A Detailed Analysis of PRINCE

Team_Illuminati

Department of Computer Science and Engineering
Indian Institute of Technology Bhilai

December 6, 2021

Outline

1. Introduction
2. Cipher Specifications
3. Observations
4. Brownie Point Nominations
5. Conclusion

Motivation

Emerging Requirements :

- Instantaneous Encryption
- Computation of ciphertext with a single clock cycle
- Low Latency
- Low hardware costs
- Low space and time overhead

PRINCE Cipher Fulfils all these requirements

Salient features

- Inspired From the FX construction
- SPN Based
- Inspired from PRESENT (Another Lightweight cipher)
- Balance between Speed/Efficiency and Security
- Considerably Less Area (In terms of gates) Than PRESENT-80 and AES-128
- The Same Core function works both in encryption and decryption (No need of Additional Logic gates)

Construction

The FX-Construction is built using a core function and 2 whitening keys which are used in pre-processing and post-processing

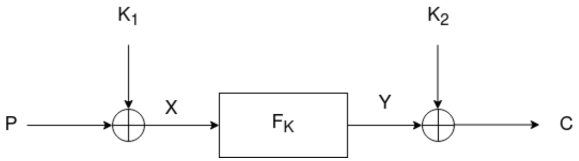
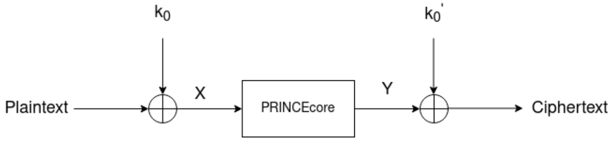


Figure: The FX Construction

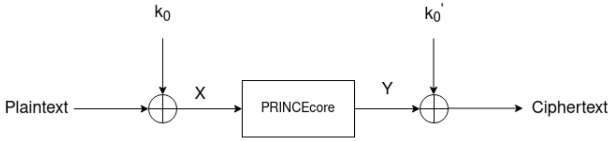
Outline

1. Introduction
2. Cipher Specifications
3. Observations
4. Brownie Point Nominations
5. Conclusion

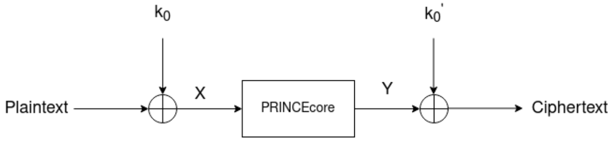
Description of PRINCE

1. PRINCE is a 64-bit block cipher with a 128-bit key consisting of 12 rounds
2. The key is split into two parts of 64 bits each i.e. k = k0||k1
3. Another part of the key k_0' is derived from k_0 by using the following relation $k_0' = (k_0 >>> 1) \bigoplus (k_0 >> 63)$
4. The keys k_0 and k_0' are used are whitening keys and the key k_1 is used as the round key for the core function.
5. 

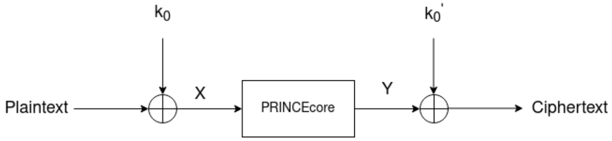
Description of PRINCE

1. PRINCE is a 64-bit block cipher with a 128-bit key consisting of 12 rounds
2. The key is split into two parts of 64 bits each i.e. k = k0||k1
3. Another part of the key k_0' is derived from k_0 by using the following relation $k_0' = (k_0 >>> 1) \bigoplus (k_0 >> 63)$
4. The keys k_0 and k_0' are used are whitening keys and the key k_1 is used as the round key for the core function.
5. 

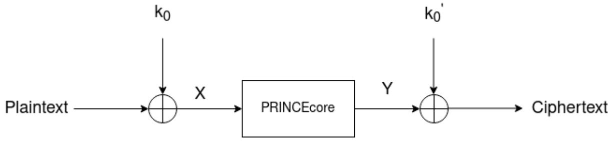
Description of PRINCE

1. PRINCE is a 64-bit block cipher with a 128-bit key consisting of 12 rounds
2. The key is split into two parts of 64 bits each i.e. k = k0||k1
3. Another part of the key k_0' is derived from k_0 by using the following relation $k_0' = (k_0 >>> 1) \bigoplus (k_0 >> 63)$
4. The keys k_0 and k_0' are used are whitening keys and the key k_1 is used as the round key for the core function.
5. 

Description of PRINCE

1. PRINCE is a 64-bit block cipher with a 128-bit key consisting of 12 rounds
2. The key is split into two parts of 64 bits each i.e. $k = k0||k1$
3. Another part of the key k_0' is derived from k_0 by using the following relation $k_0' = (k_0 >>> 1) \bigoplus (k_0 >> 63)$
4. The keys k_0 and k_0' are used are whitening keys and the key k_1 is used as the round key for the core function.
5. 

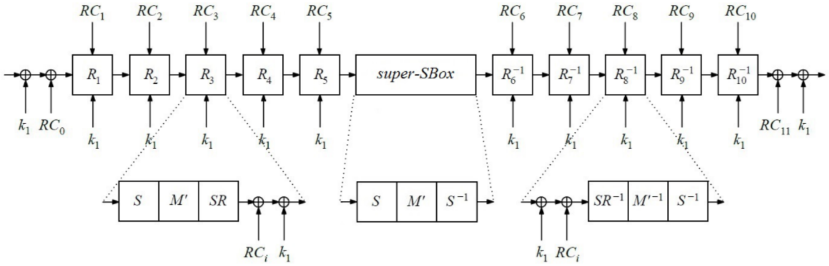
Description of PRINCE

1. PRINCE is a 64-bit block cipher with a 128-bit key consisting of 12 rounds
2. The key is split into two parts of 64 bits each i.e. k = k0||k1
3. Another part of the key k_0' is derived from k_0 by using the following relation $k_0' = (k_0 >>> 1) \bigoplus (k_0 >> 63)$
4. The keys k_0 and k_0' are used are whitening keys and the key k_1 is used as the round key for the core function.
5. 

Rounds of Prince

1. PRINCE consists of 12 rounds which include the following :
   1. First 5 rounds or the "forward rounds"
   2. The middle round which is termed as 2 rounds
   3. The last 5 rounds or the "backward rounds"

.



Specifications of Prince

- S-Layer :- Prince uses a 4-bit S-box. It is as follows :

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
S[x]	b	f	3	2	a	c	9	1	6	7	8	0	e	5	d	4

- Linear Layer :- This layer provides diffusion. It is a combined layer consisting of a shift row followed by a 64×64 matrix multiplication.
- Round Constants :- There are 12 round constants from RC_0 to RC_{11} such that $RC_i \oplus RC_{11-i} = \alpha$.
  Here, $\alpha =$ c0ac29b7c97c50dd
- Key Addition :- The 64-bit k_1 is xored with the state.

Specifications of Prince

- The Round Constants Used are as follows :

RC_0	0000000000000000
RC_1	13198a2e03707344
RC_2	a4093822299f31d0
RC_3	082efa98ec4e6c89
RC_4	452821e638d01377
RC_5	be5466cf34e90c6c
RC_6	7ef84f78fd955cb1
RC_7	85840851f1ac43aa
RC_8	c882d32f25323c54
RC_9	64a51195e0e3610d
RC_{10}	d3b5a399ca0c2399
RC_{11}	c0ac29b7c97c50dd

Specifications of Prince

α Reflection property

- $RC_i \oplus RC_{11-i} = \alpha$
- $RC_1, \ldots, RC_5$ and α have been derived from the fraction part of π

- The Linear Layer M consists of 2 parts :
  - Shift Rows :- This performs a circular shift operation on the rows similar to that used in AES. The mapping of the Shift Row operation is as follows :

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

  - M' Layer :- This is a 64 $\times$ 64 binary matrix representated as follows :

$$M' = \begin{pmatrix} M_0 & 0 & 0 & 0 \\ 0 & M_1 & 0 & 0 \\ 0 & 0 & M_1 & 0 \\ 0 & 0 & 0 & M_0 \end{pmatrix}$$

Specifications of Prince

- The constituent 16 × 16 matrices M_0 and M_1 further contain 4 × 4 matrices as follows :
- $\vec{M}^{(0)} = \begin{bmatrix} \mathrm{M}_0 & \mathrm{M}_1 & \mathrm{M}_2 & \mathrm{M}_3 \\ \mathrm{M}_1 & \mathrm{M}_2 & \mathrm{M}_3 & \mathrm{M}_0 \\ \mathrm{M}_2 & \mathrm{M}_3 & \mathrm{M}_0 & \mathrm{M}_1 \\ \mathrm{M}_3 & \mathrm{M}_0 & \mathrm{M}_1 & \mathrm{M}_2 \end{bmatrix}$
- $\vec{M}^{(1)} = \begin{bmatrix} \mathrm{M}_1 & \mathrm{M}_2 & \mathrm{M}_3 & \mathrm{M}_0 \\ \mathrm{M}_2 & \mathrm{M}_3 & \mathrm{M}_0 & \mathrm{M}_1 \\ \mathrm{M}_3 & \mathrm{M}_0 & \mathrm{M}_1 & \mathrm{M}_2 \\ \mathrm{M}_0 & \mathrm{M}_1 & \mathrm{M}_2 & \mathrm{M}_3 \end{bmatrix}$

Specifications of Prince

- The smaller 4×4 matrices used are as follows :
- $$M_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad M_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
  $$M_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad M_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Point to Note

The Linear Layer Used in Prince is an involution. This implies that it is its own inverse. This fact helps in symmetric decryption with little to no overhead over the original encryption function.

Outline

1 Introduction

2 Cipher Specifications

3 Observations

4 Brownie Point Nominations

5 Conclusion

Differential Cryptanalysis of Round Reduced Prince

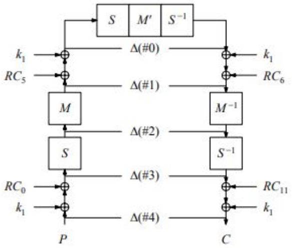
1. The difference is studied between the trail from middle to plaintext and trail from middle to ciphertext.
2. n-x-n architecture
3. 2^{32} 64-bit values for x so that M' has no effect
4. No input values x such that $x \xrightarrow{S^{-1}M'S} x \oplus \alpha$. Hence use truncated difference.
5. Sbox has a bias of $2^{-1.27}$

DDT of Sbox

in/out	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	No. of solutions
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	4	0	0	2	0	2	0	4	2	0	2	0	0	0	0	6
2	0	2	0	4	0	0	0	2	2	0	0	0	0	4	2	0	6
3	0	0	0	0	0	2	2	0	2	2	2	2	2	0	0	2	8
4	0	2	2	4	2	2	0	0	2	0	2	0	0	0	0	0	7
5	0	0	2	2	0	2	0	2	0	2	0	2	2	2	0	0	8
6	0	0	2	2	0	2	2	0	0	2	0	2	0	0	4	0	7
7	0	0	2	0	0	0	2	0	2	0	4	0	0	2	2	2	7
8	0	0	2	0	4	2	0	0	2	2	0	2	0	2	0	0	7
9	0	0	2	2	0	0	0	0	0	2	2	0	4	2	0	2	7
a	0	0	0	2	2	4	0	4	2	0	0	0	0	0	0	2	6
b	0	2	0	0	4	0	0	2	0	0	0	2	2	0	2	2	7
c	0	4	0	0	0	2	2	0	0	0	2	2	2	0	2	0	7
d	0	2	0	0	0	0	0	2	0	4	2	0	0	2	2	2	7
e	0	0	2	0	0	0	4	2	0	0	0	2	2	2	0	2	7
f	0	0	2	0	2	0	2	2	0	0	2	0	2	0	2	2	8

S-box allows 106 out of 256 possible input-output trails. Average non zero values in DDT is $256/106 = 2.415$.

Inside-out attack on 2 rounds



The differential trail

$$\xrightarrow[p=2^{-32}]{S^{-1},M',S} 0 \xrightarrow[p=1]{\oplus k1 \oplus RC_5/RC_6} \alpha \xrightarrow[p=1]{M^{-1}} \beta \xrightarrow[p=1]{S^{-1}} \gamma \xrightarrow[p=1]{\oplus k1 \oplus RC_{11}/RC_0} \gamma \oplus \alpha$$

Inside-out attack on 2 rounds

How the difference propagates:

- **Key and Round addition:**Difference changes by α.
  $\triangle(\#0) = \alpha \oplus \triangle(\#1)$
- **M or M^{-1}:** We can tell with probability 1 the resulting difference
- **Sbox:** Non-linear. Tell how difference propagates based on DDT.

Inside-out attack on 2 rounds

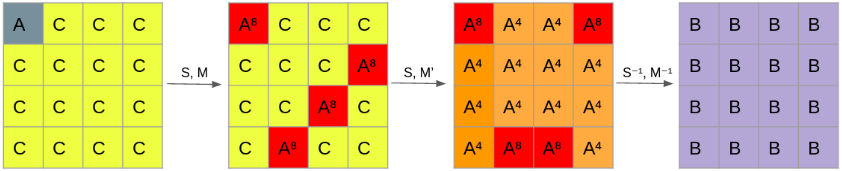
1. Choose 2^{32} plaintexts. We have $\gamma_i \oplus \alpha = P_i \oplus C_i$
2. By applying the M'^1 layer to $\alpha{=}(c0ac||29b7||c97c||50dd)$, we get $\beta{=}$ $(42a3||356a||5d3a||0fe3)$ with probability 1
3. From β we can get potential values of γ. This turns out to be 6x8x..x6$=2^{41.38}$. Filter out plaintext and ciphertext pairs and expected value to remain is $2^{9.38}$
4. 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 $(\#3)_i^j$: $(\#3)_i^j = P_i \oplus RC0 \oplus k1$.
5. For each $(\#3)_i^j$ guess $(k_1)_i^j$ and verify if computed cipher is actual ciphertext C_i
6. Full complexity - $2^{32.44}$, memory complexity - 2^{32} by storing plaintext, ciphertext and data complexity 2^{32} for plaintexts.

Integral Attack on Round Reduced PRINCE

1. We know that integral cryptanalysis works for block ciphers with substitution-permutation network and PRINCE falls in this category.
2. 3.5 round integral distinguisher
3. Notion of Active nibble
4. 4-round attack

Integral Attack on Round Reduced PRINCE

The 3.5 round distinguisher



How M' affects an active nibble?

4-round Integral Attack

Recovery of $k_0' \oplus k_1$

- Take 2^4 plaintexts with one active nibble
- Make a guess for nible of $k_0' \oplus k_1$ and decrypt partially through last Sbox
- Check if nibble is balanced or not
- Repeat this for 16 nibbles
- To remove false positives use 5 sets

4-round Integral Attack

Recovery of k_1

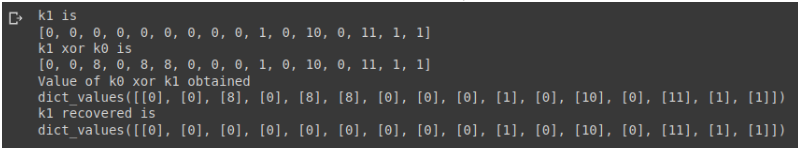
- Start with 5 sets of 2^4 plaintexts with 4 active nibbles and peel of last round using $k_0' \oplus k_1$
- Notion of 2.5 round distinguisher
- Invert linear layer and partially decrypt through 1 sbox
- Check if nibble is balanced
- Data complexity is 2x5x$2^4 \approx 2^7$, Time complexity is 16x2x5x$2^4 \approx 2^{11}$

Outline

1. Introduction
2. Cipher Specifications
3. Observations
4. Brownie Point Nominations
5. Conclusion

4-Round Integral Attack

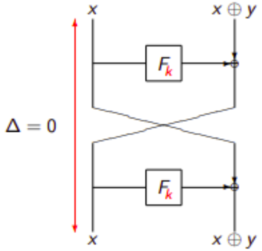
We have implemented the attack in python.



```
k1 is
[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 10, 0, 11, 1, 1]
k1 xor k0 is
[0, 0, 8, 0, 8, 8, 0, 0, 0, 1, 0, 10, 0, 11, 1, 1]
Value of k0 xor k1 obtained
dict_values([[0], [0], [8], [0], [8], [8], [0], [0], [0], [1], [0], [10], [0], [11], [1], [1]])
k1 recovered is
dict_values([[0], [0], [0], [0], [0], [0], [0], [0], [0], [1], [0], [10], [0], [11], [1], [1]])
```

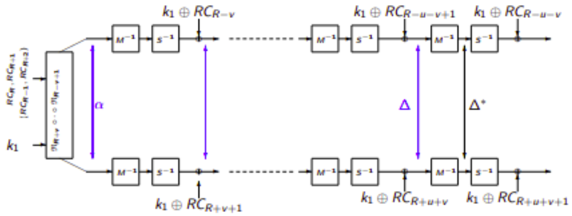
$\alpha-$Reflection.

- Previous Works on Reflection attacks
- It has been applied on some ciphers and hash functions with



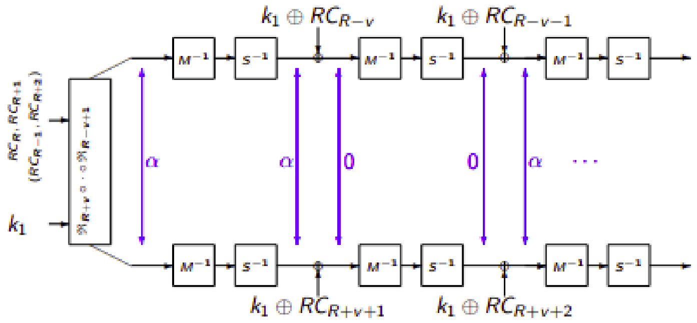
  Feistel construction
- Using Probabilistic approach rather than deterministic approach.

$\alpha-$Reflection.



- To maximize P_c, we use
  1. Cancellation idea.
  2. Branch and Bound Algorithm.

Cancellation Idea



- $\rho = Pr_x[S(X) \oplus S(X+\alpha)] = M^{-1}(\alpha)$ there is an iterative characterstic over four rounds of PRINCE cipher.

Cancellation idea vs Branch and Bound Algorithm

- Cancellation Idea

α	Δ^*	w(Δ^*)	${}^{\mathcal{P}}C_4$	Data Compl.	Time Compl
0x8400400800000000	0x8800400400000000	4	2^{-22}	$2^{57.95}$	$2^{71.37}$
0x8040000040800000	0x8080000040400000	4	2^{-22}	$2^{57.95}$	$2^{71.37}$
0x0000408000008040	0x0000404000008080	4	2^{-22}	$2^{57.95}$	$2^{71.37}$
0x0000000048008004	0x0000000044008008	4	2^{-22}	$2^{57.95}$	$2^{71.37}$
0x0000440040040000	0x0000440040040000	4	2^{-22}	$2^{60.27}$	$2^{73.69}$
0x8008000000008800	0x8008000000008800	4	2^{-22}	$2^{60.27}$	$2^{73.69}$

- Branch and Bound Algorithm

α	Δ^*	w(Δ^*)	${}^{\mathcal{P}}C_4$	Data Compl.	Time Compl
0x0108088088010018	0x0000001008000495	5	2^{-26}	$2^{62.78}$	$2^{80.2}$
0x0088188080018010	0x00000100c09d0008	5	2^{-26}	$2^{62.78}$	$2^{80.2}$
0x0108088088010018	0x000000100800d8cc	6	2^{-26}	$2^{62.83}$	$2^{84.25}$
0x0001111011010011	0x1101100110000100	7	2^{-28}	$2^{63.45}$ (a=32)	$2^{88.87}$

$\alpha-$Reflection Property

- PRINCE Cipher has a symmetric nature.
- $RC_i \oplus RC_{11-i} =$ constant, where RC is round constant, and $0 \leq i \leq 11$
- For a key$(k_0||k_0^{'}||k_1)$ $D_{k_0||k_0^{'}||k_1}(.)=E_{k_0,k_0^{'},k_1+\alpha}(.)$

Impact of construction implementing $\alpha-$Reflection Property

- If the decomposition of core cipher is independent from the key, then use the attack consisting of two plaintext-ciphertext pairs (m, c) $(m^{'}, c^{'})$ such that $m \oplus c = m^{'} \oplus c^{'}$ where, $m^{'} = E^{-1}_{k_0, k_0^{'}, k_1}(m \oplus k_0 \oplus k_2)$
- Such a collision could be found if the attacker has an access to $2^{\frac{n+1}{2}}$ known plaintext-ciphertext pairs and provides a value of $k_0 \oplus k_2$

Impact of construction implementing $\alpha-$Reflection Property

- A more relevant attack method consists in using the fact that the core cipher may have a peculiar cycle decomposition for some weak key.
- It is worth noticing that this attack applies to DESX and allows to detect the use of the four weak keys of DES for which DES is an involution.
- For the class of keys such that $k_1^{'} = k_1 \oplus \alpha$, it holds that $F^1_{(k_1||k_1^{'})} = F_{(k1||k_1^{'})}$, that is, the core cipher is an involution. This class of weak keys can then be easily detected.

Outline

1. Introduction
2. Cipher Specifications
3. Observations
4. Brownie Point Nominations
5. Conclusion

Conclusion

- Prince uses FX construction. k_0 and k_0' are used as whitening keys whereas k_1 is the 64-bit key for a 12-round block cipher referred to as $PRINCE_{CORE}$.
- One of the most critical and expensive operations of the cipher is the substitution, where we use the same Sbox 16 times (rather than having 16 different Sboxes). Therefore, the implementation of PRINCE started with a search for the most suitable Sbox for the target design specifications.

Conclusion

- In implementing the reflexive property, we do not consider related key-attacks here in the classical sense of enlarging the power of an adversary. But without a careful choice, the construction we used for implementing the reflection property might result in key-recovery attacks for certain weak-key classes, as soon as the core cipher is vulnerable to related key-attacks

Thanks

Team Members

- Name - Chodipilli Yadava Kishore, Roll No. - 11940310
- Name - Mulukala Vivek, Roll No. - 11940720
- Name - Srilekha Kadambala, Roll No. - 11941190

Implementation Info

- Github Link: `https://github.com/srilekhaK9120/PRINCE`