Linear Cryptanalysis of PRINTcipher — Trails and Samples Everywhere

Martin Ågren and Thomas Johansson

Dept. of Electrical and Information Technology, Lund University,
P.O. Box 118, 221 00 Lund, Sweden
martin.agren@eit.lth.se

Abstract. PRINTcipher is a recent lightweight block cipher designed by Knudsen et al. Some noteworthy characteristics are a burnt-in key, a key-dependent permutation layer and identical round keys. Independent work on PRINTcipher has identified weak key classes that allow for a key recovery — the obvious countermeasure is to avoid these weak keys at the cost of a small loss of key entropy. This paper identifies several larger classes of weak keys. We show how to distinguish classes of keys and give a 28-round linear attack applicable to half the keys. We show that there are several similar attacks, each focusing on a specific class of keys. We also observe how some specific properties of PRINTcipher allow us to collect several samples from each plaintext–ciphertext pair. We use this property to construct an attack on 29-round PRINTcipher applicable to a fraction 2^{-5} of the keys.

Keywords: cryptanalysis, block cipher, linear cryptanalysis, finding samples, key bit distinguisher

1 Introduction

Over the last few years, a number of hardware-efficient block ciphers have been proposed. Some noteworthy examples are HIGHT [6], PRESENT [3], and KATAN and KTANTAN [5]. One of the most recent designs to appear is PRINTcipher [8]. It is designed by Knudsen et al. and is quite similar to the well-studied PRESENT. All rounds use the same key and differ only by a round counter. The linear layer is partly key-dependent and as a result, 48-bit PRINTcipher uses keys of 80 bits, while 96-bit PRINTcipher uses 160-bit keys. We will focus exclusively on PRINTcipher-48 in this paper, noting that very similar results can be derived for PRINTcipher-96.

Our first observation relates to the key-dependent permutation: we show how there exist several linear trails in PRINTcipher that are biased for some keys but unbiased for most keys, allowing us to distinguish between classes of keys. In order to attack several rounds of PRINTcipher, we need to find many samples. Our second observation uses the identical round-structure, including identical keys, to obtain several samples per plaintext–ciphertext pair. By guessing key

bits to do partial encrypting and decrypting, we eventually reach 29 rounds of 48.

Two recent attacks are similar to our work in that they identify classes of weak keys. As a fundamental idea behind PRINTCIPHER is that the key is burnt into the device, it is straightforward to protect against these attacks by avoiding the weak keys. Avoiding the 2^{52} keys attacked in [9] the size of the key space shrinks from 2^{80} to $2^{80} - 2^{52} \approx 2^{80}$ so the entropy is still 80 bits in a practical sense. Similarly, to protect against the attack in [4] the number of keys needs to be lowered to approximately $2^{79.8}$ so there is a loss of one fifth of a bit. In this independent paper, we find several classes that are very probable (e.g., probability one half), and even avoiding only the largest classes leads to a key space of size approximately 2^{78}, meaning two bits of the key entropy are effectively lost. This makes our observations very interesting compared to the previously published results.

This paper is organized as follows: Section 2 describes PRINTCIPHER. Section 3 introduces linear cryptanalysis and discusses the importance of finding many samples. Some initial, basic observations are given in Section 4, before Section 5 gives our fundamental observation: a key bit distinguisher on 23 rounds of PRINTCIPHER. Section 6 then derives attacks on 27 and 28 rounds of PRINTCIPHER. In Section 7, we extend our observation to show that several classes of weak keys exist, making the attack very general, and to show how one can find many samples. In Section 8, we use our ability to find many samples to provide an attack on 29-round PRINTCIPHER. Section 9 concludes the paper.

2 A Description of PRINTCIPHER

We focus entirely on PRINTCIPHER-48, which uses blocks of 48 bits and 80-bit keys.

The 48-bit plaintext is loaded into the state, where we denote the 48 bit positions as (b, c), $0 \leq b < 16, 0 \leq c < 3$. The leftmost bit, also referred to as the most significant bit (msb), is $(15, 2)$ while $(0, 0)$ is the least significant bit (lsb).

There are 48 rounds where each round uses a round constant RC_i, $i = 0, \ldots, 47$ (see Table 1), a 48-bit xor-key $K^{\oplus}$ (the same in all rounds) and a 32-bit permutation key K^{π} (the same in all rounds). Each round consists of key addition, standard permutation, round constant addition, key-dependent permutation and an S-box, see Fig. 1. The S-box is given in Table 2 and takes input (x_2, x_1, x_0) to produce output (y_2, y_1, y_0).

Table 1. The round constants RC_i.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
RC_i	01	03	07	0F	1F	3E	3D	3B	37	2F	1E	3C	39	33	27	0E	1D	3A	35	2B	16	2C	18	30
i	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
RC_i	21	02	05	0B	17	2E	1C	38	31	23	06	0D	1B	36	2D	1A	34	29	12	24	08	11	22	04

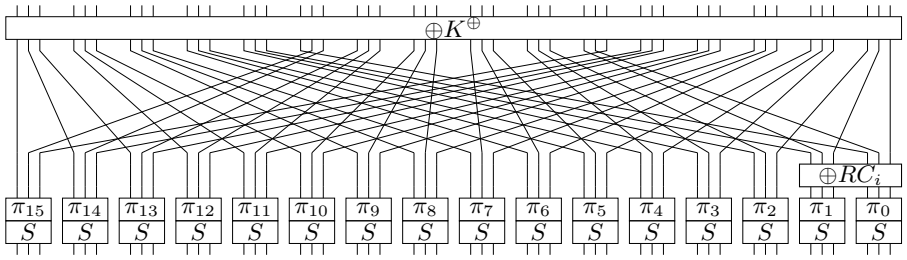

Fig. 1. One round of PRINTcipher.

Table 2. The S-box as values $S(\boldsymbol{x}) = \boldsymbol{y} = (y_2, y_1, y_0)$ corresponding to $\boldsymbol{x} = (x_2, x_1, x_0)$. As an example, $S(1,0,0) = S(4) = 7 = (1,1,1)$.

$\boldsymbol{x}$	0	1	2	3	4	5	6	7
$S(\boldsymbol{x})$	0	1	3	6	7	4	5	2

We denote the plaintext $P = p_{47}, \ldots, p_0$ and the ciphertext (state) after r rounds of encryption ($0 < r \leq 48$) by $C^r = c^r_{47}, \ldots, c^r_0$.

2.1 The Key

We split the key $K = K^{\oplus}||K^{\pi} = (k^{\oplus}_{47}, \ldots, k^{\oplus}_0)||(k^{\pi}_{31}, \ldots, k^{\pi}_0)$ into an xor-key $K^{\oplus}$ (48 bits) and a permutation key K^{π} (32 bits).

2.2 The Standard Permutation Π

A "large" permutation Π is applied to the state of PRINTcipher. Each bit, at position (b, c), is moved to $(t \bmod 16, \lfloor t/16 \rfloor)$ where $t = 3b + c$. The permutation is given in Appendix A and can also be seen in Fig. 1.

2.3 The Key-Dependent Permutation π

A "small" permutation π_b is applied to each (disjoint) triplet of bits in the state of PRINTcipher. The new positions of bits $(b,2), (b,1), (b,0)$ are $(b, c^b_2), (b, c^b_1), (b, c^b_0)$, respectively, where (c^b_2, c^b_1, c^b_0) are determined by the two key bits $(k^{\pi}_{2b+1}, k^{\pi}_{2b})$. This mapping is given in Table 3. Note in particular how one of the permutations is trivial while the others fix one bit while switching the two remaining bits. Thus, the two permutations that shift the three-bit word cyclically have been excluded from PRINTcipher and can not be selected by the key.

2.4 Other Notation

Let $I_r = r \bmod 2$ indicate whether r is even or odd. Partial encryption in rounds $r_1, \ldots, r_2 - 1$ is denoted by ϕ_{r_1, r_2}. Similarly, partial decryption in rounds $r_2 - 1, \ldots, r_1$ is denoted $\phi^{-1}_{r_1, r_2}$.

Table 3. The key-dependent permutation. The bits at positions $(b,2),(b,1),(b,0)$ are moved to positions $(b,c_2^b),(b,c_1^b),(b,c_0^b)$, respectively where (c_2^b,c_1^b,c_0^b) are determined by $(k^\pi_{2b+1},k^\pi_{2b})$.

$(k^\pi_{2b+1},k^\pi_{2b})$	(c_2^b,c_1^b,c_0^b)
$(0,0)$	$(2,1,0)$
$(0,1)$	$(1,2,0)$
$(1,0)$	$(2,0,1)$
$(1,1)$	$(0,1,2)$

2.5 Existing Work on PRINTcipher

Abdelraheem et al. have given a differential attack on 22-round PRINTcipher [1]. Using the entire code book, they study the single-bit differentials in order to learn how the bits are permuted through the entire cipher, i.e., r rounds. Finding the rth root of this permutation then gives them the single-round permutation $\pi \circ \Pi$ and thus K^π.

We note that it is straightforward to invert the last S-box upon retrieving a ciphertext (it is present only to make hardware implementations smaller as it does not require any special logic for the last round as in e.g., AES). Thus, an attacker can extend the 22-round attack to 23 rounds at very low cost: the S-box only has to be inverted if the three bits in its output are the only bits that have a difference.

Two very recent publications reach further than 22 rounds. At CRYPTO, Leander et al. [9] showed how an "invariant subspace attack" allowed for a class of 2^{52} keys to be distinguished regardless of the number of rounds, so in particular for the full PRINTcipher. At SAC, Karakoç et al. [4] combined differential and linear cryptanalysis to reach 29 rounds on 4.54% and 31 rounds on .036% of the keys.

3 Linear Cryptanalysis

Originally introduced by Matsui [11], linear cryptanalysis has since been applied to a large number of cryptographic primitives in many different fashions. In the original form, the goal is to find some biased linear relation on bits in the progressing state of the cipher. If key bits involved in partially encrypting and/or decrypting are guessed correctly, the bias should be observable, while for wrong guesses, the bias should not appear. As a result, the partial key guess can be verified and the rest of the key found through an exhaustive search. The end result is an attack faster than exhaustive search, but the cost is that one needs to access many plaintext–ciphertext pairs (see Section 3.1).

While it is common to study trails on linear combinations of bits in plaintext and ciphertext,

$$P\left(\boldsymbol{\alpha}\cdot P=\boldsymbol{\beta}\cdot C+\boldsymbol{\gamma}\cdot K\right)=\frac{1}{2}\pm\epsilon,$$

where $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ are bitmasks, the most simple case is to study single-bit trails such as

$$P\left(p_{47} = c_{47}\right) = \frac{1}{2} + \epsilon.$$

This paper will exclusively deal with single-bit trails, possibly involving the xor of one bit of key, although it is no doubt possible to find many more trails by using multiple-bit trails. The reason we do this is that the single-bit trails appear very naturally in PRINTCIPHER.

We refer to ϵ as the *bias* of the trail, and an attacker will naturally try to find relations with as large bias as possible. In the PRINTCIPHER specification, the designers show that the optimal linear trails over r rounds of PRINTCIPHER have probability $\frac{1}{2} + 2^{-r-1}$, i.e., bias 2^{-r-1}. In this paper, we will exclusively look at such optimal trails.

The piling-up lemma [11] tells us how the bias diminishes over more rounds. In our context it means that piling two optimal trails on r_1 and r_2 rounds into one trail on $r = r_1 + r_2$ rounds, results in a bias of $2^{-r-1} = 2^{-r_1-r_2-1}$. This is not surprising: piling two optimal trails results in an optimal trail.

Every time we look at (e.g.,) $p_{47} \oplus c_{47}$, we actually look at a *sample*, a bit that is picked from some distribution. By looking at sufficiently many samples, we can make a sufficiently good guess on which distribution we are dealing with.

3.1 On the Importance of Finding Many Samples

In order to distinguish between two distributions on $\{0, 1\}$, one with $\text{Prob}(1) = \frac{1}{2} + \epsilon$ and one with $\text{Prob}(1) = \frac{1}{2}$, it is commonly accepted [11, 2] that one needs ϵ^{-2} samples. One actually needs $\alpha\epsilon^{-2}$ samples, but the constant α is small enough to be ignored: this allows for easier analysis and comparisons of cryptanalytic results. In this paper, we will always need to obtain 2^{2r+2} samples.

An attacker can only access 2^{48} different plaintext–ciphertext pairs on PRINTCIPHER, which seems to indicate that only 2^{48} samples can be found and that only 23-round trails can be used, i.e., less than half the number of rounds. If we want to use a trail on $(23 + s)$ rounds, we need to obtain $2^{2(23+s)+2} = 2^{48+2s}$ samples, i.e., 2^{2s} samples per plaintext–ciphertext pair.

In this paper, we will note how some particular features of PRINTCIPHER allow us to find trails where we can access several samples per plaintext–ciphertext pair. We also see how these samples are independent (enough) to make them usable in a cryptanalytic setting.

We will only consider iterated trails, i.e., trails beginning and ending at a common bit position. This is for simplicity: iterated trails can be used to trivially create trails on larger numbers of rounds. One can also see that by using iterated trails, the number of distinct π_b involved in the trail is kept to a minimum, which keeps the involved number of key bits decently small.

Recall that a sample s_j is a bit obtained by comparing a plaintext bit to a ciphertext bit (more generally, linear combinations). Crucial in linear cryptanalysis is counting how many samples are 1 resp. 0, i.e., deriving the sum $S = \sum_j s_j$. Kaliski and Robshaw [7] note that if one can find several linear approximations

that involve the exact same key bits, i.e., the same bitmask $\boldsymbol{\gamma}$, so that one can get several counts $S^i = \sum_j s^i_j$, one can use a weighted sum of these counts S^i — this measurement has the same expected value but a smaller variance. In particular, when the bias is the same for all linear approximations, the weighted sum is simply the average, which up to a multiplicative constant is the same as the overall number of samples that are 1, i.e., $\sum_{i,j} s^i_j$. It is then natural to think of the different s^i_j (with varying i and j) as different samples from the same underlying distribution.

4 Some Initial Observations

4.1 The S-Box

Some single-bit trails are available on the S-box and through the remainder of this paper, we will focus on these three:

$$\text{Prob}(y_2 = x_2) = \text{Prob}(y_1 = x_1) = \text{Prob}(y_0 = x_0 \oplus 1) = \frac{1}{2} + 2^{-2}.$$

They are conveniently all from x_i to y_i, which is not strictly necessary but simplifies the presentation of the subsequent observations and attacks.

4.2 The Permutation π_b and the S-box

Table 4. The S-box evaluated for all possible permutations on the input.

(x_2, x_1, x_0)	$S(x_2, x_1, x_0)$	$S(x_1, x_2, x_0)$	$S(x_2, x_0, x_1)$	$S(x_0, x_1, x_2)$
$(0,0,0)$	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$
$(0,0,1)$	$(0,0,1)$	$(0,0,1)$	$(0,1,1)$	$(1,1,1)$
$(0,1,0)$	$(0,1,1)$	$(1,1,1)$	$(0,0,1)$	$(0,1,1)$
$(0,1,1)$	$(1,1,0)$	$(1,0,0)$	$(1,1,0)$	$(1,0,1)$
$(1,0,0)$	$(1,1,1)$	$(0,1,1)$	$(1,1,1)$	$(0,0,1)$
$(1,0,1)$	$(1,0,0)$	$(1,1,0)$	$(1,0,1)$	$(1,0,0)$
$(1,1,0)$	$(1,0,1)$	$(1,0,1)$	$(1,0,0)$	$(1,1,0)$
$(1,1,1)$	$(0,1,1)$	$(0,1,1)$	$(0,1,1)$	$(0,1,1)$

One can quite easily see that with $(y_2, y_1, y_0) = S(x_2, x_1, x_0)$ and $(y'_2, y'_1, y'_0) = S(x_2, x_0, x_1)$, we always have $y_2 = y'_2$, see Table 4. This means that if we are only interested in tap 2 out of the S-box, it does not matter if x_1, x_0 are swapped or not before entering the S-box.

As a consequence, if we

- know three bits that enter $S \circ \pi_b$,
- want to know y_2 out of the S-box, and
- need to guess the permutation π_b, i.e., $(k^\pi_{2b+1}, k^\pi_{2b})$,

then we only need to make three guesses on π_b.

The same property shows up on y_0 also, but not on y_1, see Table 4. We will use this observation to reduce the amount of guesswork we need to perform during partial encryption. We will use the notation π_b^3 to mark that we only guess a ternary digit, a trit, for π_b due to these properties.

Similarly, when we guess for a partial decryption, we often do not need to guess the whole permutation π_b, i.e., two bits, but only how it permutes one particular bit. We will (e.g.,) use the notation $\pi_b(2)$ to indicate that we only guess how the bit 2 is permuted by π_b.

5 A Key Bit Distinguisher

5.1 General Attack Idea

We will use a variant of linear cryptanalysis: we study single-bit trails that are biased for certain classes of keys and non-biased for other keys. As a very non-detailed example, consider a trail from the left-most bit to the left-most bit. It is readily apparent from Fig. 1 that such a trail exists and that it is iterated (although it is of course not obvious from the figure that it has a bias). We claim that we can distinguish individual bits of K^π using this trail: it is biased for half the keys and non-biased for the other half. Thus, if we can distinguish between these two distributions (i.e., if the bias is large enough and we have sufficiently many samples) we can determine the value of this key bit.

The sample trail considered here is "simple" as it is apparent to the naked eye, but it is possible to find several such trails over considerable numbers of rounds. As a consequence, there exist many classes of weak keys in PRINTcipher.

5.2 A Detailed Example

We now describe how to distinguish between two distributions: one where k_{30}^π is zero, and one where it is one. This allows for a partial-key recovery, i.e., learning one bit of the key, faster than brute force.

Note that $\mathit{\Pi}(15, 2) = (15, 2)$, and that for two of four keys, $\pi_{15}(2) = 2$. This happens precisely when $k_{30}^\pi = 0$ (see Table 5).

Table 5. How the individual bits $(2, 1, 0)$ are moved by the key-dependent permutation π_b, and for which keys $(k_{2b+1}^\pi, k_{2b}^\pi)$ it happens.

Bit Move	Possible Keys	Bit Move	Possible Keys	Bit Move	Possible Keys
$0 \to 0$	$(0,0)$,$(0,1)$	$1 \to 0$	$(1,0)$	$2 \to 0$	$(1,1)$
$0 \to 1$	$(1,0)$	$1 \to 1$	$(0,0)$,$(1,1)$	$2 \to 1$	$(0,1)$
$0 \to 2$	$(1,1)$	$1 \to 2$	$(0,1)$	$2 \to 2$	$(0,0)$,$(1,0)$

Thus, with $k_{30}^\pi = 0$, $(\pi \circ \mathit{\Pi})(15, 2) = (15, 2)$. The probability that this bit then passes the S-box unaltered is $\frac{3}{4}$, so after a single round of encryption, we

have

$$\text{Prob}(c_{47}^1 = p_{47} \oplus k_{47}^{\oplus}) = \frac{1}{2} + 2^{-2}.$$

For two rounds, we have

$$\text{Prob}(c_{47}^2 = p_{47}) = \frac{1}{2} + 2^{-3}$$

as the key xors cancel and with the use of the piling-up lemma. Generalizing to any even number of rounds, we have

$$\text{Prob}(c_{47}^r = p_{47}) = \frac{1}{2} + 2^{-r-1}.$$

For PRINTcipher on 22 rounds, we would need almost the entire code book, 2^{46} plaintext–ciphertext pairs.

We can also use the full code book, of size 2^{48}, to attack 23 rounds. We then have an odd number of rounds, and the key bit $k_{47}^{\oplus}$ shows up, so we utilize the relation

$$\text{Prob}(c_{47}^r = p_{47} \oplus k_{47}^{\oplus}) = \frac{1}{2} + 2^{-r-1}, \tag{1}$$

with $r = 23$. Things then get slightly more tricky, as we learn more about the key but need to distinguish between three distributions:

1. $c_{47}^r = p_{47}$ with probability $\frac{1}{2}$, implying $k_{30}^{\pi} = 1$.
2. $c_{47}^r = p_{47}$ with "high" probability, implying $k_{30}^{\pi} = 0$ and $k_{47}^{\oplus} = 0$.
3. $c_{47}^r = p_{47}$ with "low" probability, implying $k_{30}^{\pi} = 0$ and $k_{47}^{\oplus} = 1$.

5.3 More Linear Trails on One Round of PRINTcipher

There are in total four iterated single-round trails, and we list them in Table 6. Some constants arise as the S-box flips bit 0 with probability $\frac{3}{4}$ rather than preserves it, and as bits of RC_i enter.

Table 6. The iterated single-round trails on PRINTcipher, extended to several rounds. All trails have bias 2^{-r-1}.

Trail	Requirement
$c_0^r = p_0 \oplus k_0^{\oplus} I_r \oplus d_r$	$k_1^{\pi} = 0$
$c_{23}^r = p_{23} \oplus k_{23}^{\oplus} I_r$	$(k_{15}^{\pi}, k_{14}^{\pi}) = (0, 1)$
$c_{24}^r = p_{24} \oplus k_{24}^{\oplus} I_r \oplus I_r$	$(k_{17}^{\pi}, k_{16}^{\pi}) = (1, 0)$
$c_{47}^r = p_{47} \oplus k_{47}^{\oplus} I_r$	$k_{30}^{\pi} = 0$
$d_r = \left(r + 1 + \sum_{0 \le i < r} RC_i\right) \bmod 2$	

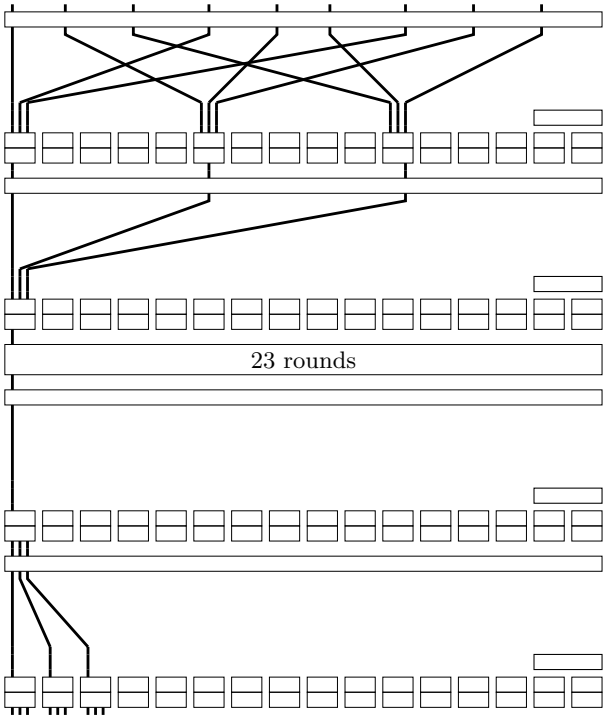

Fig. 2. Performing two rounds of partial encryption and decryption to access the bits at position $(15, 2)$.

6 Guessing Keybits for Partial Encryption and Decryption

The above observation can be used as-is to mount an attack on 23-round PRINTcipher, recovering up to three bits of the key, but it is straightforward to derive an even more powerful attack on 27 rounds of PRINTcipher: if a guessed partial key is correct, we should observe the bias, while if the guess is bad, the behaviour should be (more) random.

First, we assume that $k_{30}^{\pi} = 0$, meaning our attack only works for a fraction 2^{-1} of the keys. Then, we aim to decrypt two rounds at the end and encrypt two rounds at the top of PRINTcipher. Thus, we need to guess the bits and trits listed in Table 7. There are in total $N = 2^{13} \cdot 3^3 \approx 2^{17.8}$ guesses. See Fig. 2 for an overview of the partial calculations.

Due to the property observed in Section 4.2, we do not need to guess k_{31}^{π}. We have assumed $k_{30}^{\pi} = 0$ to fix $\pi_{15}(2) = 2$ and this is enough to predict tap 2 out of the S-box. It does not matter whether π_{15} is trivial or swaps bits 0 and 1.

We call the plaintext (resp. ciphertext) bits that affect the partial encryption (resp. decryption) to the bits we are interested in *active*. There are nine active

Table 7. The bits and trits required for encryption, decryption, and both, when encrypting/decrypting two rounds to access the bits at position $(15, 2)$.

Encryption	$k^{\oplus}_{42}$, $k^{\oplus}_{37}$, $k^{\oplus}_{31}$, $k^{\oplus}_{26}$, $k^{\oplus}_{21}$, $k^{\oplus}_{15}$, $k^{\oplus}_{10}$, $k^{\oplus}_{5}$, π_{10}, π^{3}_{5}
Decryption	$k^{\oplus}_{46}$, $k^{\oplus}_{45}$, $\pi_{14}(2)$, $\pi_{13}(2)$
Both	$k^{\oplus}_{47}$

bits in the plaintext and nine in the ciphertext. For a plaintext–ciphertext pair (P, C) we can collect these bits into an eighteen-bit word $w = (p_{47}, p_{31}, \ldots, c_{39})$.

We describe the attack: Acquire all 2^{48} plaintext–ciphertext pairs (P^j, C^j). Categorize them according to the active bits, i.e., for each possible word w, count how often it appears. Denote these counters R_w. This is the data collection part of the attack.

We then begin analyzing the data. For each plaintext–ciphertext pair and for each guess of key material, denoted by G_i, $0 \leq i < N$, we will calculate two rounds of encryption and decryption, $\hat{P}^j = \phi_{0,2}(G_i, P^j)$, $\hat{C}^j = \phi^{-1}_{25,27}(G_i, C^j)$, and count how often $\hat{c}^j_{47} = \hat{p}^j_{47} \oplus k^{\oplus\, i}_{47}$. This is done using N counters S_i. An efficient way of doing this [10] is to use the counters R_w. For each word w and each keyguess G_i, we do the partial calculations $\hat{P}^{i,w} = \phi_{0,2}(G_i, P_w)$, $\hat{C}^{i,w} = \phi^{-1}_{25,27}(G_i, C_w)$. P_w (C_w) is some plaintext (ciphertext) which has the correct active bits as determined by w. If $\hat{c}^{i,w}_{47} = \hat{p}^{i,w}_{47} \oplus k^{\oplus\, i}_{47}$, we add R_w to S_i.

By sorting all S_i, we can get a ranking of the different guesses. We pick the most likely guess, brute force all non-guessed bits and hopefully recover the key. Otherwise, we pick the second most likely guess, etc. The exact number of bits that need to be brute forced will be different for different guesses: where we guessed a trit (e.g.,) $\pi_{14}(2)$, we will have recovered one or two bits of $(k^{\pi}_{29}, k^{\pi}_{28})$. As long as the correct guess is ranked on the upper half of the sorted list of counters, the entire key will be found faster than what can be expected from a brute force (2^{78}).

The counters R_w are used for saving time [10]. Several other improvements can be made, also from [10]: We should not make $2^{18} \cdot N$ partial encryptions and decryptions. First, the plaintext and ciphertext operations can be separated completely, so that we only need to make $2^9 \cdot N + 2^9 \cdot N$ encryptions/decryptions. Second, since the overlap in encryption and decryption with respect to the guessed bits is very small, we only need to perform $2^9 \cdot N_e + 2^9 \cdot N_d$ encryptions/decryptions where N_e (N_d) is the number of key guesses that actually affect the encryption (decryption). Third, doing two complete PRINTCIPHER rounds in both directions is unneccessary as we only need to perform "partial rounds", i.e., use some small number of S-boxes.

6.1 Experimental Results

We have implemented this attack on $7+4 = 11$ rather than $23+4 = 27$ rounds of PRINTCIPHER. This means that we guess the same bits and perform the same

partial encryptions, but that the bias is larger so that it is feasible for us to perform many attacks in order to gather statistics.

It turns out that over 2^{13} different weak keys, the attack works with probability 0.78. That is, almost four times out of five, the correct key ranks on the upper half of the sorted list of S_i.

6.2 Analyzing the Attack Complexity

The attack consists of data collection and data analysis. The latter in turn consists of 1) deriving two sets of counters, N_e for encryption and N_d for decryption, and 2) combining these to find N counters. If the number of active bits in the plaintext (ciphertext) is denoted a_e (a_d) and the number of active S-boxes in encryption (decryption) is denoted A_e (A_d), the time complexities are given by

$$\begin{aligned} T^{collect} &= \epsilon^{-2}, \\ T^{count} &= \frac{2^{a_e} N_e A_e + 2^{a_d} N_d A_d}{16 \cdot r}, \\ T^{combine} &= 2^{a_e + a_d} N. \end{aligned}$$

The first two measurements are normalized to r-round PRINTCIPHER evaluations, while the last describes the number of "simple" bit and integer operations needed to calculate the counters S_i.

For the specific attack detailed above, we have $N_e = 2^{11} \cdot 3$ and $N_d = 2^3 \cdot 3^2$. Since $(a_e, a_d, A_e, A_d) = (9, 9, 4, 4)$, the complexities turn out at

$$\begin{aligned} T^{collect} &= 2^{2 \cdot 24} = 2^{48}, \\ T^{count} &= \frac{2^9 \cdot 2^{11} \cdot 3 \cdot 4 + 2^9 \cdot 2^3 \cdot 3^2 \cdot 4}{16 \cdot 27} \approx 2^{15}, \\ T^{combine} &= 2^{9+9} \cdot N = 2^{18} \cdot 2^{13} \cdot 3^3 \approx 2^{36}. \end{aligned}$$

This suggests that the most time consuming part is the data collection where we need to generate and look at 2^{48} plaintext–ciphertext pairs.

6.3 Reaching the Limit: 28 Rounds

We note that in the attacks on 27 rounds, guessing and encrypting is more expensive than guessing and *de*crypting: during decryption, we first invert S, and then only need to control one bit in some π_b. On the other hand, during encryption, we need to fully control the permutation, so that we can calculate all three bits that go into the S-box. This leads to more expensive guesswork on π_b and especially on $K^{\oplus}$. Thus, the natural approach for extending the attack by one round is to add another round in the partial decryption.

In Table 8, we list the bits and trits involved in partially decrypting and encrypting from 28 rounds to 23. The attack requires $N = 2^{18} \cdot 3^8 \approx 2^{30.7}$ guesses,

partitioned as $N_e = 2^{11} \cdot 3$ and $N_d = 2^9 \cdot 3^8$. With $(a_e, a_d, A_e, A_d) = (9, 27, 4, 13)$, we have

$$\begin{aligned} T^{collect} &= 2^{2 \cdot 24} = 2^{48}, \\ T^{count} &= \frac{2^9 \cdot 2^{11} \cdot 3 \cdot 4 + 2^{27} \cdot 2^9 \cdot 3^8 \cdot 13}{16 \cdot 28} \approx 2^{44}, \\ T^{combine} &= 2^{9+27} \cdot N = 2^{36} \cdot 2^{18} \cdot 3^8 \approx 2^{67}. \end{aligned}$$

28 rounds seems to be the best we can do: using a single trail, we have not been able to go beyond 28 rounds while keeping the attack costs below exhaustive search.

Table 8. The bits and trits required for encryption, decryption, and both, when encrypting/decrypting two/three rounds to access the bits at position $(15, 2)$. Note the overlap in π_{10}.

Encryption	$k^{\oplus}_{37}$, $k^{\oplus}_{31}$, $k^{\oplus}_{26}$, $k^{\oplus}_{21}$, $k^{\oplus}_{15}$, $k^{\oplus}_{10}$, $k^{\oplus}_{5}$, π_{10}, π^3_5
Decryption	$k^{\oplus}_{46}$, $k^{\oplus}_{45}$, $k^{\oplus}_{44}$, $k^{\oplus}_{43}$, $k^{\oplus}_{41}$, $k^{\oplus}_{40}$, $k^{\oplus}_{39}$, $\pi_{14}(2)$, $\pi_{13}(2)$, $\pi_{12}(2)$, $\pi_{11}(2)$, $\pi_{10}(2)$, $\pi_9(2)$, $\pi_8(2)$, $\pi_7(2)$
Both	$k^{\oplus}_{42}$, $k^{\oplus}_{47}$

7 On More Rounds of PRINTCIPHER: Complementary Trails

We generalize our observation slightly and give an example two-round trail: with $(k^{\pi}_{23}, k^{\pi}_{22}, k^{\pi}_{6}) = (1, 1, 0)$,

$$\text{Prob}(c^2_{11} = p_{11} \oplus k^{\oplus}_{11} \oplus k^{\oplus}_{35}) = \frac{1}{2} + 2^{-3}.$$

Note in particular how there is a complementary trail,

$$\text{Prob}(c^2_{35} = p_{35} \oplus k^{\oplus}_{11} \oplus k^{\oplus}_{35}) = \frac{1}{2} + 2^{-3},$$

see Fig. 3. The complementary trail depends on the exact same key configuration and allows us to collect *two* samples with every plaintext–ciphertext pair. We show in Section 8.1 that this works, i.e., the samples can be considered as independent.

We do not give all the two-round trails on PRINTCIPHER, as we will not use them in the remainder of the paper. We only note that due to the structure of PRINTCIPHER, every S-box is used precisely once so far in the paper: either in one trail on one round (S-boxes 0, 7, 8, 15), or in two complementary trails on two rounds.

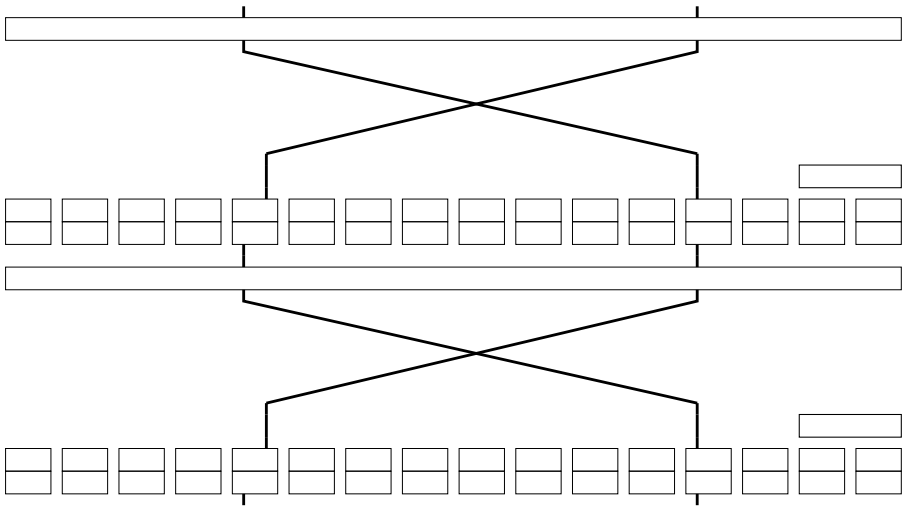

Fig. 3. Two complementary trails on two rounds of PRINTCIPHER. Both trails are activated by $(k^{\pi}_{23}, k^{\pi}_{22}, k^{\pi}_{6}) = (1, 1, 0)$.

Table 9. The number of iterated trails over various number of rounds.

#Rounds	#Trails	#Rounds	#Trails	#Rounds	#Trails	#Rounds	#Trails
1	4	5	154	9	5806	13	138662
2	16	6	424	10	13366	14	283810
3	28	7	1040	11	30430	15	560608
4	96	8	2584	12	65808	16	1075000

As a particular four-round trail that we will use later, we give

$$\text{Prob}(c^4_{37} = p_{37} \oplus k^{\oplus}_{37} \oplus k^{\oplus}_{17} \oplus k^{\oplus}_{4} \oplus k^{\oplus}_{12} \oplus 1) = \frac{1}{2} + 2^{-5}, \tag{2}$$

which is activated by $(k^{\pi}_{25}, k^{\pi}_{24}, k^{\pi}_{10}, k^{\pi}_{9}, k^{\pi}_{3}) = (1, 0, 0, 0, k^{\pi}_{2})$.

The total number of iterated trails over various number of rounds are given in Table 9.

7.1 More Attacks on 27/28 Rounds

We can use basically any trail on 23 rounds to create attacks on 27/28 rounds. Do note that the trail on bit $(15, 2)$ is very nice as the partial encryptions and decryptions involve few bits of $K^{\oplus}$ and K^{π}, due to S-box reuse. Most other trails involve more guesswork. As an example, using (2) 5.75 times yields

$$\text{Prob}(c^{25}_{12} = c^{2}_{37} \oplus k^{\oplus}_{12} \oplus 1) = \frac{1}{2} + 2^{-24},$$

from which we can build an attack on 28 rounds. We have $N = 2^{27} \cdot 3^6 \approx 2^{36.5}$, $N_e = 2^{14} \cdot 3$, $N_d = 2^{13} \cdot 3^7$ and $(a_e, a_d, A_e, A_d) = (9, 27, 4, 13)$. The complexities are

$$\begin{aligned} T^{collect} &= 2^{2 \cdot 24} = 2^{48}, \\ T^{count} &= \frac{2^9 \cdot 2^{14} \cdot 3 \cdot 4 + 2^{27} \cdot 2^{13} \cdot 3^7 \cdot 13}{16 \cdot 28} \approx 2^{46}, \\ T^{combine} &= 2^{9+27} \cdot N = 2^{36} \cdot 2^{27} \cdot 3^6 \approx 2^{73}. \end{aligned}$$

The bits and trits guessed are listed in Appendix B.

7.2 On False Positives

By piling the single-round trail on the left-most bit, we see that e.g., $\text{Prob}(c_{47}^{10} = p_{47}) = \frac{1}{2} + 2^{-11}$ when $k_{30}^{\pi} = 0$. However, there are several other ways of obtaining this distribution.

All in all, there are 102 different trails from $(15, 2)$ to $(15, 2)$ over ten rounds, each corresponding to a different class of keys. This means that a biased distribution can be explained by any of these trails, and thus any of these classes. Due to this, an attacker will prefer to use short, iterated trails involving few bits of K^{π}.

8 Using Complementary Trails to Distinguish on 24-Round Trails

We will now construct 24-round trails with bias 2^{-25}. By using trails that allow four samples per plaintext–ciphertext pair, we can get in total 2^{50} samples, allowing us to distinguish the distribution.

The best iterated trails on 24 rounds are given in Table 10. They are "best" in the sense that they use a small number of key bits (5), yet allow four complementary trails each, so that we can get the required number of samples. In fact they are constructed from iterated four-round trails, that we have piled in order to cancel the bits that appear from $K^{\oplus}$ (cf. Section 5.2).

8.1 Samples are Independent (Enough)

The connection between the bias ϵ and the required number of samples ϵ^{-2} relies on the independence of the samples, and it is not obvious that the samples we pick are independent. Most cryptanalysis simply assumes that the samples are independent, or at least independent enough for the attacks to still be possible. Verifying the independence through simulation is common, at least on a smaller number of rounds or reduced-size versions of the algorithm ("PRINTCIPHER-12"), where it is practically possible.

We need to be a little bit more wary than usual as we pick several samples from the same plaintext–ciphertext pair — it is not hard to realize that the

Table 10. The iterated trails on eight rounds ($r = 8$) composed from four-round iterated trails, depending only on five bits of K^π. All trails have bias 2^{-r-1}, and the constants e_j^r arise from the round constants RC_i. The trails are easily extended to e.g., 24 rounds ($r = 24$), in which case only the constants need to be rechecked. (The symmetrically inclined reader have ample reasons to admire this table.)

Trail	S-boxes	Trail	S-boxes
$c_4^r = p_4 \oplus e_4^r$	4,12,5,1,4,12,5,1,...	$c_{10}^r = p_{10}$	10,14,11,3,10,14,11,3,...
$c_{12}^r = p_{12} \oplus e_{12}^r$	12,5,1,4,12,5,1,4,...	$c_{30}^r = p_{30}$	14,11,3,10,14,11,3,10,...
$c_{17}^r = p_{17} \oplus e_{17}^r$	1,4,12,5,1,4,12,5,...	$c_{35}^r = p_{35}$	3,10,14,11,3,10,14,11,...
$c_{37}^r = p_{37} \oplus e_{37}^r$	5,1,4,12,5,1,4,12,...	$c_{43}^r = p_{43}$	11,3,10,14,11,3,10,14,...
Key class		Key class	
$(k_{25}^\pi, k_{24}^\pi, k_{10}^\pi, k_9^\pi, k_3^\pi) = (1, 0, 0, 0, k_2^\pi)$		$(k_{29}^\pi, k_{22}^\pi, k_{21}^\pi, k_7^\pi, k_6^\pi) = (k_{28}^\pi, 0, 0, 0, 1)$	
Trail	S-boxes	Trail	S-boxes
$c_7^r = p_7$	7,7,6,2,7,7,6,2,...	$c_{24}^r = p_{24}$	8,9,13,8,8,9,13,8,...
$c_{18}^r = p_{18}$	2,7,7,6,2,7,7,6,...	$c_{25}^r = p_{25}$	9,13,8,8,9,13,8,8,...
$c_{22}^r = p_{22}$	6,2,7,7,6,2,7,7,...	$c_{29}^r = p_{29}$	13,8,8,9,13,8,8,9,...
$c_{23}^r = p_{23}$	7,6,2,7,7,6,2,7,...	$c_{40}^r = p_{40}$	8,8,9,13,8,8,9,13,...
Key class		Key class	
$(k_{15}^\pi, k_{14}^\pi, k_{13}^\pi, k_{12}^\pi, k_5^\pi) = (1, 1, 1, 0, k_4^\pi)$		$(k_{27}^\pi, k_{19}^\pi, k_{18}^\pi, k_{17}^\pi, k_{16}^\pi) = (k_{26}^\pi, 0, 1, 1, 1)$	
Constants ($r = 8$)		Constants ($r = 24$)	
$(e_4^8, e_{12}^8, e_{17}^8, e_{37}^8) = (1, 1, 1, 1)$		$(e_4^{24}, e_{12}^{24}, e_{17}^{24}, e_{37}^{24}) = (1, 0, 1, 1)$	

calculations behind the four samples have affected each other, and it is not impossible that samples obtained from the same plaintext–ciphertext pair are so dependent that they do not contribute (much) more than one sample from an information-theoretic point of view. If this is the case, we would not be able to exploit any bias smaller than (about) 2^{-23}.

Thus, we have done the following on eight-round PRINTcipher: We use 2^{18} plaintext–ciphertext pairs to derive equally many samples on bit $(1, 1)$, and from this we guess whether the key is in the upper-left class from Table 10 by comparing the number of samples that are 1 to some pre-defined threshold derived to yield a 50% success rate. This gives false positives/negatives with probabilities 0.03/0.50, respectively. Similar probabilities are observed for the three complementary trails, when used one on one.

If we instead use only 2^{16} plaintext–ciphertext pairs, but pick 2^2 samples from each pair, we are able to carry out the attack with seemingly unchanged success: The probabilities of false positives/negatives are 0.02/0.50. These results have been obtained by attacking 2^{14} keys from each class and are listed in Table 11.

8.2 Partial Encryption and Decryption for 29 Rounds

Similar to in Section 6, we aim to guess key bits for partial encryptions and decryptions. Previously, we were able to add five rounds in this way to construct

Table 11. The attack in Section 8.1 was carried out on 2^{14} different keys using either one sample per plaintext–ciphertext pair or four samples per pair but fewer pairs. "True Pos. Ratio" shows how frequently a key belonging to the keyclass was identified as such. Similarly, "True Neg. Ratio" shows how often a key not belonging to the keyclass was correctly excluded.

Trail	Pairs	Samples/Pair	True Pos. Ratio	True Neg. Ratio
$c_4^8 = p_4 \oplus 1$	2^{18}	1	0.50	0.97
$c_{12}^8 = p_{12} \oplus 1$	2^{18}	1	0.50	0.97
$c_{17}^8 = p_{17} \oplus 1$	2^{18}	1	0.51	0.97
$c_{37}^8 = p_{37} \oplus 1$	2^{18}	1	0.51	0.97
all four	2^{16}	2^2	0.51	0.98

a 28-round attack using a 23-round trail. Now, using the 24-round trails, we reach 29 rounds. Again, we use the upper-left key class in Table 10.

The key observation is that we can divide all of the work, so that we deal with the four trails completely independently. If we number the trails $j = 1, 2, 3, 4$, we will have time complexities

$$\begin{aligned} T_j^{collect} &= \epsilon^{-2}, \\ T_j^{count} &= \frac{2^{a_e^j} N_e^j A_e^j + 2^{a_d^j} N_d^j A_d^j}{16 \cdot r}, \\ T_j^{combine} &= 2^{a_e^j + a_d^j} N. \end{aligned}$$

for producing the four different lists of counters S_i^j. In order to combine all counters S_i^j into N-many counters S_i we need to do N rather simple operations. Note that we have made related, but not identical, guesses on the permutations, e.g., by guessing $\pi_{14}(2)$ when using trail 1 and $\pi_{14}(0)$ when using trail 2. Some care must be taken here, but it does not affect the cost of this step, which remains at $T^{finalize} = N$ quite simple operations.

Specific guesses are listed in Appendix B. All $(a_e^j, a_d,^j , A_e^j, A_d^j) = (9, 27, 4, 13)$. The total attack complexities are

$$\begin{aligned} T^{collect} &= 2^{2 \cdot 24} = 2^{48}, \\ T^{count} &= \sum_j T_j^{count} \approx 2^{50}, \\ T^{combine} &= \sum_j T_j^{combine} \approx 2^{76}, \\ T^{finalize} &= N = 2^{62} \cdot 3^3 \approx 2^{67}. \end{aligned}$$

Although brute force costs 2^{75}, as we assume five bits of the key, we claim that 2^{76} "simple" operations compare favorably to 2^{75} evaluations of 29-round PRINTCIPHER.

Let us briefly comment on the possibility of using 25-round trails with bias 2^{-26}: if we can get 16 samples per plaintext–ciphertext pair, we have the necessary 2^{52} samples. As we need to involve all π_b, we would put restrictions on at least 16 bits of the key. This puts the brute force cost at 2^{64} or lower, which seems to be too low for the attack to be meaningful. Another obstacle to this attack is that the complementary trails are not completely identical, as different bits of $K^{\oplus}$ will appear.

Table 12. A summary of the explicit attacks on 27-, 28- and 29-round PRINTcipher presented in this paper. r denotes the length of the trail(s) used, and R denotes that R-round PRINTcipher is attacked. 'Enc' ('dec') tells how many rounds are partially encrypted (decrypted). T^{count} and $T^{combine}$ are rounded to the nearest integer power of two.

Trail	r	enc	dec	R	Key fraction	$T^{collect}$	T^{count}	$T^{combine}$
$c_{47}^{25} = c_{47}^{2} \oplus k_{47}^{\oplus}$	23	2	2	27	2^{-1}	2^{48}	2^{15}	2^{36}
$c_{47}^{25} = c_{47}^{2} \oplus k_{47}^{\oplus}$	23	2	3	28	2^{-1}	2^{48}	2^{44}	2^{67}
$c_{12}^{25} = c_{37}^{2} \oplus k_{12}^{\oplus} \oplus 1$	23	2	3	28	2^{-5}	2^{48}	2^{46}	2^{73}
$c_{4}^{26} = c_{4}^{2}$ and more	24	2	3	29	2^{-5}	2^{48}	2^{50}	2^{76}

9 Conclusion

Table 12 summarizes the attacks on 27–28 rounds of PRINTcipher outlined in this paper. Several more attacks are available for several more key classes.

We note some particular observations that all arise from the structure of PRINTcipher and the use of the exact same round key throughout the cipher:

- When there is a non-decomposable, iterated r-round trail there are in fact r complementary trails, allowing r samples per plaintext–ciphertext pair.
- When we guess for a partial encryption/decryption, there is overlap between the bits that activate the trail and those we need for encryption/decryption.

With this work, linear cryptanalysis has reached 28 rounds of PRINTcipher. We have used weak key classes which means that we need to carry out several attacks in parallel in order to have a high probability of success. However, we have seen that there are many large key classes and in particular several of them only depend on one or two bits of the key. This means our results "invalidate" several more keys than previous results on PRINTcipher. The exception is the differential attack which worked on all keys but only reached 23 rounds.

We have exclusively studied PRINTcipher-48, but our observations are no doubt applicable to PRINTcipher-96 as well, where it seems reasonable that our techniques could be used to reach around 52–55 rounds. Another area of future research could be to look at the linear hull effect. The work in [9] and [4]

suggest that the linear hull of PRINTcipher can behave in unexpected ways. It might be possible to cause peculiar effects to arise, e.g., by fixing more bits of the key, in order to reach further into PRINTcipher with linear cryptanalysis.

As a further research direction, we note that by inverting the $(S \circ \pi_b)$ where π_b is (partly) assumed, the number of active bits could be reduced. The technique would apply to all attacks in this paper, but the full gain of this remains to be determined.

The complementary trails that arise in PRINTcipher are very interesting, and allowed us to add one round to the attacks, albeit for a smaller class of keys. It would be very interesting to see if this complementary property could lead to more observations on PRINTcipher.

Acknowledgment

This work was supported by the Swedish Foundation for Strategic Research (SSF) through its Strategic Center for High Speed Wireless Communication at Lund. The authors wish to thank the anonymous reviewers whose comments helped improve the paper.

References

1. M. A. Abdelraheem, G. Leander, and E. Zenner. Differential cryptanalysis of round-reduced PRINTcipher: Computing roots of permutation. In A. Joux, editor, *Fast Software Encryption 2011*, Lecture Notes in Computer Science, pages 1–17. Springer-Verlag, 2011.
2. T. Baignères, P. Junod, and S. Vaudenay. How far can we go beyond linear cryptanalysis? In *Advances in Cryptology—ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 432–450. Springer-Verlag, 2004.
3. A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In *Cryptographic Hardware and Embedded Systems—CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer-Verlag, 2007.
4. F. Karakoç, H. Demirci, and A. E. Harmancı. Combined differential and linear cryptanalysis of reduced-round PRINTcipher. In *Selected Areas in Cryptography—SAC 2011*, To be published in Lecture Notes in Computer Science. Springer-Verlag, 2011.
5. C. De Cannière, O. Dunkelman, and M. Knežević. KATAN and KTANTAN — a family of small and efficient hardware-oriented block ciphers. In *Cryptographic Hardware and Embedded Systems—CHES 2009*, volume 5747, pages 272–288. Springer-Verlag, 2009.
6. D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B.-S. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, and S. Chee. HIGHT: A New Block Cipher Suitable for Low-Resource Device. In *Cryptographic Hardware and Embedded Systems—CHES 2006*, volume 4249 of *Lecture Notes in Computer Science*, pages 46–59. Springer-Verlag, 2006.
7. B. S. Kaliski and M. J. B. Robshaw. Linear cryptanalysis using multiple approximations. In Y. Desmedt, editor, *Advances in Cryptology—CRYPTO'94*, volume 839 of *Lecture Notes in Computer Science*, pages 26–39. Springer-Verlag, 1994.

8. L. Knudsen, G. Leander, A. Poschmann, and M. Robshaw. PRINTcipher: A block cipher for ic-printing. In S. Mangard and F-X. Standaert, editors, *Cryptographic Hardware and Embedded Systems—CHES 2010*, volume 6225 of *Lecture Notes in Computer Science*, pages 16–32. Springer, 2010.
9. G. Leander, M. A. Abdelraheem, H. AlKhzaimi, and E. Zenner. A cryptanalysis of PRINTcipher: The invariant subspace attack. In P. Rogaway, editor, *Advances in Cryptology—CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 206–221. Springer-Verlag, 2011.
10. M. Matsui. The first experimental cryptanalysis of the Data Encryption Standard. In Y. Desmedt, editor, *Advances in Cryptology—CRYPTO'94*, volume 839 of *Lecture Notes in Computer Science*, pages 1–11. Springer-Verlag, 1994.
11. M. Matsui. Linear cryptanalysis method for DES cipher. In T. Helleseth, editor, *Advances in Cryptology—EUROCRYPT'93*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer-Verlag, 1994.

A The Standard Permutation

Table 13 lists the key-independent permutation in PRINTcipher.

Table 13. The standard permutation Π in PRINTcipher. Bits at positions 'In' are moved to positions 'Out'.

In	Out	In	Out	In	Out	In	Out
$(0,0)$	$(0,0)$	$(4,0)$	$(12,0)$	$(8,0)$	$(8,1)$	$(12,0)$	$(4,2)$
$(0,1)$	$(1,0)$	$(4,1)$	$(13,0)$	$(8,1)$	$(9,1)$	$(12,1)$	$(5,2)$
$(0,2)$	$(2,0)$	$(4,2)$	$(14,0)$	$(8,2)$	$(10,1)$	$(12,2)$	$(6,2)$
$(1,0)$	$(3,0)$	$(5,0)$	$(15,0)$	$(9,0)$	$(11,1)$	$(13,0)$	$(7,2)$
$(1,1)$	$(4,0)$	$(5,1)$	$(0,1)$	$(9,1)$	$(12,1)$	$(13,1)$	$(8,2)$
$(1,2)$	$(5,0)$	$(5,2)$	$(1,1)$	$(9,2)$	$(13,1)$	$(13,2)$	$(9,2)$
$(2,0)$	$(6,0)$	$(6,0)$	$(2,1)$	$(10,0)$	$(14,1)$	$(14,0)$	$(10,2)$
$(2,1)$	$(7,0)$	$(6,1)$	$(3,1)$	$(10,1)$	$(15,1)$	$(14,1)$	$(11,2)$
$(2,2)$	$(8,0)$	$(6,2)$	$(4,1)$	$(10,2)$	$(0,2)$	$(14,2)$	$(12,2)$
$(3,0)$	$(9,0)$	$(7,0)$	$(5,1)$	$(11,0)$	$(1,2)$	$(15,0)$	$(13,2)$
$(3,1)$	$(10,0)$	$(7,1)$	$(6,1)$	$(11,1)$	$(2,2)$	$(15,1)$	$(14,2)$
$(3,2)$	$(11,0)$	$(7,2)$	$(7,1)$	$(11,2)$	$(3,2)$	$(15,2)$	$(15,2)$

B Bits Involved in Given Attacks

Tables 14 and 15 lists what key material is guessed for the attacks on 28 and 29 rounds using the four-round trail 5.75 and 6 times.

Table 14. The bits and trits required for encryption, decryption, and both, when encrypting/decrypting two/three rounds to access the bits at position $(12, 1)/(4, 0)$.

Encryption	$k^{\oplus}_{46}$, $k^{\oplus}_{44}$, $k^{\oplus}_{41}$, $k^{\oplus}_{30}$, $k^{\oplus}_{28}$, $k^{\oplus}_{25}$, $k^{\oplus}_{9}$, $k^{\oplus}_{4}$, π^{3}_{14}, π_9,
Decryption	$k^{\oplus}_{38}$, $k^{\oplus}_{37}$, $k^{\oplus}_{19}$, $k^{\oplus}_{18}$, $k^{\oplus}_{17}$, $k^{\oplus}_{16}$, $k^{\oplus}_{15}$, $k^{\oplus}_{13}$, k^{π}_{8}, $\pi_{15}(0)$, $\pi_{14}(0)$, $\pi_{13}(0)$, $\pi_{12}(0)$, $\pi_6(1)$, $\pi_3(1)$, $\pi_2(1)$, $\pi_0(1)$
Both	$k^{\oplus}_{36}$, $k^{\oplus}_{20}$, $k^{\oplus}_{14}$, $k^{\oplus}_{12}$

Table 15. The bits and trits guessed in the attack on 29-round PRINTcipher by encrypting/decrypting two/three rounds.

j	Pos.	N^j	Bits	
1	$(12, 1)$	$2^{31} \cdot 3^6$	Enc.	$k^{\oplus}_{9}$, $k^{\oplus}_{12}$, $k^{\oplus}_{14}$, $k^{\oplus}_{20}$, $k^{\oplus}_{25}$, $k^{\oplus}_{28}$, $k^{\oplus}_{30}$, $k^{\oplus}_{36}$, π_9, π^{3}_{14}
			Dec.	$k^{\oplus}_{0}$, $k^{\oplus}_{1}$, $k^{\oplus}_{2}$, $k^{\oplus}_{3}$, $k^{\oplus}_{5}$, $k^{\oplus}_{15}$, $k^{\oplus}_{16}$, $k^{\oplus}_{17}$, $k^{\oplus}_{37}$, $k^{\oplus}_{45}$, $k^{\oplus}_{46}$, $k^{\oplus}_{47}$, π_0, $\pi_2(0)$, $\pi_3(0)$, π^{3}_{11}, $\pi_{13}(2)$, $\pi_{14}(2)$, π_{15}
			Both	$k^{\oplus}_{4}$, $k^{\oplus}_{41}$, $k^{\oplus}_{44}$, $k^{\oplus}_{46}$
2	$(5, 2)$	$2^{23} \cdot 3^9$	Enc.	$k^{\oplus}_{1}$, $k^{\oplus}_{7}$, $k^{\oplus}_{21}$, $k^{\oplus}_{23}$, $k^{\oplus}_{28}$, $k^{\oplus}_{33}$, $k^{\oplus}_{37}$, $k^{\oplus}_{39}$, $k^{\oplus}_{44}$, k^{π}_{2}, π^{3}_{7}
			Dec.	$k^{\oplus}_{3}$, $k^{\oplus}_{4}$, $k^{\oplus}_{9}$, $k^{\oplus}_{10}$, $k^{\oplus}_{11}$, $k^{\oplus}_{13}$, $k^{\oplus}_{14}$, $k^{\oplus}_{15}$, $k^{\oplus}_{16}$, $\pi_0(1)$, $\pi_3(0)$, k^{π}_{11}, $\pi_9(0)$, $\pi_{10}(0)$, $\pi_{11}(0)$, $\pi_{13}(0)$, $\pi_{14}(0)$, $\pi_{15}(0)$
			Both	$k^{\oplus}_{5}$, $k^{\oplus}_{12}$, $k^{\oplus}_{17}$
3	$(4, 0)$	$2^{21} \cdot 3^6$	Enc.	$k^{\oplus}_{1}$, $k^{\oplus}_{4}$, $k^{\oplus}_{6}$, $k^{\oplus}_{22}$, $k^{\oplus}_{28}$, $k^{\oplus}_{33}$, $k^{\oplus}_{44}$, k^{π}_{2}
			Dec.	$k^{\oplus}_{13}$, $k^{\oplus}_{14}$, $k^{\oplus}_{15}$, $k^{\oplus}_{16}$, $k^{\oplus}_{18}$, $k^{\oplus}_{19}$, $k^{\oplus}_{37}$, $\pi_0(1)$, $\pi_2(1)$, $\pi_3(1)$, k^{π}_{8}, $\pi_{13}(0)$, $\pi_{14}(0)$, $\pi_{15}(0)$
			Both	$k^{\oplus}_{12}$, $k^{\oplus}_{17}$, $k^{\oplus}_{20}$, $k^{\oplus}_{36}$, $k^{\oplus}_{38}$, $\pi^{3}_{6} = \pi_6(2)$
4	$(1, 1)$	$2^{29} \cdot 3^7$	Enc.	$k^{\oplus}_{0}$, $k^{\oplus}_{1}$, $k^{\oplus}_{5}$, $k^{\oplus}_{11}$, $k^{\oplus}_{16}$, $k^{\oplus}_{17}$, $k^{\oplus}_{21}$, $k^{\oplus}_{27}$, $k^{\oplus}_{32}$, $k^{\oplus}_{33}$, π_0, k^{π}_{2}, π^{3}_{11}
			Dec.	$k^{\oplus}_{4}$, $k^{\oplus}_{12}$, $k^{\oplus}_{13}$, $k^{\oplus}_{14}$, $k^{\oplus}_{36}$, $k^{\oplus}_{38}$, $k^{\oplus}_{39}$, $k^{\oplus}_{40}$, $k^{\oplus}_{41}$, $k^{\oplus}_{42}$, $k^{\oplus}_{44}$, k^{π}_{8}, $\pi_6(2)$, $\pi_7(2)$, $\pi_8(2)$, $\pi_9(2)$, $\pi_{10}(2)$, $\pi_{11}(2)$, $\pi_{13}(2)$, $\pi_{14}(2)$
			Both	$k^{\oplus}_{37}$, $k^{\oplus}_{43}$, π_{11}
Overall		$2^{62} \cdot 3^3$		$k^{\oplus}_{0}$, …, $k^{\oplus}_{7}$, $k^{\oplus}_{9}$, …, $k^{\oplus}_{23}$, $k^{\oplus}_{25}$, $k^{\oplus}_{27}$, $k^{\oplus}_{28}$, $k^{\oplus}_{30}$, $k^{\oplus}_{32}$, $k^{\oplus}_{33}$, $k^{\oplus}_{36}$, …, $k^{\oplus}_{47}$, π_0, k^{π}_{2}, π_2, π_3, k^{π}_{8}, k^{π}_{11}, $\pi^{3}_{6} = \pi_6(2)$, $\pi^{3}_{7} = \pi_7(2)$, $\pi_8(2)$, π_9, π_{10}, π_{11}, π_{13}, π_{14}, π_{15}