PRINCE CIPHER
Term Paper by Team -.././cipher

Amruta Gokhale[1] (12040140)
Ankita Kumari[2] (12040220)
Nidhi Sinchana SR[3] (12040970)

Indian Institute of Technology,Bhilai, amrutagokhale@iitbhilai.ac.in
Indian Institute of Technology,Bhilai, ankitakumari@iitbhilai.ac.in
Indian Institute of Technology,Bhilai, nidhisinchana@iitbhilai.ac.in

Abstract. In this paper, we have the detailed description and implementation of PRINCE Cipher.We also have the knowledge of theoretical and practical attacks which can be implemented on our cipher.It allows encryption of data within one clock cycle. It holds that decryption for one key parallels to encryption with related key. This property which is referred to as $\alpha - reflection$ is of independent interest and we demonstrate its robustness against generic attacks.

Keywords: α-reflection · Differential · Integral · FX Construction

1 Introduction

A need of low-cost cryptosystems for several fast-growing applications has drawn great attention to the area of lightweight cryptographic primitives.A good trade-off between security and efficiency is a particularly challenging task. Some well established algorithms may not meet the basic requirements of constrained devices — low cost hardware implementation, low power usage and latency.Recently, a new lightweight block cipher called PRINCE has been proposed.This cipher was designed to reach an extremely low-latency encryption and instant response time.Also it takes only one clock for execution of their cipher unlike previous implementations.

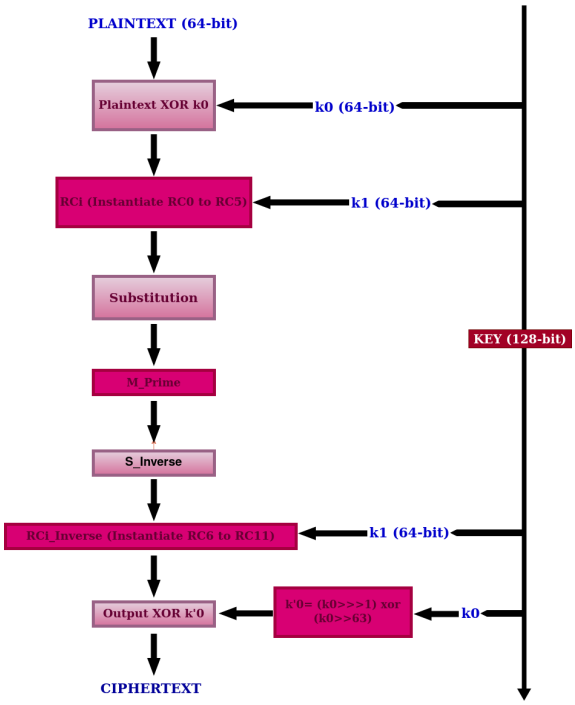
2 Prince Cipher and its features

The PRINCE cipher is a Substitution-Permutation Network (SPN) that iterates over a fixed number of rounds. However, unlike AES, PRINCE only comes in one key size: 128 bits. Moreover, the amount of data which is encrypted is fairly small — only 64 bits (though this is typical of a lightweight cipher). The reason we are encrypting half the block size is actually because all of our operations only work on nibbles.

- ✣ SPN based
- ✣ Inspired from FX construction
- ✣ Inspired from PRESENT (Another lightweight cipher)
- ✣ Balance between Speed/Efficiency and security
- ✣ Considerably Less Area than PRESENT-80 and AES-128

- The same Core function works both in encryption and decryption (No need of Additional Logic gates)

3 Block Diagram

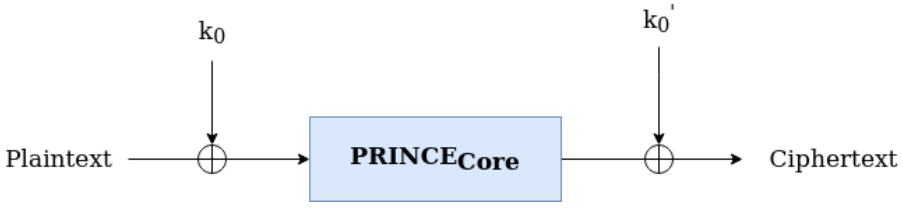


4 Description of Prince Cipher

4.1 Construction

The key is split into two parts of 64 bits each, k = k0||k1 and extended to 192 bits by the mapping.The FX-Construction is built using a core function and 2 whitening keys are used in pre-processing and post-processing.

$$(k_0||k_1) \rightarrow (k_0||k_0^{'}||k_1) := (k_0||(k_0 >>> 1) \oplus (k_0 >> 63)||k_1)$$



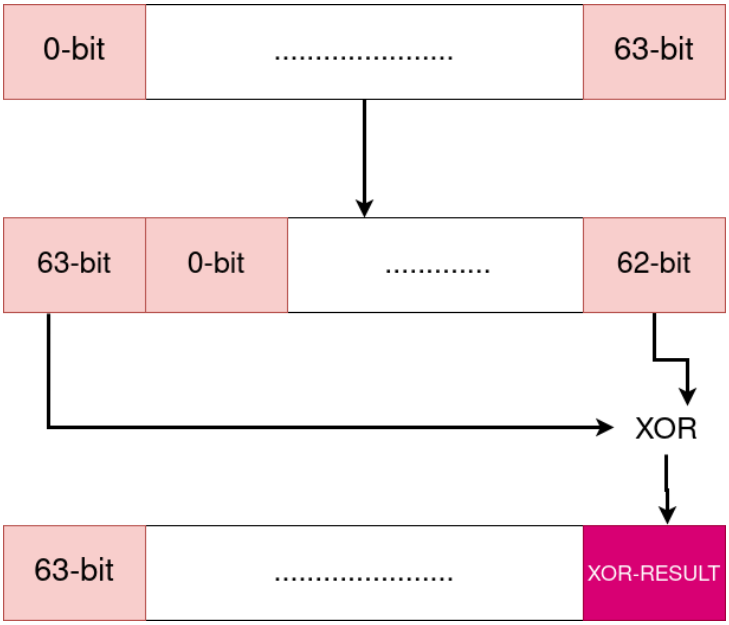


Figure 1: Key Whitening

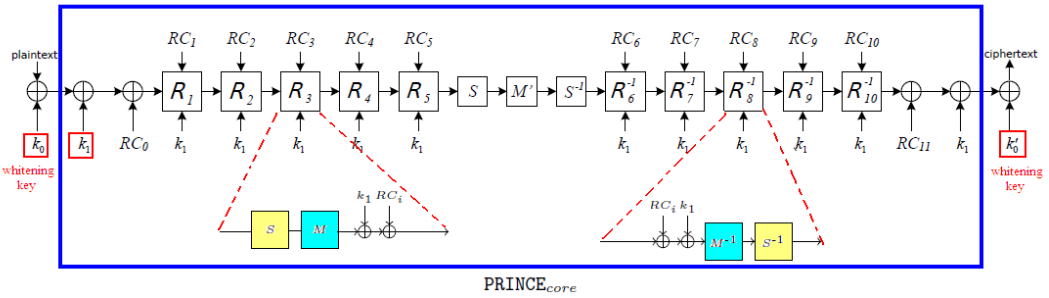
4.2 Rounds of Prince

❑ PRINCE consists of 12 rounds which include the following:

→ First 5 rounds ot the "Forward rounds"

→ The middle round which is termed as 2 rounds

→ The last 5 rounds or the "backward rounds"



Each round of PRINCEcore consist of a key addition, an Sbox-layer,a linear layer, and the addition of a round constant.

❶ Key Addition(k_i): Each round function utilizes basic key addition i.e., $\oplus$ with the 64-bit subkey.

❷ S-layer:Prince cipher uses 4-bit S-box.

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

❸ Linear Layer:This layer provides diffusion.It is combined layer consisting of a shift row followed by 64x64 matrix multiplication.

❹ Round Constants: There are 12 round constants from RC_0 to RC_{11} such that $RC_i \oplus RC_{11-i} = \alpha$.

RC_0	00000000000000000
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

5 Implementing Prince Cipher

5.1 Choosing the Sbox

Choosing the Sbox according to the area and critical path is important because it leads to decrease the cost of encryption.In order to ensure the security of the resulting design,an Sbox $S : F_2^4 \to F_2^4$ for the Prince Cipher has to fulfill the following criteria :

✓ The maximal probability of a differential is $\frac{1}{4}$.

✓ There are exactly 15 differentials with probability $\frac{1}{4}$.

✓ The maximal absolute bias of a linear approximation is $\frac{1}{4}$.

✓ here are exactly 30 linear approximations with absolute bias $\frac{1}{4}$.

There are 8 Sboxes that follow above criteria which are given below:

S0	0	1	2	D	4	7	F	6	8	C	5	3	A	E	B	9
S1	0	1	2	D	4	7	F	6	8	C	9	B	A	E	5	3
S2	0	1	2	D	4	7	F	6	8	C	B	9	A	E	3	5
S3	0	1	2	D	4	7	F	6	8	C	B	9	A	E	5	3
S4	0	1	2	D	4	7	F	6	8	C	E	B	A	9	3	5
S5	0	1	2	D	4	7	F	6	8	E	B	A	5	9	C	3
S6	0	1	2	D	4	7	F	6	8	E	B	A	9	3	C	5
S7	0	1	2	D	4	7	F	6	8	E	C	9	5	B	A	3

$\alpha-$ reflection

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

For a key $(k_0||k_0^{'}||k_1)$, $D_{(k_0||k_0^{'}||k_1)}(.) = E_{(k_0^{'}||k_0||k_1)}(.)$

Therefore same function can be used for encryption as well as decryption with a little bit of change.

5.2 Linear Layer

The linear layer M includes the multiplication by a 64x64 matrix $M^{'}$ and a Shift Rows (SR) transformation.

$$M = SR.M^{'}$$

$$M^{'}(x_0||x_1||x_2||x_3) = M_0.(x_0)||M_1.(x_1)||M_1.(x_2)||M_0.(x_3)$$

Above equation explains how the multiplication with the matrix M ' is performed, where M_0 and M_1 are two different matrixes of 16x16 that have the following property: after the multiplication, each output bit depends only on three bits of the input value.$(x_0||x_1||x_2||x_3)$ represents the 64-bit state.

Shift Rows permutes the 16 nibbles according to Table given below:

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

6 Attacks

6.1 Integral Crpytanalysis

6.1.1 4-Round Integral Attack

We are performing the integral attack on round-reduced PRINCE.

➔ **FOR EVEN ROUNDS**: Add same number of rounds before and after the middle rounds.

➔ **FOR ODD ROUNDS**: Add one extra round at the start.

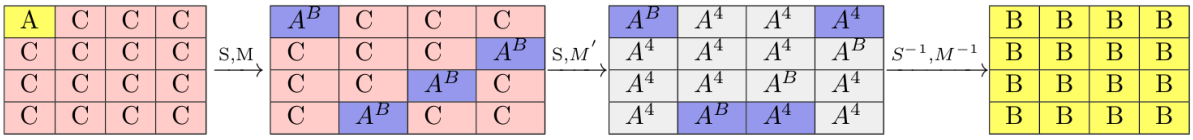
We start with one of the sixteen nibbles as active in the plaintext. Then, we estimate the key with the help of partial decryption since we know that all nibbles are balanced after 3.5 rounds. Hence start with the following:

➔ 1-nibble containing the 'ALL PROPERTY'

➔ remaining containing the 'CONSTANT PROPERTY'

➔ Ending with all balanced nibbles at the end of 3.5 rounds

- ➔ Balance property after 3.5 rounds will be destoyed because of the transformation done by S-Box. We guess the value of (k0 $\oplus$ k1)(let us say 'y') and undergo partial decryption till the final S-Box to verify if all nibbles are balanced
- ➔ We go with 5 sets of 16 plaintexts each and recognize that particular y which is present in right key guesses for all sets
- ➔ After identifying y, we should get $k_1 \oplus k_0$
- ➔ Beginning with 4 active nibbles as shown in the diagram below in the second block, again we go with 5 sets of 16 plaintexts
- ➔ Now, with the identified y value, remove fourth round by performing XOR with the ciphertext.
- ➔ Next, perform the linear layer inversion and for all possible key guesses of nibbles in k_1, undergo partial decryption using 1st S-Box
- ➔ The one which provides a **balanced nibble** is the correct nibble key guess
- ➔ Then we recover $k_0^{'}$ using y and k_1
- ➔ We can recover k_0 from $k_0^{'}$ by performing a set of *linear operations*.



Data Complexity: $2 \times 5 \times 2^4 = 2^7$ since we have chosen five sets of 24 plaintexts two times in the attack above.

Time Complexity:$16 \times 2 \times 5 \times 2^4 = 2^{11}$. Since Each alteration of S-box is one operation; One for each nibble, S-Box is performed sixteen times for each plaintext. Total Plaintexts are $2 \times 5 \times 2^4$.

6.2 Differential Attack

In differential attacks, we study between the trail from middle to plaintext and trail from middle to ciphertext. This Version has a n–x–n construction. Important points about this are given below:

- ➔ **Pre-whitening** with k_1 and RC_0
- ➔ n-forward rounds
- ➔ Super SBox part [S, $M^{'}$, S^{-1}] (middle part)
- ➔ n-backward rounds
- ➔ **Pre-whitening** with k_1 and RC_{11}
  - ✦ There exists four nibbles (2^8) each which pass through M (0) and M (1) without change
  - ✦ Hence, there are 2^{32} states which can go unaffected through $M^{'}$-layer

6.2.1 SBOX ANALYSIS

➔ DDT for S-Box of Prince Cipher:

```
[[16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  4  0  0  2  0  2  0  4  2  0  2  0  0  0  0]
 [ 0  2  0  4  0  0  0  2  2  0  0  0  0  4  2  0]
 [ 0  0  0  0  0  2  2  0  2  2  2  2  2  0  0  2]
 [ 0  2  2  4  2  2  0  0  2  0  2  0  0  0  0  0]
 [ 0  0  2  2  0  2  0  2  0  2  0  2  2  2  0  0]
 [ 0  0  2  2  0  2  2  0  0  2  0  2  0  0  4  0]
 [ 0  0  2  0  0  0  2  0  2  0  4  0  0  2  2  2]
 [ 0  0  2  0  4  2  0  0  2  2  0  2  0  2  0  0]
 [ 0  0  2  2  0  0  0  0  0  2  2  0  4  2  0  2]
 [ 0  0  0  2  2  4  0  4  2  0  0  0  0  0  0  2]
 [ 0  2  0  0  4  0  0  2  0  0  0  2  2  0  2  2]
 [ 0  4  0  0  0  2  2  0  0  0  2  2  2  0  2  0]
 [ 0  2  0  0  0  0  0  2  0  4  2  0  0  2  2  2]
 [ 0  0  2  0  0  0  4  2  0  0  0  2  2  2  0  2]
 [ 0  0  2  0  2  0  2  2  0  0  2  0  2  0  2  2]]
```

```
For input difference 0 --> Output difference is 1
For input difference 1 --> Output difference is 6
For input difference 2 --> Output difference is 6
For input difference 3 --> Output difference is 8
For input difference 4 --> Output difference is 7
For input difference 5 --> Output difference is 8
For input difference 6 --> Output difference is 7
For input difference 7 --> Output difference is 7
For input difference 8 --> Output difference is 7
For input difference 9 --> Output difference is 7
For input difference 10 --> Output difference is 6
For input difference 11 --> Output difference is 7
For input difference 12 --> Output difference is 7
For input difference 13 --> Output difference is 7
For input difference 14 --> Output difference is 7
For input difference 15 --> Output difference is 8
```

The differential uniformity is 4 and differential branch number is 2.

➔ LAT for S-Box of Prince Cipher:

```
[[ 8  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 -2 -2  2 -2  0  4 -4  0 -2  2 -2 -2  0  0]
 [ 0  0 -4  0 -4  0  0  0 -2 -2  2 -2 -2  2  2  2]
 [ 0  4 -2 -2  2 -2  0  0  2  2  4  0  0  0  2 -2]
 [ 0  0 -4  4  2  2  2  2  2  2 -2 -2  0  0  0  0]
 [ 0  0  2  2  0 -4 -2  2  2 -2  0 -4 -2 -2  0  0]
 [ 0 -4  0  0 -2 -2  2 -2  0  4  0  0 -2 -2  2 -2]
 [ 0  0  2 -2  0  0 -2  2  0  4 -2 -2  0  4  2  2]
 [ 0 -2 -2  0  4 -2 -2 -4 -2  0  0 -2  2  0  0  2]
 [ 0 -2  0  2  2  0 -2  0  2  0  2  4 -4  2  0  2]
 [ 0  2  2  4  0  2 -2  0 -4  2  2  0  0 -2  2  0]
 [ 0 -2  0 -2  2  4 -2  0  0 -2  0 -2 -2  0  2 -4]
 [ 0 -2 -2  0 -2  0 -4  2  0  2  2  0  2  0 -4 -2]
 [ 0 -2  0  2  0 -2  0  2  0 -2  0  2  4  2  4 -2]
 [ 0 -2  2  0  2  0  4  2 -2  0  4 -2  0  2 -2  0]
 [ 0  2  0  2  0 -2  0 -2 -2  0 -2  0 -2  4 -2 -4]]
```

```
For input difference 0 --> Output difference is 1
For input difference 1 --> Output difference is 10
For input difference 2 --> Output difference is 10
For input difference 3 --> Output difference is 10
For input difference 4 --> Output difference is 10
For input difference 5 --> Output difference is 10
For input difference 6 --> Output difference is 10
For input difference 7 --> Output difference is 10
For input difference 8 --> Output difference is 10
For input difference 9 --> Output difference is 10
For input difference 10 --> Output difference is 10
For input difference 11 --> Output difference is 10
For input difference 12 --> Output difference is 10
For input difference 13 --> Output difference is 10
For input difference 14 --> Output difference is 10
For input difference 15 --> Output difference is 10
```

➔ After summing up the numbers in solutions column for each, we get 106 (for 256 input pairs, 106 output differences exist).
Mean non zero value in DDT = $\frac{256}{106}$= **2.415**

Table 1: A Table comparing Sboxes

Group Name	Cipher Name	Sbox SIze	Differential Uniformity	Differential branch number
gugu gaga	Midori	4-bit	4	2
SPL Encrypted	GIFT	4-bit	6	2
Hope we "3" SDVV	Serpent	4-bit	4	3
-.././cipher	Prince	4-bit	4	2
TechHeist 3.0	Pride	4-bit	4	2
Rook	Ascon	5-bit	8	3
Three Amigos	Klein	4-bit	4	2
Decryptor	PHOTON-beetle	4-bit	4	3
cryptoducks	LED	4-bit	4	3
Ping 999+	Elephant	4-bit	4	3
Kryptonian	Wage	8-bit	8	2

7 Automated Cryptanalysis

It is the study of optimizing (minimizing or maximizing) a linear objective function subject to linear inequalities involving decision variables.

➔ First set of Constraints:
Firstly, to ensure S-box to be active when any one of its input is 1.
$x_{00} - a_{00} \leq 0$
$x_{01} - a_{00} \leq 0$
$x_{02} - a_{00} \leq 0$
$x_{03} - a_{00} \leq 0$

➔ Second set of Constarints:
When S-Box is active, one of its input must be 1:
$x_{00} + x_{01} + x_{02} + x_{03} - a_{00} \geq 0$

8 Software Application

The software application that we made implemented an app for making confessions anonymously. It was made using node.js, ejs, javascript. First, a user registers for an account. The password given by the user is stored in a database as a hash value using the bcrypt module of npm. Then the user can decide amongst two options : either submitting a confession or seeing what has confessed to him or her. When the user submits a confession, it is converted to hex and then encrypted using the Prince cipher. Then it is stored in the database as ciphertext. When the user wants to see his or her confessions, the data is obtained from the database, decrypted and then shown.

9 Conclusion

We have successfully implemented **Prince Cipher** in python(prince.py) and also have detailed Sbox analysis(SboxAnalysis.ipynb).The detailed discussion of integral and differential cryptanalysis of prince cipher is also done in the term paper.

9.1 Brownie Points

- We have added ECB and CBC modes of operation in software implementation.
- We have added hashing in our Software Application.

References

- PRINCE – A Low-latency Block Cipher for Pervasive Computing Applications [Refer Here]
- Security Analysis of PRINCE [Refer Here]
- INTEGRAL CRYPTANALYSIS OF ROUND-REDUCED PRINCE CIPHER [Refer Here]
- PRINCE Cipher Rundown - Chris Dare [Refer Here]