Advanced cryptology (GBX9SY06)
Exercises

2018-11-30

Part I

The goal of this exercise is to find and analyse an attack for the fictional PONEY block cipher. PONEY was designed to be resistant against standard differential and linear cryptanalysis; unfortunately, it suffers from a serious distinguisher that succeeds with a very high probability, *independent of the number of rounds* used in the cipher. However, the success of the attack is conditioned on the key belonging to a certain subset of *weak keys*.

1 Poney specifications

PONEY is a lightweight cipher, with 64-bit keys and 64-bit blocks. Its round function is the composition of the parallel application of sixteen 4-bit S-boxes, a bit permutation, and a round-key addition.

1.1 The S-box of Poney

The S-box PS of PONEY is the first entry of [Saa11, Table 1], which has the lowest possible differential uniformity and linearity for a 4-bit S-box (respectively 4 and 8). As such, it is expected to offer "optimal" resistance against differential and linear cryptanalysis. This S-box is given in Table 1.

Table 1: The S-box of PONEY

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
PS(x)	0	1	2	3	4	6	8	A	5	B	C	F	7	9	D	E

The algebraic normal form of the S-box is given in Table 2, where x_i (resp. y_i) denotes the i^{th} most-significant bit of the input (resp. output) of PS. The designers were satisfied with the fact that it is of maximal degree on every output bit.

Table 2: The ANF of the S-box of PONEY

$$
\begin{aligned}
y_0 &= x_0x_1x_3 + x_0x_2x_3 + x_0x_2 + x_0x_3 + x_0 + x_1x_2x_3 + x_1x_3 + x_3 \\
y_1 &= x_0x_2x_3 + x_0x_2 + x_0x_3 + x_1x_2 + x_1x_3 + x_1 + x_2x_3 \\
y_2 &= x_0x_1x_3 + x_0x_3 + x_1x_2x_3 + x_1x_2 + x_2x_3 + x_2 + x_3 \\
y_3 &= x_0x_1x_3 + x_0x_3 + x_1x_2x_3 + x_1x_2 + x_1x_3
\end{aligned}
$$

1.2 The bit permutation of Poney

The bit permutation used in PONEY is given in Table 3 to Table 6.

Table 3: The bit permutation of PONEY (0–15)

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PP(x)	56	53	14	59	20	57	38	35	24	25	54	3	32	33	18	19

Table 4: The bit permutation of PONEY (16–31)

x	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
PP(x)	44	29	46	15	8	9	26	43	28	17	2	51	4	13	10	11

Table 5: The bit permutation of PONEY (32–47)

x	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
PP(x)	36	1	6	47	40	41	62	23	16	5	50	55	48	49	34	7

Table 6: The bit permutation of PONEY (48–63)

x	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
PP(x)	60	21	42	27	12	61	22	31	0	37	30	39	52	45	58	63

1.3 The key schedule of Poney

The key schedule of PONEY is quite simple. The i^{th} round key k_i is given by i successive applications of the permutation given in Table 7 performed on the 4-bit nibbles of the master key k_0. Or equivalently, $k_i = \text{PKS}(k_{i-1})$. This permutation has order 140, which means that all round keys will be distinct up to at most that many rounds.

Table 7: The key schedule nibble permutation of PONEY

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

1.4 Summary

A depiction of PONEY's round function is given in Figure 1. The designers claim that PONEY with 128 rounds should be a secure cipher.

2 Questions

Q1: What surprising behaviour can you observe in the S-box when the two most-significant bits of its input are set to zero? How does the ANF explain this? What is the degree of the S-box when restricted to such inputs?

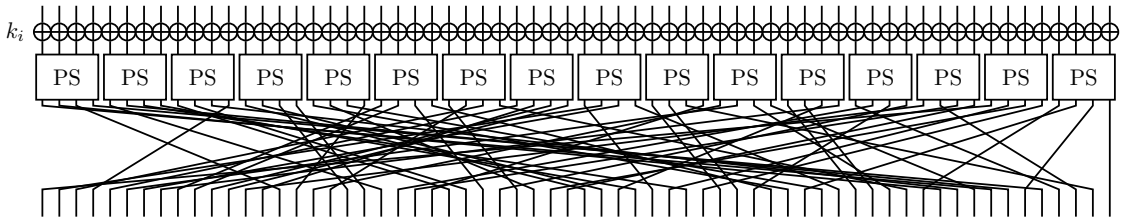

Figure 1: One round of PONEY

Q2: Carefully inspect the bit permutation of PONEY. What can you say about the image of the bits modulo four (equivalently, what can you say about the image of i if it is the j^{th} output bit of an S-box).

Q3: What happens after one round of PONEY if all inputs to the S-boxes have their two highest bits equal to zero? Under what condition for the round keys does this property hold for an arbitrary number of rounds?

Q4: Is the round key property identified in the previous question preserved by the actual key schedule of PONEY? (That is, if if the property holds for k_i, does it hold for k_{i+1}, and under what conditions?) How many master keys exist such that all of their derived round keys have this property? What is the probability that this happens for a uniformly random master key?

Q5: Give a distinguisher for PONEY of unit time and data complexity that provides an advantage ≈ 1 when the master key has the property identified above. Does the success of this distinguisher depend on the number of rounds?

Comments

Although PONEY was specifically designed to be weak against the attack described above, some actual cipher proposals such as PRINTCIPHER have been attacked in a similar way [LAAZ11].

Part II

Exercise 1: MISTY S-box differential branch number

We start with the following definition.

Definition 1. The *differential branch number* DBN of an n-bit S-box $\mathcal{S} : \{0,1\}^n \to \{0,1\}^n$ is defined as:

$$\min_{\{(a,b)\neq(0,0)|\,\delta_{\mathcal{S}}(a,b)\neq 0\}} \mathrm{wt}(a) + \mathrm{wt}(b),$$

where $\mathrm{wt}(x)$ is the Hamming weight of x and $\delta_{\mathcal{S}}(a,b) = \#\{x \in \mathbb{F}_2^n | \, \mathcal{S}(x) + \mathcal{S}(x+a) = b\}$.

In other words, the DBN of an S-box is the minimum input/output weight of a non-trivial difference that may be propagated with non-zero probability.

Q1: Show that if $\mathcal{S}$ is invertible, then its DBN is at least two.

We now recall the MISTY construction for obtaining a $2n$-bit S-box $\mathcal{S}$ from three n-bit ones $\mathcal{S}_0$, $\mathcal{S}_1$, $\mathcal{S}_2$. Define:

1. $x_1^L := x_0^R; \quad x_1^R := \mathcal{S}_0(x_0^L) \oplus x_0^R;$

2. $x_2^L := x_1^R; \quad x_2^R := \mathcal{S}_1(x_1^L) \oplus x_1^R;$
3. $x_3^L := x_2^R; \quad x_3^R := \mathcal{S}_2(x_2^L) \oplus x_2^R;$

and $\mathcal{S}(x_0^L||x_0^R) = x_3^L||x_3^R$. The resulting S-box is invertible if $\mathcal{S}_{0,1,2}$ are.

Q2: Show that if $\mathcal{S}_{0,1,2}$ are invertible and have DBN 3, then $\text{DBN}(\mathcal{S}) = 3$.
(Hint: there are only two cases to consider. It may be helpful to draw the "circuit" of $\mathcal{S}$ in function of the smaller S-boxes.)

Q3: What advantage can there be for an S-box to have a DBN of three (or more) when used in an SPN block cipher with a bit permutation as linear layer (like PRESENT or Poney)? What about the case of an SPN whose linear layer is defined over n-bit quantities (like the AES)?

Exercise 2: Noekeon's Theta mapping

In this exercise, we study some properties of the Theta mapping of the Noekeon block cipher [DPAR00], which is defined through the following program.[1]

```
#define ROTL8(x) (((x) << 8) ^ ((x) >> 24))
#define RORL8(x) (((x) >> 8) ^ ((x) << 24))

void Theta(uint32_t a[4])
{
        uint32_t temp;
        temp = a[0] ^ a[2]; temp ^= ROTL8(temp); temp ^= ROTR8(temp);
        a[1] ^= temp;
        a[3] ^= temp;
        temp = a[1] ^ a[3]; temp ^= ROTL8(temp); temp ^= ROTR8(temp);
        a[0] ^= temp;
        a[2] ^= temp;
}
```

Q1: Show that Theta is $\mathbb{F}_2$-linear.

Q2: Show that Theta is involutive (i.e. is its own inverse). (Hint: Write a symbolic execution of two consecutive runs of the program, and keep track of each register's value.)

Q3: Is it necessary for the mapping $x \mapsto x$ `^ ROTL8(`x`) ^ ROTR8(`x `^ ROTL8(`x`))` to be invertible for Theta to be invertible? What does the entire structure of Theta look like (use a drawing)?

References

[DPAR00] Joan Daemen, Michaël Peeters, Gilles Van Assche, and Vincent Rijmen. The Noekeon Block Cipher. Nessie Proposal, October 2000.

[LAAZ11] Gregor Leander, Mohamed Ahmed Abdelraheem, Hoda AlKhzaimi, and Erik Zenner. A Cryptanalysis of PRINTcipher: The Invariant Subspace Attack. In Phillip Rogaway, editor, *Advances in Cryptology — CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 206–221. Springer, 2011.

[Saa11] Markku-Juhani O. Saarinen. Cryptographic Analysis of All 4 × 4-Bit S-Boxes. In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography — SAC 2011*, volume 7118 of *Lecture Notes in Computer Science*, pages 118–133. Springer, 2011.

[1] The actual mapping also involves a round key addition, that we omit here for simplicity.