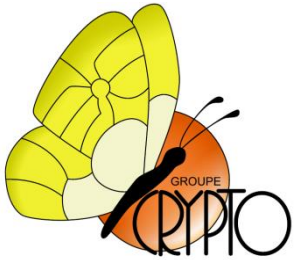
Simple Key Enumeration (and Rank Estimation) using Histrogams: an Integrated Approach

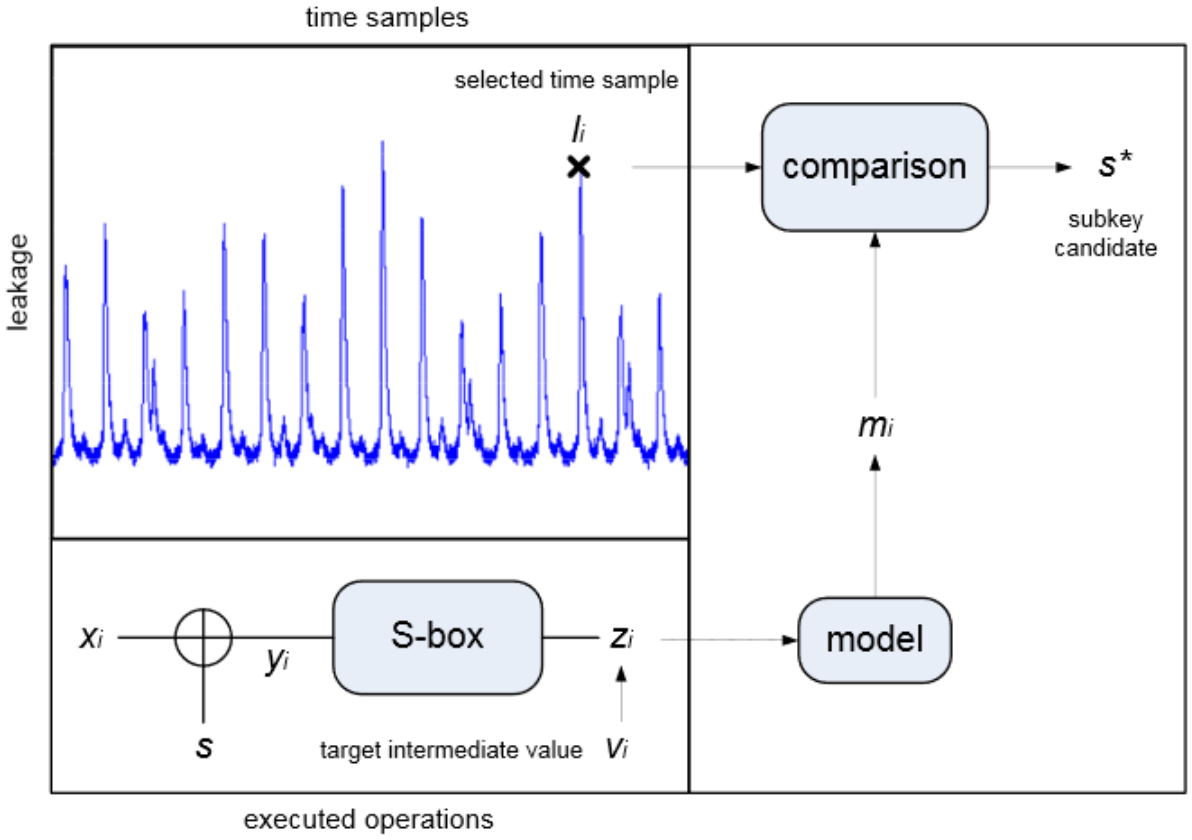
Romain Poussier, François-Xavier Standaert: Université catholique de Louvain, Belgium
Vincent Grosso: Rurh-Universität Bochum, Germany



Outline

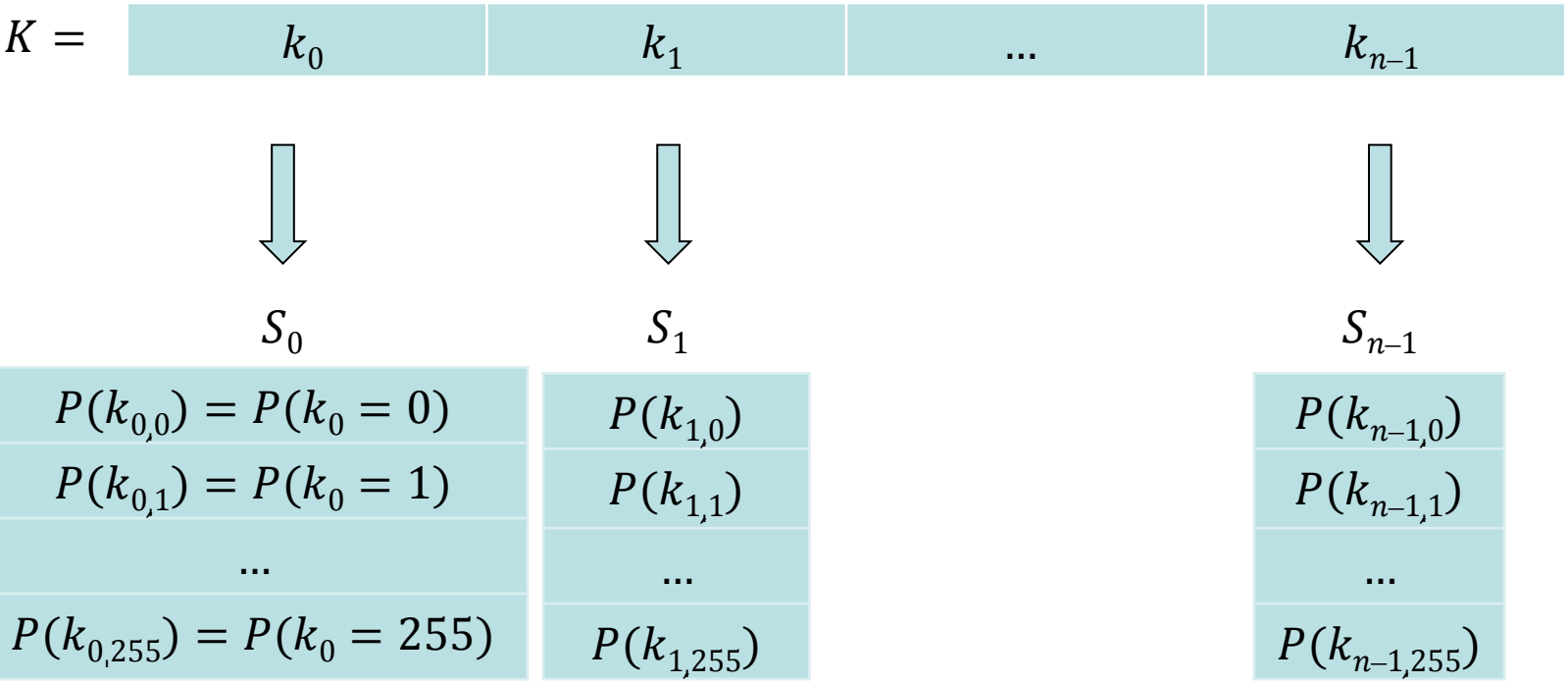
- Side channel attacks (SCA)
- Rank estimation
- Key enumeration
- Experimental results
- Conclusion

SCA principle



Divide and conquer SCA results

Real key: $k = k_0, \ldots, k_{n-1}$



Exploitation of SCA results

3 cases

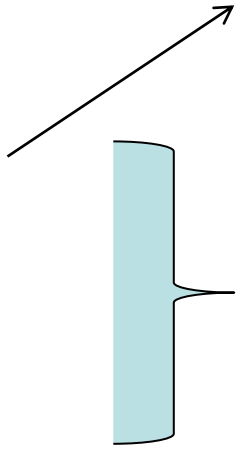
Direct recovery	Enumeration	Rank Estimation
• Enough SCA Information	• Not enough SCA Information • Enough computational power	• Not enough SCA Information • Not enough computational power
	Evaluator/Atta-cker	Evaluator

Outline

- Side channel attacks (SCA)
- Rank estimation
- Key enumeration
- Experimental results
- Conclusion

Literature

- 2013: Veyrat-Charvillon et al.
- 2014: Ye et al.
- 2015: Glowacz et al.
- 2015: Bernstein et al.
- 2015: Martin et al.
- 2015: Duc et al.
- 2015: Poussier et al.



Building block of our enumeration algorithm

Less than 1 bit of accuracy within second(s)

Glowacz et al. algorithm steps

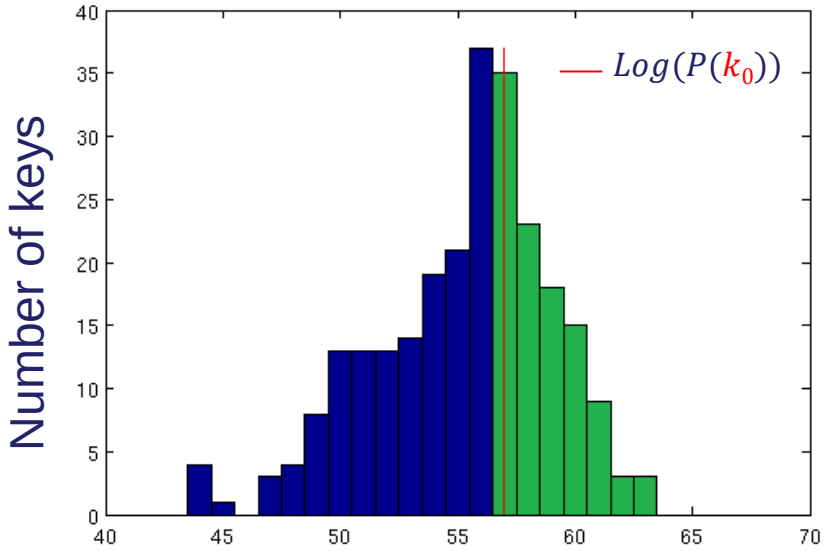
1) Building histograms from SCA results

2) Combining histograms (convolution)

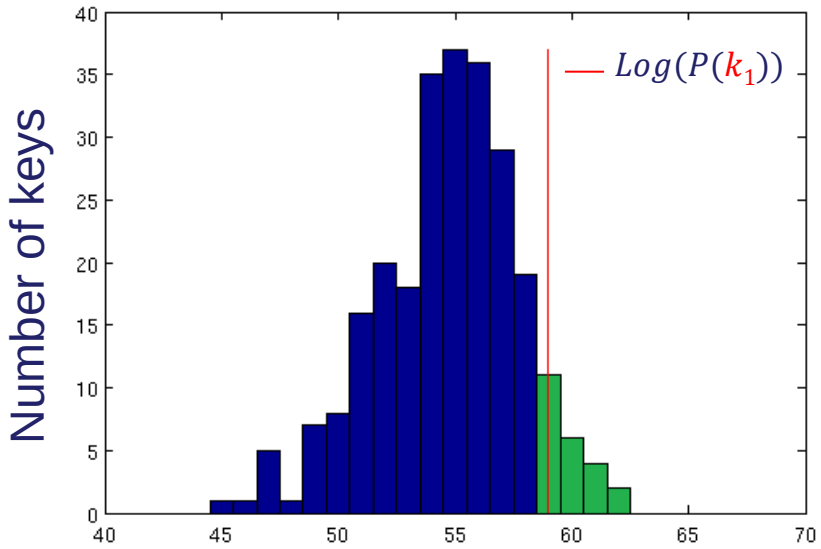
3) Counting the rank

Step 1. Building histograms

subkey 0

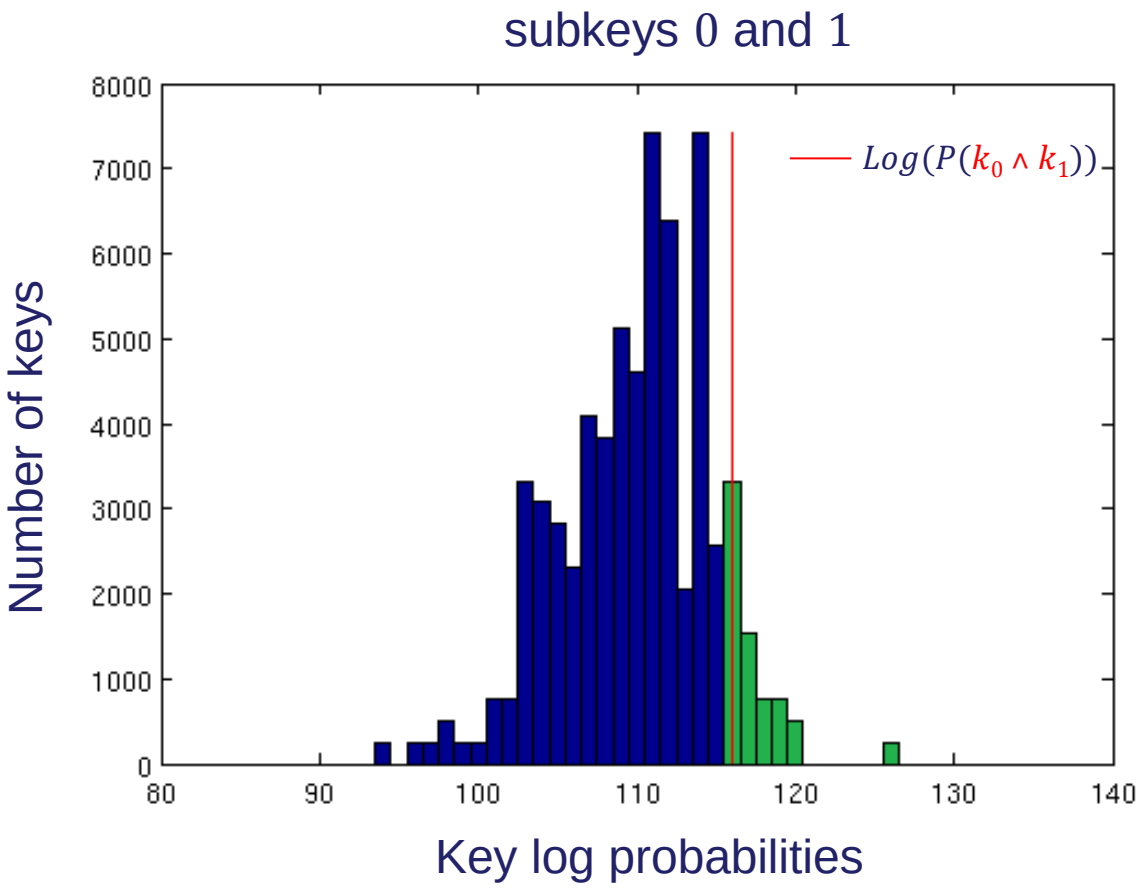


subkey 1



The rank is the number of keys in the green zone

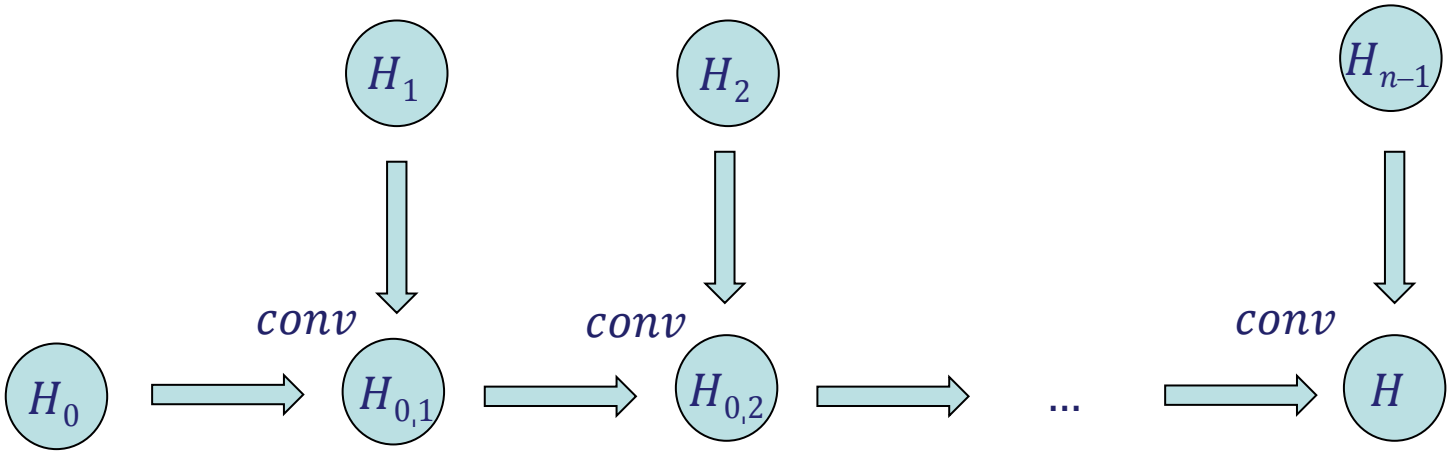
Step 2. Combining histograms



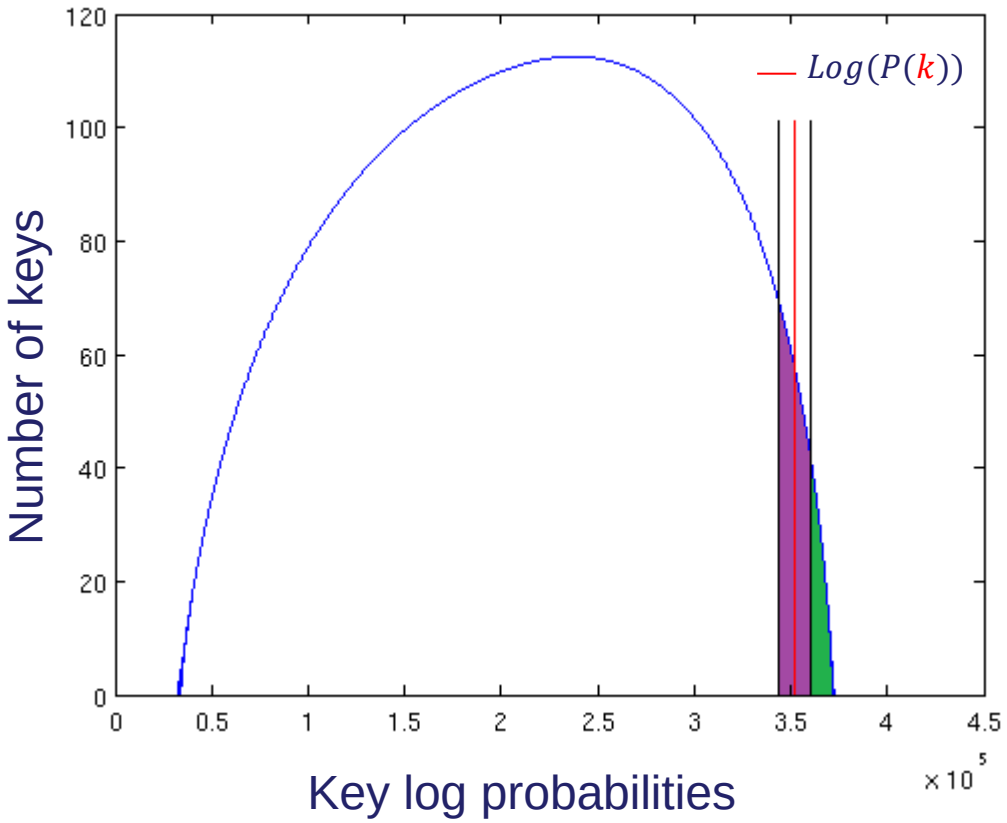
Let B_k denotes the bin where the real key lies

Step 2. Combining histograms

- Goal: estimate $H = H(S)$ from $H(S_0), \ldots, H(S_{n-1}) = H_0, \ldots, H_{n-1}$



Step 3. Counting the rank



+ Higher bound $= B_k + \lceil \frac{n}{2} \rceil$

Lower bound $= B_k - \lceil \frac{n}{2} \rceil$

Tightness

Outline

- Side channel attacks (SCA)
- Rank estimation
- Key enumeration
- Experimental results
- Conclusion

Literature

- 2012: Veyrat-Charvillon et al.
  - Optimal
  - Memory intensive
  - Sequential
- 2015: Bogdanov et al.
  - Suboptimal, no bounds
  - Memory efficient
  - Parallelization
- 2015: Martin et al.
  - Suboptimal, bounds
  - Memory efficient
  - Parallelization
- 2015: David et al.
  - Suboptimal, no bounds
  - Memory efficient
  - Sequential
- 2016: Histogram enumeration.
  - Suboptimal, **easy** bounds
  - Memory efficient
  - **Easy** parallelization

Histogram enumeration steps

1) Building histograms from SCA results

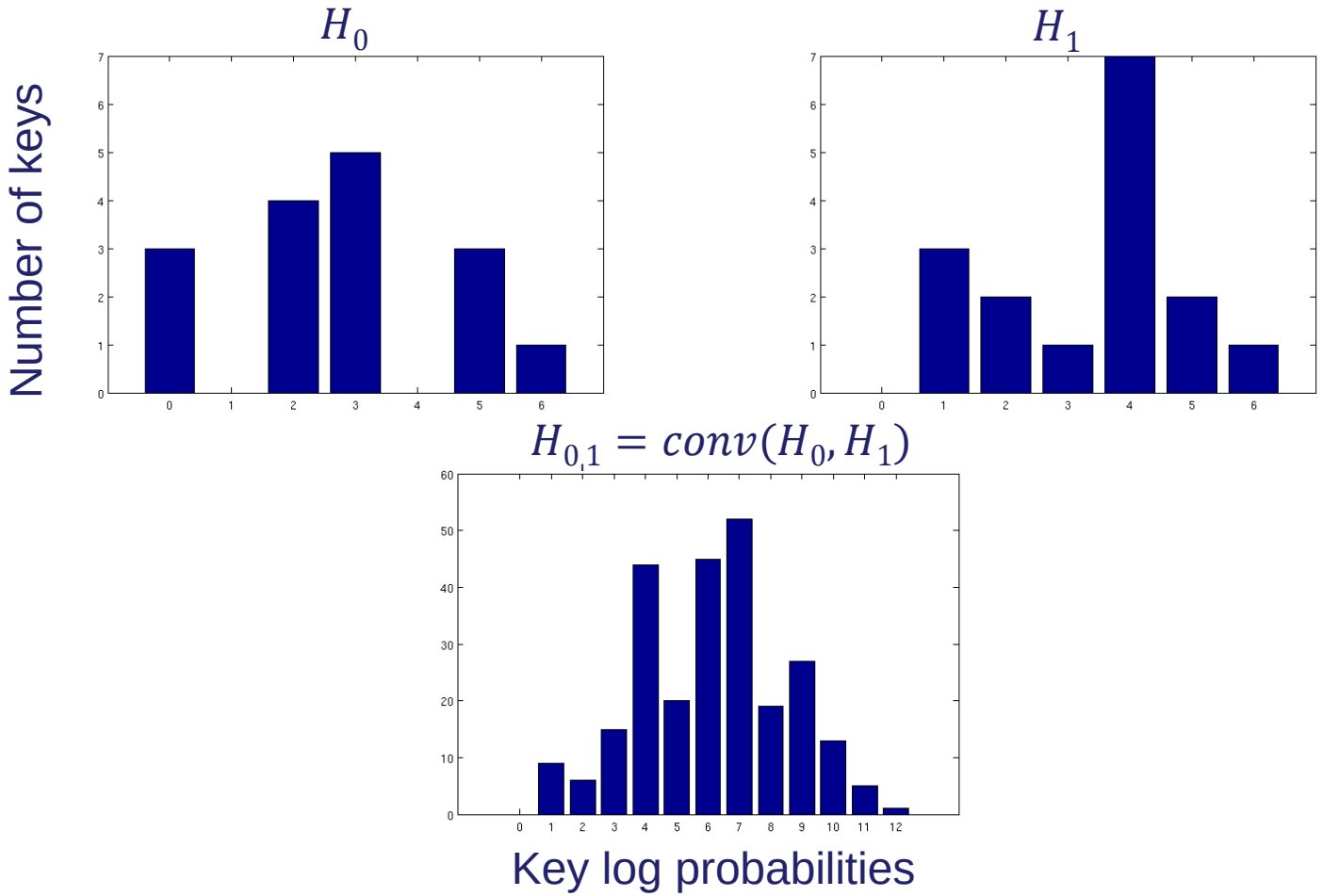
2) Combining histograms (convolution)

3) Bounds search = counting the rank

Same as Glowacz et al. algorithm

4) Enumeration (Backtracking)

Building phase: step 1 and 2.

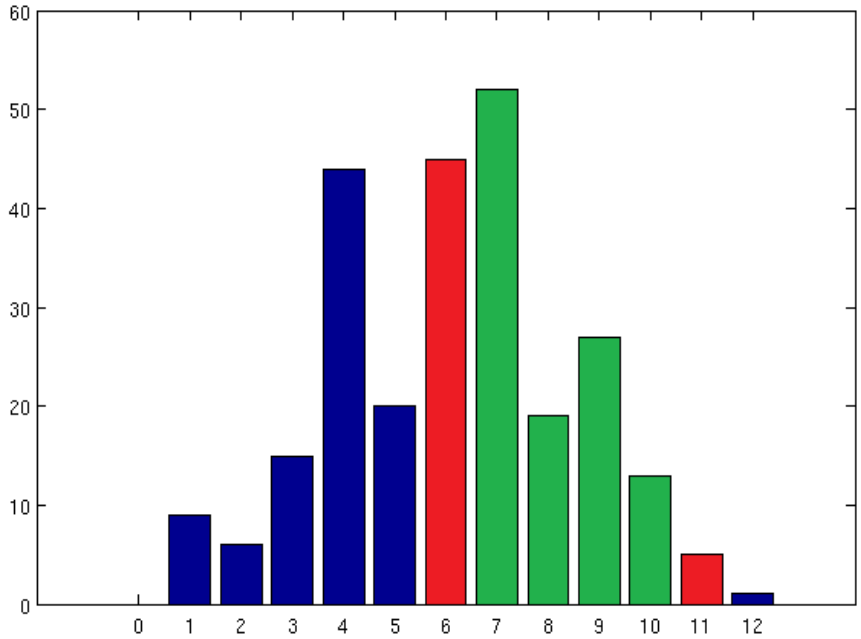


Step 3. Bounds search

Rounded ranks between 10 and 100

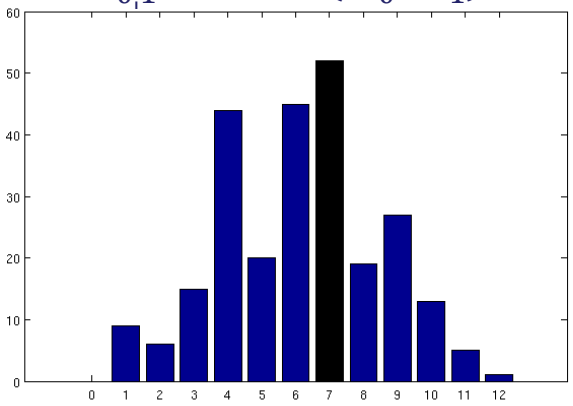
Rounded ranks between 10 and 100 + bounds

$$H_{0,1} = [0,9,6,15,44,20,45,52,19,27,13,5,1]$$



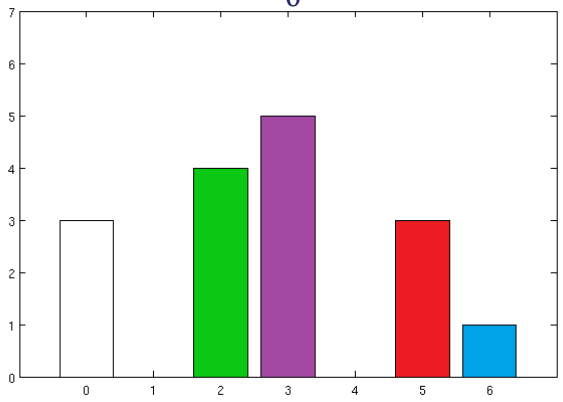
Step 4. Backtracking: two subkeys case

$H_{0,1} = conv(H_0, H_1)$

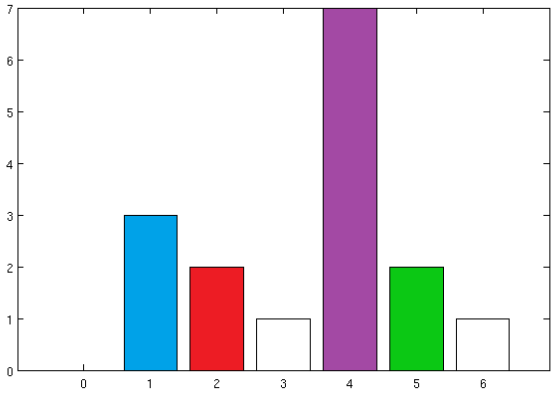


same colors: $H_0(i) + H_1(j) = 7$.
$\longrightarrow$ Possible key combination with a score of 7 in $H_{0,1}$ (black).

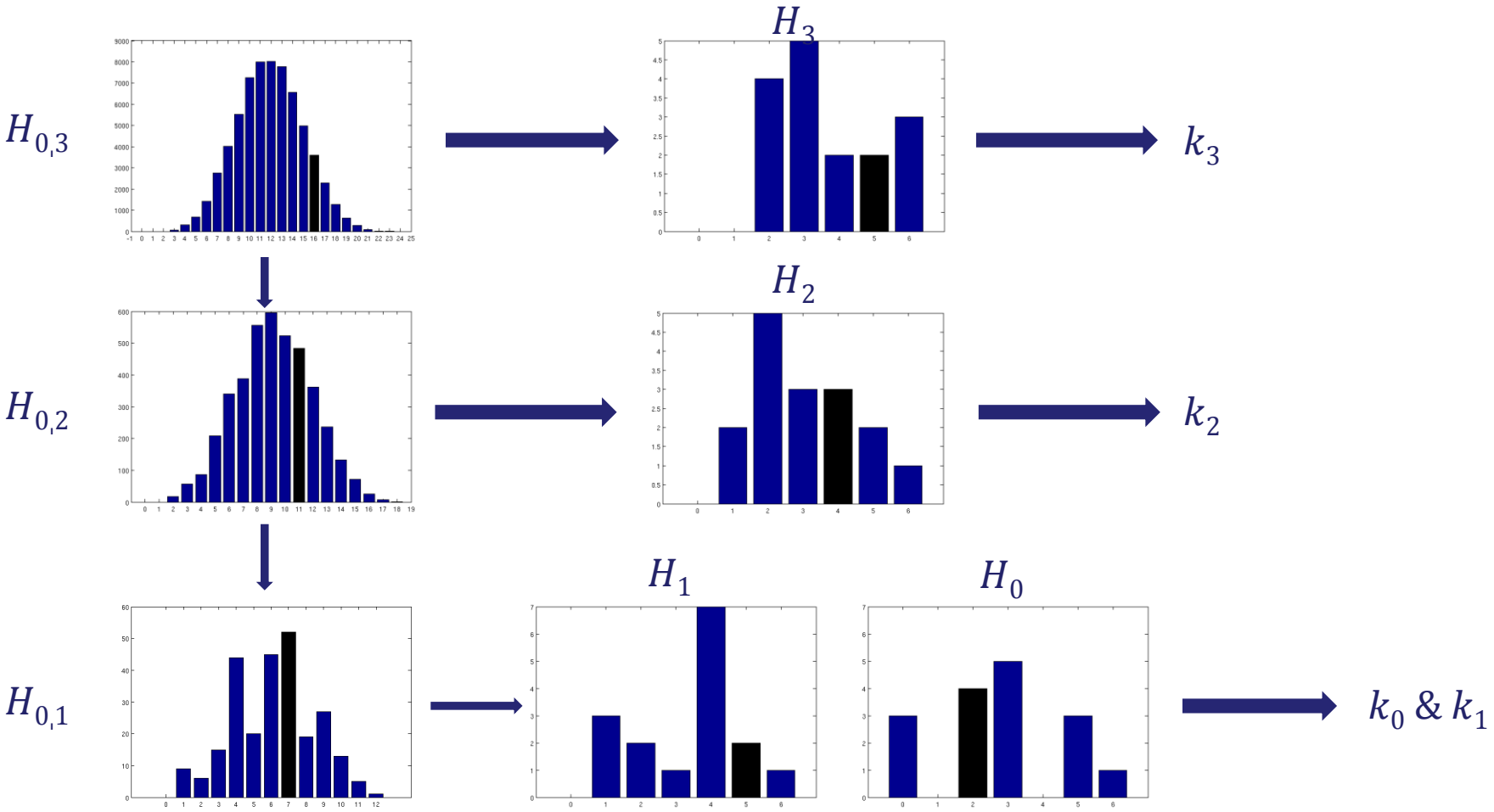
H_0



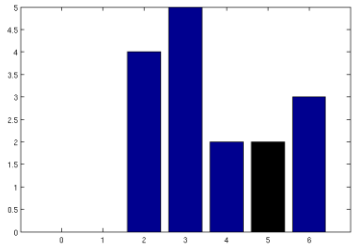
H_1



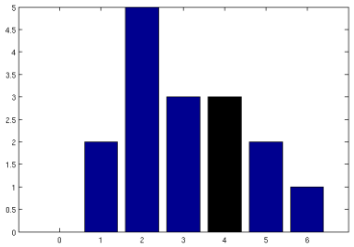
Step 4. Backtracking: recurrence



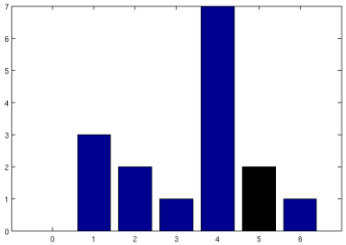
Factorization of the output



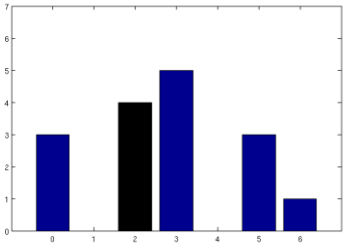
$k_3 \in \{0\text{x}22, 0\text{x}53\}$



$k_2 \in \{0x32, 0x42, 0xA4\}$



$k_1 \in \{0\text{x}8\text{E}, 0\text{x}\text{C}6\}$



$k_0 \in \{0x03, 0xFA, 0xE2, 0x52\}$

Factorized lists: 4 lists of 2,3,2 and 4 elements

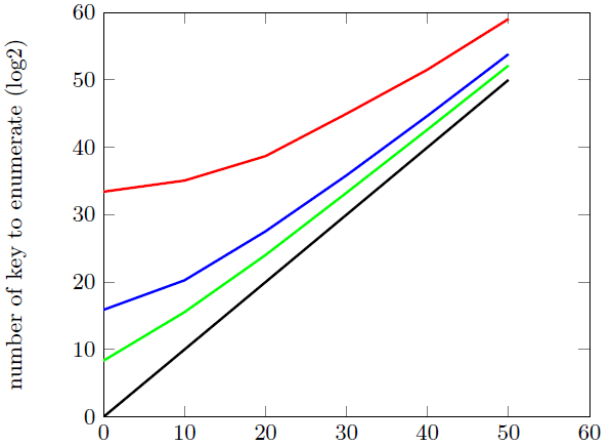
$(0x22, 0x32, 0x8E, 0x03)$
$(0x53, 0x32, 0x8E, 0x03)$ $\longrightarrow$ Defactorized list: 1 list of 2x3x2x4 = 48 key candidates
$(0x22, 0x42, 0x8E, 0x03)$
….

Outline

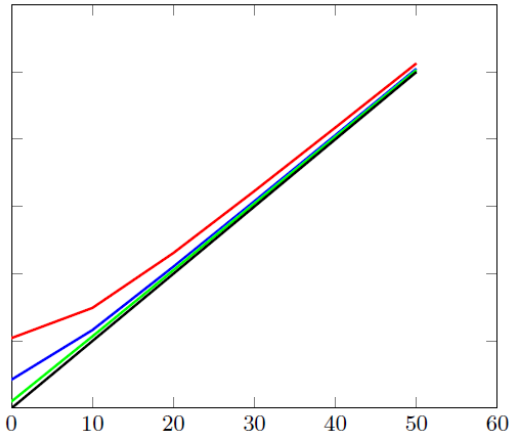
- Side channel attacks (SCA)
- Rank estimation
- Key enumeration
- Experimental results
- Conclusion

Accuracy

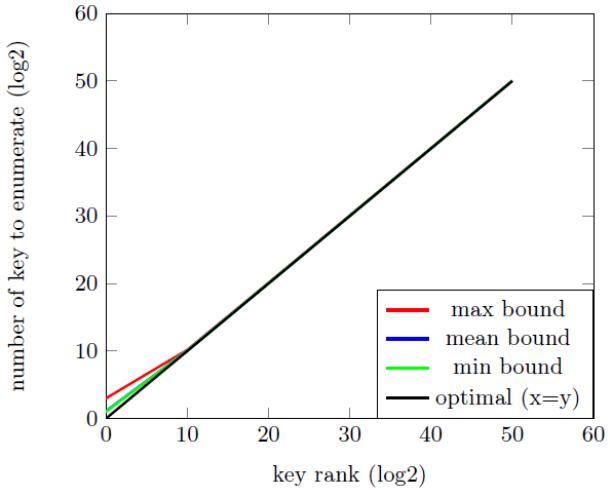
256 bins



2048 bins



65536 bins

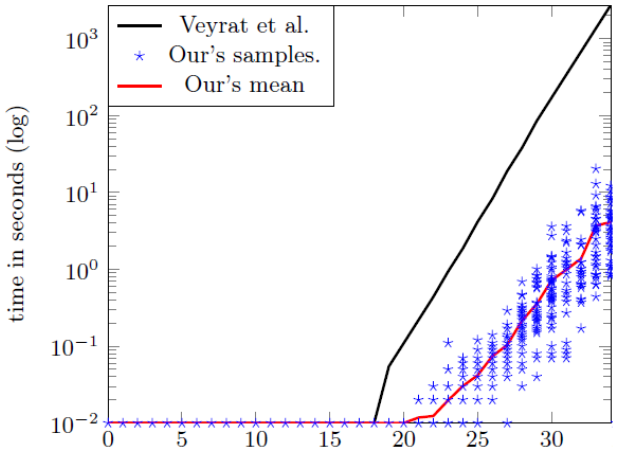


Y-coordinate: number of keys to enumerate to guarantee an enumeration up to an exact key rank given by the X-coordinate.

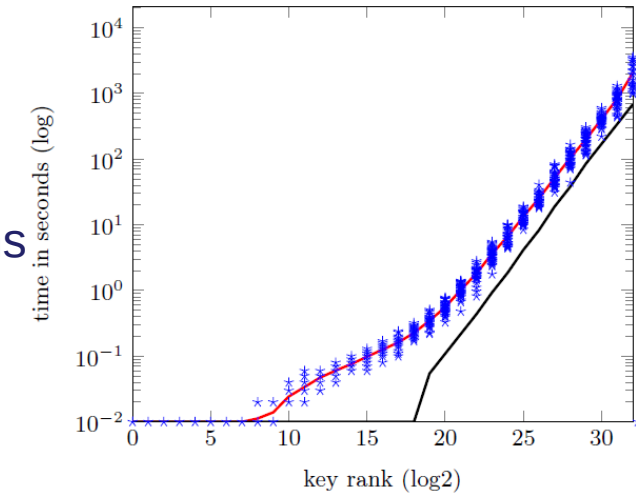
Common to all rounded enumeration algorithms.

Speed

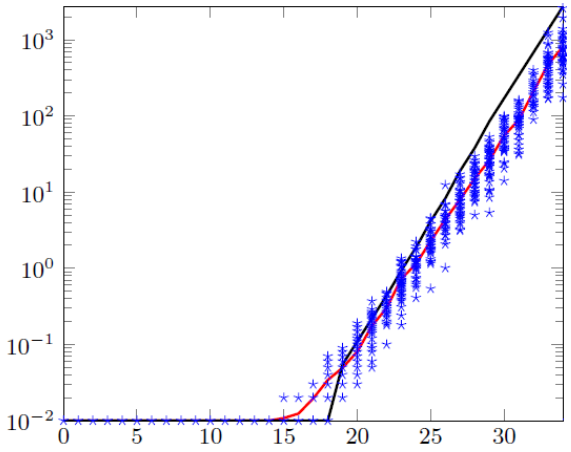
256 bins



65536 bins



2048 bins



Y-coordinate: time to enumerate up to the depth given by the X-coordinate.

Consequences on key testing

Tradeoff between precision and testing capabilities

Local attacker
Computing bound < 2^{40}

Organization: big computing infrastructure
Computing bound > 2^{40}

Consequences on key testing

Tradeoff between precision and testing capabilities

Local attacker
Computing bound < 2^{40}

- Software enumeration and key testing.

Organization: big computing infrastructure
Computing bound > 2^{40}

Consequences on key testing

Tradeoff between precision and testing capabilities

Local attacker
Computing bound < 2^{40}

- Software enumeration and key testing.

Key testing on the fly: maximize accuracy ⟶ 2048 or 65536 bins and defactorized lists

Organization: big computing infrastructure
Computing bound > 2^{40}

Consequences on key testing

Tradeoff between precision and testing capabilities

Local attacker
Computing bound < 2^{40}

- Software enumeration and key testing.

Key testing on the fly: maximize accuracy ⟶ 2048 or 65536 bins and defactorized lists

Organization: big computing infrastructure
Computing bound > 2^{40}

- Software enumeration and hardware key testing

Consequences on key testing

Tradeoff between precision and testing capabilities

Local attacker
Computing bound < 2^{40}

- Software enumeration and key testing.

Key testing on the fly: maximize accuracy ⟶ 2048 or 65536 bins and defactorized lists

Organization: big computing infrastructure
Computing bound > 2^{40}

- Software enumeration and hardware key testing

Separate enumeration and key testing: Minimize bandwidth ⟶ 256 bins for a high factorization

Conclusion

- Yet another key enumeration algorithm:
  - Simple bounds
  - Key factorization (as Martin et al.)
  - Easy parallelization balancing (cf. paper)
  - Simple construction

- Open source with code examples can be downloaded at : http://perso.uclouvain.be/fstandae/PUBLIS/172.zip
  - Read the paper to get more insight on the parameters

Thank you for your attention !