# On the Security of the Core of PRINCE Against Biclique and Differential Cryptanalysis

Farzaneh Abed, Eik List, Stefan Lucks

Bauhaus-University Weimar, Germany
{farzaneh.abed,eik.list,stefan.lucks}@uni-weimar.de

**Abstract.** PRINCE is a modern involutive lightweight cipher which was proposed by Rechberger *et al.* in 2012. PRINCE uses 64-bit core cipher, PRINCE$_{core}$, which holds the major encryption logic and is wrapped by two key additions. Thus, the security of the cipher is mainly depending on the security properties of the core. In this paper, we present an independent-biclique attack on the full version and also a differential inside-out cryptanalysis on the round-reduced version of PRINCE$_{core}$.

**Keywords:** block ciphers, lightweight, biclique, differential cryptanalysis

## 1 Specification of PRINCE

PRINCE is a lightweight cipher with a state size of 64 bits and a key length of 128 bits. Its structure follows the so-called $FX$ construction principle [4], where one part of the key is used for a core cipher $F$, which contains the major encryption process, and the remaining parts are used for whitenings before and after the core: $FX_{k,k_1,k_2} = k_2 \oplus F_k(x \oplus k_1)$. In PRINCE, the 128-bit key $k$ is split into two 64-bit words, first

$$k = k_0 || k_1,$$

before it is expanded to 192 bits by the mapping

$$k = (k_0||k_1) \rightarrow (k_0||k_0'||k_1) := (k_0||(k_0 \ggg 1) \oplus (k_0 \gg 63)||k_1)$$

$k_1$ is then used for the core; the remaining words, $k_0$ and the derived value $k_0'$, are used to wrap the core with two key additions, the pre- and post-whitening.
The core cipher, PRINCE$_{core}$, is a block cipher of its own with key and state lengths of 64 bits each. It employs an involutive structure which, in the beginning, consists of two XORs with the key and a round constant, followed by five forward rounds, a middle layer, five backward rounds and at the end, again two XORs with a round constant and a key. Figure 1 shows the schematic view of the core. Every round in PRINCE contains five operations:

- **An S-box-layer** $S$: Every byte in the internal state is replaced by using a 4 × 4-bit S-box.
- **A linear layer** $M'$: In the linear layer, the state is multiplied by 64 × 64-matrix. More precisely, there are two $16 \times 16$ submatrices $M_0$ and $M_1$ which are arranged on the diagonal of a bigger matrix, where every submatrix affects a 16-bit chunk $x_i$ of the 64-bit state $x = (x_1 \| x_2 \| x_3 \| x_4)$:

$$M'(x) = \begin{pmatrix} M_0 & 0 & 0 & 0 \\ 0 & M_1 & 0 & 0 \\ 0 & 0 & M_1 & 0 \\ 0 & 0 & 0 & M_0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = M_0(x_1) \| M_1(x_2) \| M_1(x_3) \| M_0(x_4)$$

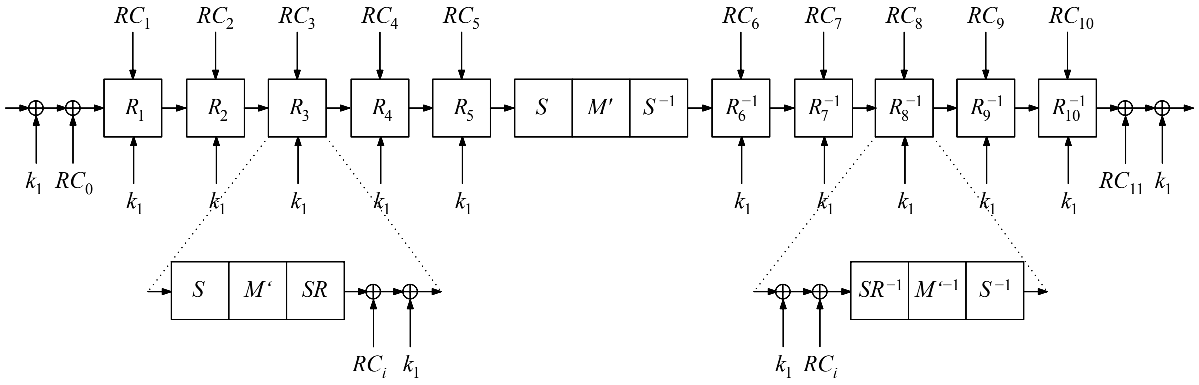


**Fig. 1.** Schematic view of the PRINCE$_{core}$ cipher.

- **A ShiftRows operation** $SR$: Works exactly same as the one in the AES cipher.
- **A bit-wise XOR with a round constant** $RC_i$, for $i \in \{0, \ldots, 11\}$.
- **A bit-wise XOR with the key** $k_1$.

The state of PRINCE$_{core}$ can be visualized as a $4 \times 4$-matrix, where every cell represents a nibble. One forward round of the cipher is depicted in Figure 2. In the backward rounds, the order of operations is the inverse of the forward part, where only the round constants differ. In the middle between the forward and backward rounds, there are three keyless operations: a forward S-box layer, a matrix multiplication with $M'$ and an inverse S-box layer. The matrix $M'$ is the same as in the $M$ operation in forward and backward rounds. To remain the involution property in the middle part, $M'$ was chosen to be self-inverting. Yet, in the round functions, involution is not necessary. So, the operation $M$ combines the matrix multiplication with an AES-like ShiftRows operation in the round to ensure quick diffusion: $M := SR \circ M'$. Like in the AES, the combination of matrix multiplication and shifting provides full diffusion after only two rounds.

The varying round constants $RC_i$ supplement the round transformation in order to prevent slide attacks. The difference between $RC_i \oplus RC_{11-i}$ is always equal to a constant value $\alpha = 0xc0ac29b7c97c50dd$. As a result of the involutive structure, software and hardware implementations can use the same encryption and decryption operations. The decryption only needs to be parametrized with the key XOR and with the round constant difference $\alpha$:

$$D_{k_0||k_0'||k_1 \oplus \alpha}(\cdot) = E_{k_0||k_0'||k_1}(\cdot).$$

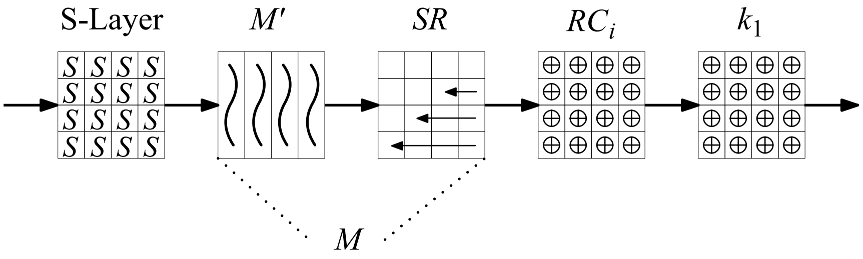


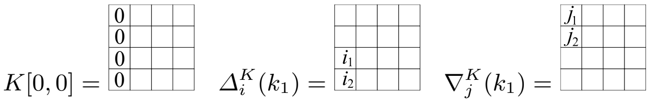
**Fig. 2.** Overview of a single forward round in PRINCE$_{core}$.

## 2 Independent-Biclique Attack on the Full PRINCE$_{core}$

In this section, we describe an independent-biclique attack on the full version of PRINCE$_{core}$. We suppose that most of our readers are familiar with biclique attacks, so we will concentrate on the details which are necessary for this specific attack. For details of the technique, we refer to the works by Khovratovich *et al.* [3] and Bogdanov *et al.* [1] which introduced bicliques for cryptanalysis.

### 2.1 Key Space Partitioning

We partition the key space with respect to the secret key $k_1$ and enumerate groups of $2^{48}$ base keys. The base keys $K[0,0]$ are all $2^{48}$ 16-nibble values with four nibbles fixed to 0, where all other nibbles in the state take on all possible values. The keys in a group $\{K[i,j]\}$ are enumerated by all possible differences $i = (i_1 \| i_2)$ and $j = (i_1 \| i_2)$ with respect to $K[0,0]$.



### 2.2 Single-Round Biclique of Dimension 8

We construct a biclique over the final round and over the final additions of the state with $k_1$ and $RC_{11}$, as shown in Figure 3. In both differentials, the two-nibble difference in $k_1$ is injected in the beginning of the round in the state, and spreads out to eight active nibbles after matrix multiplication in the $M'$-layer. In the forward trails, the key addition after the final round leads to ten active nibbles in the ciphertexts $C_i$. Since we fix $C_0$ for all key groups, the data complexity is upper bounded by $2^{40}$ ciphertexts.

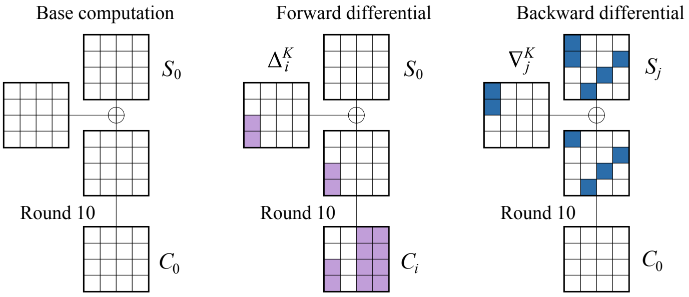


**Fig. 3.** Biclique for PRINCE$_{core}$ in round 11 with $\Delta_i$- and $\nabla_j$-differentials.

### 2.3 Matching over 9 Rounds

For the remaining rounds 1-9, we use the matching-with-precomputations approach which was introduced in [1]. We match in two nibbles of the state after the middle part, *i.e.,* before round 6. In the precomputation step, first, in forward direction, we compute from the plaintexts $P_i$ to the matching state $\overrightarrow{v_{i,0}}$ under the keys $K[i,0]$:

$$P_i \xrightarrow{K[i,0]} \overrightarrow{v_{i,0}} \quad \forall i \in \{0,\ldots,2^8-1\}.$$

Similarly, in backward direction, we compute from the states $S_j$ in the beginning of the biclique to the matching states $\overleftarrow{v_{0,j}}$ under the keys $K[0,j]$:

$$\overleftarrow{v_{0,j}} \xleftarrow{K[0,j]} S_j \quad \forall j \in \{0,\ldots,2^8-1\}.$$

The $2^8$ precomputed values $\overrightarrow{v_{i,0}}$ and $2^8$ values $\overleftarrow{v_{0,j}}$ are then stored. For all further values $\overrightarrow{v_{i,j}}$ and $\overleftarrow{v_{i,j}}$, we use the precomputed values and recompute only those parts which differ from the stored one. The differences result from the usage of the keys $K[i,j]$ instead of $K[i,0]$ or $K[0,j]$, respectively:

$$P_i \xrightarrow{K[i,j]} \overrightarrow{v_{i,j}} \overset{?}{=} \overleftarrow{v_{i,j}} \xleftarrow{K[i,j]} S_j \quad \forall i,j \in \{0,\ldots,2^8-1\}.$$

If we could apply $\overrightarrow{v_{i,j}} \overset{?}{=} \overleftarrow{v_{i,j}}$ for some $K[i,j]$, then $K[i,j]$ yields a potential secret-key candidate. These $2^{16}-2^9$ recomputations make up the major summand in the total computational complexity. The matching and the parts which need to be recomputed are illustrated in Figure 4. As we can see from the figure, in forward part, the key injection has affected the full state after two rounds. But since we match only in a portion of the state $v$, only three rounds, 3-5, have to be considered.

The recomputation costs consist of the number of matrix multiplications, S-box calls, shifts and some XOR operations. The matrix multiplications and S-boxes operations certainly have the largest impact on the complexity. To represent this effort in the best way, we consider a single number and concentrate on the S-boxes which need to be recomputed, following the argumentation in [1].

In forward direction, one needs to recompute $2+5+16+16+16+4+2=61$ S-boxes; Figure 4 shows the active nibbles in the $S$ operations in the states directly after the key additions. There, we need to include the $S^{-1}$ operation at the end of the middle part. In backward direction, the relevant states are also located directly after the key additions which means, $2+8+4=14$ S-boxes need to be considered. So, in total, one has to recompute $61+16=77$ S-boxes in the matching part.

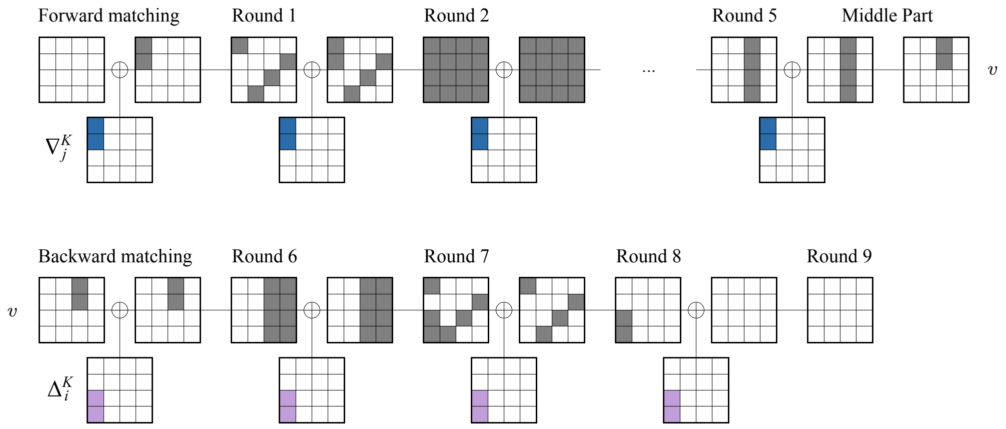


**Fig. 4.** Recomputations for PRINCE$_{core}$ in forward and backward direction.

## 2.4 Complexity of the Attack

PRINCE$_{core}$ uses the $S/S^{-1}$ layer twelve times in the round transformation, which sums up to $12 \cdot 16 = 192$ S-boxes in the full cipher. For a key group of $2^{16}$ keys, $C_{recomp}$ is therefore equal to $2^{16} \cdot \frac{77}{192} \approx 2^{14.68}$ full encryptions. The effort for constructing one biclique $C_{biclique}$ is equivalent to $2^9$ computations of one out of twelve rounds or $2^{5.42}$ full encryptions. Further, the complexity of precomputations is given by computing eleven out of eleven rounds $2^8$ times or $2^{7.88}$ encryptions. Thus, the total effort sums up to

$$2^{48} \cdot (2^{5.42} + 2^{7.88} + 2^{14.68} + 2^8 + 2^8) = 2^{62.72}.$$

The data complexity of this attack is $2^{40}$ and we need to store $2^8$ texts per group for the precomputations.

### 2.5 Discussion

The designers of PRINCE considered the resistance to manyfold classical attacks in their security analysis [2], including meet-in-the-middle and biclique cryptanalysis. They argued that due to the linear layer, the cipher achieves full diffusion after two rounds, which limits meet-in-the-middle attacks to maximally four rounds (cf. [2, Appendix C.4]). Further, according to their studies, the authors stated that independent bicliques could be constructed for up to two rounds which limited biclique attacks without an exhaustive component to six rounds and, since biclique attacks over the entire cipher had to cover four additional rounds, the exhaustive part of an attack on the full PRINCE$_{core}$ would be dominant and the advantage would not be significantly more than a factor of 2.

In this work, our study confirms their statements regarding the length of bicliques as we could construct bicliques over two rounds at most. For the shown biclique attack, we decided to construct a biclique over a single round to lower the data complexity, since two applications of the linear layer would make the entire state active. Though, a complete application of an attack in this work is an essential help to better understanding. In the matching part, the key differences lead to full diffusion after two rounds, but, due to the partial matching, we have to consider only three rounds (3-5) in full. Further, our computation from rounds 10-7 in inverse direction allows us to pass the full round 10 without any recomputations. By extending the biclique to two rounds, the attack complexity could be reduced even further; in an optimal case, the matching part would require one round or 16 S-boxes less which would decrease the recomputation effort to about $2^{16} \cdot \frac{61}{192} \approx 2^{62.35}$ full encryptions.

The shown biclique attack is fairly a straight-forward application of the generic biclique approach to PRINCE which does not violate the security claimed by designers.

## 3 Differential Cryptanalysis on Round-reduced PRINCE$_{core}$

We are interested in studying differences between the trail which propagate from the middle to the plaintext $P$ and the trail from the middle to the ciphertext $C$. We target only symmetrically reduced versions of PRINCE$_{core}$. So, we call a reduced version of the cipher with

- a pre-whitening with $k_1$ and $RC_0$,
- $n$ forward rounds,
- the middle part,
- $n$ backward rounds and
- a post-whitening with $k_1$ and $RC_{11}$

an $n$-x-$n$ construction. Prior, we note a few basic observations on which our analysis relies.

1. There are $2^8$ out of the $2^{16}$ 16-bit values which pass the multiplication with $M_0$ without change. Equivalently, there are $2^8$ 16-bit values which pass the multiplication with $M_1$ without affect. Hence, there are $(2^8)^4 = 2^{32}$ possible 64-bit values $x = (x_1 \| x_2 \| x_3 \| x_4)$ for which

the application of the entire $M'$ layer has no effect:

$$M'(x) = M'(x_1\|x_2\|x_3\|x_4) = M_0(x_1)\|M_1(x_2)\|M_1(x_3)\|M_0(x_4) = (x_1\|x_2\|x_3\|x_4).$$

As a consequence, these $2^{32}$ values, pass the entire middle part and the wrapping key addition $(AK \circ S^{-1} \circ M' \circ S \circ AK)$ without any changes, since the inverse S-box layer and the key additions at the end just reverse the actions of their forward applications.

2. As an adaptation, we studied if there are any input values $x$ to the middle part

$$x \xrightarrow{S^{-1}\circ M'\circ S} x \oplus \alpha,$$

where $\alpha = (c0ac\|29b7\|c97c\|50dd)$. We found that there are no such values $x$, because there is no input value which has a difference with its output of $0x50dd$ in the last four nibbles. So, we studied the number of values $x'$ which have a truncated difference $\alpha' \approx \alpha$ with their middle part outputs

$$x' \xrightarrow{S^{-1}\circ M'\circ S} x' \oplus \alpha',$$

where
- $\alpha' = (c0a\cdot\|29\cdot 7\|c\cdot 7c\|\cdot 0dd)$ has 17920 solutions,
- $\alpha' = (\cdot 0ac\|29b\cdot\|c9\cdot c\|5\cdot dd)$ has 38720 solutions,
- $\alpha' = (c\cdot ac\|\cdot 9b7\|c97\cdot\|50\cdot d)$ has 26880 solutions,
- $\alpha' = (c0\cdot c\|2\cdot b7\|\cdot 97c\|50d\cdot)$ has 92160 solutions,

and $(\cdot)$ can denote any 4-bit value.

3. The S-box of PRINCE$_{core}$ has a bias of $2^{-1.27}$ (and so has its inverse). Each of the 16 possible 4-bit input differences $\beta = a \oplus b$ $(a, b, \beta \in \{0,1\}^4)$ maps, depending on the value of $\beta$, only to 1, 6, 7, or 8 possible output differences $\gamma = S[a] \oplus S[b]$. These are summarized in Table 2 in the Appendix A. In total, the S-box allows 106 out of 256 possible input-output trails, *i.e.,* a given arbitrary 4-bit input difference $\beta$ will lead to only $\frac{106}{16} = 6.625 \approx 2^{2.73}$ output differences in average. From the maximum of 8 solutions follows, that a given 64-bit input difference $\delta$ can map to $8^{16} = 2^{48}$ output differences at most, and to $(2^{2.73})^{16} \approx 2^{43.65}$ in average.

4. In the case we are given a valid S-box trail $\beta \xrightarrow{S^{-1}} \gamma$ for unknown $\beta, \gamma \in \{0,4\}^4$, the number of quartets $(a, b, c, d \in \{0,4\}^4)$ which can built it,

$$a \oplus b = \beta \xrightarrow{S^{-1}} \gamma = c \oplus d,$$

is only 2.42 in average. From the Appendix A, we can see that there are either one, two or four solutions per trail. There is once a single solution $(\beta = 0 \Leftrightarrow \gamma = 0)$, 90 times two solutions and 16 times four valid solutions, which is equivalent to $\frac{1+90\cdot 2+15\cdot 4}{1+90+16} = \frac{256}{106} \approx 2.42 \approx 2^{1.27}$ solutions in average.

### 3.1 Inside-Out Attack on Two Rounds of PRINCE$_{core}$

Here, we propose a differential trail for a 1-x-1 construction reduced to one forward round, the middle part, and one backward round, as illustrated in Figure 5. The differential trail is given by:

$$\xrightarrow[p=2^{-32}]{S^{-1}\circ M'\circ S} \underset{\Delta(\#0)}{0} \xrightarrow[p=1]{(\oplus k_1)\circ(\oplus RC_5/RC_6)} \underset{\Delta(\#1)}{\alpha} \xrightarrow[p=1]{M^{-1}} \underset{\Delta(\#2)}{\beta} \xrightarrow[p=1]{S^{-1}} \underset{\Delta(\#3)}{\gamma} \xrightarrow[p=1]{(\oplus RC_0/RC_{11})\circ(\oplus k_1)} \underset{\Delta(\#4)}{\gamma \oplus \alpha.}$$

The steps of the attack can be described as follows:

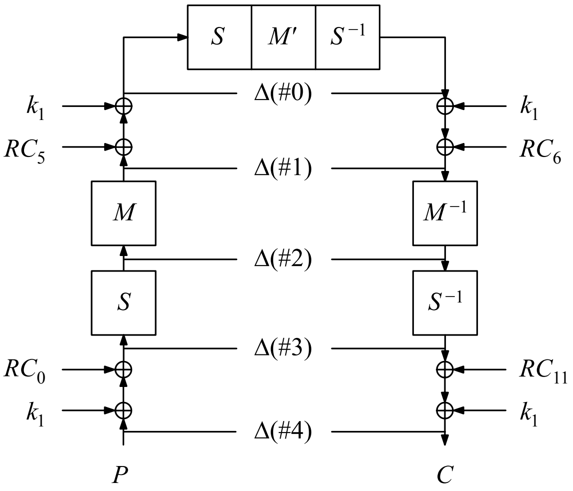


**Fig. 5.** States in differential trails for PRINCE$_{core}$.

1. **Preparation:** It applies that the inverse matrix multiplication with $M'^{-1}$ transforms $\alpha = (c0ac\|29b7\|c97c\|50dd)$ to $\beta = (42a3\|356a\|5d3a\|0fe3)$ with probability 1:

$$Pr[\alpha \xrightarrow[p=1]{M'^{-1}} \beta] = 1$$

   As we can see from Table 2 in Appendix A, for the difference $\beta = (42a \dots e3)$ there are $6 \cdot 8 \cdot 7 \cdot \dots \cdot 7 \cdot 6 \approx 2^{41.38}$ possible output differences $\gamma$. Since every 4-bit S-box operates independently, we only need to store $6 + 8 + 7 + \dots + 7 + 6 = 103$ 4-bit values.
2. **Oracle queries:** Choose $2^{32}$ different plaintexts $P_i$ and request the corresponding ciphertexts $C_i$ from an encryption oracle. From Observation (1) follows that we can expect to have one pair $(P_i, C_i)$ for which applies $\Delta(\#0) = 0$, *i.e.,* that its values before and after the middle part are equal.
3. **Derive $\gamma_i$:** For all pairs $(P_i, C_i)$ derive $\gamma_i = \Delta(\#3)$:

$$\gamma_i \oplus \alpha = P_i \oplus C_i \quad \rightsquigarrow \quad \gamma_i = P_i \oplus C_i \oplus \alpha.$$

4. **Discard mismatching pairs:** Discard all pairs $(P_i, C_i)$, if $\gamma_i$ does not belong to the $2^{41.38}$ valid differences. For any of our 64-bit differences $\gamma_i$, we can expect that it belongs to the valid differences by random with a probability of $2^{-64} \cdot 2^{41.38}$, so we expect to have

$$2^{32} \cdot 2^{-64} \cdot 2^{41.38} \approx 2^{9.38}$$

   remaining pairs.
5. **Derive solutions for the S-box trails:** For every nibble in every remaining $P_i$, we lookup all possible solutions $a, b, c, d \in \{0,1\}^4$ with $a \oplus b = \gamma_i \xrightarrow{S} \beta = c \oplus d$.
   There are $(2^{1.27})^{16} \approx 2^{20.35}$ solutions in average for every $P_i$ for state $(\#3)_i$, which we enumerate by $(\#3)_i^j$:

$$(\#3)_i^j = P_i \oplus RC_0 \oplus k_1.$$

   So, we have $2^{9.38} \cdot 2^{20.35} \approx 2^{29.73}$ potential values $(\#3)_i^j$.
6. **Derive key candidates:** For every value $(\#3)_i^j$, we can derive a key candidate $(k_1)_i^j$:

$$(k_1)_i^j = P_i \oplus RC_0 \oplus (\#3)_i^j.$$

7. **Eliminate false positives:** For every key candidate $(k_1)_i^j$, check if it is the correct key by encrypting any plaintext $P_j \neq P_i$, from which we know the correct encryption result $C_j$ from the oracle:
$$E_{(k_1)_i^j}(P_j) \overset{?}{=} C_j, \quad j \neq i.$$

The computational complexity of the attack is composed by $2^{32}$ encryptions of the oracle, $2^{32}$ XORs to compute $\gamma_i$, $2^{32}$ simple lookups if the difference is valid, $2^{29.73}$ XORs to derive the $k_1$ candidates and finally, $2^{29.73}$ full encryptions to identify the correct key. Since we have eight XORs in our reduced 1-x-1 construction, we can overestimate the effort for one XOR- or lookup-operation with $2^{-3}$ encryptions each. The full complexity is given by

$$2^{32} + 2^{32-3} + 2^{29.73-3} + 2^{29.73} \leq 2^{32.44}$$

full encryptions. The memory complexity is given by storing $2^{32}$ text pairs $(P, C)$ and the data complexity is given by $2^{32}$ chosen plaintexts.

### 3.2 Inside-Out Attack on Four Rounds of PRINCE$_{core}$

In this section, we target to create a similar attack on a 2-x-2 construction, as shown in Figure 6. The differential trail is given by:

$$\begin{tikzcd}
 & \Delta(\#0) & \Delta(\#1) & \Delta(\#2) & \Delta(\#3) \\
{} \arrow[r, "S^{-1}\circ M'\circ S", "p=1"'] & ? \arrow[r, "(\oplus k_1)\circ(\oplus RC_5/RC_6)", "p=1"'] & ? \arrow[r, "M^{-1}", "p=1"'] & ? \arrow[r, "S^{-1}", "p\leq 2^{-49}"'] & \alpha' \\
{} \arrow[r, "(\oplus k_1)\circ(\oplus RC_4/RC_7)", "p=1"'] & \alpha'' \arrow[r, "M^{-1}", "p=1"'] & \beta \arrow[r, "S^{-1}", "p=1"'] & \gamma \arrow[r, "(\oplus RC_0/RC_{11})\circ(\oplus k_1)", "p=1"'] & \gamma \oplus \alpha, \\
 & \Delta(\#4) & \Delta(\#5) & \Delta(\#6) & \Delta(\#7)
\end{tikzcd}$$

with

$$\begin{aligned}
\alpha' &= (c0a \cdot \| 29 \cdot 7 \| c \cdot 7c \| \cdot 0dd) \\
\alpha'' &= (000 \cdot \| 00 \cdot 0 \| 0 \cdot 00 \| \cdot 000) \\
\beta &= (000 \cdot \| 000 \cdot \| 000 \cdot \| 000 \cdot) \\
\gamma &= (000 \cdot \| 000 \cdot \| 000 \cdot \| 000 \cdot)
\end{aligned}$$

The steps of the attack can then be described as follows:

1. **Oracle queries:** Choose $2^{48}$ different plaintexts $P_i$ and request the corresponding ciphertexts $C_i$ from an encryption oracle. Since 16 bits in $\alpha'$ are not specified, we can expect to have one pair $(P_i, C_i)$ for which applies that $\Delta_i(\#3) = \alpha'$.
2. **Derive $\gamma_i$:** For all pairs $(P_i, C_i)$ derive $\gamma_i$:
$$\gamma_i \oplus \alpha = P_i \oplus C_i \quad \rightsquigarrow \quad \gamma_i = P_i \oplus C_i \oplus \alpha.$$
3. **Discard mismatching pairs:** Discard pair $(P_i, C_i)$, if the three leftmost columns of $\gamma_i$ are not all zeroes. Since we match in 48 bits, we can expect to have only a few pairs for which this criterion is fulfilled. Arguing with the cumulative binomial distribution with $n = 2^{48}, p = 2^{-48}$ and $k = 8$, we can say that we expect $X \leq 8$ matches with probability $Pr[X \leq 8] \geq 0.999$.
4. **Derive solutions for the S-box trails:** Again, for every remaining $P_i$, we can derive the possible solutions $a, b, c, d \in \{0,1\}^4$ with $a \oplus b = \gamma_i \xrightarrow{S} \beta = c \oplus d$. For each of the 12 nibbles in the three leftmost columns of the state $(\#6)_i^j$, we need to consider all 16 possible values per nibble. For the four nibbles in the rightmost column, there are $(2^{1.27})^4 \approx 2^{5.08}$ solutions per pair. So, we can estimate that we need to compute $2^3 \cdot 2^{48} \cdot 2^{5.08} \approx 2^{56.08}$ potential values $(\#6)_i^j$.

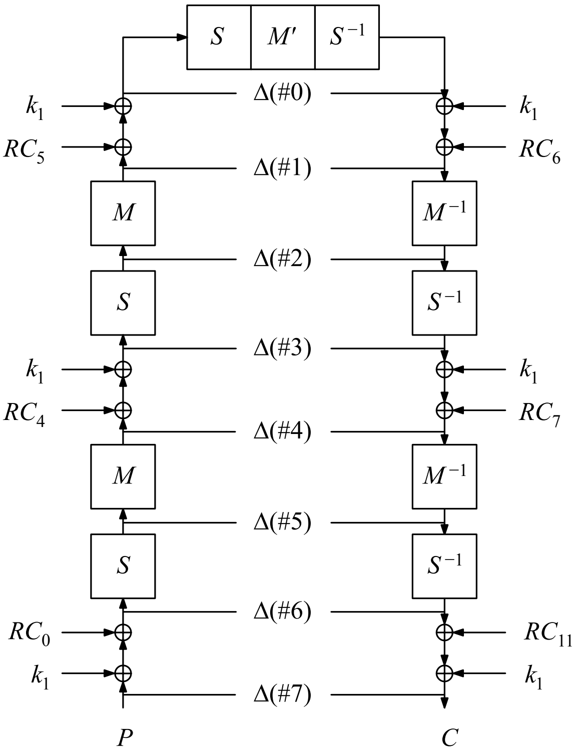


**Fig. 6.** States in differential trails for PRINCE$_{core}$.

5. **Derive key candidates:** For each of the obtained values of the state $(\#6)_i^j$, we can derive a key candidate $k_1$:
$$(k_1)_i^j = P_i \oplus RC_0 \oplus (\#6)_i^j.$$
6. **Eliminate false positives:** For every key candidate, check if it is the correct key by encrypting any plaintext $P_j \neq P_i$, from which we know the correct encryption result $C_j$ from the oracle:
$$E_{(k_1)_i^j}(P_j) \overset{?}{=} C_j, \quad j \neq i.$$

The attack requires $2^{48}$ full encryptions, $2^{48}$ XORs to compute $\gamma_i$, $2^{48}$ lookups, and $2^{56.08}$ XORs to compute $k_1$ and $2^{56.08}$ encryptions to identify the correct key. Again, we upperbound the effort for XORs and lookups with $2^{-3}$ two-round encryptions each. The full complexity is given by
$$2^{48} + 2^{48-3} + 2^{56.08-3} + 2^{56.08} \leq 2^{56.26}$$
full encryptions. The memory complexity is given by storing $2^{48}$ text pairs $(P, C)$ and the data complexity is given by $2^{48}$ chosen plaintexts.

## 4 Conclusion

The design of PRINCE$_{core}$ combines the secure operations of the AES with a small footprint for limited devices. Since 64 bits of the key material are only used in a wrapper, the security of the cipher depends essentially on the security of the core. The core was the target in our attacks, for which the results are summarized in Table 1. At the time of writing, these results are the first ones for PRINCE.

| Type | Rounds | Comp. | Data | Memory |
|---|---|---|---|---|
| | | complexity | complexity (CP) | complexity |
| Differential | 2 | $2^{32.44}$ | $2^{32}$ | $2^{32}$ |
| Differential | 4 | $2^{56.26}$ | $2^{48}$ | $2^{48}$ |
| Biclique | 10 (full) | $2^{62.72}$ | $2^{40}$ | $2^{8}$ |

**Table 1.** Attacks on PRINCE$_{core}$ in this work. CP = chosen plaintexts.

Our biclique analysis of PRINCE$_{core}$ basically confirms the conceptual assumptions of the designers, saying that independent bicliques are limited to two rounds and attacks will reduce the effort for testing the whole search space to around one half. Though, our work in a complete application of an attack can deliver some corrections: first, we gained an advantage of $2^{-1.28}$; second, there are only three rounds which need to be fully considered in the recomputation part; third, an adversary does not need to recompute the round which is next to the biclique. Moreover, we only used a single-round biclique, so there may be more efficient bicliques which can reduce the computational effort even further.

Regarding the field of differential cryptanalysis, the involution structure of the core seems attractive for inside-out attacks which study trails from the middle to the ends. In our work, we demonstrated two (though still very limited) differential trails over two and four rounds, respectively. We highlighted that the current implementation of the multiplication matrix allows $2^{32}$ values to pass the multiplication without affect. Further, we studied the S-box in detail and used its bias to recover the key from given S-box trails. At the end, we note that, neither our biclique nor our differential attacks can violate the claimed security by designers.

## References

1. Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. Biclique Cryptanalysis of the Full AES. Cryptology ePrint Archive, Report 2011/449, 2011. `http://eprint.iacr.org/`.
2. Julia Borghoff, Anne Canteaut, Tim G”uneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalcin. PRINCE – A Low-latency Block Cipher for Pervasive Computing Applications. Cryptology ePrint Archive, Report 2012/591, 2012. `http://eprint.iacr.org/`.
3. Dmitry Khovratovich, Christian Rechberger, and Alexandra Savelieva. Bicliques for Preimages: Attacks on Skein-512 and the SHA-2 Family. Cryptology ePrint Archive, Report 2011/286, 2011. `http://eprint.iacr.org/`.
4. Joe Kilian and Phillip Rogaway. How to Protect DES Against Exhaustive Key Search (an Analysis of DESX). *J. Cryptology*, 14(1):17–35, 2001.

# A Differential Trails for the Inverse S-box of PRINCE$_{core}$

| $\beta \downarrow / \gamma \rightarrow$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f | #solutions |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 solutions |
| 1 | 0 | 4 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 4 | 2 | 0 | 0 | 6 solutions |
| 2 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 2 | 2 | 8 solutions |
| 3 | 0 | 0 | 4 | 0 | 4 | 2 | 2 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 6 solutions |
| 4 | 0 | 2 | 0 | 0 | 2 | 0 | 0 | 0 | 4 | 0 | 2 | 4 | 0 | 0 | 0 | 2 | 6 solutions |
| 5 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 2 | 0 | 4 | 0 | 2 | 0 | 0 | 0 | 7 solutions |
| 6 | 0 | 2 | 0 | 2 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 2 | 0 | 4 | 2 | 7 solutions |
| 7 | 0 | 0 | 2 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 4 | 2 | 0 | 2 | 2 | 2 | 7 solutions |
| 8 | 0 | 4 | 2 | 2 | 2 | 0 | 0 | 2 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 7 solutions |
| 9 | 0 | 2 | 0 | 2 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 0 | 0 | 4 | 0 | 0 | 7 solutions |
| a | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 4 | 0 | 2 | 0 | 0 | 2 | 2 | 0 | 2 | 7 solutions |
| b | 0 | 2 | 0 | 2 | 0 | 2 | 2 | 0 | 2 | 0 | 0 | 2 | 2 | 0 | 2 | 0 | 8 solutions |
| c | 0 | 0 | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 4 | 0 | 2 | 2 | 0 | 2 | 2 | 7 solutions |
| d | 0 | 0 | 4 | 0 | 0 | 2 | 0 | 2 | 2 | 2 | 0 | 0 | 0 | 2 | 2 | 0 | 7 solutions |
| e | 0 | 0 | 2 | 0 | 0 | 0 | 4 | 2 | 0 | 0 | 0 | 2 | 2 | 2 | 0 | 2 | 7 solutions |
| f | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 2 | 0 | 2 | 2 | 2 | 0 | 2 | 2 | 2 | 8 solutions |

**Table 2.** Summary of possible differential trails $\beta \xrightarrow{S^{-1}} \gamma$ for the inverse S-box of PRINCE$_{core}$.