

# A Related-Key Chosen-IV Distinguishing Attack on Full Sprout Stream Cipher

Yonglin Hao

Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

haoyl14@mails.tsinghua.edu.cn

**Abstract.** Sprout is a new lightweight stream cipher proposed at FSE 2015. According to its designers, Sprout can resist time-memory-data trade-off (TMDTO) attacks with small internal state size. However, we find a weakness in the updating functions of Sprout and propose a related-key chosen-IV distinguishing attack on full Sprout. Our attack enable the adversary to detect non-randomness on full 320-round Sprout with a practical complexity (no more than  $2^{20}$  key-IV pairs).

## 1 Introduction

Recently at FSE 2015, Frederik and Vasily proposed a new design of stream cipher called Sprout [1]. They proved that Sprout can resist the time-memory-data trade-off (TMDTO) attacks despite of its extraordinarily small internal state size. They also claimed that Sprout has the same secure margin with the famous Grain-128a stream cipher [2].

**Our contribution.** However, we have found a weakness in the updating functions of Sprout. Based on our finding, we manage to launch a practical related-key chosen-IV distinguishing attack on full Sprout. Our method can distinguish Sprout from random stream with no more than  $2^{20}$  key-IV pairs.

**Organization of the Paper.** Section 2 provides the description of Sprout and some notations used in this paper. Section 3 give our findings about Sprout. In Section 4, we describe the procedure of our attack. Finally, we summarize our paper in Section 5.

## 2 Preliminary

The Sprout stream cipher generates keystream bits from 70 public IV bits, denoted by  $v_0, \dots, v_{79}$ , and an 80 secret key bits, denoted by  $x_0, \dots, x_{79}$ . It consists of a 40-bit NFSR and a 40-bit LFSR, denoted by  $\mathbf{N}$  and  $\mathbf{L}$  respectively.  $\mathbf{N}$  and  $\mathbf{L}$  are first initialized as follows:

$$\begin{aligned}\mathbf{N} &= (n_0, \dots, n_{39}) = (v_0, \dots, v_{39}), \\ \mathbf{L} &= (l_0, \dots, l_{39}) = (v_{40}, \dots, v_{69}, 1, \dots, 1).\end{aligned}$$

For  $t \geq 0$ , the feedback function of LFSR is defined as:

$$l_{t+40} = l_t + l_{t+5} + l_{t+15} + l_{t+20} + l_{t+25} + l_{t+34} \quad (1)$$

and that of NFSR is

$$\begin{aligned}n_{t+40} &= k_t^* + l_t + c_t^4 + n_t + n_{t+13} + n_{t+19} + n_{t+35} + n_{t+39} \\ &\quad + n_{t+2}n_{t+25} + n_{t+3}n_{t+5} + n_{t+7}n_{t+8} + n_{t+14}n_{t+21} + n_{t+16}n_{t+18} \\ &\quad + n_{t+22}n_{t+24} + n_{t+26}n_{t+32} + n_{t+33}n_{t+36}n_{t+37}n_{t+38} \\ &\quad + n_{t+10}n_{t+11}n_{t+12} + n_{t+27}n_{t+30}n_{t+31}.\end{aligned} \quad (2)$$

where the key  $k_t^*$  is assigned according to a round key function as follow:

$$k_t^* = \begin{cases} x_t & 0 \leq t \leq 79 \\ (x_{t \bmod 80}) \cdot (l_{t+9} + l_{t+21} + l_{37} + n_{t+9} + n_{t+20} + n_{t+29}), & t \geq 80 \end{cases} \quad (3)$$

and the counter bit  $c_t^4$  is the 4-th significant bit of the integer  $(t \bmod 80)$ . The output function is defined as

$$z_t = \sum_{j \in B} n_{t+j} + l_{t+30} + h(t) \quad (4)$$

where  $B = \{1, 6, 15, 17, 23, 28, 34\}$  and

$$h(t) = n_{t+4}l_{t+6} + l_{t+8}l_{t+10} + l_{t+32}l_{t+17} + l_{t+19}l_{t+23} + n_{t+4}l_{t+32}n_{t+38}.$$

After  $l_0, \dots, l_{39}$  and  $n_0, \dots, n_{39}$  are settled, the Sprout state then runs 320 initialization rounds not producing an output but feeding the output back into both LFSR and NFSR. So the actual output bits are  $z_{320}, z_{321}, \dots$

Before our descriptions, we give some notations used throughout this paper.

**Bit** We use small letters to represent 0-1 bits. More specifically, the 80 secret key bits are represented by letter  $x_i$  ( $i = 0, \dots, 79$ ) and the 70 public IV bits are represented by letter  $v_j$  ( $j = 0, \dots, 69$ ).

**Vector** We denote the 0-1 vectors by bold letters such as  $\mathbf{k}$  for the 80-bit key and  $\mathbf{v}$  for 70-bit IV. We also denote the internal states of NFSR and LFSR after  $t$  rounds as

$$\begin{aligned} \mathbf{N}_t &= (n_t, \dots, n_{t+39}) \\ \mathbf{L}_t &= (l_t, \dots, l_{t+39}). \end{aligned}$$

**Specific Bit** We refer to the  $i$ -th ( $i$  starts from 0) bit of vector  $\mathbf{n}$  as  $\mathbf{n}[i]$ . For example, the  $i$ -th bit of secret key  $\mathbf{k}$  defined above, we have  $\mathbf{k}[i] = x_i$  ( $i = 0, \dots, 79$ ).

Besides, for  $t \geq 80$ , we specifically denote the value  $\lambda_t$  as

$$\lambda_t = l_{t+9} + l_{t+21} + l_{37} + n_{t+9} + n_{t+20} + n_{t+29}.$$

The definition of  $\lambda_t$  is for the convenience of our interpretations in the following section.

### 3 Our Findings

Before illustrate our finding on Sprout, we give the following definition.

**Definition 1. (Dual Key-IV Pairs)** For any secret key  $\mathbf{k} = (x_0, \dots, x_{79})$  and public IV  $\mathbf{v} = (v_0, \dots, v_{69})$ , we define their dual key  $\hat{\mathbf{k}}$  and IV  $\hat{\mathbf{v}}$  as follows:

$$\begin{aligned} \hat{\mathbf{k}} &= (x_0 + 1, x_1, \dots, x_{79}) \\ \hat{\mathbf{v}} &= (v_0 + 1, v_1, \dots, v_{69}). \end{aligned}$$

We refer to key-IV pair  $(\hat{\mathbf{k}}, \hat{\mathbf{v}})$  as the “dual pair” of  $(\mathbf{k}, \mathbf{v})$ .

In the remainder of this paper, we denote the intermediate state bits  $n_i, l_i, z_i$  as the bits deduced from  $(\mathbf{k}, \mathbf{v})$  and denote  $\hat{n}_i, \hat{l}_i, \hat{z}_i$  as their counterparts deduced from the dual pair  $(\hat{\mathbf{k}}, \hat{\mathbf{v}})$ .  $\mathbf{N}_t$  ( $\mathbf{L}_t$ ) and  $\hat{\mathbf{N}}_t$  ( $\hat{\mathbf{L}}_t$ ) are defined in the same way.

Our main finding about Sprout can be summarized as Proposition 1.

**Proposition 1.** For any randomly chosen key-IV pair  $(\mathbf{k}, \mathbf{v})$  and its dual pair  $(\hat{\mathbf{k}}, \hat{\mathbf{v}})$ , after  $80t$  Sprout rounds ( $t \geq 1$ ), we have

$$Pr\{z_{80t} = \hat{z}_{80t}\} = 2^{-1} + 2^{-t} \quad (5)$$

*Proof.* In the first 80 rounds,  $x_0$  only takes part in the updating of  $n_{40}$ . We found that

$$n_{40} = (v_0 + x_0) + C = (v_0 + x_0 + 1 + 1) + C = \hat{n}_{40}$$

where  $C$  is irrelevant with  $v_0$  and  $x_0$ . So we have

$$Pr\{n_{40} = \hat{n}_{40}\} = Pr\{\mathbf{L}_1 = \hat{\mathbf{L}}_1\} = Pr\{\mathbf{N}_1 = \hat{\mathbf{N}}_1\} = 1.$$

In the following 79 rounds, besides  $\mathbf{L}_1$  and  $\mathbf{N}_1$  only the rest 79 identical key bits are involved in the updating process, so we have

$$Pr\{\mathbf{L}_j = \hat{\mathbf{L}}_j\} = Pr\{\mathbf{N}_j = \hat{\mathbf{N}}_j\} = 1.$$

for  $j = 1, \dots, 79$  and (5) holds when  $t = 1$ .

Then, when updating  $n_{120}$  ( $\hat{n}_{120}$ ), the bit  $\mathbf{k}[0]$  ( $\hat{\mathbf{k}}[0]$ ) will not be involved with probability

$$Pr\{\lambda_{80} = \hat{\lambda}_{80} = 0\} = Pr\{n_{120} = \hat{n}_{120} = 0\} = 2^{-1}.$$

If we  $n_{120} = \hat{n}_{120}$ , we can also deduce that

$$Pr\{\mathbf{L}_{80+j} = \hat{\mathbf{L}}_{80+j} | n_{120} = \hat{n}_{120}\} = Pr\{\mathbf{N}_{80+j} = \hat{\mathbf{N}}_{80+j} | n_{120} = \hat{n}_{120}\} = 1.$$

for  $j = 1, \dots, 79$  and (5) holds when  $t = 2$ .

More generally, for  $t > 1$  and  $j = 1, \dots, 79$ , we have

$$Pr\{\mathbf{L}_{80t+j} = \hat{\mathbf{L}}_{80t+j} | \bigwedge_{i=1}^{t-1} (n_{80i+40} = \hat{n}_{80i+40})\} = Pr\{\mathbf{N}_{80t+j} = \hat{\mathbf{N}}_{80t+j} | \bigwedge_{i=1}^{t-1} (n_{80i+40} = \hat{n}_{80i+40})\} = 1.$$

and

$$Pr\{\bigwedge_{j=1}^{t-1} (n_{80j+40} = \hat{n}_{80j+40})\} = Pr\{\bigwedge_{j=1}^{t-1} (\lambda_{80j} = \hat{\lambda}_{80j} = 0)\} = 2^{-(t-1)}.$$

So we can deduce that

$$Pr\{z_{80t} = \hat{z}_{80t}\} = 2^{-1} Pr\{\bigwedge_{j=1}^{t-1} (\lambda_{80j} = \hat{\lambda}_{80j} = 0)\} + Pr\{\bigwedge_{j=1}^{t-1} (\lambda_{80j} = \hat{\lambda}_{80j} = 0)\} = 2^{-1} + 2^{-t}$$

which is exactly (5). □

## 4 Related-Key Chosen-IV Attack on Full Sprout

With Proposition 1, we can naturally conduct a related-key chosen-IV distinguishing attack on full 320-round Sprout stream cipher. With practical complexity, we can distinguish full Sprout from random 0-1 stream. The procedure is as follows:

1. We randomly select  $m$  key-IV pairs  $(\mathbf{k}_i, \mathbf{v}_i)$  and compute their dual pairs  $(\hat{\mathbf{k}}_i, \hat{\mathbf{v}}_i)$  ( $i = 1, \dots, m$ ) as Definition 1.
2. For all  $i = 1, \dots, m$ , compute their first output bit of Sprout stream, denoted as  $z_{320}^i$  and  $\hat{z}_{320}^i$ . Count the number of  $i$ 's satisfying  $z_{320}^i = \hat{z}_{320}^i$  and denoted by  $\xi$ , which means

$$\xi := \#\{i | z_{320}^i = \hat{z}_{320}^i, i = 1, \dots, m\}.$$

According to Proposition 1, the Sprout output streams satisfy

$$\lim_{m \rightarrow \infty} \frac{\xi}{m} = \frac{1}{2} + \left(\frac{1}{2}\right)^4 = 0.5625.$$

For random bit streams, the ratio  $\frac{\xi}{m}$  will approach 0.5 instead.

Since the bias for full 320 round Sprout is  $2^{-4}$ , the success probability is significant enough for  $m \sim \tilde{O}(2^4)$ . For precise consideration, we set  $m = 2^{20}$  and experimentally verified the correctness of our attack. Some of the statistics are shown in Table 1.

## 5 Conclusion

In this paper, we find a weakness in the design of newly proposed stream cipher Sprout. Based on the weakness, we manage to launch a related-key chosen-IV attack on the full cipher with practical complexity and 100% success probability. We suggest that the designers of Sprout should revisit their design by changing the updating functions of NFSR.

**Table 1.** Experiment Verifications ( $m = 2^{20}$ )

Output	$\frac{\xi}{m}$
$z_{80}$	1
$z_{160}$	0.749512
$z_{240}$	0.635986
$z_{320}$	0.563721

## References

1. Armknecht, F., Mikhalev, V.: On lightweight stream ciphers with shorter internal states, FSE (2015)
2. Ågren, M., Hell, M., Johansson, T., Meier, W.: Grain-128a: a new version of grain-128 with optional authentication. International Journal of Wireless and Mobile Computing **5**(1) (2011) 48–59