Optimal Collision Side-Channel Attacks

Cezary Glowacz

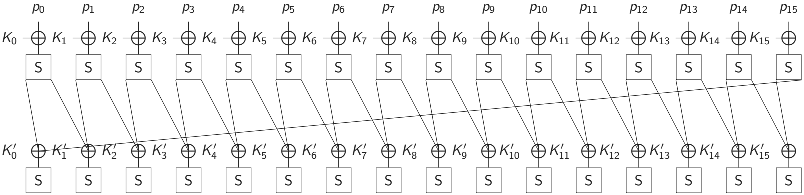
Telekom Security
Bonn

Vincent Grosso

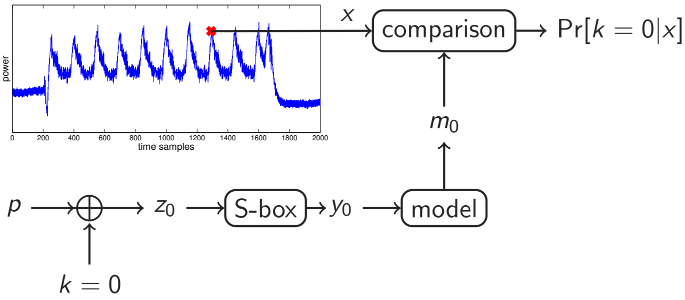
CNRS/laboratoire Hubert Curien
Saint-Étienne

Side-channel attacks

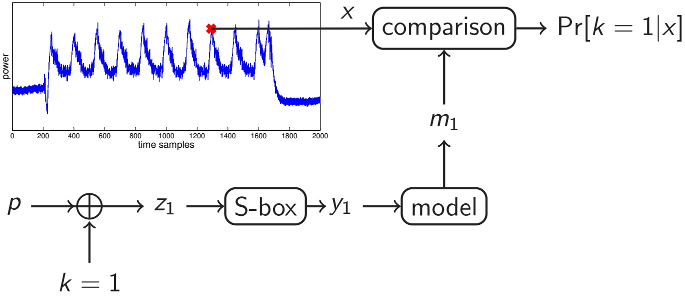
Block ciphers



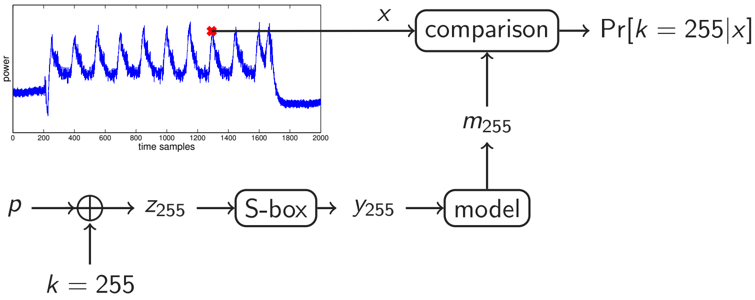
Side-channel attacks: divide-and-conquer strategy



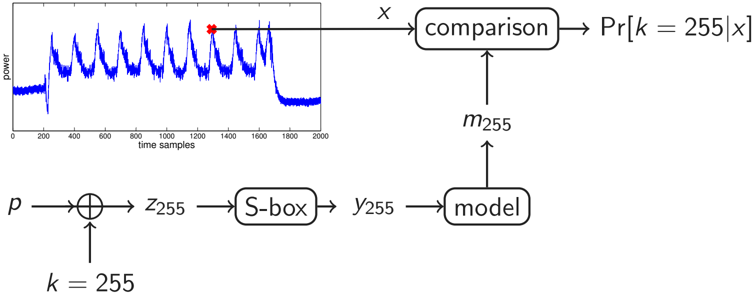
Side-channel attacks: divide-and-conquer strategy



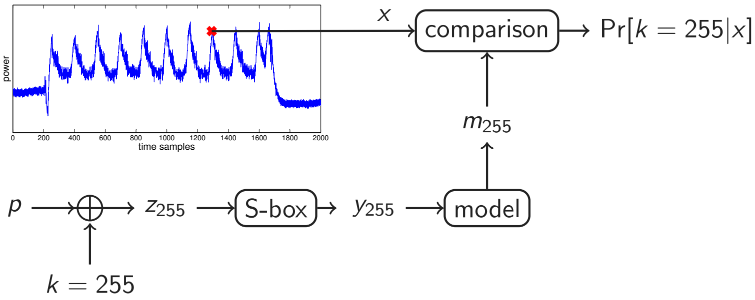
Side-channel attacks: divide-and-conquer strategy



Side-channel attacks: divide-and-conquer strategy



$$16 \times 2^8 < 2^{128}$$



$$16 \times 2^8 < 2^{128}$$

Today: comparison and combination

Model

Profiled:

- Require a similar device to build the model
- Maximum likelihood ⇒ optimal approach if the model is good

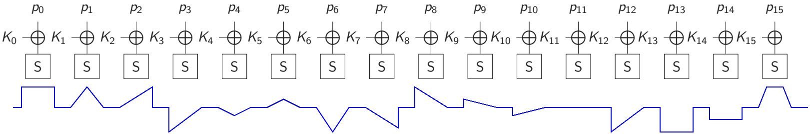
Non profiled:

- No need of a device
- Optimality dependent on the model use

Collision attacks

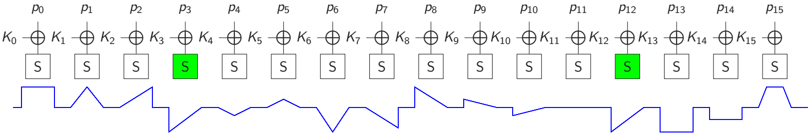
Collision attacks

Principle of collision attacks



Principle of collision attacks

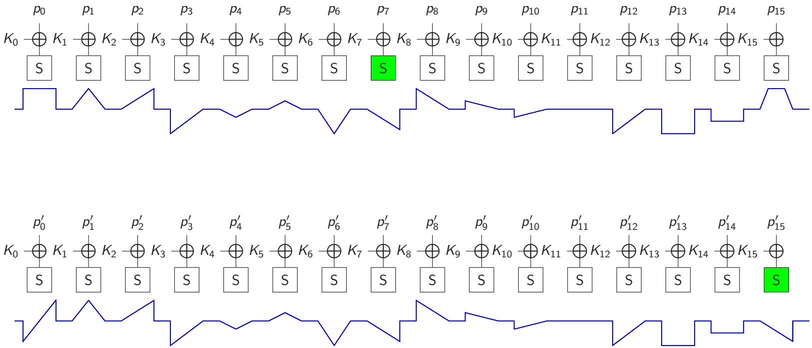
If the same data is proceed the leakages should be similar



- $K_3 \oplus p_3 = K_{12} \oplus p_{12} \Rightarrow K_3 \oplus K_{12} = p_3 \oplus p_{12}$

Principle of collision attacks

If the same data is proceed the leakages should be similar



- $K_3 \oplus p_3 = K_{12} \oplus p_{12} \Rightarrow K_3 \oplus K_{12} = p_3 \oplus p_{12}$
- $K_7 \oplus p_7 = K_{15} \oplus p'_{15} \Rightarrow K_7 \oplus K_{15} = p_7 \oplus p'_{15}$

Recovering the key from collisions

Extract 15 (independent) relations

$$K_i \oplus K_j = \Delta K_{i,j}$$

Then enumerate all 2^8 candidates for K_0 and recover the valid key

$$K_0, \ldots, K_{15}$$

$$\begin{pmatrix} \Delta_{2,9} \\ \Delta_{2,4} \\ \Delta_{7,8} \\ \Delta_{1,4} \\ \Delta_{1,5} \\ \Delta_{2,6} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} K_1 \\ K_2 \\ K_3 \\ K_4 \\ K_5 \\ K_6 \\ K_7 \\ K_8 \\ K_9 \end{pmatrix}$$

Limitations

- Noisy leakages $\rightsquigarrow$ false relation $\rightsquigarrow$ inconsistent system
  - Perform averaging
  - use all leakages

$$\rho_{k^{(l_1)},k^{(l_2)}}\left(x^{(l_1)},x^{(l_2)}\right)$$

$$\mathcal{D}_{sto.coll} = \underset{\tilde{\mathrm{k}}\in(\mathbb{F}_2^n)^L}{\operatorname{argmax}} \sum_{u\in\mathbb{F}_2^n} \frac{\left(\sum_{l=1}^{L}\sum_{q=1\ldots Q|t_q\oplus k^{(l)}=u} x_q^{(l)}\right)^2}{\sum_{l=1}^{L}\sum_{q=1\ldots Q|t_q\oplus k^{(l)}=u} 1}$$

- Noisy leakages: How to detect collision
  - Optimal formula when the distribution of the leakage function values is known

Optimal distinguisher

Maximum likelihood derivation

$$\mathcal{D}_{opt} = \underset{\tilde{\mathrm{k}} \in (\mathbb{F}_2^n)^L}{\operatorname{argmax}} \prod_{q=0}^{2^n-1} \prod_{l=1}^{L} f_{\sigma^2}\left(x_q^{(l)} - \varphi\left(t_q^{(l)} \oplus k^{(l)}\right)\right),$$

Maximum likelihood derivation

$$\mathcal{D}_{opt} = \underset{\tilde{\mathrm{k}} \in (\mathbb{F}_2^n)^L}{\operatorname{argmax}} \prod_{q=0}^{2^n-1} \prod_{l=1}^{L} f_{\sigma^2}\left(x_q^{(l)} - \varphi\left(t_q^{(l)} \oplus k^{(l)}\right)\right),$$

derivation under known distribution p of leakage function values φ is given by:

$$D_{opt.fun.p} = \underset{\tilde{\mathrm{k}} \in (\mathbb{F}_2^n)^L}{\operatorname{argmax}} \prod_{q=0}^{2^n-1} \int \left(\prod_{l=1}^{L} f_{\sigma^2}\left(x_{q \oplus k^{(l)}}^{(l)} - \varphi\right)\right) dp(\varphi),$$

Maximum likelihood derivation

$$\mathcal{D}_{opt} = \underset{\tilde{k}\in(\mathbb{F}_2^n)^L}{\operatorname{argmax}} \prod_{q=0}^{2^n-1} \prod_{l=1}^{L} f_{\sigma^2}\left(x_q^{(l)} - \varphi\left(t_q^{(l)} \oplus k^{(l)}\right)\right),$$

derivation under known distribution p of leakage function values φ is given by:

$$D_{opt.fun.p} = \underset{\tilde{k}\in(\mathbb{F}_2^n)^L}{\operatorname{argmax}} \prod_{q=0}^{2^n-1} \int \left(\prod_{l=1}^{L} f_{\sigma^2}\left(x_{q\oplus k^{(l)}}^{(l)} - \varphi\right)\right) dp(\varphi),$$

dp taken as a density of Gaussian distributionand balanced set-up of traces:

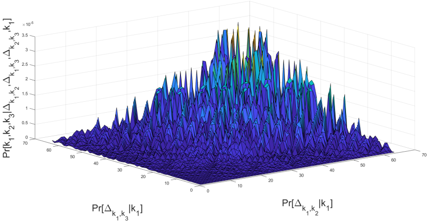
$$D_{opt.fun.gauss} = \underset{\tilde{k}\in(\mathbb{F}_2^n)^L}{\operatorname{argmax}} \sum_{q=0}^{2^n-1} \sum_{l_1=1}^{L} \sum_{l_2=l_1+1}^{L} \left(x_{q\oplus k^{(l_1)}}^{(l_1)} \times x_{q\oplus k^{(l_2)}}^{(l_2)}\right).$$

Limitations

$D_{opt.fun.gauss}$ and $\mathcal{D}_{sto.coll}$ require to perform a search over $\tilde{\mathrm{k}} \in (\mathbb{F}_2^n)^L$

For correlation enhanced attacks similar problem appear as local maximum will not give global maximum

$$\Delta K_{i,j} = \Delta K_{i,k} \oplus \Delta K_{k,j}$$



Can we use divide and conquer strategy for optimal collision attack?

Optimal combining strategy

Compute the maximum using divide and conquer

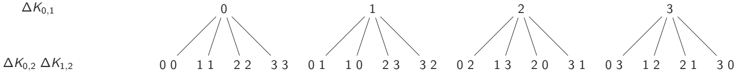
- Look at all valid differential tuples

$\Delta K_{0,1}$	0	1	2	3

Optimal combining strategy

Compute the maximum using divide and conquer

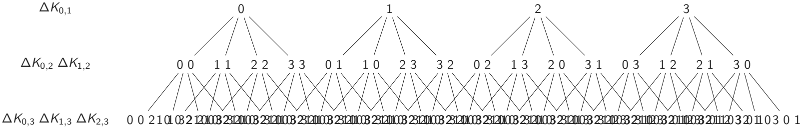
- Look at all valid differential tuples



Optimal combining strategy

Compute the maximum using divide and conquer

- Look at all valid differential tuples

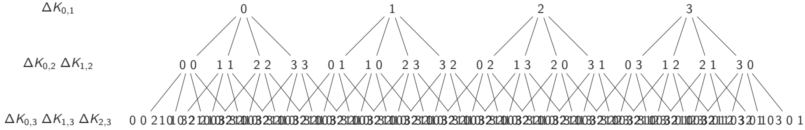


- No path can be cut off

Optimal combining strategy

Compute the maximum using divide and conquer

- Look at all valid differential tuples



- No path can be cut off
- Computational cost $\simeq 2^{120}$
- Testing solutions 2^8

Maximum algorithms

Maximum algorithms

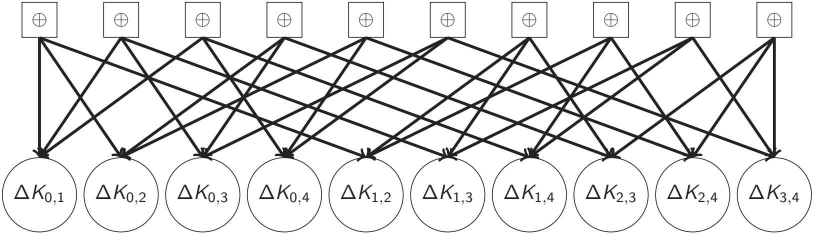
Previous solutions

Belief propagation (Gérard and Standaert)

Represent the equations as a graph

$$\Delta K_{i,j} = \Delta K_{i,k} \oplus \Delta K_{k,j}$$

$$K_i \oplus K_j = K_i \oplus K_k \oplus K_k \oplus K_j$$

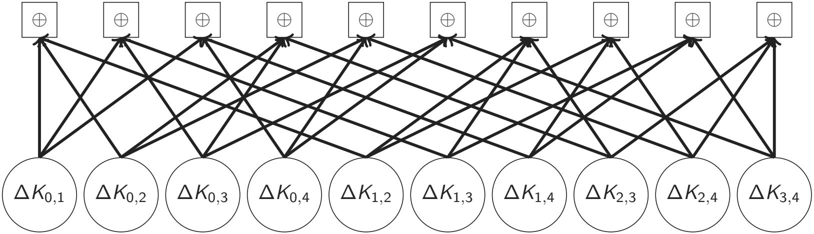


- Propagation of information by iterating messages exchange
- From function nodes to variable nodes
- From variable node to function nodes

Belief propagation (Gérard and Standaert)

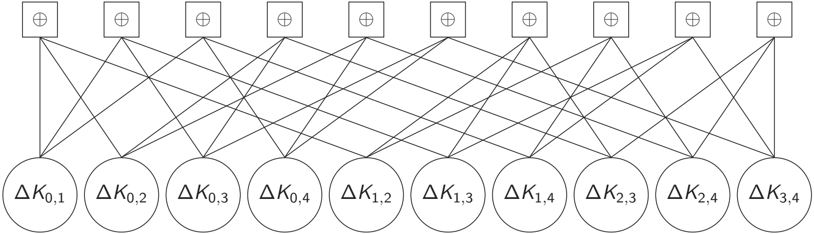
Represent the equations as a graph

$$\Delta K_{i,j} = \Delta K_{i,k} \oplus \Delta K_{k,j}$$
$$K_i \oplus K_j = K_i \oplus K_k \oplus K_k \oplus K_j$$



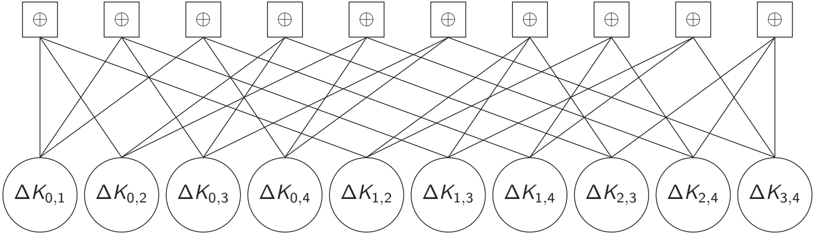
- Propagation of information by iterating messages exchange
- From function nodes to variable nodes
- From variable node to function nodes

Belief propagation (Gérard and Standaert)



- number of variables: $\dfrac{n \times (n-1)}{2} = 120$
- number of functions: $\dfrac{n \times (n-1) \times (n-2)}{6} = 560$
- number of edges per variable $n - 2 = 14$

Belief propagation (Gérard and Standaert)



- number of variables: $\frac{n \times (n-1)}{2} = 120$
- number of functions: $\frac{n \times (n-1) \times (n-2)}{6} = 560$
- number of edges per variable $n - 2 = 14$

- Computational cost $1680 \times 256 \times 8 \times$ number of loops (update of XOR node in $nlog(n)$)
- Testing solutions 2^8

Branch-and-bound (Wiemers and Klein)

- For each i, j compute:

$$Score(\Delta K_{l_1,l_2} = \delta) = \rho(\{x_0^{(l_1)}, \ldots, x_{255}^{(l_1)}\}, \{x_{0\oplus\delta}^{(l_2)}, \ldots, x_{255\oplus\delta}^{(l_2)}\})$$

$\Delta K_{0,1}$	0	1	2	3

Branch-and-bound (Wiemers and Klein)

- For each i, j compute:

$$Score(\Delta K_{l_1,l_2} = \delta) = \rho(\{x_0^{(l_1)}, \ldots, x_{255}^{(l_1)}\}, \{x_{0\oplus\delta}^{(l_2)}, \ldots, x_{255\oplus\delta}^{(l_2)}\})$$

- Keep highest differential scores

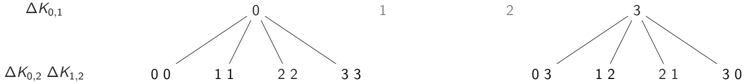
$\Delta K_{0,1}$	0	1	2	3

Branch-and-bound (Wiemers and Klein)

- For each i, j compute:

$$Score(\Delta K_{l_1,l_2} = \delta) = \rho(\{x_0^{(l_1)}, \ldots, x_{255}^{(l_1)}\}, \{x_{0\oplus\delta}^{(l_2)}, \ldots, x_{255\oplus\delta}^{(l_2)}\})$$

- Keep highest differential scores
- Compute sum of the scores if we add the next key

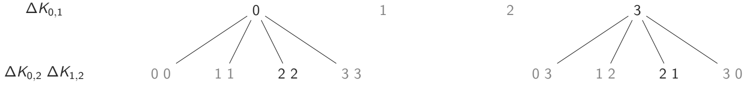


Branch-and-bound (Wiemers and Klein)

- For each i, j compute:

$$Score(\Delta K_{l_1,l_2} = \delta) = \rho(\{x_0^{(l_1)}, \ldots, x_{255}^{(l_1)}\}, \{x_{0\oplus\delta}^{(l_2)}, \ldots, x_{255\oplus\delta}^{(l_2)}\})$$

- Keep highest differential scores
- Compute sum of the scores if we add the next key
- Keep highest differential scores

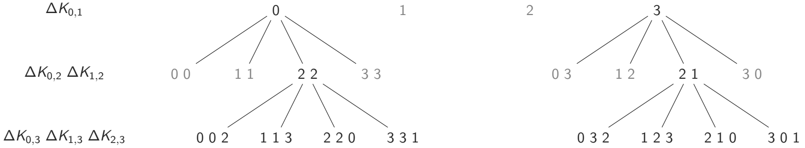


Branch-and-bound (Wiemers and Klein)

- For each i, j compute:

$$Score(\Delta K_{l_1,l_2} = \delta) = \rho(\{x_0^{(l_1)}, \dots, x_{255}^{(l_1)}\}, \{x_{0\oplus\delta}^{(l_2)}, \dots, x_{255\oplus\delta}^{(l_2)}\})$$

- Keep highest differential scores
- Compute sum of the scores if we add the next key
- Keep highest differential scores



- Computational cost $15 \times 256 \times$ number of elements kept
- Testing solutions $2^8 \times$ number of elements kept

Limitations of existing solutions

- There is not clear rule about combining scores
  - Sum of scores
  - Product of Bayesian extension (only valid asymptotically)
- No link to optimal strategy
- Attacks only, no evaluation

Proposal

Greedy approach with random start

$\Delta K_{0,1}$	0	1	2	3

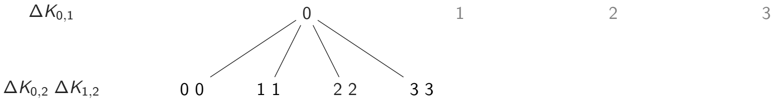
Greedy approach with random start

- Keep the highest differential score

$\Delta K_{0,1}$	0	1	2	3

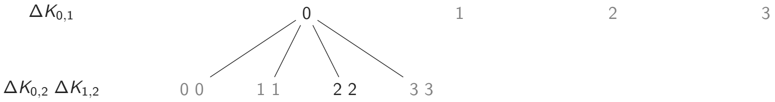
Greedy approach with random start

- Keep the highest differential score
- Compute sum of the scores if we add the next key



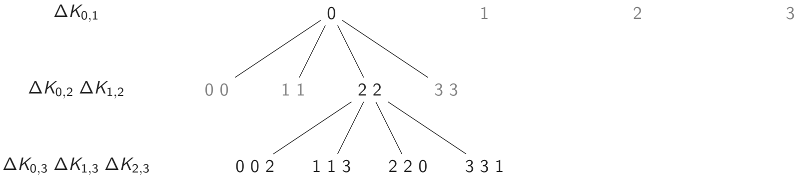
Greedy approach with random start

- Keep the highest differential score
- Compute sum of the scores if we add the next key
- Keep the highest differential score



Greedy approach with random start

- Keep the highest differential score
- Compute sum of the scores if we add the next key
- Keep the highest differential score



Greedy approach with random start

- Keep the highest differential score
- Compute sum of the scores if we add the next key
- Keep the highest differential score
- Start from another differential

$\Delta K_{4,7}$	0	1	2	3

Greedy approach with random start

- Keep the highest differential score
- Compute sum of the scores if we add the next key
- Keep the highest differential score
- Start from another differential

$\Delta K_{4,7}$	0	1	2	3

- Computational cost $15 \times 256\times$ number of random starts
- Testing solutions 2^8

Upper bound on optimal strategy

- Greedy algorithm output a system with score S_G
- Evaluation: score of the key S_K is known
  - $S_K < S_G$: optimal first order success rate will fail
  - $S_K \geq S_G$: optimal first order success rate may succeed

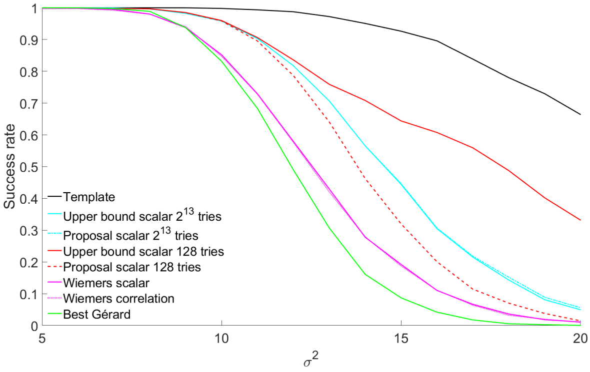
Upper bound on optimal strategy

- Greedy algorithm output a system with score S_G
- Evaluation: score of the key S_K is known
  - $S_K < S_G$: optimal first order success rate will fail
  - $S_K \geq S_G$: optimal first order success rate may succeed

$$SR_G \leq SR_O \leq UB_G$$

If $|UB_G - SR_G| < \epsilon \Rightarrow$ greedy approach $\simeq$ optimal approach

Results



- Close to optimal
- Better than previous solution
- Worse than template

Limitations of greedy approach

- Greedy approach will failed if all the max $\Delta K_{i,j}$ are incorrect
- Limited to "first-order" success rate
- Requires similar leakage for every S-box

Conclusion

- Optimal collision attack derived from maximum likelihood when distribution of leakage function values is known
- Optimal collision attacks sum of scalar product when leakage function values are drawn from a Gaussian distribution
- Greedy approach to find maximum is close to optimal
- First bound on optimal strategy
- Close result for 80% success rate with limited computational cost

Thanks!

Questions?

Comments?