Unified and Optimized Linear Collision Attacks and Their Application in a Non-Profiled Setting

Benoît Gérard and François-Xavier Standaert

CryptoGroup - Université catholique de Louvain - Belgium

CHES 2012 - September the 10th

Power Analysis Attacks

Divide & Conquer

- Differential Power Analysis
- Correlation Power Analysis
- Template Attack
- ...

Alternatives

- Algebraic side-channel attacks
- Side-channel collision attacks

Motivations

- Getting rid of the leakage model
- Main idea

  same output ⇒ same leakage.

  leakage model choice → similarity metric choice

- [Schramm et al. '03] collision in the f function of DES
- [Bogdanov '07] collision between S-box computations
  - Software: table implementation of an S-boxes
  - Hardware: high area constraints → S-box reuse

This work

Target: linear collision attacks

1. Enhancing collision attacks
   - Handling errors generically
   - Exploiting non-colliding events

2. Collision-attack relevance
   - Comparison with **unprofiled** attacks
   - Software context

Overview

Linear Collision Attacks

Linear Collision and Coding Theory

Experiments

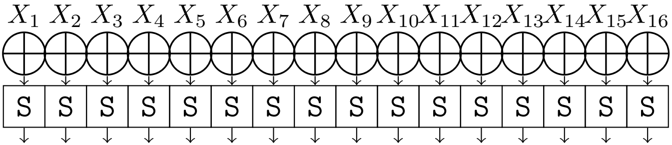
Overview

Linear Collision Attacks

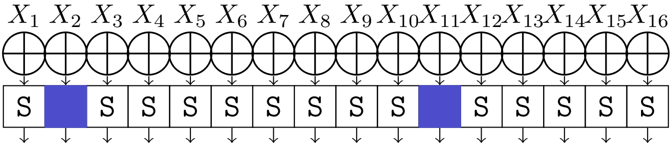
Linear Collision and Coding Theory

Experiments

Linear Collision Attacks Principle

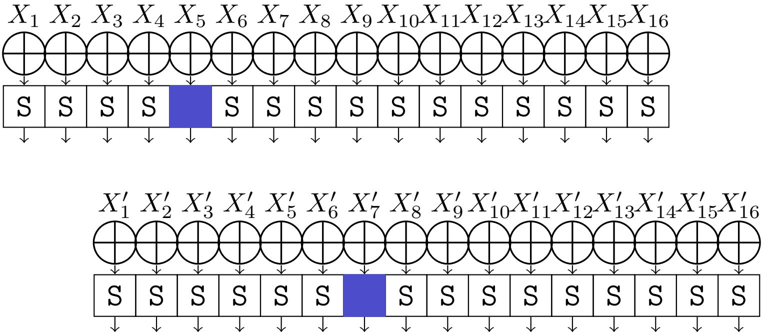


Linear Collision Attacks Principle



$$X_2 \oplus K_2 = X_{11} \oplus K_{11} \quad \Longrightarrow \quad K_2 \oplus K_{11} = X_2 \oplus X_{11}$$

Linear Collision Attacks Principle



$$X_5 \oplus K_5 = X'_7 \oplus K_7 \quad \Longrightarrow \quad K_5 \oplus K_7 = X_5 \oplus X'_7$$

Recovering the Key from Collisions

$$\left\{\begin{array}{lcl} K_1 \oplus K_5 & = & \Delta K_{1,5} \\ K_1 \oplus K_2 & = & \Delta K_{1,2} \\ K_2 \oplus K_8 & = & \Delta K_{2,8} \\ K_3 \oplus K_4 & = & \Delta K_{3,4} \\ K_6 \oplus K_7 & = & \Delta K_{6,7} \end{array}\right.$$

Recovering the Key from Collisions

$$\begin{cases} K_1 \oplus K_5 & = & \Delta K_{1,5} \\ K_1 \oplus K_2 & = & \Delta K_{1,2} \\ K_2 \oplus K_8 & = & \Delta K_{2,8} \\ K_3 \oplus K_4 & = & \Delta K_{3,4} \\ K_6 \oplus K_7 & = & \Delta K_{6,7} \end{cases}$$

2^{24} values for (K_1, K_3, K_6)

$\Downarrow$

2^{24} keys $(K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_8)$

instead of 2^{64}

Limitations

1. Information available only if a collision occurs
   - Non-colliding event also brings information
2. Errors
   - Inconsistency in the system
   - Undetectable erroneous system

Techniques to enhance the attack

- [Bogdanov '08] binary and ternary voting
- [Moradi et al. '10] determining $\Delta K_{a,b}$ using correlation

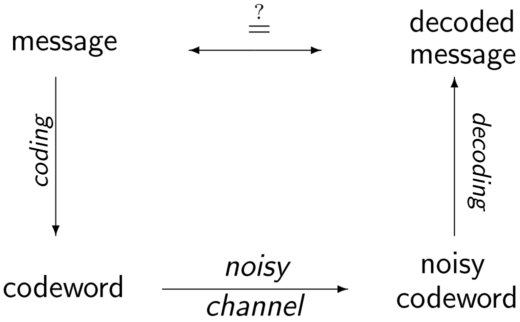
Overview

Linear Collision Attacks

Linear Collision and Coding Theory

Experiments

Decoding Problem



Coding: adding redundancy to the message

Collision Attacks as a Decoding Problem

key
$(K_1, \ldots, K_{16})$

Collision Attacks as a Decoding Problem

key

$(K_1, \ldots, K_{16})$

↓

$(\Delta K_{a,b})_{1 \leq a < b \leq 16}$

Collision Attacks as a Decoding Problem

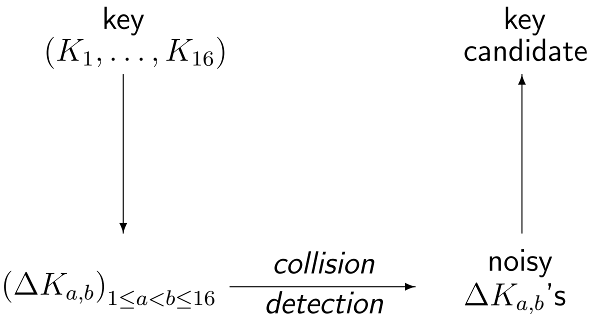
key

$(K_1, \ldots, K_{16})$

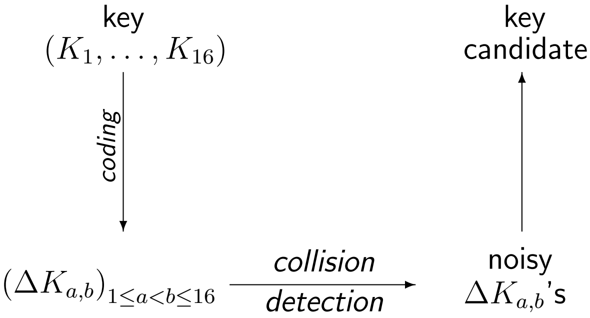
↓

$(\Delta K_{a,b})_{1 \leq a < b \leq 16}$ —*collision detection*→ noisy $\Delta K_{a,b}$'s

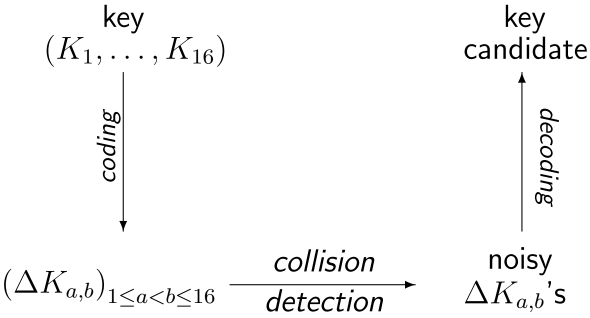
Collision Attacks as a Decoding Problem



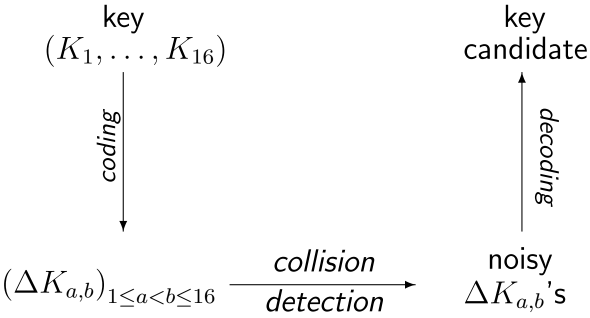
Collision Attacks as a Decoding Problem



Collision Attacks as a Decoding Problem



Collision Attacks as a Decoding Problem

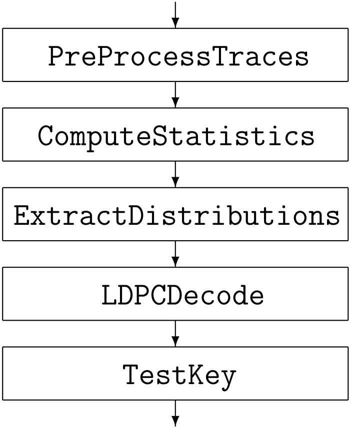


$\Rightarrow$ why not using a decoding algorithm?

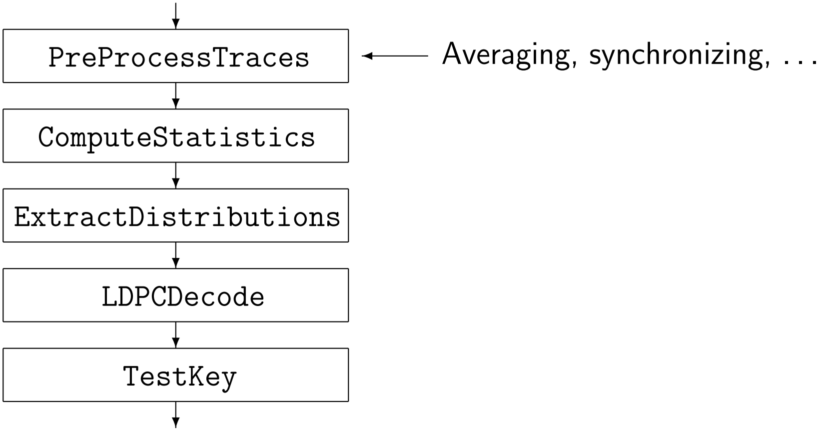
Contributions

1. General framework
2. Enhancement of current tools
   - LDPC **soft** decoding
   - Bayesian extensions
3. Experiments on software implementations

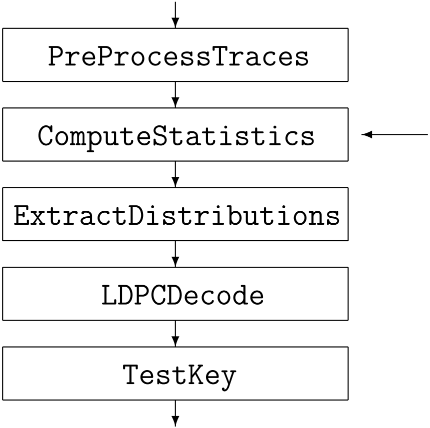
General Framework for Collision Attacks



General Framework for Collision Attacks



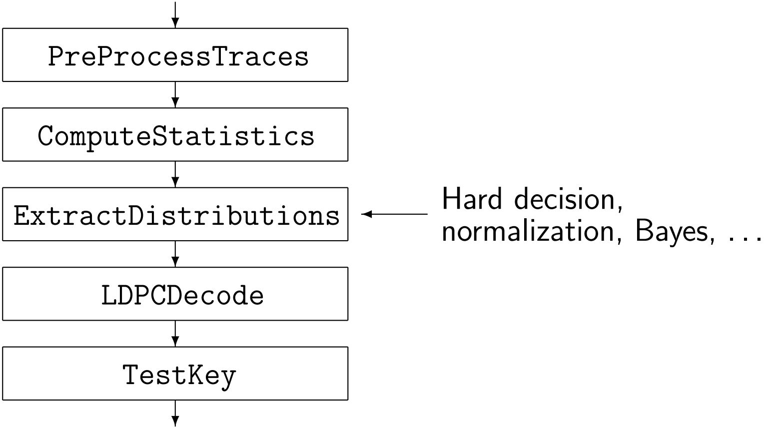
General Framework for Collision Attacks



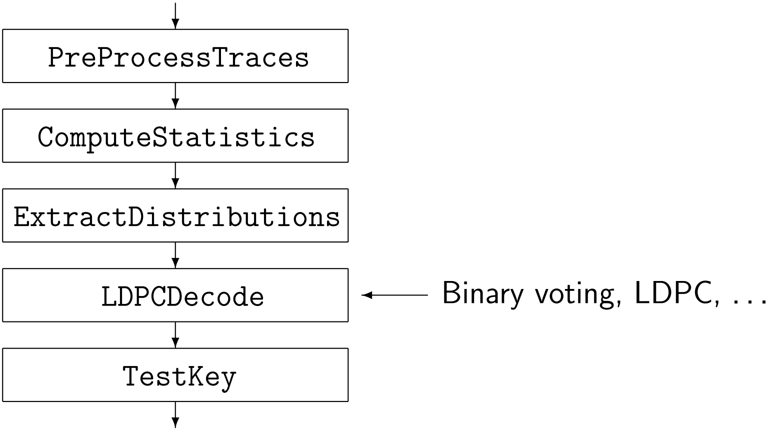
"Collision detection":

Pearson correlation
Euclidean distance

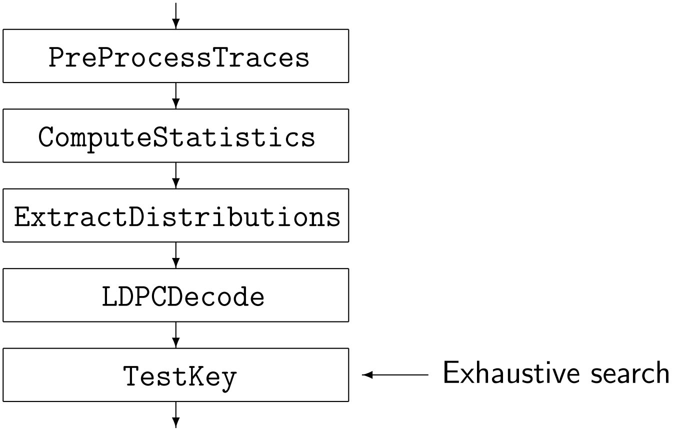
General Framework for Collision Attacks



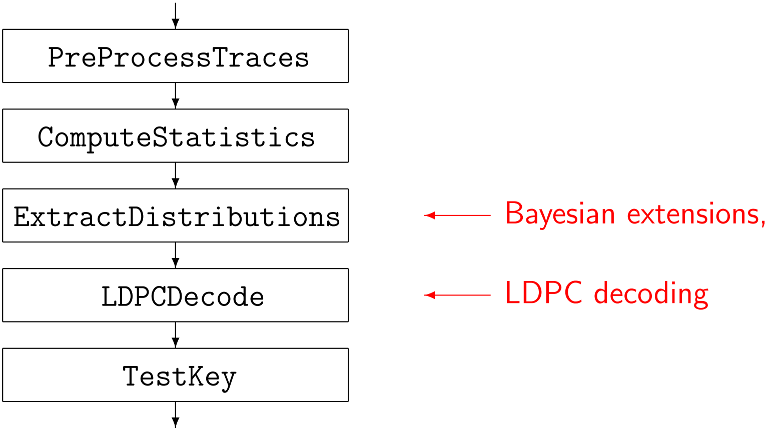
General Framework for Collision Attacks



General Framework for Collision Attacks



General Framework for Collision Attacks



Collision-Attack LDPC Code

$$\Delta K \stackrel{\text{def}}{=} (\Delta K_{1,2}, \ldots, \Delta K_{15,16})$$

- 120 $\Delta K's \rightarrow$ dimension-15 subspace
- Constraints involving 3 positions

$$\underbrace{\Delta K_{a,b}}_{\cancel{K_a} \oplus K_b} \oplus \underbrace{\Delta K_{a,c}}_{\cancel{K_a} \oplus K_c} = \underbrace{\Delta K_{b,c}}_{K_b \oplus K_c}$$

Collision-Attack LDPC Code

$$\Delta K \stackrel{\text{def}}{=} (\Delta K_{1,2}, \ldots, \Delta K_{15,16})$$

- 120 $\Delta K's \rightarrow$ dimension-15 subspace
- Constraints involving 3 positions

$$\underbrace{\Delta K_{a,b}}_{\cancel{K_a} \oplus K_b} \oplus \underbrace{\Delta K_{a,c}}_{\cancel{K_a} \oplus K_c} = \underbrace{\Delta K_{b,c}}_{K_b \oplus K_c}$$

Sparse
constraints

Collision-Attack LDPC Code

$$\Delta K \stackrel{\text{def}}{=} (\Delta K_{1,2}, \ldots, \Delta K_{15,16})$$

- 120 $\Delta K's \rightarrow$ dimension-15 subspace
- Constraints involving 3 positions

$$\underbrace{\Delta K_{a,b}}_{\cancel{K_a} \oplus K_b} \oplus \underbrace{\Delta K_{a,c}}_{\cancel{K_a} \oplus K_c} = \underbrace{\Delta K_{b,c}}_{K_b \oplus K_c}$$

Sparse constraints $\Longrightarrow$ LDPC code

Collision-Attack LDPC Code

$$\Delta K \stackrel{\text{def}}{=} (\Delta K_{1,2}, \ldots, \Delta K_{15,16})$$

- 120 $\Delta K's \rightarrow$ dimension-15 subspace
- Constraints involving 3 positions

$$\underbrace{\Delta K_{a,b}}_{\cancel{K_a} \oplus K_b} \oplus \underbrace{\Delta K_{a,c}}_{\cancel{K_a} \oplus K_c} = \underbrace{\Delta K_{b,c}}_{K_b \oplus K_c}$$

Sparse constraints $\Longrightarrow$ LDPC code $\Longrightarrow$ efficient decoding algorithm

Bayesian extensions

$$\left.\begin{array}{l}\text{scores } (s_1, \ldots, s_n) \\ \Pr[S|\text{coll}] \\ \Pr[S|\text{non-coll}]\end{array}\right\} \xrightarrow{\textit{Bayes}} \Pr[\Delta K_{a,b} = \delta]$$

Unprofiled setting:

- Theoretical models
- On-line parameter estimation

Bayesian extensions

1. Euclidean distance (normalized distance)

$$\mathrm{NED}(T, T') = \sum_j \frac{\left(T_j - T'_j\right)^2}{2\sigma_j^2}$$

2. Correlation-enhanced (Fisher transform)

$$\mathrm{Fisher}(c) = \mathrm{arctanh}(c)$$

Overview

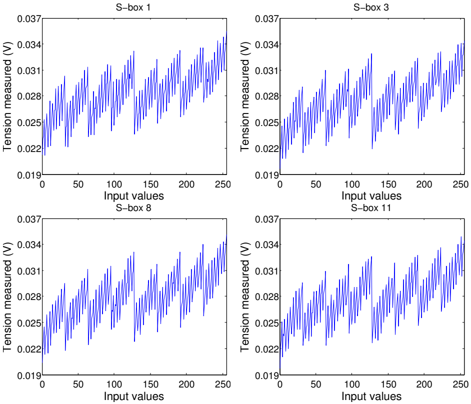
Linear Collision Attacks

Linear Collision and Coding Theory

Experiments

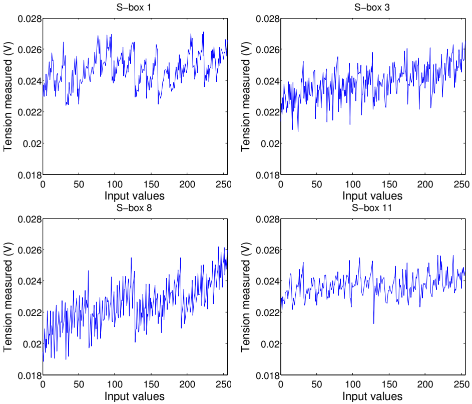
Reference implementation

Reference
mov SR, STxy
mov ZL, SR
lpm SR, Z
mov STxy,SR

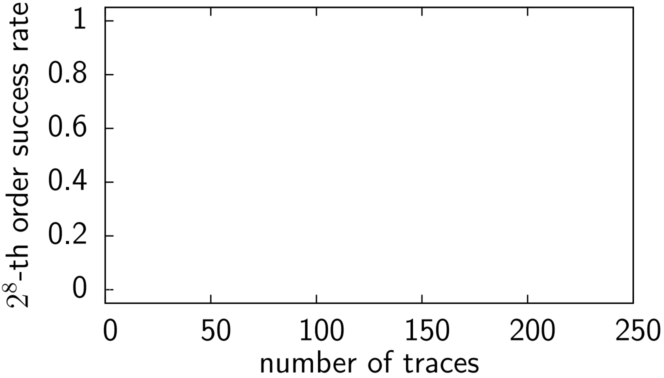


Furious implementation

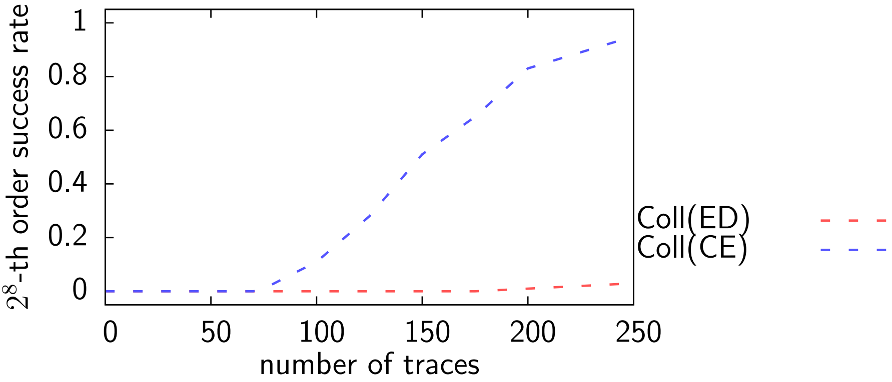
Furious
mov H1, ST21
mov ZL, ST22
lpm ST21, Z
mov ZL, ST23
lpm ST22, Z



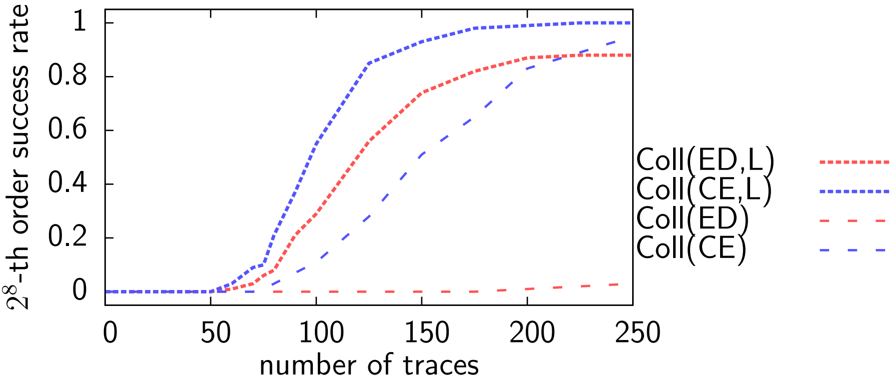
Attacking the Reference Implementation



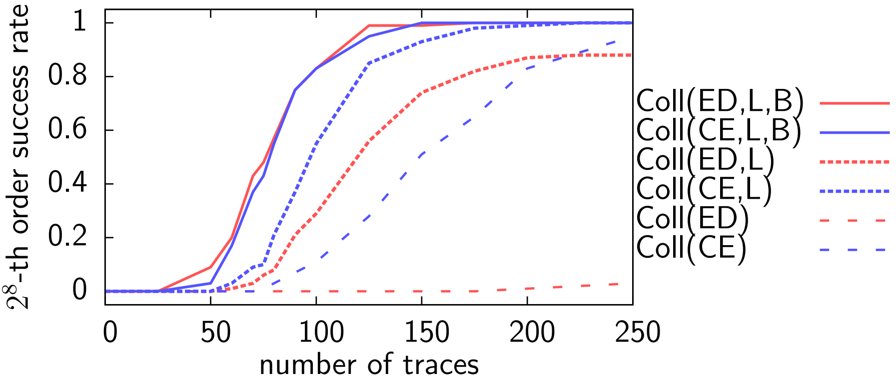
Attacking the Reference Implementation



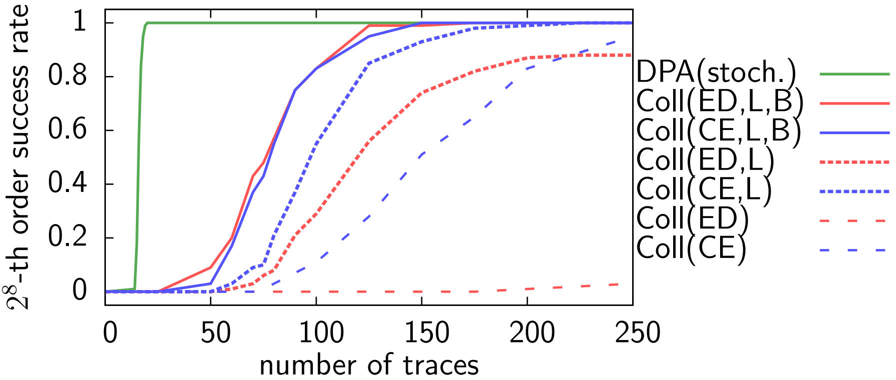
Attacking the Reference Implementation



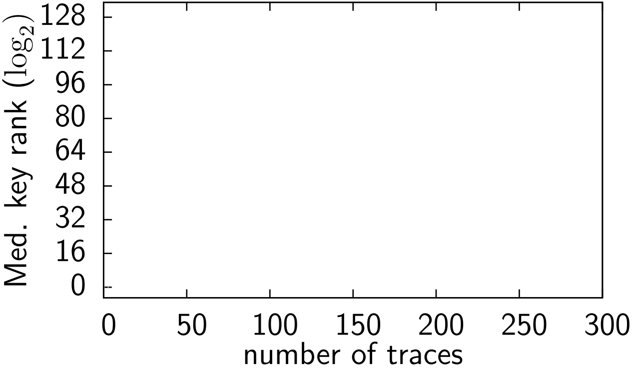
Attacking the Reference Implementation



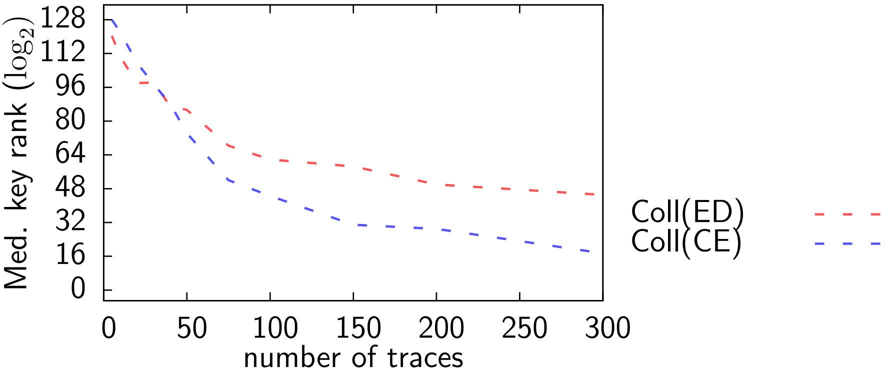
Attacking the Reference Implementation



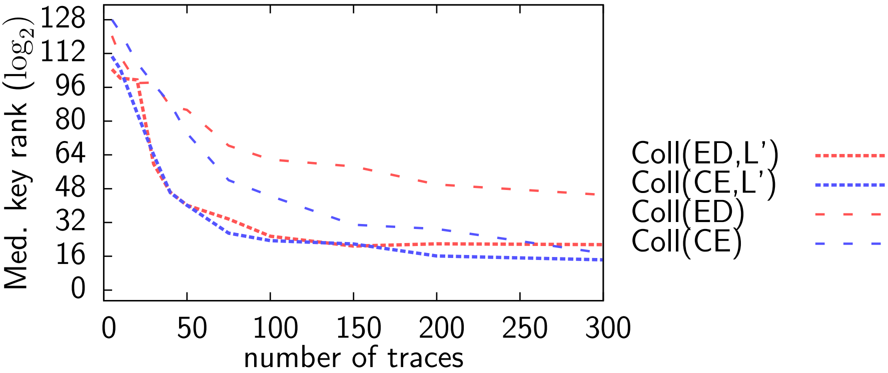
Attacking the Furious Implementation



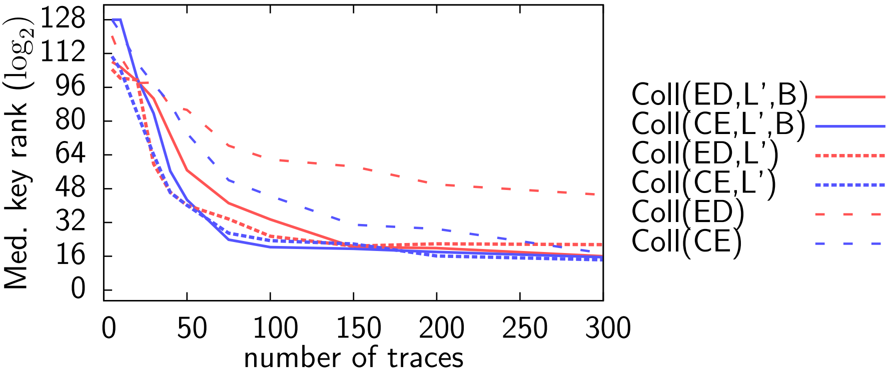
Attacking the Furious Implementation



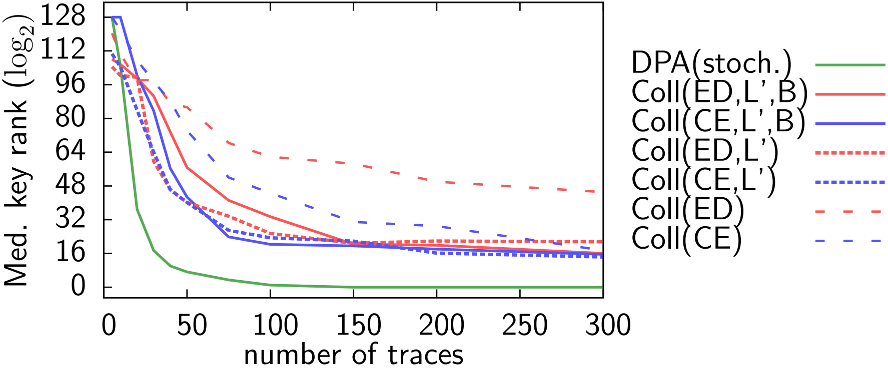
Attacking the Furious Implementation



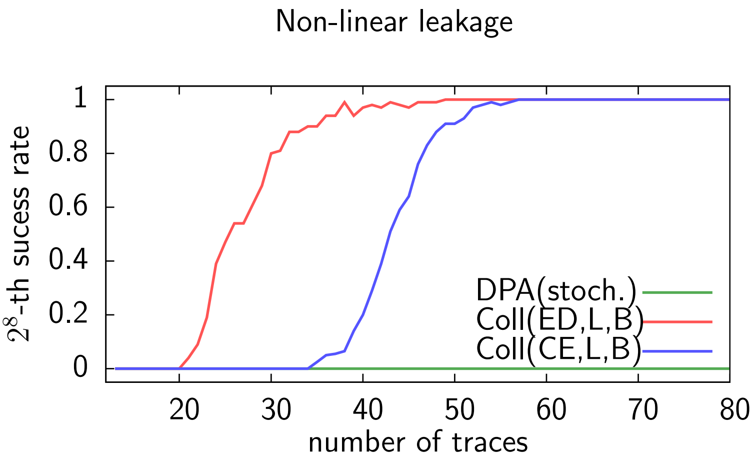
Attacking the Furious Implementation



Attacking the Furious Implementation



Theoretical context



Conclusions

Collision attacks as a decoding problem

- General framework
- Improvements of former attacks
  - Soft decoding algorithm
  - Bayesian extensions

Experiments performed

- Usually less efficient than stochastic DPA
- May be useful in challenging implementation contexts

Perspectives

- List decoding
- Application to masked implementations
- Application to non-linear collisions