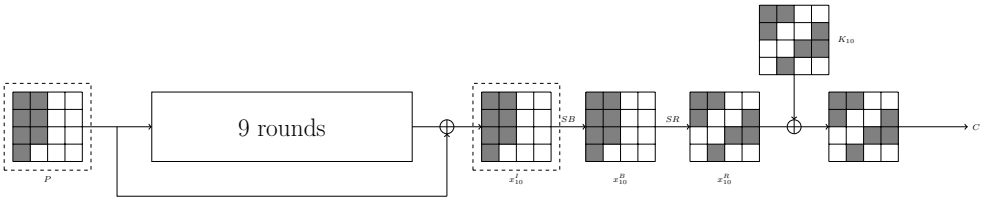


Figure 19: Key-recovery attack for Dual-AES-PRF$_{9,1}$.

$\Delta x_2^I = MC \circ SR(\Delta x_1^S)$, and the difference Δx_2^S of x_2^S as $\Delta x_2^S = SR^{-1} \circ MC^{-1}(\Delta x_2^O)$. Then, the values of x_2^I and x_2^S at bytes $[0:3]$ can be retrieved. Thus, the value of K_1 at bytes $[0:3]$ can be computed as $K_1 = MC \circ SR(x_1^S) \oplus x_2^I$.

Then, we enumerate all 2^{32} differences of x_1^S at bytes $[3, 4, 9, 14]$. With a similar analysis, we are able to obtain the value of K_0 at bytes $[3, 4, 9, 14]$, and the value of K_1 at bytes $[4:7]$ under each guess.

Since $K_1[5] = K_1[1] \oplus K_0[5]$, we obtain 2^{56} 120-bit information of the subkeys by combining the two sets composed of 2^{32} candidates. At last, all master keys that are compatible with the candidates are tested exhaustively against a maximum of two plaintext-ciphertext pairs.

In total, the data complexity of this attack is 2^{64} known plaintexts. The time complexity, which is dominated by the data collection phase and the exhaustive search phase, is 2^{65}. The memory complexity, which is dominated by the collision searching phase, is 2^{64}.

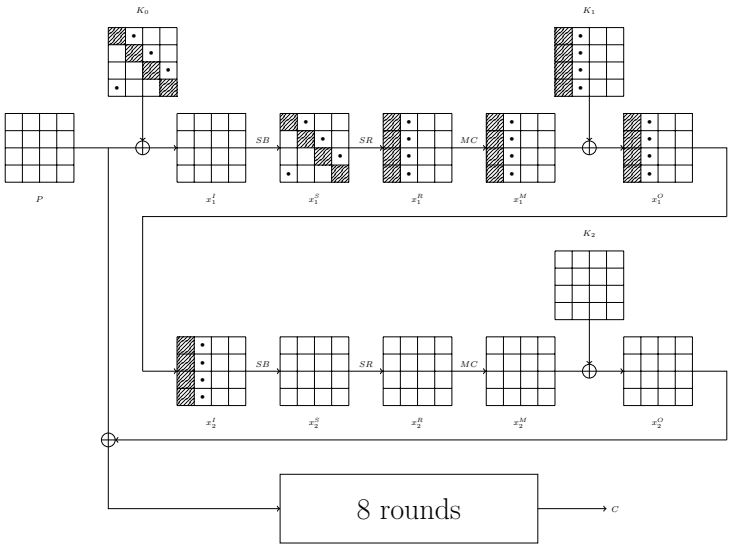


Figure 20: Key-recovery attack for Dual-AES-PRF$_{2,8}$.

B.4 Differential Attack for Dual-AES-PRF$_{3,7}$

An illustration for the key-recovery procedure can be found in Fig. 21.

Consider a sturcture of 2^{32} plaintexts such that the 4 gray bytes shown in Fig. 21 are active, while the remaining bytes in white are fixed to constants. From one structure of 2^{32} plaintexts, we are able to construct $2^{32} \times (2^{32} - 1)/2 \approx 2^{63}$ pairs. In order to obtain one collision pair at the output, we need to create 2^{128} pairs. Thus, 2^{65} structures are required. If we find a collision at the output, we know the difference at x_3^O.

For the collision pair P_1 and P_2 satisfying the input difference, we enumerate all 2^{32} input differences at x_2^I and 2^{32} output differences at x_2^S. Denote S as the AES S-box. The expected number of solutions of the equation $S(x) \oplus S(x \oplus \Delta x) = \Delta y$ in x for randomly chosen Δx and Δy is 1. Since we already know the differences of x_2^I and x_2^S at bytes $[0:3]$, we are able to obtain the values of x_2^I and x_2^S at bytes $[0:3]$.

Then, we compute the difference Δx_1^S at x_1^S as $\Delta x_1^S = SR^{-1} \circ MC^{-1}(\Delta x_2^I)$. Note that the difference Δx_1^I at x_1^I is known, we are able to retrieve the values of x_1^I and x_1^S at bytes $[0, 5, 10, 15]$. Thus, the value of K_0 at bytes $[0, 5, 10, 15]$ can be obtained, since $K_0 = P \oplus x_1^I$.

Now, we compute the value of x_1^M at bytes $[0:3]$ with the 4-byte known value of x_1^S. Combining the known value of x_2^I, we are able to get the value of K_1 at bytes $[0:3]$.

After that, we compute the difference Δx_3^I at x_3^I as $\Delta x_3^I = MC \circ SR(\Delta x_2^S)$, and the difference Δx_3^S at x_3^S as $\Delta x_3^S = SR^{-1} \circ MC^{-1}(\Delta x_3^O)$. With the input and output differences of the third SubBytes operation, the values of x_3^I and x_3^S are determined. Then the value of x_2^E at bytes $[0, 7, 10, 13]$ can be computed with $x_2^E = MC^{-1}(x_3^I)$. In addition to the known bytes of x_2^S, the value of EK_2 at bytes $[0, 7, 10, 13]$ can be obtained, naturally.

For each fixed Δx_2^I and Δx_2^S, we are able to retrieve 96-bit information of the subkey, and all master key that are compatible with are tested exhaustively against a maximum of two plaintext-ciphertext pairs.

In total, the data complexity of this attack is 2^{97} chosen plaintexts. The time complexity, which is dominated by the data collection phase, is also 2^{97}. In the collision searching phase, each structure can be analyzed, independently. Thus, the memory complexity is 2^{32}.

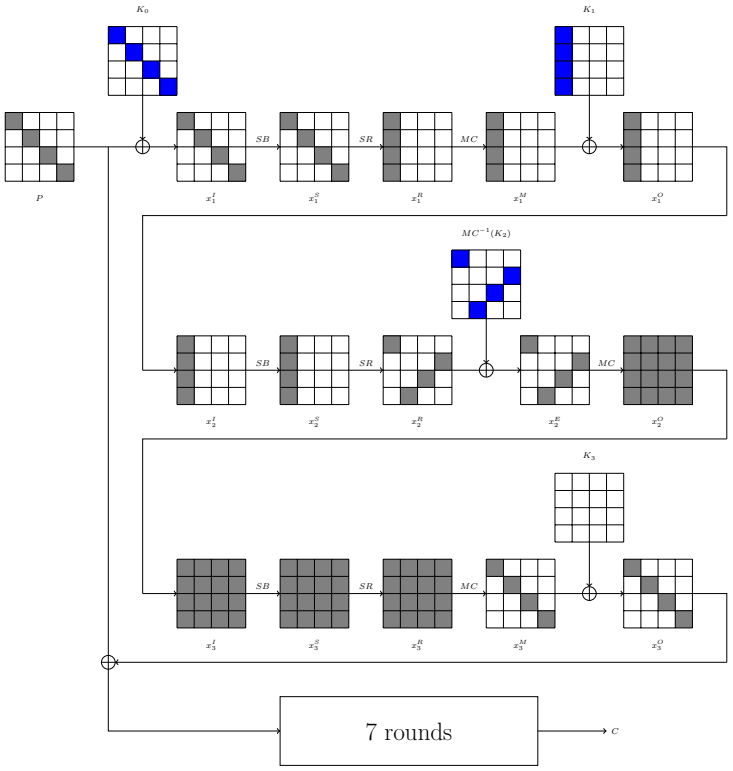


Figure 21: Key-recovery attack for Dual-AES-PRF$_{3,7}$.