Cryptanalysis of PRINT cipher

By: Tushar Vatsal

---

1 Abstract

In this paper, we talk about radiofrequency identification (RFID) cards, integrated circuit printing, or IC-printing and then consider the need and cryptographic implications of the Print Cipher. We have two versions of the Print Cipher depending on their block size: Print Cipher-48 and Print Cipher-96. And then we move towards security goals and security analysis where, concerning the key cryptanalytic methods known, we evaluate the protection of our proposal.

2 Introduction

The radiofrequency identification (RFID) system has an RFID reader and an RFID tag. RFID tags are a form of tracking system that identifies objects using smart barcodes. There are three types of RFID tags, namely, Active tag, Passive tag, and Semipassive tag based on their power supply. An active tag has its power supply, whereas a passive lag has no power supply, and a semipassive tag has its power supply but to transmit feedback signal to the RFID reader, it depends on the power supplied by the RFID reader. An active tag has a maximum range, whereas a passive and semi-passive tag has quite much the same range but less than that of an active tag. RFID cards or tags use radiofrequency technology to transmit data from a tag to a reader which then transmits the information to an RFID computer program.

The cryptographic algorithm provides confidentiality and maintains the integrity of the information, but due to exceptional physical and economic constraints, we must leave behind conventional cryptography because conventional cryptographic techniques require higher resources, but the problem with this RFID tag is that they have minimal resources like limited computation, limited memory, power, and small size. For example, the RSA algorithm of 1204 bits, which will consume most of the resources of RFID, which already have minimal resources. Therefore, for other functions, RFID won't have adequate resources. So it is not possible to implement the RSA algorithm of 1204 bits on RFID. For the proper functioning of the RFID, any cipher with more significant gate equivalence (GE) will create a problem. Only PRINT cipher and EPCBC cipher can be used for the RFID having a corresponding chip area of 402/726 GE and 1008 GE.

This paper takes into account technological advancements, such as integrated circuit printing or IC printing. The technology remains in its infancy and it has yet to fully understand its true potential. But the claimed advantages include the ability to print on thin and flexible materials and being much cheaper than silicon-based deployments because the conventional manufacturing process is bypassed. Since the main driver for IC printing is cost-effective, the typically mentioned application areas overlap closely with the typical lightweight cryptography domains. Indeed in the development of cheap RFID tags, one of the applications is IC-printing. Therefore, some of the strategies suggested for traditional RFID tags and those that will be used on printed tags have a lot in common.

For many factors, block ciphers create a natural starting point. They may not only be used in several different ways, but we feel a little more relaxed with them as a group. Analysis and design. That said we are working right at the edge of proven practice for such extreme environments as IC-printing and we are forced to consider and highlight some interesting problems. This is the objective behind the PRINT cipher of the block cipher.

3 The design approach to PRINT cipher

We can conceptually imagine that an "encryption computation" and a "subkey computation" are required inside a block cipher. The short-cuts we can build for the first are minimal, as we are restricted by the attentions of the cryptanalyst. For the most part, this means that proposals for a given security level and a given set of parameters for the block cipher will occupy almost the same area. If we were to reduce the space occupied by an execution, then the block size would most likely be reduced. Nevertheless, for the "subkey calculation," things are a little different and it is not always clear precisely how a key should be used. This emphasizes two different problems.

The first problem is whether in an application a key is likely to be modified. For RFID applications, it is very unlikely that one will want to change the key. Indeed, some other work on RFID implementation has shown that the overhead can be important in promoting a key shift.

The second problem is the exact shape of the key schedule. Some block ciphers, e.g. IDEA has a very simple key schedule in which subkeys are produced by user-supplied key sampling bits. The benefit of this method is that the subkey computations require no working memory. Some key schedule computation is available from other lightweight block ciphers, e.g. PRESENT, while another CGEN proposal proposes to use no key schedule, without any sampling or extra computation, the user-supplied key is used.

4 Implementation

PRINTcipher is a block ciher with b-bit blocks, b ϵ {48, 96}. PRINTcipher has an effective key length of 5/3 * b bits and a total of b rounds. For Ex- PRINTcipher-48 has 48-bit blocks, uses and (5/3*40) = 80-bit key and consists of 48 rounds. Similarly, PRINTcipher-96 operates on 96-bit blocks, uses a (5/3 * 96) = 160-bit key and consists of 96 rounds.

Each round of PRINTcipher consists of the following steps:

1) Bitwise exclusive-or (xor) of cipher state is done with a round key
2) The cipher state is mixed up using a fixed linear diffusion layer.
3) Bitwise exclusive-or (xor) of an n-least significant bit of cipher state is done with a round counter, where n = $log_2 r$, r is no of rounds.
4) Three-bit input to S-box is first permutated in a key-dependent manner.
5) The cipher state is passed through b/3 non-linear S-box layer.

Key Xor: Bitwise xor of the cipher state is done with a b-bit subkey(sk1) from a total of 5/3 * b bits effective key. This subkey is identical in all rounds.

Linear Diffusion: The permutation-layer is a simple bit permutation. In general, an SP-network with block size b and s bit S-boxes, bit permutation is given by:

$$p(i) = s \times i \; mod(b-1) \; ; for \; 0 \leq i \leq (b-2)$$
$$p(i) = b-1; for \; i = b-1$$

In case of PRINTcipher, we have 3 bit s-boxes, so s = 3. Hence in PRINT cipher bit *i* of the current state is moved to bit position *P(i)* in the following way:

$p(i) = 3 \times i \, mod(b - 1) \; ; for \; 0 \leq i \leq (b - 2)$
$p(i) = b - 1; for \; i = b - 1$

3) Round Counter RC_i: The round counter RC_i for $1 \;<=\; i \;<=\; r$ is combined using xor to the least significant bits of the current state. The values of the round counter are generated using an n-bit shift-register (n = [$log_2 r$]) in the following way. In case of PRINTcipher-48, r = 48. So, n = [$log_2 48$] = 6. Let's denote the state of the shift register as Xn-1 | | ………….| |X1 | |X0 (in case of PRINTcipher-48, state of shift register is denoted as X5| |X4| |X3| |X2| |X1| |X0). The shift register is initialised to all zeros. In case of PRINTcipher-48, shift register is initialised as 000000. Round counter RCi is computed as:

t = 1 + Xn-1 + Xn-2
$X_i = X_{i-1}$ for n-1 ≥ i ≥ 1
$X_0 = t$

Example:
Sequence of RCi for PRINTcipher-48 in hexadecimal notation:
RC1 :
X = 0 | | 0 | | 0 | | 0 | | 0 | | 0
t = 1 + 0 + 0 = 1
$X \;=\; 0 \,||\, 0 \,||\, 0 \,||\, 0 \,||\, 0 \,||\, 1$
RC1 = 01

RC2 :
t = 1 + 0 + 0 = 1
$X \;=\; 0 \,||\, 0 \,||\, 0 \,||0 \,||\, 1 \,||\, 1$
RC2 = 03

RC3:
t = 1 + 0 + 0 = 1
$X \;=\; 0 \,||\, 0 \,||\, 0 \,||\, 1 \,||\, 1 \,||\, 1$
RC3 = 07

RC4:
t = 1 + 0 + 0 = 1
$X \;=\; 0 \,||\, 0 \,||\, 1 \,||\, 1 \,||\, 1 \,||\, 1$
RC4 = 0F

RC5:

t = 1 + 0 + 0 = 1
$X = 0 \,||\, 1 \,||\, 1 \,||\, 1 \,||\, 1 \,||\, 1$
RC5 = 1F

RC6:

t = 1 + 0 + 1 = 0
$X = 1 \,||\, 1 \,||\, 1 \,||\, 1 \,||\, 1 \,||\, 0$
RC6 = 3E

RC7:

t = 1
$X = 1 \,||\, 1 \,||\, 1 \,||\, 1 \,||\, 0 \,||\, 1$
RC7 = 3D

4) Keyed Permutation: We have 3-bit S-boxes and hence 3-bit input to it. The set of 3-bit input is first permuted among themselves and then fed to s-box. And these permutations are done in a key-dependent manner.

The 5/3 b-bit user-supplied key k is considered as consisting of two subkey components k = sk1||sk2 where sk1 is b bits long and sk2 is 2/3 b-bits long. The first subkey is used, unchanged, within the xor layer of each and every round. The second subkey sk2 is used to generate the key-dependent permutations in the following way. The 2/3 b bits are divided into b/3 sets of two bits and each two-bit quantity a1||a0 is used to pick one of four of the available permutations. Specifically, the three input bits c2||c1||c0 are permuted to give the following output bits according to the value of a1||a0.

a1||a0	
00	C2 || c1 || c0
01	C1 || c2 || c0
10	C2 || c0 || c1
11	C0 || c1 || c2

5) S-box layer: A total of 3-bit to 3-bit S-box is used b/3 times in parallel. Each set of three input bit is applied to the S-box layer to get 3-bit output. The S-box is used in the following way:

x	0	1	2	3	4	5	6	7
S[x]	0	1	3	6	7	4	5	2

Keyed permutation with S-box layer:

Applying bit-permutation:

a1|| a 0		**x**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
	Binary		**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
00	**c2||C1||C0**	**L0[X]**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
01	**C1||C2||C0**	**L1[X]**	**0**	**1**	**4**	**5**	**2**	**3**	**6**	**7**
10	**C2||C0||C1**	**L2[X]**	**0**	**2**	**1**	**3**	**4**	**6**	**5**	**7**
11	**C0||C1||C2**	**L3[X]**	**0**	**4**	**2**	**6**	**1**	**5**	**3**	**7**

Applying the S-box layer:

L[x]	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
V0[x]	**0**	**1**	**3**	**6**	**7**	**4**	**5**	**2**
V1[x]	**0**	**1**	**7**	**4**	**3**	**6**	**5**	**2**
V2[x]	**0**	**3**	**1**	**6**	**7**	**5**	**4**	**2**
V3[x]	**0**	**7**	**3**	**5**	**1**	**4**	**6**	**2**

6 Conclusion

We have considered the technology of IC-printing in this paper and we have seen how it could impact the cryptography we use. We also specifically suggested the PRINT cipher lightweight block cipher, which directly takes advantage of this modern manufacturing strategy. Of course, it must be stressed that PRINTcipher-48 is intended not to be appropriate for deployment, but rather to be an object of study. It is also meant to be a catalyst for those who may be involved in this emerging technology being considered. We definitely agree that IC-features printing may be a fascinating line of work and we believe that it helps to highlight a variety of intriguing cryptographic design issues, most importantly how best to use a cipher key.

7 References:

1) Knudsen L., Leander G., Poschmann A., Robshaw M.J.B. (2010) PRINTcipher: A Block Cipher for IC-Printing. In: Mangard S., Standaert FX. (eds) Cryptographic Hardware and Embedded Systems, CHES 2010. CHES 2010. Lecture Notes in Computer Science, vol 6225. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-15031-9_2

2) Cryptanalysis of PRINTcipher(a youtube video): https://youtu.be/wB7LxBLbBJo

3) Leander G., Abdelraheem M.A., AlKhzaimi H., Zenner E. (2011) A Cryptanalysis of PRINTcipher: The Invariant Subspace Attack. In: Rogaway P. (eds) Advances in Cryptology – CRYPTO 2011. CRYPTO 2011. Lecture Notes in Computer Science, vol 6841. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-22792-9_12