Rank estimation for large key

Vincent Grosso
November 15, 2017

Institute for Computing and Information Sciences – Digital Security
Radboud University Nijmegen
1. Key enumeration/Rank estimation

2. Previous solution

3. New solution
Key enumeration/Rank estimation
Each cell contains 2 piles of coins.

For each column the first pile contains the same amount of coins:

\[ P_1(a, 1) = P_1(b, 1) = P_1(1) \]

For each row the second pile contains the same amount of coins:

\[ P_2(a, 1) = P_2(a, 2) = P_2(a) \]

You can take piles of \( n \) cells, how to maximize the profit?
Each cell contains 2 piles of coins.
Each cell contains 2 piles of coins.

- For each column the first pile contains the same amount of coins
  \[ P_1(a, 1) = P_1(b, 1) = P_1(1) \]
Each cell contains 2 piles of coins.

- For each column the first pile contains the same amount of coins
  \[ P_1(a, 1) = P_1(b, 1) = P_1(1) \]

- For each row the second pile contains the same amount of coins
  \[ P_2(a, 1) = P_2(a, 2) = P_2(a) \]
Generous King

Each cell contains 2 piles of coins.

- For each column the first pile contains the same amount of coins
  \( P_1(a, 1) = P_1(b, 1) = P_1(1) \)

- For each row the second pile contains the same amount of coins
  \( P_2(a, 1) = P_2(a, 2) = P_2(a) \)

You can take piles of \( n \) cells, how to maximize the profit?

How many better solution

How to many cells have more coins than a specified cell?
The problem can be generalized

- with higher number of rows/columns
- higher dimension
Side-channel attacks

\[ x \xleftarrow{k = 0} z_0 \xrightarrow{\bigoplus} S\text{-box} \xrightarrow{} y_0 \xrightarrow{\text{model}} \Pr[k = 0] \]

Divide-and-conquer strategy.
Side-channel attacks

\[ S-box \oplus x \rightarrow \Pr[k = 1] \rightarrow m_1 \rightarrow \text{model} \]

\( k = 1 \)

Divide-and-conquer strategy.
Side-channel attacks

Divide-and-conquer strategy.
## SCA result

<table>
<thead>
<tr>
<th>$k_0$</th>
<th>$k_1$</th>
<th>$k_2$</th>
<th>$\cdots$</th>
<th>$k_{15}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x2b, 0.125</td>
<td>0x23, 0.128</td>
<td>0x23, 0.325</td>
<td></td>
<td>0x45, 0.347</td>
</tr>
<tr>
<td>0xcd, 0.100</td>
<td>0x51, 0.045</td>
<td>0xde, 0.204</td>
<td></td>
<td>0xdc, 0.210</td>
</tr>
<tr>
<td>0xae, 0.050</td>
<td>0xff, 0.035</td>
<td>0xfe, 0.036</td>
<td></td>
<td>0x83, 0.151</td>
</tr>
<tr>
<td>0x12, 0.025</td>
<td>0x2b, 0.025</td>
<td>0x21, 0.029</td>
<td></td>
<td>0x13, 0.435</td>
</tr>
</tbody>
</table>

Real key.
Previous solution
### The histogram

<table>
<thead>
<tr>
<th>Candidate</th>
<th>Pr</th>
<th>$k_1$</th>
<th>bin</th>
<th>Pr</th>
<th>$k_2$</th>
<th>bin</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.6643</td>
<td>-0.5901</td>
<td>1</td>
<td>0.0012</td>
<td>-9.7027</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>0.2588</td>
<td>-1.9501</td>
<td>1</td>
<td>0.0011</td>
<td>-9.8283</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>0.0313</td>
<td>-4.9977</td>
<td>2</td>
<td>0.3588</td>
<td>-1.4787</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0.0412</td>
<td>-4.6012</td>
<td>2</td>
<td>0.0713</td>
<td>-3.8100</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0.0001</td>
<td>-13.2877</td>
<td>4</td>
<td>0.5643</td>
<td>-0.8255</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0.0020</td>
<td>-8.9658</td>
<td>3</td>
<td>0.0012</td>
<td>-9.7027</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>0.0013</td>
<td>-9.5873</td>
<td>3</td>
<td>0.00005</td>
<td>-14.2877</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>0.0010</td>
<td>-9.9658</td>
<td>3</td>
<td>0.00205</td>
<td>-8.9302</td>
<td>3</td>
</tr>
</tbody>
</table>

![Histogram $h_1$](image1)

![Histogram $h_2$](image2)
Mix result

Perform convolution of histogram

\[ \text{conv}(h_1, h_2)[i] = \sum_{j=0}^{i} h_1[j] h_2[i - j] \]
Limitation for larger keys

For large number of dimension we perform convolution on larger and larger histograms: could be costly.

![Graph showing execution time vs. number of subkeys for different bins sizes and languages (matlab and C).](image-url)
New solution
Batching

Keep the size of the histogram constant

 batching(\text{conv}(h_1, h_2))
Our solution has a complexity linear in the number of subkeys
Tightness

![Graph showing tightness vs. number of subkeys]

- **FSE’15 $2^{16}$ bins matlab**
- **Our solution $2^{16}$ bins matlab**

- FSE’15
- **Our solution**
Tightness

![Graph showing tightness versus number of subkeys]

- **FSE'15**: $2^{16}$ bins matlab
- **Our solution**: $2^{16}$ bins matlab

<table>
<thead>
<tr>
<th># subkeys</th>
<th>FSE'15 2^{16} bins matlab</th>
<th>Our solution 2^{16} bins matlab</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>140</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FSE’15

Our solution
Similar tightness

![Graph showing execution time versus number of subkeys for different solutions.]

- FSE’15 (1 bit)
- FSE’15 (0.3 bit)
- Our solution (1 bit)
- Our solution (0.3 bit)
Thanks