Learning With Physical Rounding for Linear and Quadratic Leakage Functions

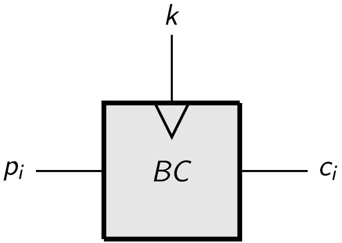
Clément Hoffmann[1], Pierrick Méaux[2], Charles Momin[1], Yann Rotella[3], François-Xavier Standaert[1], Balazs Udvarhelyi[1]

[1] Crypto Group, ICTEAM Institute, UCLouvain, Louvain-la-Neuve, Belgium
[2] Luxembourg University, SnT, Luxembourg
[3] Université Paris-Saclay, UVSQ, CNRS, Laboratoire de mathématiques de Versailles, 78000,Versailles, France

Outline

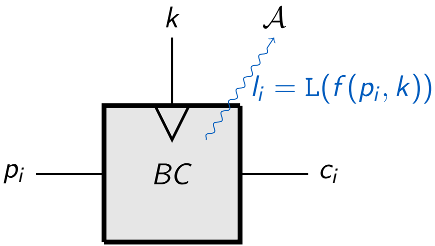
Outline

Block cipher SCA security (DPA setting)



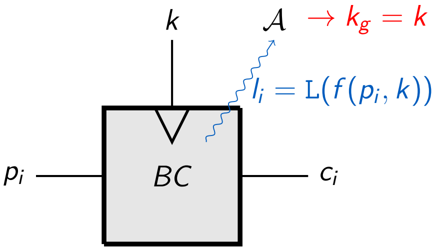
- Same long-term k for each p_i

Block cipher SCA security (DPA setting)



- Same long-term k for each p_i
- $\forall p_i \neq p_j$, l_i provides new information on k

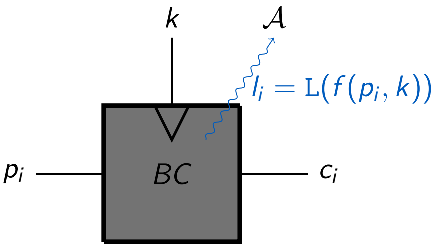
Block cipher SCA security (DPA setting)



- Same long-term k for each p_i
- $\forall p_i \neq p_j$, l_i provides new information on k

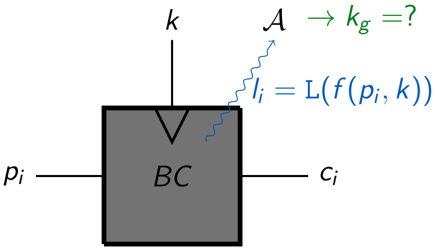
$\rightarrow$ $\mathcal{A}$ uses **several** l_is to guess k (**DPA**)

Block cipher SCA security (DPA setting)



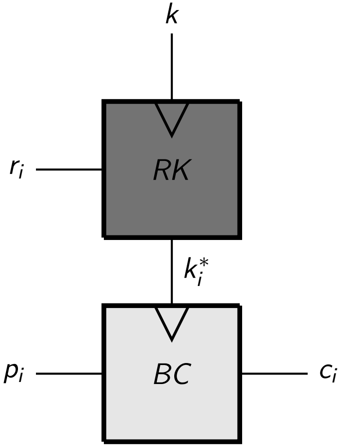
- Same long-term k for each p_i
- $\forall p_i \neq p_j$, l_i provides new information on k
  - $\rightarrow$ $\mathcal{A}$ uses **several** l_is to guess k (**DPA**)
- Popular contermeasure: masking

Block cipher SCA security (DPA setting)



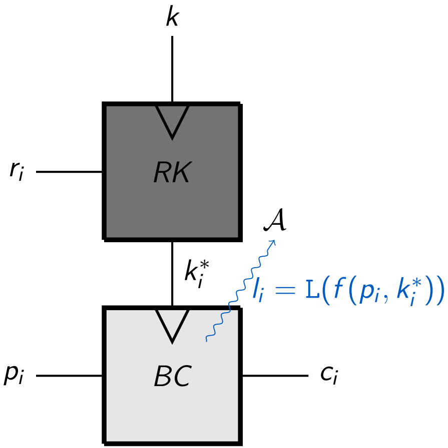
- Same long-term k for each p_i
- $\forall p_i \neq p_j$, l_i provides new information on k
  - $\rightarrow$ $\mathcal{A}$ uses **several** l_is to guess k (**DPA**)
- Popular contermeasure: masking
  - $\rightarrow$ hard to guess k from l_is

Block cipher SCA security (SPA setting)



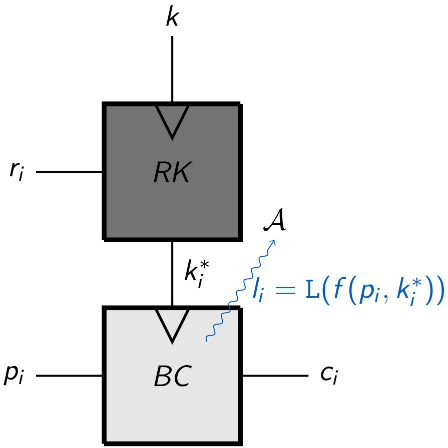
- Two primitives:
  - $\rightarrow$ RK easy to mask (DPA)
  - $\rightarrow$ BC unprotected
- Fresh k_i^* for each r_i

Block cipher SCA security (SPA setting)



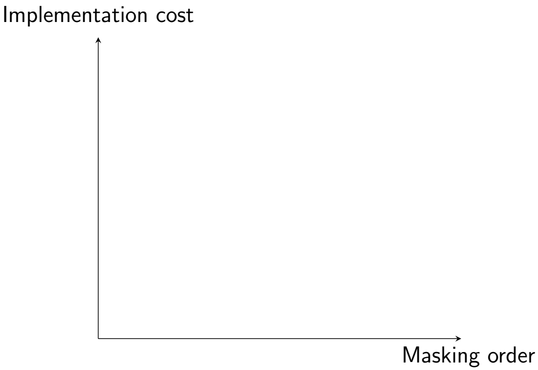
- Two primitives:
  - $\rightarrow$ *RK* easy to mask (DPA)
  - $\rightarrow$ *BC* unprotected
- Fresh k_i^* for each r_i
  - $\rightarrow$ $\mathcal{A}$ uses a **few** l_i to get k_i^* (**SPA**)

Block cipher SCA security (SPA setting)

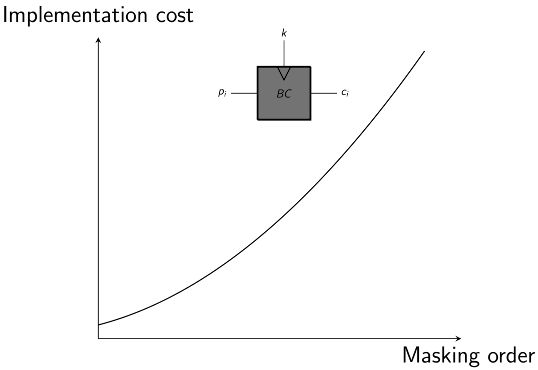


- Two primitives:
  - $\rightarrow$ RK easy to mask (DPA)
  - $\rightarrow$ BC unprotected
- Fresh k_i^* for each r_i
  - $\rightarrow$ $\mathcal{A}$ uses a **few** l_i to get k_i^* (**SPA**)
  - $\rightarrow$ In practice: hard for HW

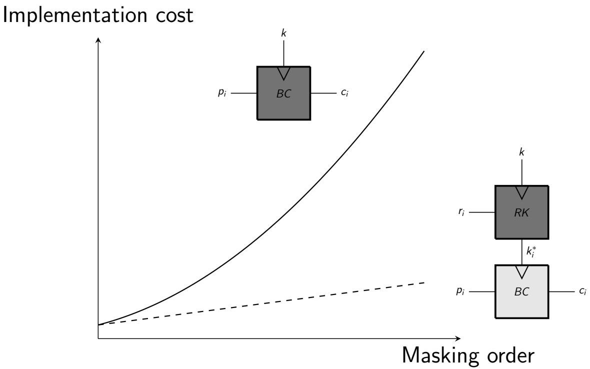
Interest of re-keying



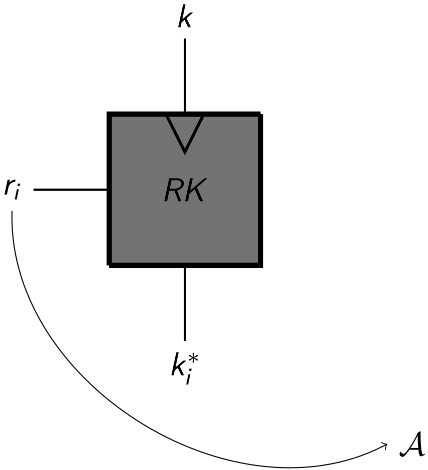
Interest of re-keying



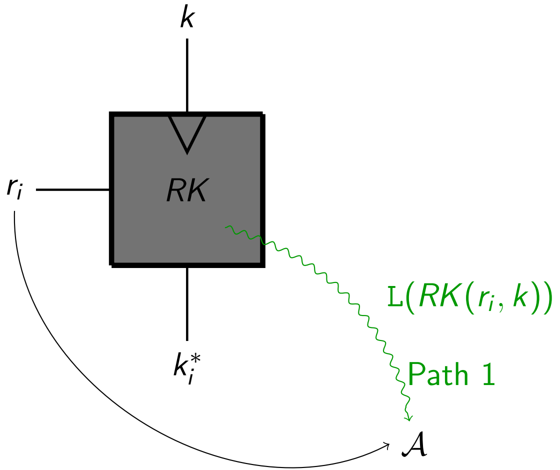
Interest of re-keying



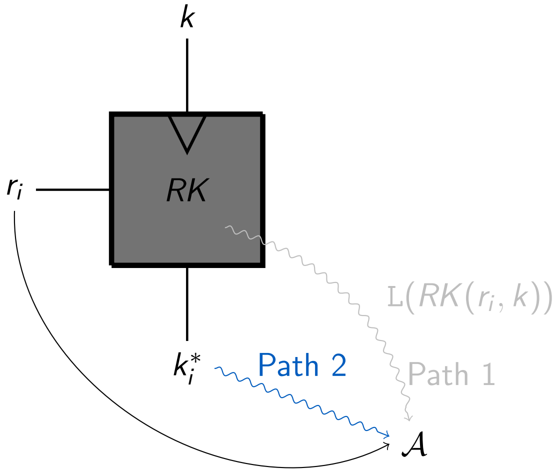
Attack path considering fresh re-keying



Attack path considering fresh re-keying

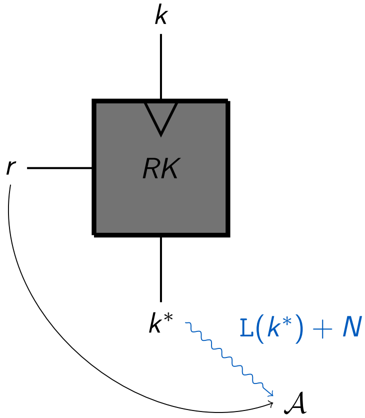


Attack path considering fresh re-keying



Adversarial model 1

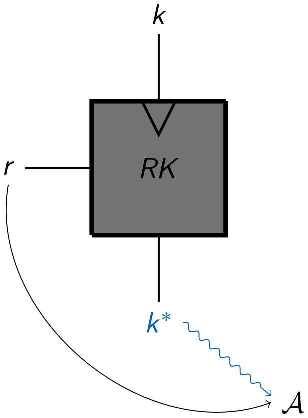
Medwed, Standaert, Großschädl and Regazzoni (2010)



- Noisy leakage
- Finite field multiplication: $k^* = r \cdot k$ over $\mathbb{F}_{2^\kappa}$ (key homomorphic)
- Efficient implementation
- Significant noise level required

Adversarial model 2

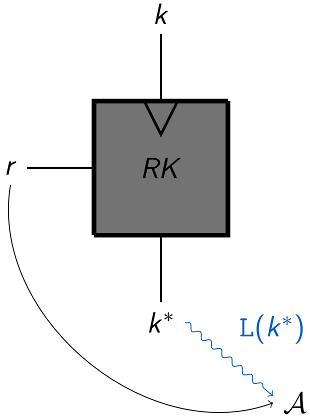
Dziembowski, Faust, Herold, Journault, Masny and Standaert (2016)



- Unbounded leakage
- wPRF with rounded inner product: $k^* = \lfloor \langle \mathbf{k}, \mathbf{r} \rangle \rfloor_p$, $\mathbf{k}, \mathbf{r} \in \mathbb{Z}_{2q}^n$ (nearly key homomorphic)
- Large key requirement (cost and perfs)

Adversarial model 3

Duval, Méaux, Momin and Standaert (2021)



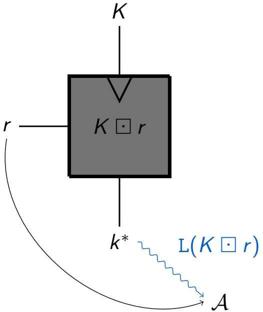
- Noise free (compressive) leakage
- Finite field matrices product: $k^* = \mathbf{K} \cdot (\mathbf{r}, 1)$, $\mathbf{r} \in \mathbb{F}_p^n$, $\mathbf{K} \in \mathbb{F}_p^{m \times (n+1)}$ (key homomorphic)
- Similar to *Crypto Dark Matter* wPRF (Boneh, Ishai, Passelègue, Sahai and Wu, 2018):

$$\mathtt{F}_{\mathbf{K}}(\mathbf{r}) = \mathsf{map}(\mathbf{K} \cdot \mathbf{r})$$

with (non-linear) map $= \mathtt{L}$

$\rightarrow$ map done by the physics (no cost)!

Learning With Physical Rounding (LWPR)

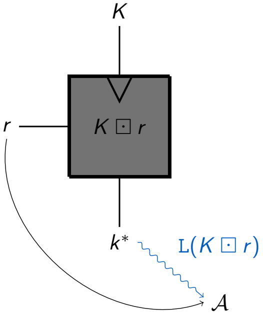


- Hard physical learning problem
  $\rightarrow$ Similarity with LWE and LWR.
- $\mathcal{A}$ try to recover $\mathbf{K}$ from samples

  $$(\mathbf{r}, \mathtt{L}(k^*)) = (\mathbf{r}, \mathtt{L}(\mathbf{K}\cdot(\mathbf{r}, 1))) = (\mathbf{r}, \mathtt{L}(\mathbf{K}\boxdot\mathbf{r}))$$

  with $\mathbf{r} \in \mathbb{F}_p^n, \mathbf{K} \in \mathbb{F}_p^{m\times(n+1)}$
- Requires an assumption on $\mathtt{L}$

Learning With Physical Rounding (LWPR)



- Hard physical learning problem
  $\rightarrow$ Similarity with LWE and LWR.
- $\mathcal{A}$ try to recover **K** from samples

$$(\mathbf{r}, \mathtt{L}(k^*)) = (\mathbf{r}, \mathtt{L}(\mathbf{K}\cdot(\mathbf{r}, 1))) = (\mathbf{r}, \mathtt{L}(\mathbf{K}\boxdot\mathbf{r}))$$

with $\mathbf{r} \in \mathbb{F}_p^n, \mathbf{K} \in \mathbb{F}_p^{m\times(n+1)}$

- Requires an assumption on $\mathtt{L}$

- CHES21 ([DMMS21]): Hamming Weight ($\mathtt{HW}$) leakage assumption only.
- This work: generalization to a class of leakage function $\mathtt{L}$

Outline

Generalization of the physical leakage model

- CHES21: $\mathtt{L} = \mathtt{HW}$
- More realistic model:

$$\mathtt{L}(k^*) = \sum_{i=1}^{n_b} \alpha_i \beta_i(k^*)$$

with $\alpha_i \in \mathbb{R}, \mathtt{L}(k^*) \in \mathbb{R}$

Generalization of the physical leakage model

- CHES21: $\mathtt{L} = \mathtt{HW}$
- More realistic model:

$$\mathtt{L}(k^*) = \sum_{i=1}^{n_b} \alpha_i \beta_i(k^*)$$

  with $\alpha_i \in \mathbb{R}, \mathtt{L}(k^*) \in \mathbb{R}$
- LWPR case: $\forall i, \alpha_i = 1, \beta_i(k^*) = k^*(i) \rightarrow \mathtt{L}(k^*) = \mathtt{HW}(k^*)$

Formal security analysis setting

- Considering that $\mathtt{L}$ can be interpreted over $\mathbb{F}_p$
  $\rightarrow$ algebraic system over $\mathbb{F}_p$ with unknowns $\mathbf{K}_{i,j}$

Formal security analysis setting

- Considering that $\mathtt{L}$ can be interpreted over $\mathbb{F}_p$
  $\rightarrow$ algebraic system over $\mathbb{F}_p$ with unknowns $\mathbf{K}_{i,j}$
- $s-$bounded pseudo-linear leakage functions (serial case):

$$\mathtt{L} \approx \mathtt{F}_a : \mathbb{F}_p \rightarrow \mathbb{F}_p, y \rightarrow \sum_{i=1}^{t} a_i \cdot y(i)$$

with $a_i \in [0, s], s \in \mathbb{F}_p, t = \lceil \log p \rceil, st < p$

Formal security analysis setting

- Considering that $\mathtt{L}$ can be interpreted over $\mathbb{F}_p$
  $\rightarrow$ algebraic system over $\mathbb{F}_p$ with unknowns $\mathbf{K}_{i,j}$
- $s-$bounded pseudo-linear leakage functions (serial case):

$$\mathtt{L} \approx \mathtt{F}_a : \mathbb{F}_p \rightarrow \mathbb{F}_p, y \rightarrow \sum_{i=1}^{t} a_i \cdot y(i)$$

  with $a_i \in [0, s], s \in \mathbb{F}_p, t = \lceil \log p \rceil, st < p$
- Hypothesis:
  - Bounded degree of $\mathtt{L}$
  - Bounded s

  $\rightarrow$ Leads to attack complexity $\geq \mathcal{O}(2^{\lambda})$

Intuition on s-bounded pseudo-linear function

- Consider $p = 7$, $t = 3$, $\mathtt{F}_a = 1 \cdot y(1) + 2 \cdot y(2) + 2 \cdot y(3)$

y	$y(i)$			$\mathtt{F}_a(y)$
0	0	0	0	0
1	1	0	0	1
2	0	1	0	2
3	1	1	0	3
4	0	0	1	2
5	1	0	1	3
6	0	1	1	4

$$\Leftrightarrow \mathtt{F}_a : \mathbb{F}_p \rightarrow \mathbb{F}_p, \quad y \rightarrow 6y^6 + 2y^5 + 5y^3 + 2y$$

$\rightarrow$ Linear over the bits $\rightarrow$ Non-linear over $\mathtt{F}_p$

- 2 main images (i.e., $2, 3$) with main preimage size $v_{\mathtt{F}_a} = 2$

Concrete attacks analysis

Exact algebraic system attack

$$\begin{bmatrix} K_{(1,1)} & K_{(1,2)} & \cdots & K_{(1,n+1)} \\ K_{(2,1)} & & & \vdots \\ \vdots & & & \vdots \\ K_{(m,1)} & \cdots & \cdots & K_{(m,n+1)} \end{bmatrix} \times \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \\ 1 \end{bmatrix} = \begin{bmatrix} k_1^* \\ k_2^* \\ \vdots \\ k_m^* \end{bmatrix}$$

$l = \mathtt{F}_a(k_1^*) = \mathtt{F}_a(K_1 \boxdot r)$

$\mathcal{A}$

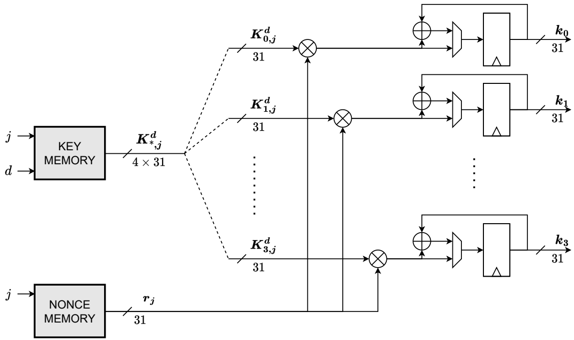
- Knowing $\mathtt{F}_a$, l, $\boldsymbol{r}$, solve for $K_{(1,*)} = K_1$
- Complexity $\approx \mathcal{O}(V_d^2) = \mathcal{O}\left(\binom{n+d}{n}^2\right)$
- $d = \deg(\mathtt{F}_a) \geq v_{\mathtt{F}_a}$

Other contributions (see paper):

- Noisy linear system complexity (non-linearity)
- Adaptation for parallel case (required)
- Worst-case s-bounded leakage

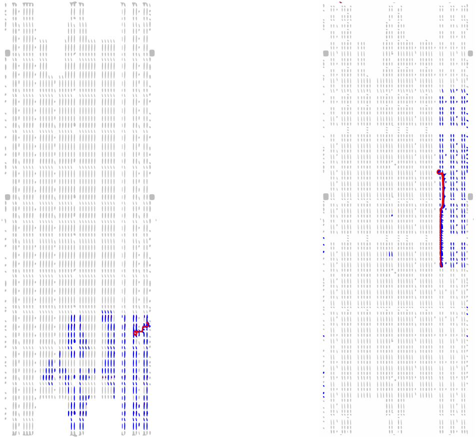
Outline

Experimental setup



- HW // implem. of LWPR
- $n = m = 4$, $p = 2^{31} - 1$
- 3 congestion levels
  - Unconstrained
  - Constrained
  - Virtually amplified

Example of congestion

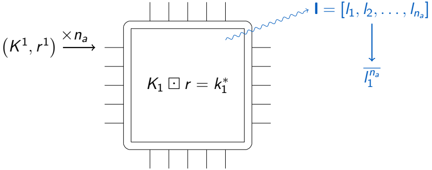


Unconstrained Constrained

- Same architecture:
  - White: unused ressource
  - Blue: used ressource
  - Red: same signal route

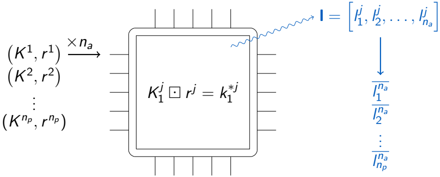
Physical L function assumptions assesment

- Noiseless linear regression model of degree 1



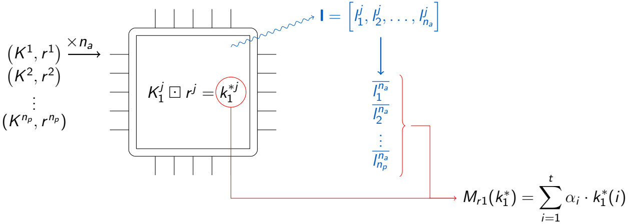
Physical L function assumptions assesment

- Noiseless linear regression model of degree 1



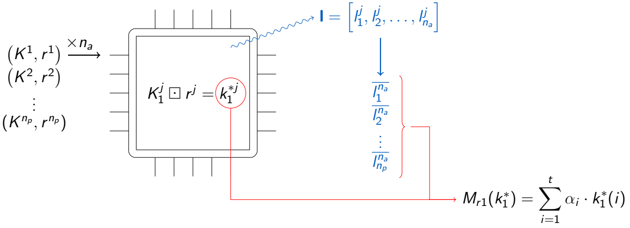
Physical L function assumptions assesment

- Noiseless linear regression model of degree 1



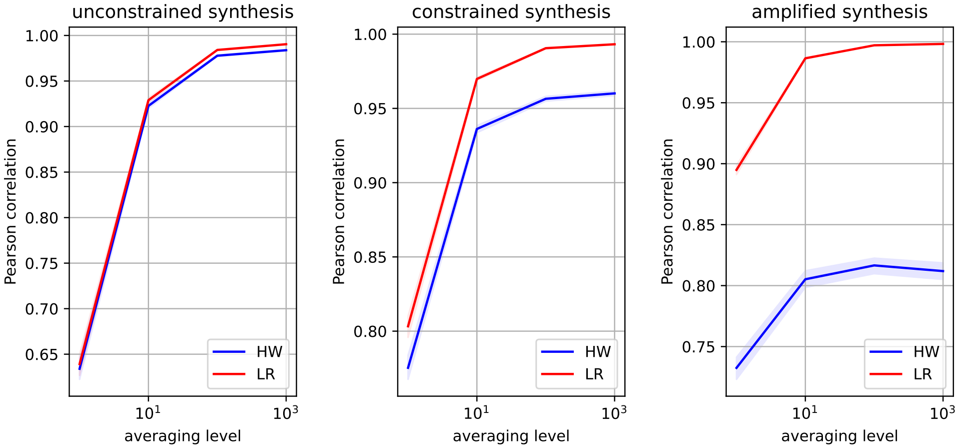
Physical L function assumptions assesment

- Noiseless linear regression model of degree 1



- Correlation based SCA security: $N = \dfrac{c}{\hat{\rho}(M_{r1}, L)^2}$

Correlation results



$\Rightarrow$ 1^{st} degree LR model captures most of the information.

Bound on the value of s

- Considering s-bounded leakage (discretized version of M_{r1} denoted M_{r1}^{s})

$$\hat{\boldsymbol{a}} = \left\lceil \boldsymbol{\alpha} \cdot \frac{s}{\max(\boldsymbol{\alpha})} \right\rceil$$

Bound on the value of s

- Considering s-bounded leakage (discretized version of M_{r1} denoted M^s_{r1})

$$\hat{\boldsymbol{a}} = \left\lceil \boldsymbol{\alpha} \cdot \frac{s}{\max(\boldsymbol{\alpha})} \right\rceil$$

- Correlation chain rules

$$\begin{aligned} \hat{\rho}\left(M^s_{r1}, L\right) &= \hat{\rho}\left(M^s_{r1}, M_{r1}\right) \cdot \hat{\rho}\left(M_{r1}, L\right) \\ &= (1-\phi) \cdot \hat{\rho}\left(M_{r1}, L\right) \end{aligned}$$

Bound on the value of s

- Considering s-bounded leakage (discretized version of M_{r1} denoted M^s_{r1})

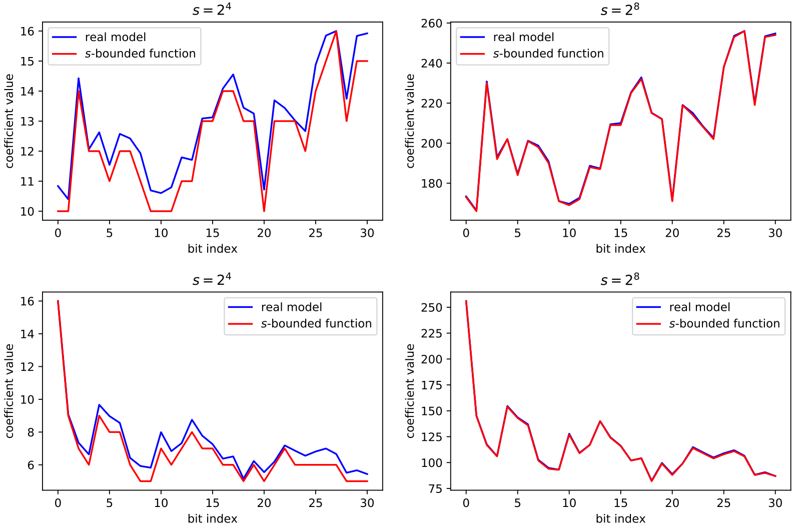
$$\hat{\boldsymbol{a}} = \left\lceil \boldsymbol{\alpha} \cdot \frac{s}{\max(\boldsymbol{\alpha})} \right\rfloor$$

- Correlation chain rules

$$\begin{aligned}\hat{\rho}\left(M^s_{r1}, L\right) &= \hat{\rho}\left(M^s_{r1}, M_{r1}\right) \cdot \hat{\rho}\left(M_{r1}, L\right) \\ &= (1 - \phi) \cdot \hat{\rho}\left(M_{r1}, L\right)\end{aligned}$$

- With $s = 2^8 \rightarrow \phi < 10^{-6}$

Discretized model coefficients



Putting things together

- From experimentation: reasonable physical leakage hypotheses
  $\rightarrow$ s-bounded physical leakage analysis sound.

- Practical implementation analyzed:
  - 124-bit k^*
  - Parallel implementation (3 congestion flavours)
  - $s = 2^{12}$

  $\rightarrow$ complexity $> \mathcal{O}(2^{124})$

- (Going further, LWPR secure for quadratic leakage function, see paper)

Outline

1. Background
2. Generalization of LWPR leakage model
3. Leakage function hypotheses validation
4. Conclusion

Conclusion

- LWPR is secure for wide class of (sound) leakage function
  - if implemented with parallelism (the more, the better).
  - when $\mathcal{A}$ follows our natural attack path.
- Open problem:
  - Analysis based on cardinality of leakage function
    $\rightarrow$ link s to quality of measurement apparatus
  - Multivariate analysis
  - Improved cryptanalysis to break LWPR
  - Integration in PQ crypto

Questions

Questions?

Supplementary

Supplementary material

Parallelism Requirement Intuition: LWPR case

$$\begin{bmatrix} K_{(1,1)} & K_{(1,2)} & \cdots & K_{(1,n+1)} \\ K_{(2,1)} & K_{(2,2)} & \cdots & K_{(2,n+1)} \\ K_{(3,1)} & K_{(3,2)} & \cdots & K_{(3,n+1)} \\ K_{(4,1)} & K_{(4,2)} & \cdots & K_{(4,n+1)} \end{bmatrix} \times \begin{bmatrix} r_1^1 \\ r_2^1 \\ \vdots \\ r_n^1 \\ 1 \end{bmatrix} = \begin{bmatrix} k_1^{*1} \\ k_2^{*1} \\ k_3^{*1} \\ k_4^{*1} \end{bmatrix} \qquad k_1^{*1} \xrightarrow{\mathtt{L}(k_1^{*1})} \mathcal{A}$$

- Serial recombination of k^*
  - one 31-bit words k_i^* per cycle.
- $\mathcal{A}$ obtains independent $\mathtt{L}(k_i^*)$
  - she can filter worst-case leakage e.g., $\mathtt{HW}(k_i^*) = 0 \rightarrow k_i^* = 0$ (with prob. $1/p$)
- $(n+1)$ w.c. observations $\rightarrow \mathbf{K}_i$ recovery
- Parallelism limits the risk ([DMMS21])

Parallelism Requirement Intuition: LWPR case

$$\begin{bmatrix} K_{(1,1)} & K_{(1,2)} & \cdots & K_{(1,n+1)} \\ K_{(2,1)} & K_{(2,2)} & \cdots & K_{(2,n+1)} \\ K_{(3,1)} & K_{(3,2)} & \cdots & K_{(3,n+1)} \\ K_{(4,1)} & K_{(4,2)} & \cdots & K_{(4,n+1)} \end{bmatrix} \times \begin{bmatrix} r_1^2 \\ r_2^2 \\ \vdots \\ r_n^2 \\ 1 \end{bmatrix} = \begin{bmatrix} k_1^{*2} \\ k_2^{*2} \\ k_3^{*2} \\ k_4^{*2} \end{bmatrix} \qquad k_1^{*2} \xrightarrow{\mathtt{L}(k_1^{*2})} \mathcal{A}$$

- Serial recombination of k^*
  - one 31-bit words k_i^* per cycle.
- $\mathcal{A}$ obtains independent $\mathtt{L}(k_i^*)$
  - she can filter worst-case leakage e.g., $\mathtt{HW}(k_i^*) = 0 \to k_i^* = 0$ (with prob. $1/p$)
- $(n+1)$ w.c. observations $\to$ $\mathbf{K}_i$ recovery
- Parallelism limits the risk ([DMMS21])

Parallelism Requirement Intuition: LWPR case

$$\begin{bmatrix} K_{(1,1)} & K_{(1,2)} & \cdots & K_{(1,n+1)} \\ K_{(2,1)} & K_{(2,2)} & \cdots & K_{(2,n+1)} \\ K_{(3,1)} & K_{(3,2)} & \cdots & K_{(3,n+1)} \\ K_{(4,1)} & K_{(4,2)} & \cdots & K_{(4,n+1)} \end{bmatrix} \times \begin{bmatrix} r_1^3 \\ r_2^3 \\ \vdots \\ r_n^3 \\ 1 \end{bmatrix} = \begin{bmatrix} k_1^{*3} \\ k_2^{*3} \\ k_3^{*3} \\ k_4^{*3} \end{bmatrix} \xrightarrow{\mathtt{L}(k_1^{*3})} \mathcal{A}$$

- Serial recombination of k^*
  - one 31-bit words k_i^* per cycle.
- $\mathcal{A}$ obtains independent $\mathtt{L}(k_i^*)$
  - she can filter worst-case leakage e.g., $\mathtt{HW}(k_i^*) = 0 \rightarrow k_i^* = 0$ (with prob. $1/p$)
- $(n+1)$ w.c. observations $\rightarrow \mathbf{K}_i$ recovery
- Parallelism limits the risk ([DMMS21])