Exploiting Linear Hull in Matsui’s Algorithm 1

Andrea Röck and Kaisa Nyberg
Department of Information and Computer Science
Aalto University, School of Science
The Seventh International Workshop on Coding and Cryptography 2011
April 11-15, 2011, Paris, France
Outline

Introduction

Direct Attack

Related Key Attack

Results from Experiments

Conclusion
Introduction
Linear Cryptanalysis [Matsui 1994]

- Key-alternating iterated block cipher \((R\) rounds):
  - Block size: \(n\) bits
  - Plain text: \(x = x_1\)
  - Key schedule: \(K \mapsto K_1, \ldots, K_R\) \((K \in \mathbb{Z}_2^\ell)\)
  - Round function: \(x_{i+1} = g(x_i \oplus K_i)\)
  - Cipher text: \(\varepsilon_K(x) = x_{R+1}\)

- Correlation over \(R\) rounds:
  \[
c_R(u, w, K) = \frac{\#\{u \cdot x = w \cdot \varepsilon_K(x)\} - \#\{u \cdot x \neq w \cdot \varepsilon_K(x)\}}{2^n}
\]

- Matsui’s Algorithm 1:
  - Use key dependency of \(c_R(u, w, K)\) to learn \(K \cdot v\)

- Matsui’s Algorithm 2:
  - Use that \(|c_{R-1}(u, w, K)| > 0\) to gain information on \(K_R\)
Example 1

- Single strong trail (like in SERPENT)

\[ c(u, w, K) = (-1)^{k_1 \oplus k_2 \oplus k_3} c_1 c_2 c_3 \]

Sign of trail-correlation depends on linear combination of key bits
Example 2 - Linear Hull

▶ Multiple strong trails (like in AES, PRESENT)

▶ The total correlation is the sum of the trail-correlations
[Nyberg 2001, Deamen and Rijmen 2002]

\[ c(u, w, K) = (-1)^{k_1 \oplus k_2 \oplus k_3} c^3 + (-1)^{k_1 \oplus k_4 \oplus k_5} (-c^3) \]
Linear Hull - Algorithm 2

- The average squared correlation of the linear approximation taken over all keys is equal to the sum of all squared trail correlations [Nyberg 1995]

- On average $|c_{R-1}(u, w, K)|$ is large enough to learn $K_R$

- For some keys, $|c_{R-1}(u, w, K)|$ is very small and the attack does not work [Murphy 2009]
Linear Hull - Algorithm 1

- Until now not analyzed

- **Example**: Two (independent) trails with trail-correlation $c$
  - For 1/4 of keys: $c(u, w, K) = -2c$
  - For 1/2 of keys: $c(u, w, K) = 0$ (Alg. 2 does not work)
  - For 1/4 of keys: $c(u, w, K) = 2c$

- Correlation gives information of the key
  - In example: we learn 1.5 bits of information
Direct Attack
Idea

- **Total correlation** can be approximated by **strong key-mask correlations**: \( c(u, w, K) \approx \sum_{v \in V} \rho(v)(-1)^{v \cdot K} \)

- **Set of strong key masks**: \( V \)

- **Key-mask correlation**: \( \rho(v)(-1)^{v \cdot K} \)

- **Possible correlations**: \( C = \{ c(u, w, K) : K \in \mathbb{Z}_2^l \} \)

- **Key classes**: \( \mathcal{K}(c) = \{ K \in \mathbb{Z}_2^l : c(u, w, K) = c \} \)

- **Goal**: For a given secret key \( K \) estimate \( c \in C \) from data such that \( K \in \mathcal{K}(c) \)
Efficient Precomputation

- How to compute $C$ and $\mathcal{K}(c)$ faster than evaluating
  $\sum_{v \in V} \rho(v)(-1)^{v \cdot K}$ for all $K \in \mathbb{Z}_2^l$?
- Let $t = \dim(\text{span}(V))$
- Can partition set of keys into $2^t$ disjoint subsets such that
  all the keys in a subset have the same correlation
  (subset $\subset \mathcal{K}(c)$ for a $c \in C$)
- Use fast Walsh-Hadamard transform
- Precomputation complexities: time $\mathcal{O}(t2^t)$, memory $\mathcal{O}(2^t)$
Statistical Test

- $|C|$-ary hypothesis testing problem: Find correct $c \in C$
- $|\mathcal{K}(c)|$ varies a lot for different $c$
  - Use a priori probabilities $\pi_c = \Pr[c(u, w, K) = c]$ of $c$ (Bayesian approach)
- Complexity depends on minimal distance in $C$:
  \[ d = \min_{\substack{c_1 \neq c_2 \in C}} |c_1 - c_2| \]
- Data complexity for error probability $P_e$
  \[ N = 8 \ln(2) \frac{\log_2(|C| - 1) - \log_2 P_e}{d^2} \]
Gained Information

- How much information do we learn?
- Average learned information: Shannon’s entropy of a priori probabilities $\pi_c$

$$h = - \sum_{c \in \mathcal{C}} \pi_c \log_2 \pi_c$$

- Special case: If all vectors in $\mathcal{V}$ linearly independent and $|\rho(\mathcal{V})| = \text{const}$: $c \in \mathcal{C}$ are binomial distributed and $O \left( \frac{1}{2} \log_2 \left( \frac{\pi e}{2} |\mathcal{V}| \right) \right)$

- Always $h \leq \log_2 |\mathcal{C}|$
Related Key Attack
Idea

- Complexity of direct attack increases with number of strong key masks $|\mathcal{V}|$
- Reduce number of relevant key masks by related key attack
- Correlation difference:

$$\Delta(K, \alpha) = c(u, w, K) - c(u, w, K \oplus \alpha)$$

$$= \sum_{v \in \mathcal{V}} (-1)^{v \cdot K} \rho(v) - \sum_{v \in \mathcal{V}} (-1)^{v \cdot (K \oplus \alpha)} \rho(v)$$

- Reduced key mask set: $\mathcal{V}_\alpha = \{ v \in \mathcal{V} : v \cdot \alpha = 1 \}$

$$\Delta(K, \alpha) = 2 \sum_{v \in \mathcal{V}_\alpha} (-1)^{v \cdot K} \rho(v)$$

- Statistical test and definition of $C_\alpha, d_\alpha, t_\alpha, h_\alpha$ equivalent to direct attack
Multiple Related Key Attack

- For a given $\mathcal{V}$ we can learn at most $t = \dim(\text{span}(\mathcal{V}))$ bits of information.

- Independent case: all vectors in $\mathcal{V}$ are linearly independent.

  - Given any $v \in \mathcal{V}$ choose $\alpha_v$ such that for all $v' \in \mathcal{V}$:
    $$\alpha_v \cdot v' = \delta_{v,v'} = \begin{cases} 1 & \text{if } v' = v \\ 0 & \text{otherwise} \end{cases}$$

  - Then $\mathcal{V}_{\alpha_v} = \{v\}$ and from $\Delta(K, \alpha_v) = 2(-1)^{v \cdot K} \rho(v)$ we learn $K \cdot v$ (as in the classical Alg. 1).

  - Applying related key attacks for all $\alpha_v, v \in \mathcal{V}$ gives us $|\mathcal{V}| = t$ bits of information.

- Can be generalized to dependent case by considering a basis of $\text{span}(\mathcal{V})$ instead of $\mathcal{V}$ to learn $\leq t$ bits.
Results from Experiments
Round Reduced PRESENT [Bogdanov et al. 2007]

- 7 round 80-bit key version of PRESENT cipher
- Key schedule is semi-linear
- Extended key $K \in \mathbb{Z}_2^{104}$: round keys depend linearly on $K$
- Multiple strong trails of correlation $2^{-2R}$ for $R$ rounds
- **Direct attack**
  - $|V| = 24$, $|C| = 13$, $t = 15$, $|\rho(v)| = 2^{-14}$, $h = 3.2$
- **Related key approach**
  - Assert that $K \oplus \alpha$ can be produced ($\alpha$ must not influences non-linear parts of the key schedule)
  - $|V_\alpha| = 9$, $|C_\alpha| = 10$, $t_\alpha = 9$, $|\rho(v)| = 2^{-14}$, $h_\alpha = 2.6$
- **Multiple related key approach**
  - Learn 14.25 bits of information
- 400 random keys and $2^{32}$ plain text blocks
- Direct attack theoretically applicable on up to 12 rounds for an 80-bit key and on up to 14 rounds for a 128-bit key
Probability of Success

- Test for 400 different keys

- Multiple related key is only correct if all key classes are correct

- Related key has higher success probability
Achieved Entropy

- **Achieved entropy**: \( \text{entropy} \times \text{success probability} \)
- **Test for 400 different keys**

![Graph showing achieved entropy vs \( \log_2(N) \)]

- For \( N \geq 2^{28} \) the **multiple related key approach** leads to best result
Conclusion
Comparison (1)

► Algorithm 1 vs. Algorithm 2 for multiple strong trails

Algorithm 1

- Targets $K$
- Works for all keys
- Data complexity inverse proportional to minimal distance $d$ between elements in $C$

Algorithm 2

- Targets $K_R$
- Works for most keys
- For about half of the keys, the data complexity is better or equal to $\mathcal{O}\left((\sum_{v \in V} \rho(v)^2)^{-1}\right)$
Comparison (2)

- Multiple related key approach vs. multidimensional linear cryptanalysis for Algorithm 1

<table>
<thead>
<tr>
<th>Setting</th>
<th>Multiple related key</th>
<th>Multidimensional</th>
</tr>
</thead>
<tbody>
<tr>
<td>One approximation with multiple strong trails</td>
<td>$m$ linearly independent approx. each with one strong trail</td>
<td></td>
</tr>
<tr>
<td>$t$ dimension of trail set $\mathcal{V}$</td>
<td>$m$ number of base approx.</td>
<td></td>
</tr>
<tr>
<td>$O\left(\max_{1 \leq i \leq t} \left(\frac{</td>
<td>C_{\alpha_i}</td>
<td>- 1}{d_{\alpha_i}^2} \right) \log P_e\right)$</td>
</tr>
<tr>
<td>Offline</td>
<td>$O(t^2 2^t)$, $m$: $O(t 2^t)$</td>
<td>$t$: $O(m 2^m)$, $m$: $O(2^m)$</td>
</tr>
<tr>
<td>Online</td>
<td>$O(t N)$, $m$: $O(t)$</td>
<td>$t$: $O(m N)$, $m$: $O(2^m)$</td>
</tr>
<tr>
<td>Inform.</td>
<td>$\sim t$ bits</td>
<td>$m$ bits</td>
</tr>
</tbody>
</table>

Equations:
- $O\left(\max_{1 \leq i \leq t} \left(\frac{|C_{\alpha_i}| - 1}{d_{\alpha_i}^2} \right) \log P_e\right)$
- $O\left(\frac{(2^m - 1) - \log P_e}{2^m \sum_{\eta \in \mathbb{Z}_2^m} (p_\eta - 2^{-m})^2}\right)$
Conclusion

▶ Application of Matsui’s Algorithm 1 on key-alternating iterated block cipher which has linear approximations with multiple strong trails
▶ Precomputation complexity increases with number of trails
▶ Data complexity is inverse proportional to minimal distance between possible correlations
▶ Related key analysis reduces number of considered trails
▶ Several key differences can be combined for a better result