Horizontal Side-Channel Attacks and Countermeasures on the ISW Masking Scheme

Alberto Battistello, Jean-Sébastien Coron, Emmanuel Prouff and Rina Zeitoun

Speaker: Guillaume Barbu

CHES ’16, Santa Barbara

August 17th 2016

Agenda

1. Context of Application of our Attack
2. Horizontal Side-Channel Attack: A First Attempt
3. Improved Horizontal Side-Channel Attack
4. Practical Experiments
5. Countermeasure

Outline

1. Context of Application of our Attack
2. Horizontal Side-Channel Attack: A First Attempt
3. Improved Horizontal Side-Channel Attack
4. Practical Experiments
5. Countermeasure

Basic Principle

Each sensitive variable x is shared into $n+1$ variables:

$$x = x_0 \oplus x_1 \oplus x_2 \oplus \cdots \oplus x_n$$

Basic Principle

Each sensitive variable x is shared into $n+1$ variables:

$$x \;=\; x_0 \oplus x_1 \oplus x_2 \oplus \cdots \oplus x_n$$

Security at order n [PR13, DFS15]

A **sufficient** condition for security at order n:

$$\sigma \cdot c \;\geqslant\; n$$

with σ the standard deviation of the side-channel observations

[PR13] *Higher-Order Side Channel Security and Mask Refreshing.* Prouff, Rivain, Eurocrypt 2013.
[DFS15] *Making Masking Security Proofs Concrete.* Duc, Faust, Standaert, Eurocrypt 2015.

Higher-Order Masking

Basic Principle

Each sensitive variable x is shared into $n+1$ variables:

$$x = x_0 \oplus x_1 \oplus x_2 \oplus \cdots \oplus x_n$$

Security at order n [PR13, DFS15]

A **sufficient** condition for security at order n:

$$\sigma \cdot c \geqslant n$$

with σ the standard deviation of the side-channel observations

$$\boxed{\text{What if } n > \sigma \cdot c \text{ ?}}$$

[PR13] *Higher-Order Side Channel Security and Mask Refreshing.* Prouff, Rivain, Eurocrypt 2013.
[DFS15] *Making Masking Security Proofs Concrete.* Duc, Faust, Standaert, Eurocrypt 2015.

Context of Application: computation of $x \cdot y$

- **Inputs:** $(x_i)_i$ and $(y_i)_i$ such that
  - $x_0 \oplus x_1 \oplus \cdots \oplus x_n = x$
  - $y_0 \oplus y_1 \oplus \cdots \oplus y_n = y$
- **Output:** $(c_i)_i$ such that
  - $c_0 \oplus c_1 \oplus c_2 \oplus \cdots \oplus c_n = x\, y$

Secure Multiplication in Higher-Order Masking Schemes

Context of Application: computation of $x \cdot y$

- **Inputs:** $(x_i)_i$ and $(y_i)_i$ such that
  - $x_0 \oplus x_1 \oplus \cdots \oplus x_n = x$
  - $y_0 \oplus y_1 \oplus \cdots \oplus y_n = y$
- **Output:** $(c_i)_i$ such that
  - $c_0 \oplus c_1 \oplus c_2 \oplus \cdots \oplus c_n = x\,y$

☞ Use ISW/RP scheme [ISW03, RP10]

[ISW03] *Private Circuits: Securing Hardware against Probing Attacks.* Ishai, Sahai, Wagner, CRYPTO'03

[RP10] *Provably Secure Higher-Order Masking of AES.* Rivain, Prouff, CHES'10.

ISW / Rivain-Prouff Countermeasure

Algorithm 1 SecMult

Require: $\bigoplus_i x_i = x$ and $\bigoplus_i y_i = y$
Ensure: shares c_i satisfying $\bigoplus_i c_i = x\,y$

```
for i = 0 to n
    for j = i + 1 to n
        r_{i,j} ← rand
        r_{j,i} ← (r_{i,j} ⊕ x_i y_j) ⊕ x_j y_i
for i = 0 to n
    c_i ← x_i y_i
    for j = 0 to n, j ≠ i do c_i ← c_i ⊕ r_{i,j}
return (c_0, c_1, ..., c_n)
```

$$\begin{pmatrix} x_0 y_0 & (r_{1,2} \oplus x_0 y_1) \oplus x_1 y_0 & (r_{1,3} \oplus x_0 y_2) \oplus x_2 y_0 \\ r_{1,2} & x_1 y_1 & (r_{2,3} \oplus x_1 y_2) \oplus x_2 y_1 \\ r_{1,3} & r_{2,3} & x_2 y_2 \end{pmatrix} \begin{array}{l} \Rightarrow \; c_0 \\ \Rightarrow \; c_1 \\ \Rightarrow \; c_2 \end{array}$$

Outline

Horizontal SCA on ISW Countermeasure

Assumption

The attacker observes the manipulation of all x_i, y_j and $x_i\,y_j$

- 1 manipulation of each $x_i\,y_j$
- $\underline{n}$ manipulations of each $\underline{x_i}$ and $\underline{y_j}$

$$\begin{pmatrix} x_0 y_0 & (r_{1,2} \oplus x_0 \underline{y_1}) \oplus x_1 y_0 & (r_{1,3} \oplus x_0 y_2) \oplus x_2 y_0 \\ r_{1,2} & x_1 \underline{y_1} & (r_{2,3} \oplus x_1 y_2) \oplus x_2 \underline{y_1} \\ r_{1,3} & r_{2,3} & x_2 y_2 \end{pmatrix} \begin{matrix} \Rightarrow & c_0 \\ \Rightarrow & c_1 \\ \Rightarrow & c_2 \end{matrix}$$

Horizontal SCA on ISW Countermeasure

Assumption: we get for $0 \leqslant i, j \leqslant n$:

$$\begin{cases} L_i & = & h(x_i) & + & B_i(\sigma/\sqrt{n}) \\ L'_j & = & h(y_j) & + & B'_j(\sigma/\sqrt{n}) \\ L''_{ij} & = & h(x_i \cdot y_j) & + & B''_{ij}(\sigma) \end{cases}$$

$h()$: Hamming weight B: Gaussian noise

Horizontal SCA on ISW Countermeasure

Assumption: we get for $0 \leqslant i, j \leqslant n$:

$$\begin{cases} L_i = h(x_i) + B_i(\sigma/\sqrt{n}) \\ L'_j = h(y_j) + B'_j(\sigma/\sqrt{n}) \\ L''_{ij} = h(x_i \cdot y_j) + B''_{ij}(\sigma) \end{cases}$$

$h()$: Hamming weight $\quad$ B: Gaussian noise

The intuition for the case $k = 1$

- If $x_i = 0 \Rightarrow \forall j \; h(x_i \cdot y_j) = 0 \Rightarrow \forall j \; L''_{ij} = B''_{ij}$

Horizontal SCA on ISW Countermeasure

Assumption: we get for $0 \leqslant i, j \leqslant n$:

$$\begin{cases} L_i &= h(x_i) &+ B_i(\sigma/\sqrt{n}) \\ L'_j &= h(y_j) &+ B'_j(\sigma/\sqrt{n}) \\ L''_{ij} &= h(x_i \cdot y_j) &+ B''_{ij}(\sigma) \end{cases}$$

$h()$: Hamming weight $\quad B$: Gaussian noise

The intuition for the case $k = 1$

- If $x_i = 0 \Rightarrow \forall j\ h(x_i \cdot y_j) = 0 \Rightarrow \forall j\ L''_{ij} = B''_{ij}$
- If $x_i = 1 \Rightarrow \forall j\ h(x_i \cdot y_j) = h(y_j) \Rightarrow \forall j\ L''_{ij} = h(y_j) + B''_{ij}$

Horizontal SCA on ISW Countermeasure

Assumption: we get for $0 \leqslant i, j \leqslant n$:

$$\begin{cases} L_i &= h(x_i) &+ B_i(\sigma/\sqrt{n}) \\ L'_j &= h(y_j) &+ B'_j(\sigma/\sqrt{n}) \\ L''_{ij} &= h(x_i \cdot y_j) &+ B''_{ij}(\sigma) \end{cases}$$

$h()$: Hamming weight $\qquad$ B: Gaussian noise

The intuition for the case $k = 1$

- If $x_i = 0 \;\Rightarrow\; \forall j\; h(x_i \cdot y_j) = 0 \;\Rightarrow\; \forall j\; L''_{ij} = B''_{ij}$
- If $x_i = 1 \;\Rightarrow\; \forall j\; h(x_i \cdot y_j) = h(y_j) \;\Rightarrow\; \forall j\; L''_{ij} = h(y_j) + B''_{ij}$

☞ Distinguish between $x_i = 0$ and $x_i = 1$

Generalization to $\mathbb{F}_{2^k}$: A first attempt

Attack Principle

1. Build templates relative to the manipulation of all values of x_i, y_j and $x_i y_j$
2. Find the x_i maximizing the probability of the observation of a given tuple $L_i, (L'_j, L''_{ij}),\ \forall j$

Generalization to $\mathbb{F}_{2^k}$: A first attempt

Finding x_i

- Compute a probability distribution for x_i:

Finding x_i

- Compute a probability distribution for x_i:
  - $L_i \overset{ML}{\Rightarrow} \Pr[L_i \mid x_i = u], \ \forall u$

Generalization to $\mathbb{F}_{2^k}$: A first attempt

Finding x_i

- Compute a probability distribution for x_i:
  - $L_i \overset{ML}{\Rightarrow} \Pr[L_i \mid x_i = u], \forall u$
- For each y_j :
  - Compute a probability distribution for y_j:

Generalization to $\mathbb{F}_{2^k}$: A first attempt

Finding x_i

- Compute a probability distribution for x_i:
  - $L_i \overset{ML}{\Rightarrow} \Pr[L_i \mid x_i = u], \ \forall u$
- For each y_j :
  - Compute a probability distribution for y_j:
    - $L'_j \overset{ML}{\Rightarrow} \Pr[L'_j \mid y_j = u], \ \forall u$

Generalization to $\mathbb{F}_{2^k}$: A first attempt

Finding x_i

- Compute a probability distribution for x_i:
  - $L_i \overset{ML}{\Rightarrow} \Pr[L_i \mid x_i = u], \forall u$
- For each y_j :
  - Compute a probability distribution for y_j:
    - $L'_j \overset{ML}{\Rightarrow} \Pr[L'_j \mid y_j = u], \forall u$
  - Compute a probability distribution for $x_i \cdot y_j$:

Generalization to $\mathbb{F}_{2^k}$: A first attempt

Finding x_i

- Compute a probability distribution for x_i:
  - $L_i \overset{ML}{\Rightarrow} \Pr[L_i \mid x_i = u],\ \forall u$
- For each y_j :
  - Compute a probability distribution for y_j:
    - $L'_j \overset{ML}{\Rightarrow} \Pr[L'_j \mid y_j = u],\ \forall u$
  - Compute a probability distribution for $x_i \cdot y_j$:
    - $L''_{ij} \overset{ML}{\Rightarrow} \Pr[L''_{ij} \mid x_i \cdot y_j = u]$

Generalization to $\mathbb{F}_{2^k}$: A first attempt

Finding x_i

- Compute a probability distribution for x_i:
  - $L_i \overset{ML}{\Rightarrow} \Pr[L_i \mid x_i = u],\ \forall u$
- For each y_j :
  - Compute a probability distribution for y_j:
    - $L'_j \overset{ML}{\Rightarrow} \Pr[L'_j \mid y_j = u],\ \forall u$
  - Compute a probability distribution for $x_i \cdot y_j$:
    - $L''_{ij} \overset{ML}{\Rightarrow} \Pr[L''_{ij} \mid x_i \cdot y_j = u]$
  - Which gives $\Pr[L'_j, L''_{ij} \mid x_i = u],\ \forall u$

oberthur TECHNOLOGIES THE M COMPANY

Generalization to $\mathbb{F}_{2^k}$: A first attempt

Finding x_i

$$f_{\mathbf{L}|X_i}((L_i,(L'_j,L''_{i,j})),x_i) =$$

$$2^{-nk} f(L_i,x_i)\cdot\prod_{j=1}^{n}\left(\sum_{y} f'_{\mathbf{L}'|Y_j}(L'_j)\cdot f''_{\mathbf{L}''|X_i,Y_j}(L''_{i,j})\right)$$

Algorithm for Attack 1

Algorithm 2 Attack 1

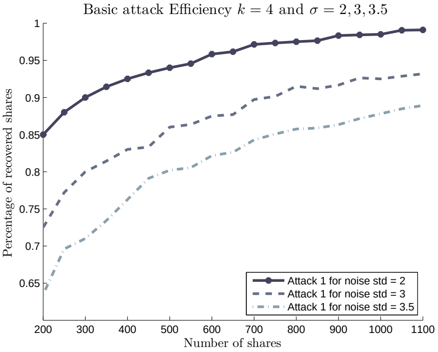
Require: Leakages L_i, L'_j, L''_{ij} for all j, noise σ, number of shares n

1: **for** $x_p = 0$ **to** $2^k - 1$
2: $\quad proba[x_p] = \log(d_{ML}(L_i, x_p, \sigma/\sqrt{n}))$
3: **for each** y_j
4: $\quad$ **for** $x_p = 0$ **to** $2^k - 1$
5: $\quad\quad proba[x_p] += \log(d_{ML}(L'_j, L''_{ij}, x_p, \sigma, n))$
6: **return** x_i with $i = indexMax(proba)$

Numerical Experiments

Number of shares n as a function of σ to succeed with probability > 0.5

σ (SNR)	0 ($+\infty$)	0.2 (25)	0.4 (6.25)	0.6 (2.77)	0.8 (1.56)	1 (1)
n	12	14	30	73	160	284



Outline

Generalization to $\mathbb{F}_{2^k}$: An iterative attack

Apply Previous Attack

for each x_i

$$\begin{cases} P_{x_0} = (\,.\,,\,.\,,\,\ldots\,,\,.\,) \\ P_{x_1} = (\,.\,,\,.\,,\,\ldots\,,\,.\,) \\ \vdots \\ P_{x_n} = (\,.\,,\,.\,,\,\ldots\,,\,.\,) \end{cases}$$

Generalization to $\mathbb{F}_{2^k}$: An iterative attack

Apply Previous Attack

for each x_i

$$\begin{cases} P_{x_0} = (. , . , \dots , .) \\ P_{x_1} = (. , . , \dots , .) \\ \vdots \\ P_{x_n} = (. , . , \dots , .) \end{cases}$$

Apply Same Attack

for each y_i

Use previous P_{x_i}

$$\begin{cases} P_{y_0} = (. , . , \dots , .) \\ P_{y_1} = (. , . , \dots , .) \\ \vdots \\ P_{y_n} = (. , . , \dots , .) \end{cases}$$

Generalization to $\mathbb{F}_{2^k}$: An iterative attack

Apply Previous Attack
for each x_i

$$\begin{cases} P_{x_0} = (. , . , \ldots , .) \\ P_{x_1} = (. , . , \ldots , .) \\ \vdots \\ P_{x_n} = (. , . , \ldots , .) \end{cases}$$

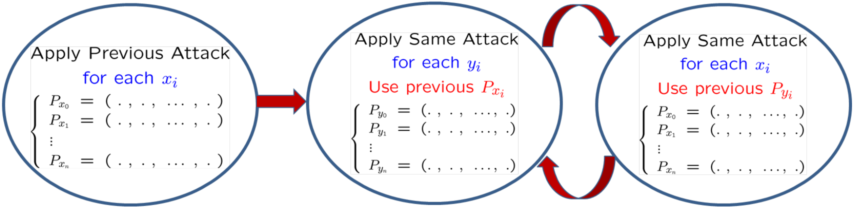
Apply Same Attack
for each y_i
Use previous P_{x_i}

$$\begin{cases} P_{y_0} = (. , . , \ldots , .) \\ P_{y_1} = (. , . , \ldots , .) \\ \vdots \\ P_{y_n} = (. , . , \ldots , .) \end{cases}$$

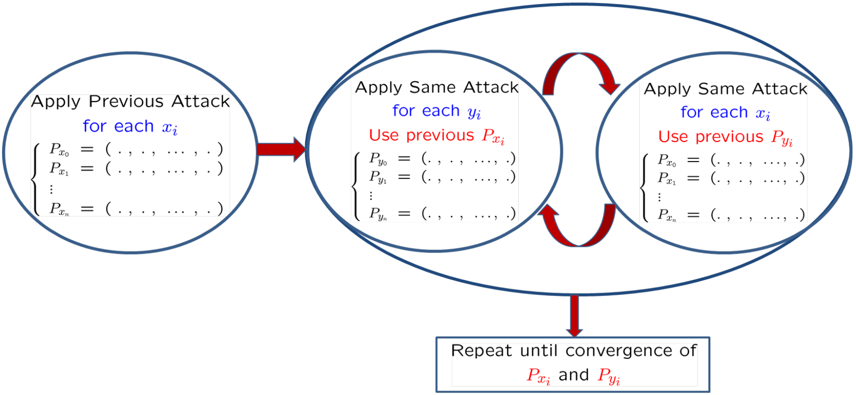
Apply Same Attack
for each x_i
Use previous P_{y_i}

$$\begin{cases} P_{x_0} = (. , . , \ldots , .) \\ P_{x_1} = (. , . , \ldots , .) \\ \vdots \\ P_{x_n} = (. , . , \ldots , .) \end{cases}$$

Generalization to $\mathbb{F}_{2^k}$: An iterative attack

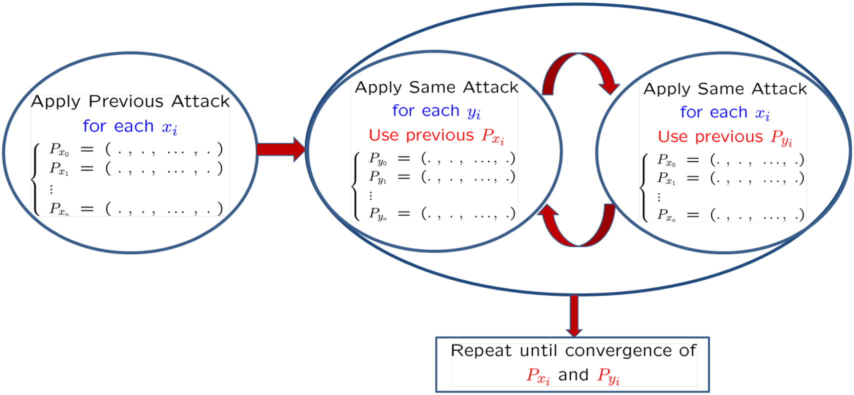


Generalization to $\mathbb{F}_{2^k}$: An iterative attack



oberthur TECHNOLOGIES THE M COMPANY

Generalization to $\mathbb{F}_{2^k}$: An iterative attack

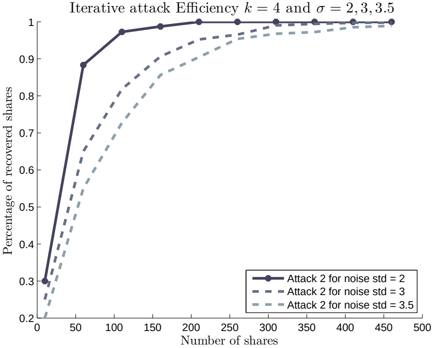


- **Improvement:** Repeat the attack by starting with y_i

Numerical Experiments

Number of shares n as a function of σ to succeed with probability > 0.5

σ $(\mathrm{SNR}_4, \mathrm{SNR}_8)$	0 $(+\infty, +\infty)$	0.2 (25, 17.67)	0.4 (6.25, 4.41)	0.6 (2.77, 1.96)	0.8 (1.56, 1.10)	1 (1, 0.7071)
n (for $\mathbb{F}_{2^4}$)	2	2	3	6	13	25
n (for $\mathbb{F}_{2^8}$)	5	6	8	11	16	21



Security at order n [PR13, DFS15]

A **sufficient** condition for security at order n: $\sigma \cdot c \geqslant n$

$\boxed{\text{What if } n > \sigma \cdot c \text{ ?}}$

[PR13] *Higher-Order Side Channel Security and Mask Refreshing.* Prouff, Rivain, Eurocrypt 2013.
[DFS15] *Making Masking Security Proofs Concrete.* Duc, Faust, Standaert, Eurocrypt 2015.

Higher-Order Masking

Security at order n [PR13, DFS15]

A **sufficient** condition for security at order n: $\sigma \cdot c \geqslant n$

$\boxed{\text{What if } n > \sigma \cdot c \text{ ?}}$

Condition for our attack to work

- Collect leakages on $(y_j, x_i y_j)$

Higher-Order Masking

Security at order n [PR13, DFS15]

A **sufficient** condition for security at order n: $\sigma \cdot c \geqslant n$

$\boxed{\text{What if } n > \sigma \cdot c \text{ ?}}$

Condition for our attack to work

- Collect leakages on $(y_j, x_i y_j)$ ☞ Second order SCA

Security at order n [PR13, DFS15]

A **sufficient** condition for security at order n: $\sigma \cdot c \geqslant n$

$$\boxed{\text{What if } n > \sigma \cdot c \text{ ?}}$$

Condition for our attack to work

- Collect leakages on $(y_j, x_i y_j)$ ☞ Second order SCA
- [CJRR99, GHR15, SVO10] ☞ $n = \mathcal{O}(\sigma^2_{y_j} \cdot \sigma^2_{x_i y_j})$

[CJRR99] *Towards sound approaches to counteract power-analysis attack.* Chari, Jutla, Rao, Rohatgi, Crypto'99.
[GHR15] *A key to success - success exponents for side-channel distinguishers.* Guilley, Heuser, Rioul, Indocrypt'15.
[SVO10] *The world is not enough: Another look on second-order DPA.* Standaert, Veyrat-Charvillon, Oswald, Gierlichs, Medwed, Kasper, Mangard, Asiacrypt'10.

Higher-Order Masking

Security at order n [PR13, DFS15]

A **sufficient** condition for security at order n: $\sigma \cdot c \geqslant n$

What if $n > \sigma \cdot c$?

Condition for our attack to work

- Collect leakages on $(y_j, x_i y_j)$ ☞ Second order SCA
- [CJRR99, GHR15, SVO10] ☞ $n = \mathcal{O}(\sigma_{y_j}^2 \cdot \sigma_{x_i y_j}^2)$
  ☞ $n = \mathcal{O}(\sigma^2)$

[CJRR99] *Towards sound approaches to counteract power-analysis attack.* Chari, Jutla, Rao, Rohatgi, Crypto'99.
[GHR15] *A key to success - success exponents for side-channel distinguishers.* Guilley, Heuser, Rioul, Indocrypt'15.
[SVO10] *The world is not enough: Another look on second-order DPA.* Standaert, Veyrat-Charvillon, Oswald, Gierlichs, Medwed, Kasper, Mangard, Asiacrypt'10.

Security at order n [PR13, DFS15]

A **sufficient** condition for security at order n: $\sigma \cdot c \geqslant n$

What if $n > \sigma \cdot c$?

Condition for our attack to work

- Collect leakages on $(y_j, x_i y_j)$ ☞ Second order SCA
- [CJRR99, GHR15, SVO10] ☞ $n = \mathcal{O}(\sigma_{y_j}^2 \cdot \sigma_{x_i y_j}^2)$

☞ $n = \mathcal{O}(\sigma^2)$

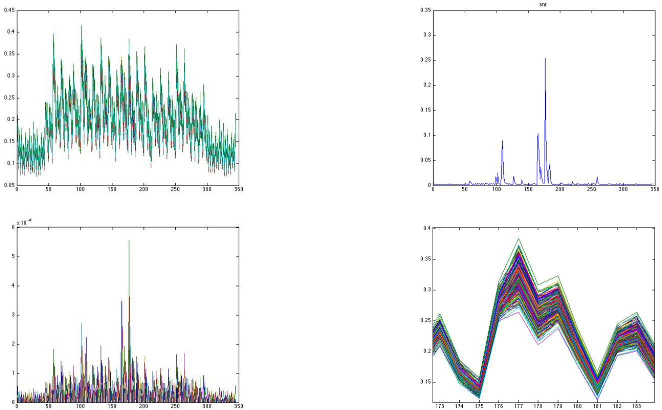
If $n > \sigma^2 \cdot c$ ☞ An **attack** is possible

Outline

1. Context of Application of our Attack
2. Horizontal Side-Channel Attack: A First Attempt
3. Improved Horizontal Side-Channel Attack
4. Practical Experiments
5. Countermeasure

Create Templates

Compute mean and variance over 200k observations for each x_i.

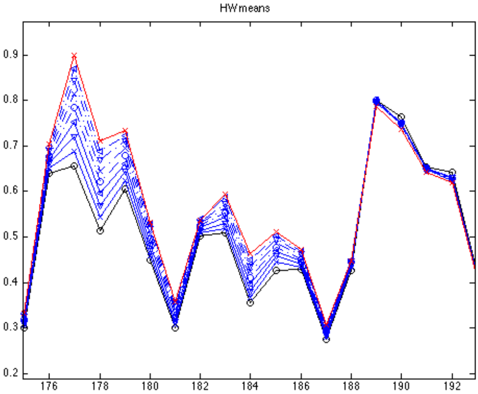


Average signal for each x_i (top left), Variance of signal for each x_i (bottom left), Signal to Noise Ratio (top right), Average signal for each x_i (Zoom on POI) (bottom right)

Insights on the Model

HW Templates

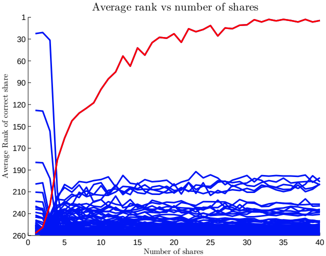
Compute mean and variance over 200k observations for each $HW(x_i)$.

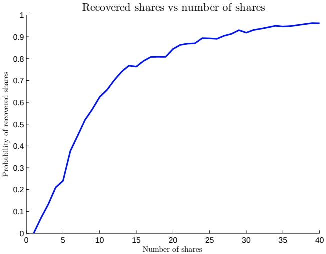


Results Attack 1 (ML)

Simple attack results

Rank and probability of success averaged over 100 repetitions ($1 \le n \le 40$).





Result

10 shares seem sufficient with KEA.

Comparison with numerical experiments

σ (SNR$_8$)	0 $(+\infty)$	1 (0.7071)	11 (0.25)
n (for numerical)	5	21	NA
n (for experiments)	NA	NA	10

Number of shares n as a function of the noise σ to succeed with $P > 0.5$ (Attack 1 with $k = 8$).

Probably the disparity with numerical experiments is due to the use of the 11 points with a multivariate attack.

Outline

Reworking ISW/RP Countermeasure

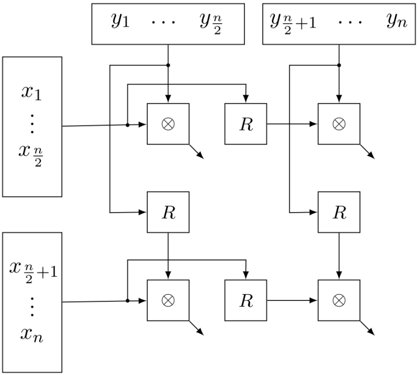
Algorithm 1 SecMult (*ISW*/*RP*)

Require: $\bigoplus_i x_i = x$ and $\bigoplus_i y_i = y$
Ensure: shares c_i satisfying $\bigoplus_i c_i = x\,y$

```
M_ij ← MatMult = (x_i · y_j)_{1≤i,j≤n}
for i = 0 to n
    for j = i + 1 to n
        r_{i,j} ← rand
        r_{j,i} ← (r_{i,j} ⊕ M_ij) ⊕ M_ji
for i = 0 to n
    c_i ← M_ii
    for j = 0 to n, j ≠ i do
        c_i ← c_i ⊕ r_{i,j}
return (c_0, c_1, ..., c_n)
```

Recursive MatMult

Split x_i, y_i into four blocks and refresh masks.



Countermeasure

Each variable is now manipulated at most twice

$$\begin{pmatrix} x_0y_0 & (r_{1,2}\oplus x_0\underline{y_1})\oplus x_1y_0 & (r_{1,3}\oplus x_2y_0)\oplus x_0y_2 & (r_{1,4}\oplus x_3y_0)\oplus x_0y_3 \\ r_{1,2} & x_1\underline{y_1} & (r_{2,3}\oplus x_2\underline{y_1})\oplus x_1y_2 & (r_{2,4}\oplus x_3\underline{y_1})\oplus x_1y_3 \\ r_{1,2} & r_{2,2} & x_2y_2 & (r_{3,4}\oplus x_2y_3)\oplus x_3y_2 \\ r_{1,4} & r_{2,4} & r_{3,4} & x_3y_3 \end{pmatrix}$$

With

$x = x \oplus RefreshMask$

$x = x \oplus RefreshMask$

$x = x \oplus RefreshMask$

$x = x \oplus RefreshMask$

Conclusion

- **Horizontal SCA** on Rivain-Prouff countermeasure
  - A **first attempt** with poor efficiency
  - An **improved attack** with more realistic results
  - **Successful experiments**

Conclusion

- **Horizontal SCA** on Rivain-Prouff countermeasure
  - A **first attempt** with poor efficiency
  - An **improved attack** with more realistic results
  - **Successful experiments**
- Can be performed if the **order n** is sufficiently **high**
  - For typical instances, about $n \approx 10$ is necessary

Conclusion / Perspectives

Conclusion

- **Horizontal SCA** on Rivain-Prouff countermeasure
  - A **first attempt** with poor efficiency
  - An **improved attack** with more realistic results
  - **Successful experiments**
- Can be performed if the **order n** is sufficiently **high**
  - For typical instances, about $n \approx 10$ is necessary

Perspectives

- Provide a proof of security of our countermeasure (against n^2 probes)
- Study the gap in between $n > \sigma c$ and $n > \sigma^2 c$
- Improve the countermeasure (cf. eprint)

Thank you for your attention!

http://eprint.iacr.org/2016/540