Key Enumeration from the Adversarial Viewpoint

When to Stop Measuring and Start Enumerating?

Melissa Azouaoui   *Romain Poussier*   *François-Xavier Standaert*   *Vincent Verneuil*

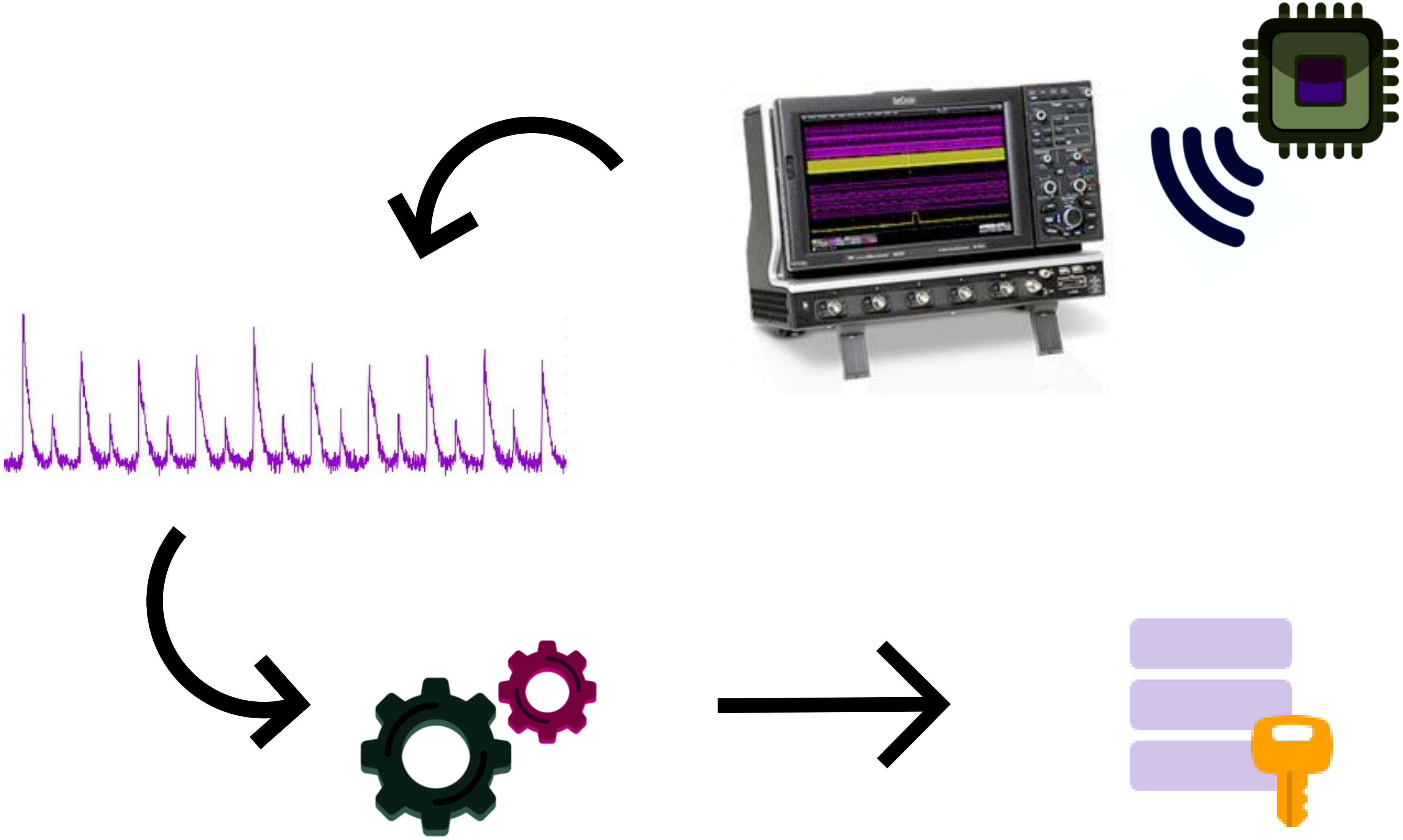
UCL
Université catholique de Louvain

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

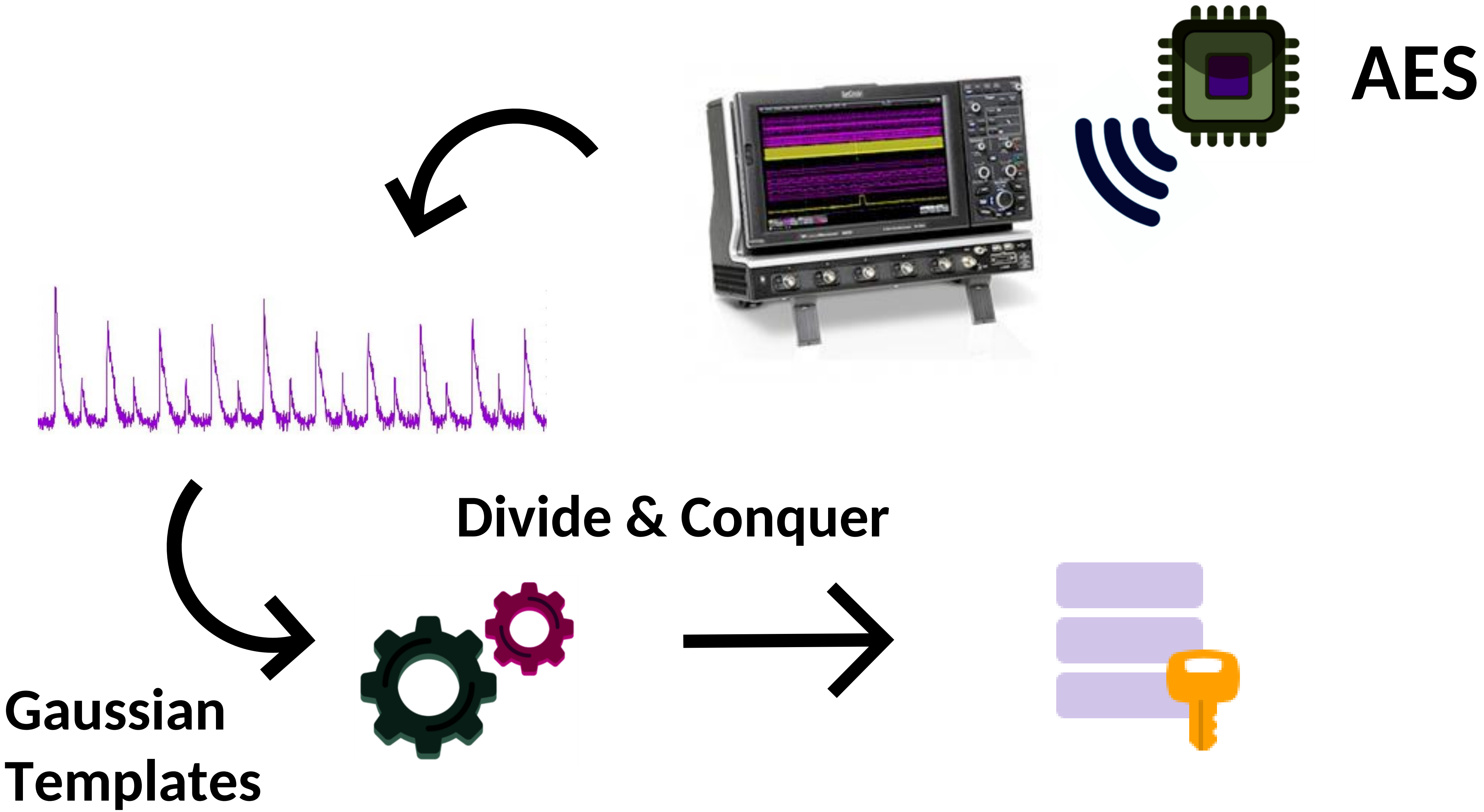
Outline

- Background: SCA, Enumeration and Rank Estimation
- Question
- Heuristic solution and comparison to related works
- Experiments
- Limitations
- Conclusion

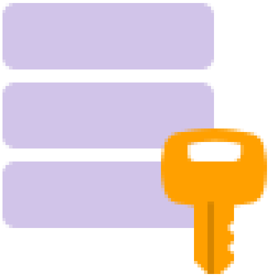
Side-channel attacks



Side-channel attacks



Side-channel attacks



Information on Subkeys

Key = $\mathbf{k_0}$ $\mathbf{k_1}$ . . . $\mathbf{k_{15}}$

Probabilities (or Scores)

Pr[k_0] = 0
Pr[k_0] = 1
. . .

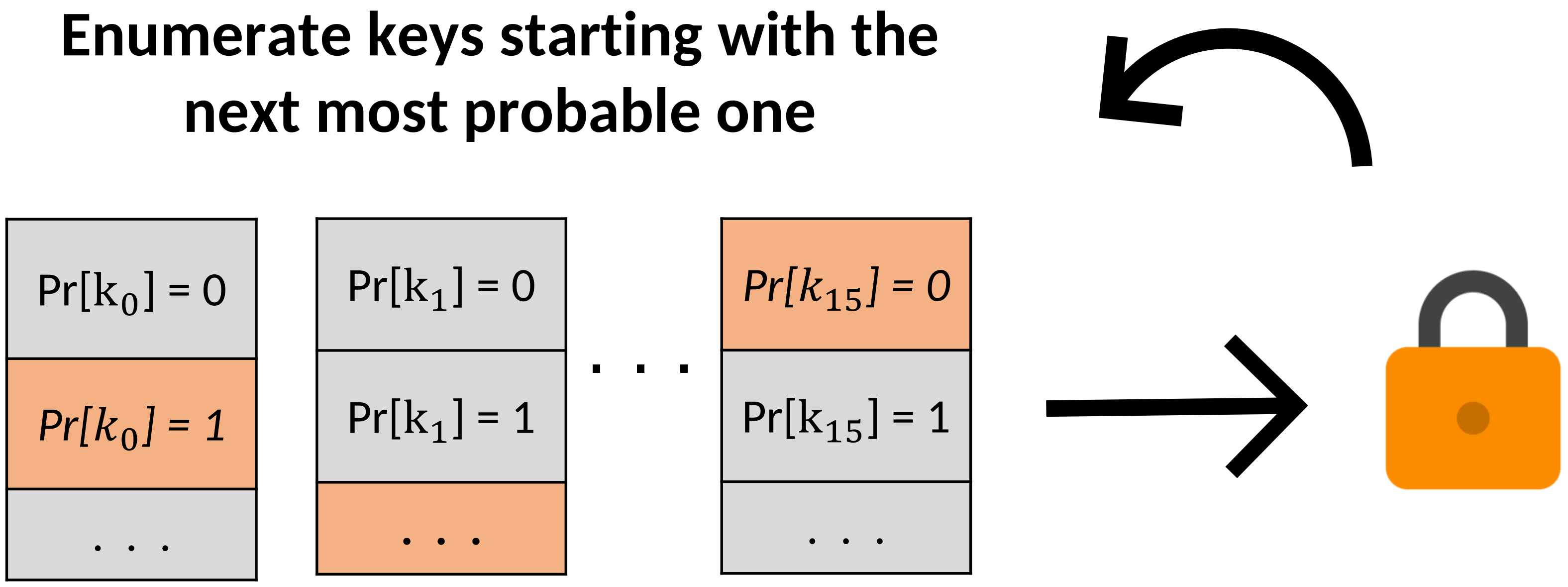
Pr[k_1] = 0
Pr[k_1] = 1
. . .

. . .

Pr[k_{15}] = 0
Pr[k_{15}] = 1
. . .

Key enumeration

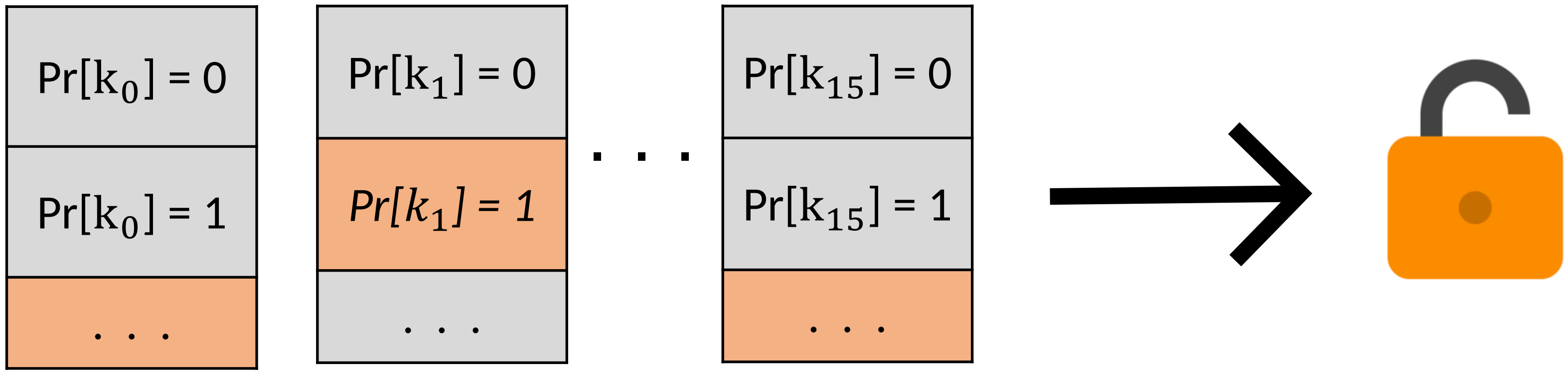
- Attacker tool
- Trade data complexity for time complexity



Key enumeration

- Attacker tool
- Trade data complexity for time complexity

Enumerate keys starting with the next most probable one



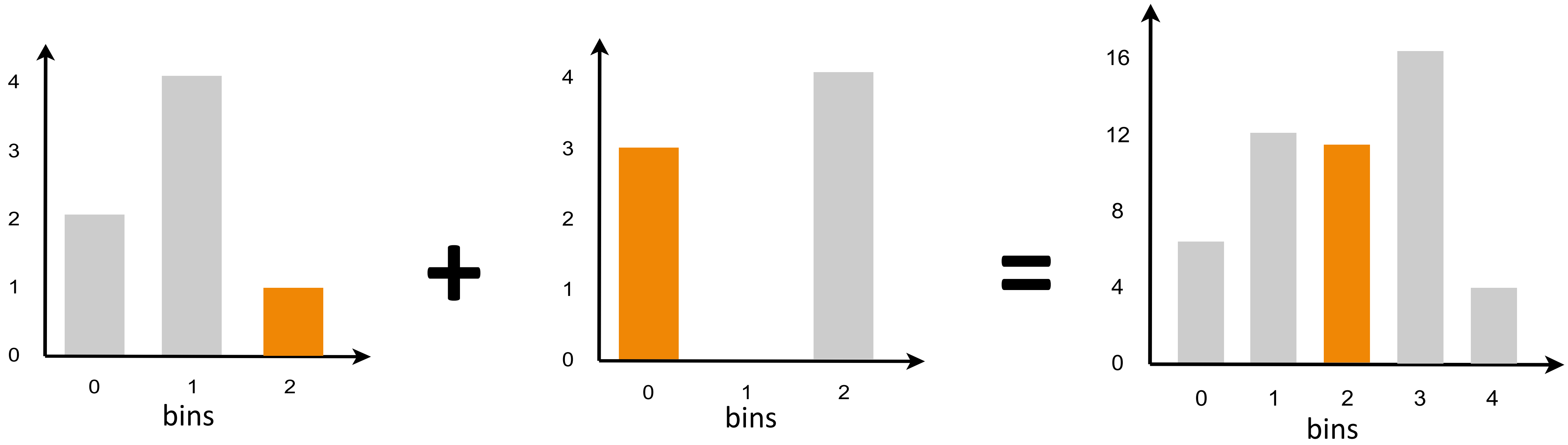
Key rank estimation

- Evaluator tool that requires the knowledge of the key
- Finds the key rank efficiently without enumeration

Histogram-based Key Rank Estimation
Glowacz *et al.* FSE 2015

Key rank estimation

Histogram-based Key Rank Estimation – FSE 2015



H_0 = hist($\mathbf{log}(\mathbf{Pr}[\mathbf{K_0}])$)

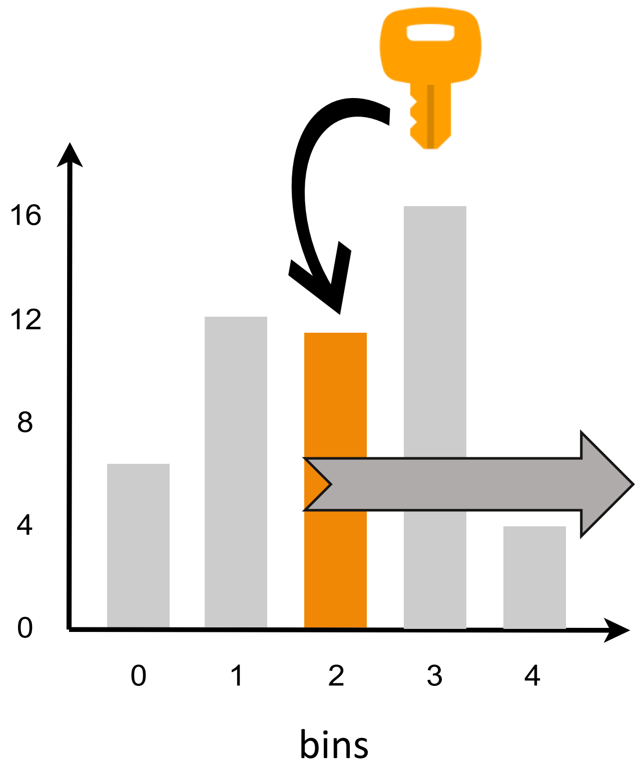
H_1 = hist($\mathbf{log}(\mathbf{Pr}[\mathbf{K_1}])$)

H_2 = hist($\mathbf{log}(\mathbf{Pr}[\mathbf{K_0}]) + \mathbf{log}(\mathbf{Pr}[\mathbf{K_1}])$)

= **conv**$(\mathrm{H}_0, \mathrm{H}_1)$

Key rank estimation

Histogram-based Key Rank Estimation – FSE 2015



RANK = # of keys in the bins with higher log probability than the correct key

Question

Practical problem:

- An attacker does not know the position of key
- An attacker does not know if enumeration will succeed for a reasonable effort

Question

Practical problem:

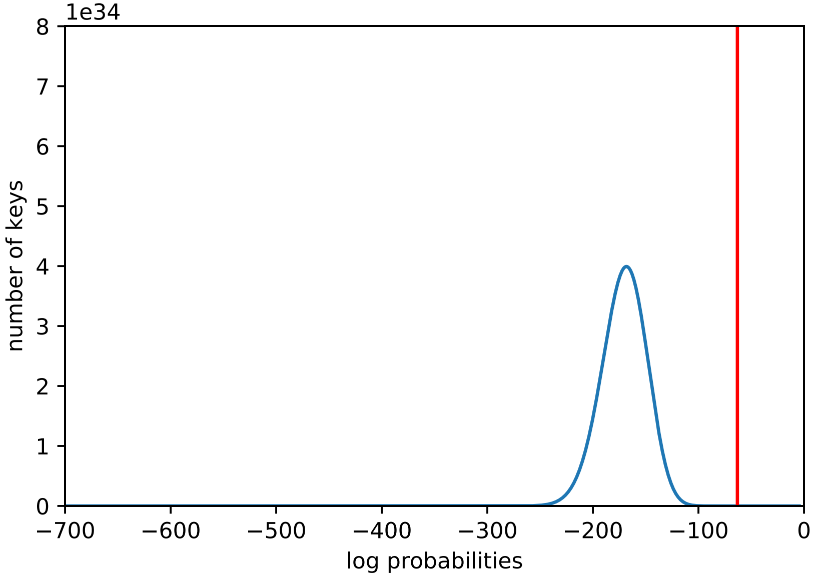
- An attacker does not know the position of key
- An attacker does not know if enumeration will succeed for a reasonable effort

How to Efficiently **approximate the rank without the knowledge of the key after an attack?**

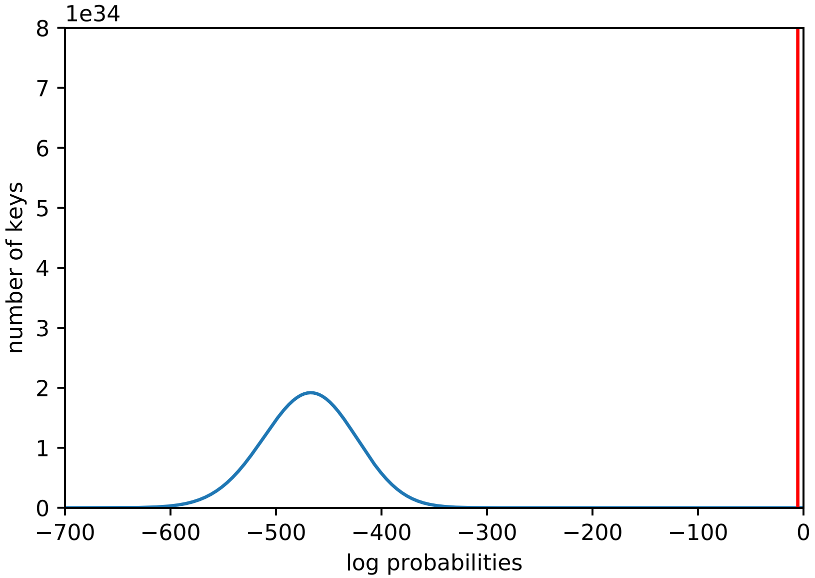
Heuristic solution

Distribution of the key candidates log probabilities. X-axis: log probabilities, Y-axis: number of keys having a certain log probability

The red vertical line correspond to the bin where the log probability of the key is



Key rank = 2^{87}



Key rank = 2^{4}

Heuristic solution

- The entropy of the key tells us *approximately* how many bits of information are left to recover
- The histogram from the FSE’15 rank estimation method is a compressed representation of the distribution of the full key

Heuristic solution

- The entropy of the key tells us *approximately* how many bits of information are left to recover
- The histogram from the FSE'15 rank estimation method is a compressed representation of the distribution of the full key

Estimate the remaining entropy of the key using the histogram

Heuristic solution

Given the histogram:

$\mathbf{bin}[i]$: center (log probability) of the i^{th} bin

$\mathbf{freq}[i]$: number of keys in the i^{th} bin

The entropy can be estimated as:

$$\widetilde{\mathbf{H}} \approx \underbrace{\sum_i \mathbf{freq}[i]}_{\text{Sum over all keys}} . \underbrace{\exp(\mathbf{bin}[i]) . \mathbf{bin}[i]}_{\Pr[\mathrm{K} = \mathrm{k}] . \log(\Pr[\mathrm{K} = \mathrm{k}])} \qquad \textbf{(1)}$$

Requires normalization s.t. $\sum_i \mathbf{freq}[i]. \exp(\mathbf{bin}[i]) = 1$

Comparison to related work

Key-agnostic Rank Estimation

Choudary and Popescu CHES'17

Bounds a GE-like metric that does not require the knowledge of the key

$\mathbf{p} = \left[\mathrm{p}_1 > \mathrm{p}_2, > \cdots > \mathrm{p}_{|\mathrm{K}|}\right]$: Sorted vector of key probabilities

$$\mathbf{GE_{kl}} = \sum_i \boldsymbol{i} \times \mathbf{p}_{\boldsymbol{i}} \qquad \textbf{(2)}$$

Comparison to related work

Difference between the $\mathbf{GE_{kl}}$ and the $\mathbf{GE}$ (used in SCA):

$$\mathbf{GE_{kl}} = \mathbf{E_{attack}} \sum_i \boldsymbol{i} \times \mathbf{p}_{\boldsymbol{i}}$$

= Expectation of the position of a key after the attack

$$\mathbf{GE} = \mathbf{E_{attack}}(\mathbf{R})$$

= Expectation of the position or rank of the correct key

The $\mathbf{GE_{kl}}$ is close to the $\mathbf{GE}$ if the templates used for the attack are perfect

Comparison to related work

We look at what happens when using this key-less GE for the single-attack case.

$$\widetilde{\mathbf{GE}}_{\mathbf{kl}} \approx \sum_{i} \underbrace{\left(\sum_{j=i} \mathbf{freq}[j]\right)}_{\text{Position}} . \underbrace{\exp(\mathbf{bin}[\boldsymbol{i}])}_{\text{Probability}} \quad \textbf{(3)}$$

Comparison to related work

What we have so far and what we want to compare:

- $\mathbf{log_2(R)}$
- $\widetilde{\mathbf{H}}$
- $\mathbf{log_2(\widetilde{GE}_{kl})}$

requires the knowledge of the key

do not require the knowledge of the key

Experiments

Gaussian template attack on the AES

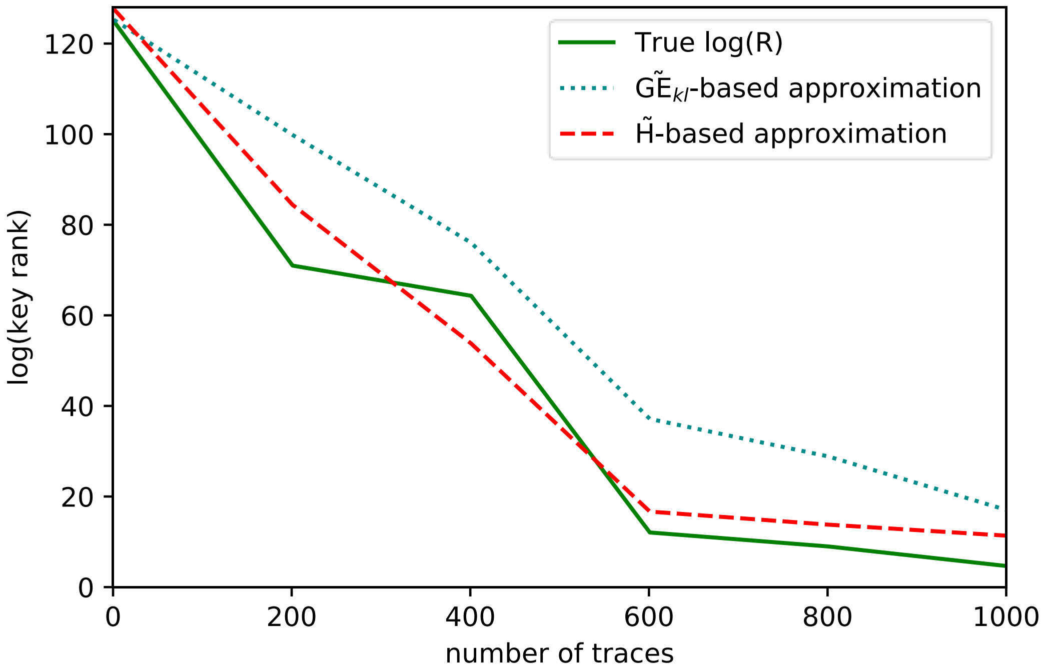
Simulated traces

- Sbox output (HW leakage, $\sigma = 10$)

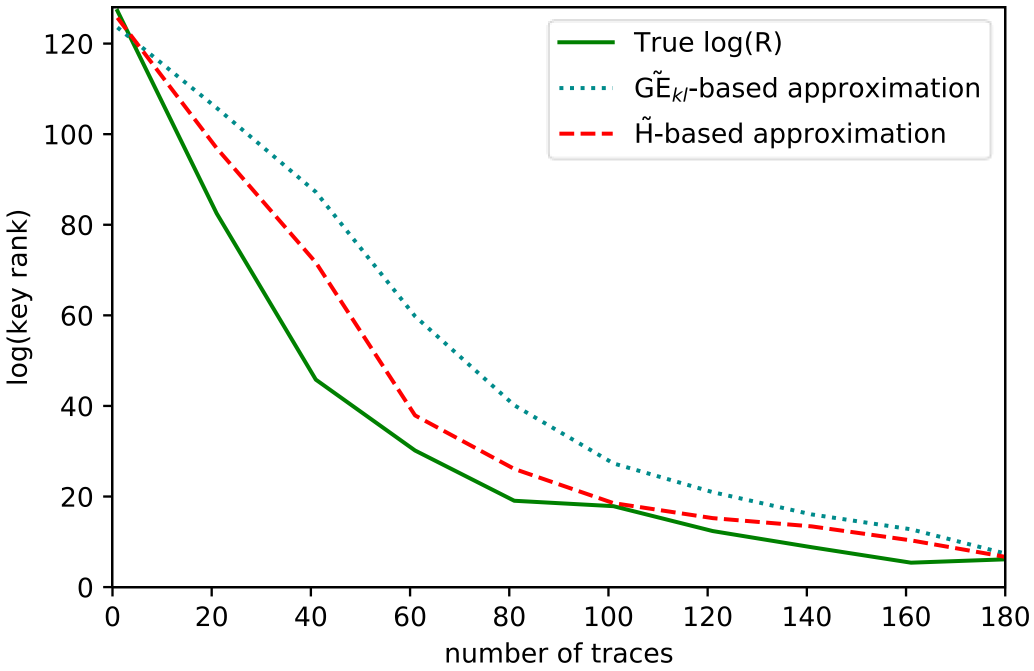
Real traces

- EM traces, ARM cortex-M3, Sbox output

Experiments: One attack



Simulated Leakages



Real Traces

Experiments: distance to the rank

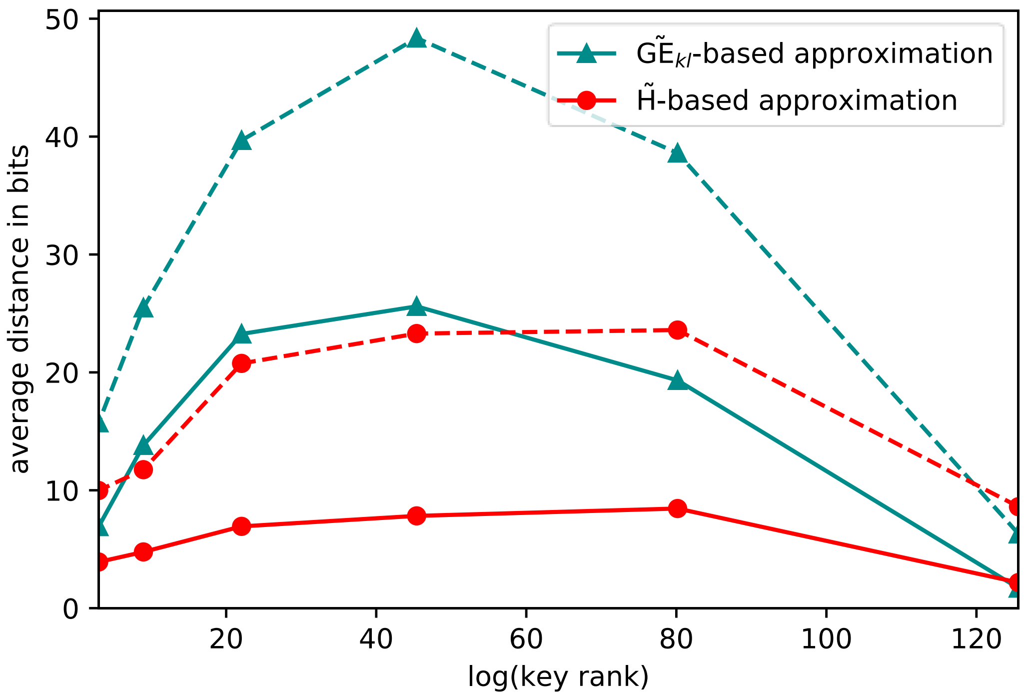
We compare:

- $\left|\log_2 \mathbf{R} \; - \widetilde{\mathbf{H}}\right|$
- $\left|\log_2 \mathbf{R} \; - \log_2 \widetilde{\mathbf{GE}}_{\mathbf{kl}}\right|$

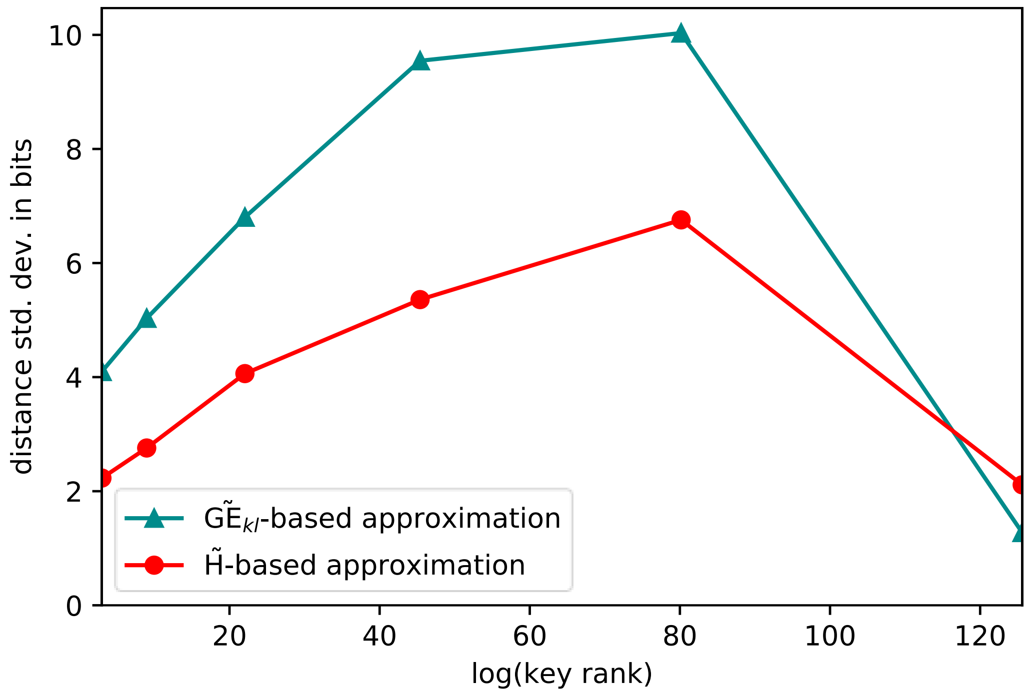
On average, over multiple iterations of the attack, for different rank values.

Experiments: distance to the rank (simulated)

—— average - - - maximum



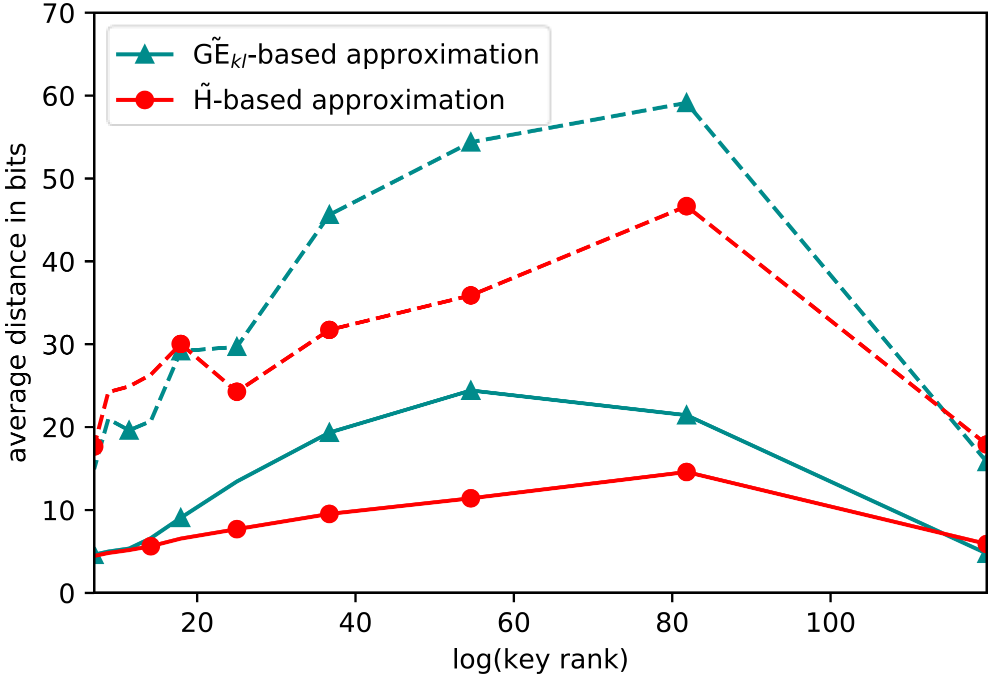
Average distance



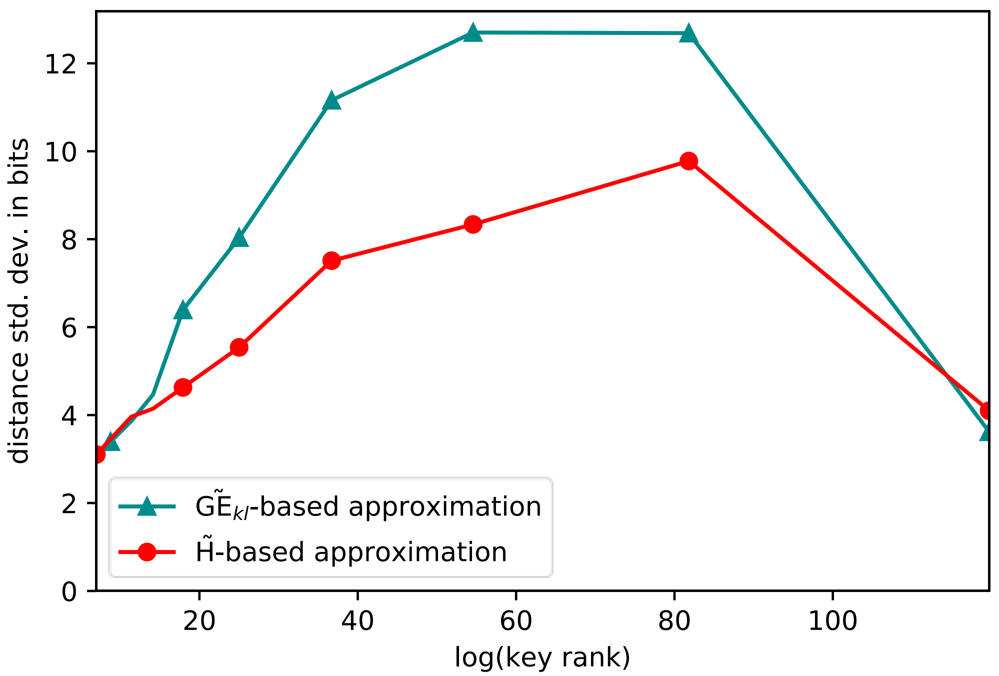
Variance of the distance

Experiments : distance to the rank (EM traces)

—— average - - - maximum



Average distance



Variance of the distance

Caveats and limitations

- Imperfect leakage characterization (for e.g. wrong assumption on the leakage model)
- Flawed attack (for e.g. wrong intermediate)

Counter-example: $b \in \mathrm{F}_2$, $b = 1$

Attack 1 ($\log_2 \mathrm{R} = 0$)
$\Pr[b = 0] = 0$
$\Pr[b = 1] = 1$

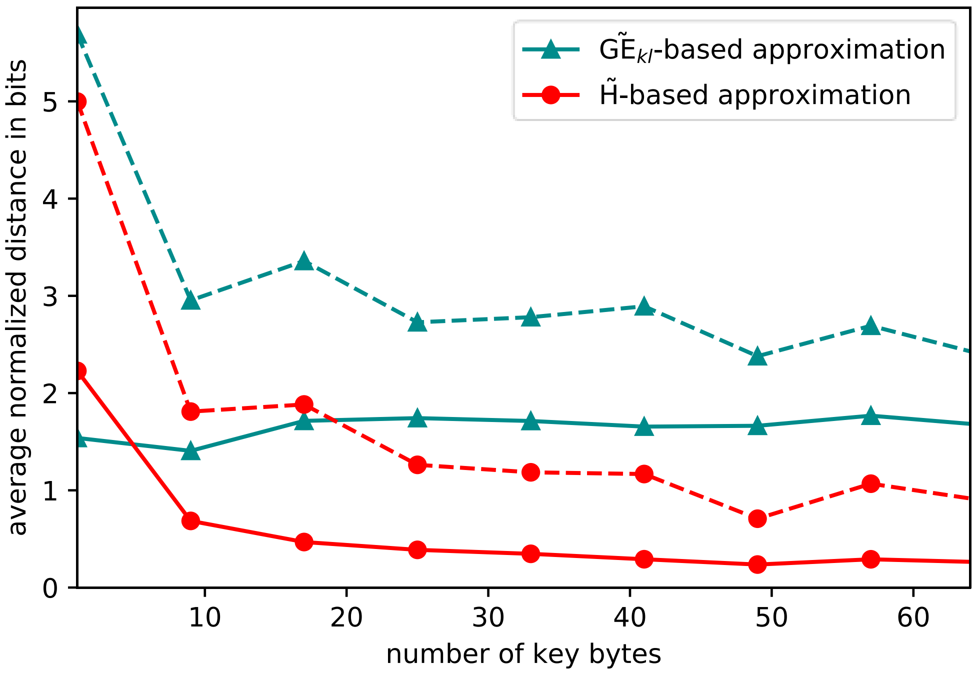
$\mathrm{H}[b] = 0$

Attack 2 ($\log_2 \mathrm{R} = 0$)
$\Pr[b = 0] = 0{,}45$
$\Pr[b = 1] = 0{,}55$

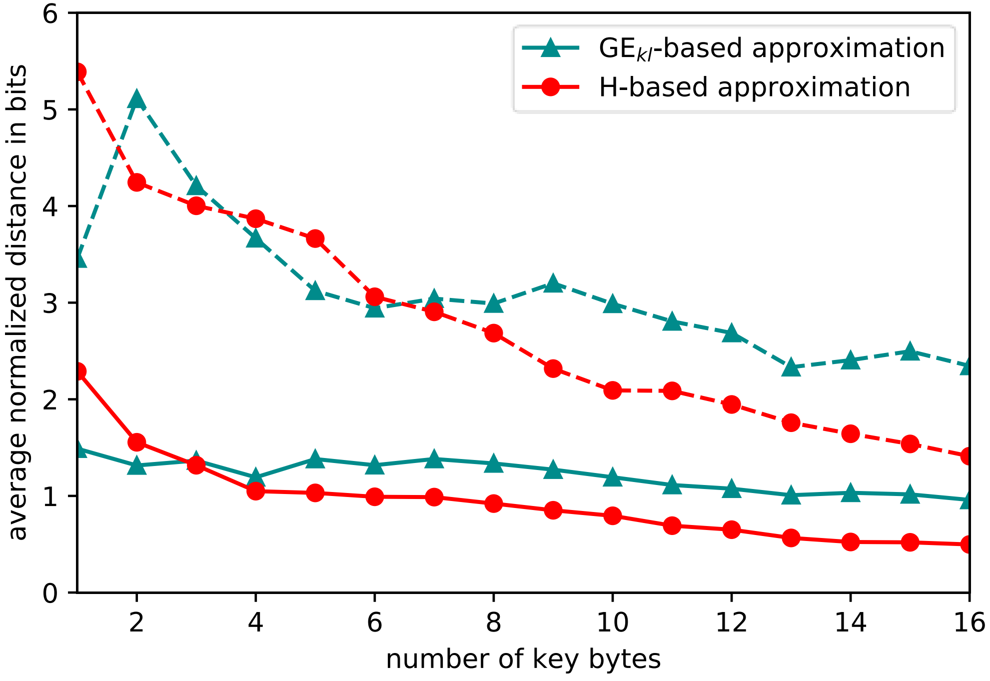
$\mathrm{H}[b] = 0{,}99277$

Experiments: impact of the number of subkeys

$$\frac{\text{average distance to } \log_2(\text{Rank})}{\text{\# of subkeys}}$$



Simulated Leakages



Real Traces

Conclusions

Efficient heuristic method to approximate the rank of the key without its knowledge for the single attack case

Future work

- Propose a more precise technique or metric to approximate the rank in the same single attack scenario
- Key-less rank approximation for score based attacks

Thank You