A Time-Memory Tradeoff
Attack Against LILI-128

Helsinki University of Technology
Laboratory for Theoretical Computer Science
P.O. Box 5400, FIN-02015 HUT, Finland
mjos@tcs.hut.fi

Abstract. In this note we discuss a novel and simple time-memory tradeoff attack against the stream cipher LILI-128. The attack defeats the security advantage of having an irregular stepping function. The attack requires \(2^{16}\) bits of keystream, a lookup table of \(2^{45}\) 89-bit words and computational effort which is roughly equivalent to \(2^{48}\) DES operations.

1 Introduction

\[2^{112}\]

1.1 Previous Work

\[2^{79}\]

\[2^{40}\]
### 1.2 Time/Memory/Data Tradeoffs

<table>
<thead>
<tr>
<th>Step</th>
<th>Output</th>
<th>( x_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_0 )</td>
<td>( \text{Output} )</td>
<td>( \text{Step} )</td>
</tr>
</tbody>
</table>

\[
z(i) = \text{Output}(x_{i-1})
\]

\[
x_i = \text{Step}(x_{i-1})
\]

**Off-line preprocessing stage.**

\[
z(i), z(i+1), \ldots, z(i + O(\log N))
\]

**On-line computation phase.**

\[O(\log N)\]

<table>
<thead>
<tr>
<th>Step</th>
<th>( O(\sqrt{N}) )</th>
<th>( O(\log N \sqrt{N}) )</th>
</tr>
</thead>
</table>

### 2 Description of LILI-128

\[
LFSR_{c_1} \quad LFSR_{d_1} \quad LFSR_{c_2} \quad LFSR_{d_2} \quad LFSR_{c_3}
\]

\[
128 = 39 + 89
\]

\[
t_0, t_1, \ldots, t_{38} \quad LFSR_{c_1} \quad t_0 \quad t_{38}
\]

---

1. In [6] the authors also discuss other keying methods for LILI-128.
**Fig. 1.** Overview of the LILI-28 keystream generator.

\[ u_0, u_1, \ldots, u_{88} \]

\[ LFSR_{c} \]

\[ x^{39} + x^{35} + x^{33} + x^{31} + x^{17} + x^{15} + x^{14} + x^2 + 1 \]

\[ LFSR_{d} \]

\[ x^{89} + x^{83} + x^{80} + x^{55} + x^{53} + x^{42} + x^{39} + x + 1. \]

\[ LFSR_{d} \]

\[ z(t) \]

\[ f_d, f_d : \mathbb{F}_2^{10} \to \mathbb{F}_2 \]

\[ z(t) = f_d(u_0, u_1, u_3, u_7, u_{12}, u_{20}, u_{30}, u_{44}, u_{65}, u_{80}). \]

\[ LFSR_{c} \]

\[ c(t) \]

\[ f_c : \mathbb{F}_2^{2} \to \mathbb{F}_2 \]

\[ c(t) = f_c(t_{12}, t_{20}) = 2t_{12} + t_{20} + 1 \]

\[ LFSR_{d} \]

\[ c(t) \]

\[ \text{before} \]

---

\[ \text{The } f_d \text{ function is specified as a 1024-entry table in the original specification [5], and is excluded from this paper since it is irrelevant to the present attack.} \]
Lemma 1. For each \( \Delta_c = 2^{39} - 1 \) times LFSR\(_c\) is clocked, LFSR\(_d\) is clocked exactly \( \Delta_d = 5 \times 2^{38} - 1 \) times.\( ^3 \)

Proof.

\[
\sum_{i=1}^{2^{39} - 1} c(t + i) = \Delta_d
\]

\[
\begin{align*}
\text{LFSR}_c & \quad 2^{39} - 1 = \Delta_c \\
(0, 0) & \quad 2^{37} - 1 \quad (0, 1) \quad (1, 0) \quad (1, 1) \quad 2^{37} \\
\end{align*}
\]

\[
1 \times (2^{37} - 1) + (2 + 3 + 4) \times 2^{37} = 1374389534719
\]

Lemma 2. LFSR\(_d\) can be stepped by \( \Delta_d \) number of positions forward or backward by performing a vector-matrix multiplication with a precomputed 89 \times 89 bit matrix over \( GF(2) \). The matrix can be constructed with roughly \( 2^{28} \) bit operations using a binary matrix exponentiation algorithm.

Proof.

\[
\Delta_d \quad GF(2^{89}) \\
2^{11.4} \quad 3949 \approx
\]

3 The Attack

3.1 Constructing the Lookup Table

\[
\begin{align*}
2^{45} & \quad \Delta_d & \quad LFSR_d & \quad 2^{46} \\
& \quad f_d
\end{align*}
\]

Analysis.

\[
2^{51.48} \quad 1 - e^{-2} = 0.8647 \\
2^{48}
\]

\( ^3 \) This lemma follows implicitly from Theorem 2 in [5]
3.2 Lookup Stage

\[z(0), z(1), \ldots, z(2^{46} - 1)\]

\[2^{46} - 44\Delta_c - 1\]

\[z(i) \mid z(i + \Delta_c) \mid \ldots \mid z(i + 44\Delta_c)\]

LFSRd \[\Delta_d \mid \Delta_c\]

\[f_d(LFSR_d) \neq z(j\Delta_c + (i \mod \Delta_c))\]

LFSRd \[\Delta_d \mid \Delta_c\]

Analysis.

LFSRd

\[1 - \left(1 - \frac{0.8647 + 2^{45}}{2^{49}}\right)^{2^{46} - 44\Delta_c} \approx 90\%\]

4 Conclusions

2^{46}

2^{45} \quad 2^{51.48}
5 Acknowledgments

References