On the Practical Security of a Leakage Resilient Masking Scheme

T. Roche
thomas.roche@ssi.gouv.fr
Joint work with E. Prouff and M. Rivain

French Network and Information Security Agency (ANSSI)
CryptoExperts

CT-RSA 2014 – Feb. 26

Side Channel Analysis

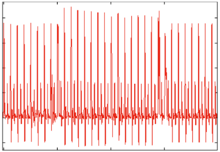
- Side Channel Attacks (SCA) appear 15 years ago
  - 1996 : Timing Attacks
  - 1998 : Power Analysis
  - 2000 : Electromagnetic Analysis
- Numerous attacks
  - 1998 : (single-bit) DPA — KocherJaffeJune 1999
  - 1999 : (multi-bit) DPA — Messerges 1999
  - 2000 : Higher-order SCA — Messerges 2000
  - 2002 : Template SCA — ChariRaoRohatgi 2002
  - 2004 : CPA — BrierClavierOlivier 2004
  - 2005 : Stochastic SCA — SchindlerLemkePaar 2006
  - 2008 : Mutual Information SCA — GierlichsBatinaTuyls 2008
  - etc.

Side Channel Analysis

- Side Channel Attacks (SCA) appear 15 years ago
  - 1996 : Timing Attacks
  - 1998 : Power Analysis
  - 2000 : Electromagnetic Analysis
- Numerous attacks
  - 1998 : (single-bit) DPA — KocherJaffeJune 1999
  - 1999 : (multi-bit) DPA — Messerges 1999
  - 2000 : Higher-order SCA — Messerges 2000
  - 2002 : Template SCA — ChariRaoRohatgi 2002
  - 2004 : CPA — BrierClavierOlivier 2004
  - 2005 : Stochastic SCA — SchindlerLemkePaar 2006
  - 2008 : Mutual Information SCA — GierlichsBatinaTuyls 2008
  - etc.

Side Channel Analysis

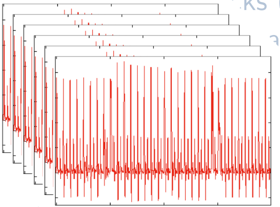
- ...cks (SCA) appear 15 years ago
  - ...Attacks
  - ...nalys...
  - ...nagne...



DES (ASIC), EM Radiations
http://www.dpacontest.org

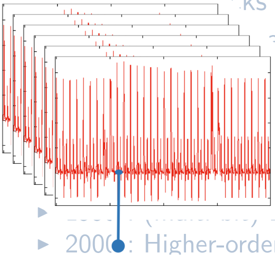
- 1998 : (single-bit) DPA — KocherJaffeJune 1999
- 1999 : (multi-bit) DPA — Messerges 1999
- 2000 : Higher-order SCA — Messerges 2000
- 2002 : Template SCA — ChariRaoRohatgi 2002
- 2004 : CPA — BrierClavierOlivier 2004
- 2005 : Stochastic SCA — SchindlerLemkePaar 2006
- 2008 : Mutual Information SCA — GierlichsBatinaTuyls 2008
- etc.

Side Channel Analysis



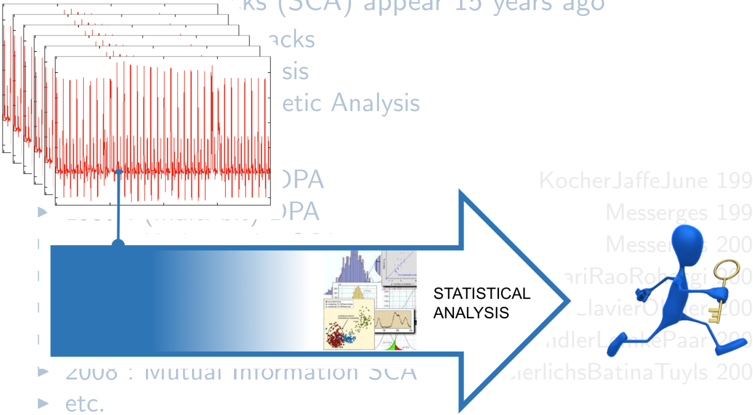
- …cks (SCA) appear 15 years ago
  - …acks
  - …sis
  - …etic Analysis
- …
  - …DPA KocherJaffeJune 1999
  - …DPA Messerges 1999
  - 2000 : Higher-order SCA Messerges 2000
  - 2002 : Template SCA ChariRaoRohatgi 2002
  - 2004 : CPA BrierClavierOlivier 2004
  - 2005 : Stochastic SCA SchindlerLemkePaar 2006
  - 2008 : Mutual Information SCA GierlichsBatinaTuyls 2008
  - etc.

Side Channel Analysis

- ...cks (SCA) appear 15 years ago
  - ...acks
  - ...sis
  - ...etic Analysis
- 

  - ...DPA — KocherJaffeJune 1999
  - ...DPA — Messerges 1999
  - 200... : Higher-order SCA — Messerges 2000
  - ChariRaoRohatgi 2002
  - BrierClavierOlivier 2004
  - 2005 : Stochastic SCA — SchindlerLemkePaar 2006
  - 2008 : Mutual Information SCA — GierlichsBatinaTuyls 2008
  - etc.

Side Channel Analysis

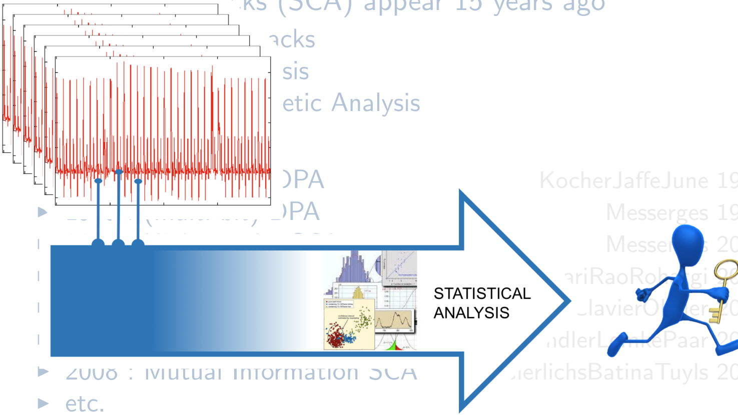


KocherJaffeJune 1999

Messerges 1999

- etc.

*d*th-order Side Channel Analysis



- ...cks (SCA) appear 15 years ago
  - ...acks
  - ...sis
  - ...etic Analysis
-
  - ...DPA KocherJaffeJune 1999
  - ...DPA Messerges 1999
  - Messer... 2000
  - ...2002
  - ...2004
  - ...2006
  - 2008 : Mutual Information SCA ...ierlichsBatinaTuyls 2008
  - etc.

Masking/Sharing Coutermeasures

Idea : consists in securing the implementation using secret sharing techniques.

- First Ideas in GoubinPatarin99 and ChariJutlaRaoRohatgi99.
- Soundness based on the following remark :

ChariJutlaRaoRohatgi, CRYPTO 1999

- Bit x masked $\mapsto x_0, x_1, \ldots, x_d$
- Leakage : $L_i \sim x_i + \mathcal{N}(\mu, \sigma^2)$
- # of leakage samples to test $\left((L_i)_i | x = 0\right) = \left((L_i)_i | x = 1\right)$:

$$q \geq O(1)\sigma^d$$

Masking/Sharing Coutermeasures

Idea : consists in securing the implementation using secret sharing techniques.

- First Ideas in GoubinPatarin99 and ChariJutlaRaoRohatgi99.
- Soundness based on the following remark :

ChariJutlaRaoRohatgi, CRYPTO 1999

- Bit x masked $\mapsto x_0, x_1, \ldots, x_d$
- Leakage : $L_i \sim x_i + \mathcal{N}(\mu, \sigma^2)$
- # of leakage samples to test $\left((L_i)_i | x = 0\right) = \left((L_i)_i | x = 1\right)$:

$$q \geq O(1)\sigma^d$$

Masking/Sharing Coutermeasures

Idea : consists in securing the implementation using secret sharing techniques.

- First Ideas in GoubinPatarin99 and ChariJutlaRaoRohatgi99.
- Soundness based on the following remark :

ChariJutlaRaoRohatgi, CRYPTO 1999

- Bit x masked $\mapsto x_0, x_1, \ldots, x_d$
- Leakage : $L_i \sim x_i + \mathcal{N}(\mu, \sigma^2)$
- \# of leakage samples to test $\big((L_i)_i | x = 0\big) = \big((L_i)_i | x = 1\big)$:

$$q \geq O(1)\sigma^d$$

Masking/Sharing Coutermeasures

Idea : consists in securing the implementation using secret sharing techniques.

- First Ideas in GoubinPatarin99 and ChariJutlaRaoRohatgi99.
- Soundness based on the following remark :

ChariJutlaRaoRohatgi, CRYPTO 1999

- Bit x masked $\mapsto x_0, x_1, \ldots, x_d$
- Leakage : $L_i \sim x_i + \mathcal{N}(\mu, \sigma^2)$
- # of leakage samples to test $\big((L_i)_i | x = 0\big) = \big((L_i)_i | x = 1\big)$:

$$q \geq O(1)\sigma^d$$

extended to *continuous leakage* by ProufRivain, EUROCRYPT 2013

DucDziembowskiFaust, to appear EUROCRYPT 2014

Probing Adversary

- Notion introduced in IshaiSahaiWagner, CRYPTO 2003
- A d^{th}-order probing adversary is allowed to observe **at most** d intermediate results during the overall algorithm processing.
  - Hardware interpretation : d is the maximum of wires observed in the circuit.
  - Software interpretation : d is the maximum of different timings during the processing.
- d^{th}-order probing adversary = d^{th}-order SCA as introduced in Messerges99.
- Countermeasures proved to be secure against a d^{th}-order probing adv. :
  - $d = 1, 2$: KocherJaffeJune99, BlömerGuajardoKrummel04, ProuffRivain07, RivainDottaxProuff08.
  - $d \geq 1$: IshaiSahaiWagner03, ProuffRoche11, GenelleProuffQuisquater11, CarletGoubinProuffQuisquaterRivain12, Coron14.

Probing Adversary

- Notion introduced in IshaiSahaiWagner, CRYPTO 2003
- A d^{th}-order probing adversary is allowed to observe **at most** d intermediate results during the overall algorithm processing.
  - Hardware interpretation : d is the maximum of wires observed in the circuit.
  - Software interpretation : d is the maximum of different timings during the processing.
- d^{th}-order probing adversary $=$ d^{th}-order SCA as introduced in Messerges99.
- Countermeasures proved to be secure against a d^{th}-order probing adv. :
  - $d = 1, 2$: KocherJaffeJune99, BlömerGuajardoKrummel04, ProuffRivain07, RivainDottaxProuff08.
  - $d \geq 1$: IshaiSahaiWagner03, ProuffRoche11, GenelleProuffQuisquater11, CarletGoubinProuffQuisquaterRivain12, Coron14.

Higher-Order Masking Schemes

Achieving security in the probing adversary model

Definition

A *dth-order masking scheme* for an encryption algorithm $c \leftarrow \mathcal{E}(m, k)$ is an algorithm

$$(c_0, c_1, \ldots, c_d) \leftarrow \mathcal{E}'\big((m_0, m_1, \ldots, m_d), (k_0, k_1, \ldots, k_d)\big)$$

- Completeness : there exists R s.t. :

$$R(c_0, \cdots, c_d) = \mathcal{E}(m, k)$$

- Security : $\forall\{iv_1, iv_2, \ldots, iv_d\} \subseteq \{\text{intermediate var. of } \mathcal{E}'\}$:

$$\Pr\big(k \mid iv_1, iv_2, \ldots, iv_d\big) = \Pr\big(k\big)$$

State Of The Art

dth-order masking schemes

- Boolean Masking $n = 2d + 1, O(d^2)$
  - [Ishai *et al.* 03] (hardware oriented)
  - $\hookrightarrow$ [Rivain-Prouff 10] [Kim *et al.* 11]
  - [Coron 14 *to appear*] (table re-computation)
- Multiplicative Masking $n = d + 1, O(d^2)$
  - [Genelle *et al.* 11]
  - (alternating Boolean and Multiplicative Masking)
- Polynomial Masking $\widetilde{O}(d^2)$
  - [Prouff-Roche 11] ($n = 2d + 1$, Glitches Resitance)
- Inner-Product Masking $O(d^2)$
  - [Balasch *et al.* 12] ($n = 2(d + 1)$, Glitches Resistance)

State Of The Art

dth-order masking schemes

- Boolean Masking $n = 2d + 1, O(d^2)$
  - [Ishai *et al.* 03] (hardware oriented)
  - $\hookrightarrow$ [Rivain-Prouff 10] [Kim *et al.* 11]
  - [Coron 14 *to appear*] (table re-computation)
- Multiplicative Masking $n = d + 1, O(d^2)$
  - [Genelle *et al.* 11]
  - (alternating Boolean and Multiplicative Masking)
- Polynomial Masking $\widetilde{O}(d^2)$
  - [Prouff-Roche 11] ($n = 2d + 1$, Glitches Resitance)
- Inner-Product Masking $O(d^2)$
  - [Balasch *et al.* 12] ($n = 2(d + 1)$, Glitches Resistance)

Mutual Information Evaluation

Hamming Weight Model and Additive Gaussian Noise

$$\mathcal{O}(Z) = HW(Z) + \mathcal{B}$$

$\mathcal{B} \leftarrow \mathcal{N}(0, \sigma)$

In this idealized model, the success rate of an optimal multi-query (HO-)SCA targeting $(Z_0, \cdots, Z_d)$ is a monotonously increasing function of

$$\mathcal{I}(\mathcal{O}(Z_0), \cdots, \mathcal{O}(Z_d); Z)$$

[Standaert *et al.* 09]

Boolean Sharing

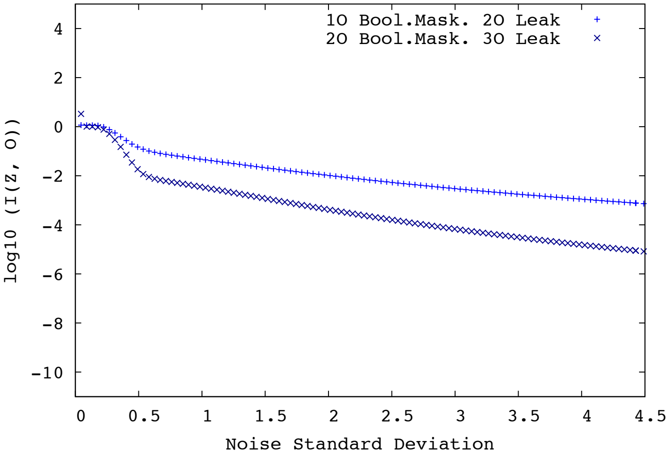
Manipulation of randomized variable

$$z \xrightarrow{\$} (z \oplus r_1 \oplus \cdots \oplus r_d, r_1, \cdots, r_d) \ ,$$

where r_i are randomly generated in $\mathrm{GF}(2^\ell)$.

Information Leaked by a d^{th}-order Boolean Sharing

8-bit variables



IP-masking DziembowskiFaust, TCC 2012

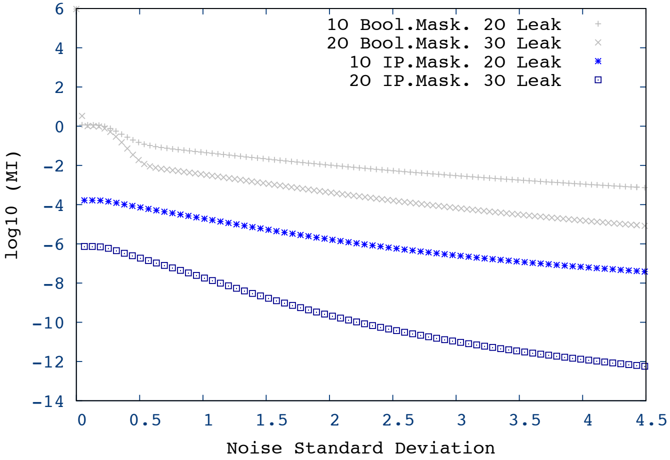
Manipulation of randomized variable

$$z \xrightarrow{\$} \left(L_1, \cdots, L_n, \frac{z \oplus \sum_{i=2}^{n} L_i R_i}{L_1}, R_2, \cdots, R_n\right)$$

where L_i are randomly generated in $\mathrm{GF}(2^\ell)^\star$
and R_i are randomly generated in $\mathrm{GF}(2^\ell)$.

Information Leaked by a d^{th}-order IP sharing

8-bit variables



IP-masking Scheme BalaschFaustGierlichsVerbauwhede, ASIACRYPT 2012

Practical Leakage Resilient Masking Scheme

- $2n$ shares for $(n-1)$ probing security
- (HO-)Glitches Attack resistant masking scheme
- Weak information leakage assuming standard Leakage Functions *e.g. HW*
- Complexity $O(n^2)$
- Proofs in the continuous bounded-range leakage model
  - $\mathcal{O}() : \{0,1\}^\ell \mapsto \{0,1\}^\lambda$ $\lambda << \ell$
  - no limit in the number of observations

IP-masking Scheme BalaschFaustGierlichsVerbauwhede, ASIACRYPT 2012

Practical Leakage Resilient Masking Scheme

- $2n$ shares for $(n-1)$ probing security
- (HO-)Glitches Attack resistant masking scheme
- Weak information leakage assuming standard Leakage Functions *e.g. HW*
- Complexity $O(n^2)$
- Proofs in the continuous bounded-range leakage model only if $n \geq 130$
  - $\mathcal{O}() : \{0,1\}^{\ell} \mapsto \{0,1\}^{\lambda}$ $\lambda << \ell$
  - no limit in the number of observations

IP-masking Scheme BalaschFaustGierlichsVerbauwhede, ASIACRYPT 2012

Inner-Product Sharing Scheme

$$z \xrightarrow{\$} (L_1, \cdots, L_n, \frac{z \oplus \sum_{i=2}^{n} L_i R_i}{L_1}, R_2, \cdots, R_n) = (\mathbf{L}_z, \mathbf{R}_z)$$

R_i in $\mathrm{GF}(2^\ell)$, L_i in $\mathrm{GF}(2^\ell)^\star$.

IP-Masking Scheme

$$\text{inputs} : \{(\mathbf{L}_A, \mathbf{R}_A), (\mathbf{L}_B, \mathbf{R}_B)\}$$

- $\texttt{RefreshMasks}(A)$: $O(n)$
- $A + B$: $O(n)$
- $xA + y$: $O(n)$
- $A \times B$: $O(n^2)$

IP-masking Scheme BalaschFaustGierlichsVerbauwhede, ASIACRYPT 2012

Inner-Product Sharing Scheme

$$z \xrightarrow{\$} (L_1, \cdots, L_n, \frac{z \oplus \sum_{i=2}^{n} L_i R_i}{L_1}, R_2, \cdots, R_n) = (\mathbf{L}_z, \mathbf{R}_z)$$

R_i in $\mathrm{GF}(2^\ell)$, L_i in $\mathrm{GF}(2^\ell)^\star$.

IP-Masking Scheme

$$\text{inputs} : \{(\mathbf{L}_A, \mathbf{R}_A), (\mathbf{L}_B, \mathbf{R}_B)\}$$

- $\texttt{RefreshMasks}(A)$: $O(n)$
- $A + B$: $O(n)$
- $xA + y$: $O(n)$
- $A \times B$: $O(n^2)$

Algorithm RefreshMasks

$\langle \mathbf{L}, \mathbf{R} \rangle$ denotes the scalar product.

Input : the $(2n, d)$-sharing $(\mathbf{L}, \mathbf{R})$ of Z.
Output: the $(2n, d)$-sharing $(\mathbf{L}^\star, \mathbf{R}^\star)$ such that $\langle \mathbf{L}^\star, \mathbf{R}^\star \rangle = \langle \mathbf{L}, \mathbf{R} \rangle$.

```
/* Refresh Masks                                                    */
```

$\mathbf{L}^\star \leftarrow (\texttt{randNonZero()})^n$;
for $i = 1$ **to** n **do**
  $A_i \leftarrow L_i \oplus L_i^\star$;

$X \leftarrow \langle \mathbf{A}, \mathbf{R} \rangle$;
$\mathbf{T} \leftarrow \texttt{IPHalfMask}(X, \mathbf{L}^\star)$;
$\mathbf{R}^\star \leftarrow \mathbf{R} \oplus \mathbf{T}$;
return $(\mathbf{L}^\star, \mathbf{R}^\star)$;

Algorithm RefreshMasks

$\langle \mathbf{L}, \mathbf{R} \rangle$ denotes the scalar product.

Input : the $(2n, d)$-sharing $(\mathbf{L}, \mathbf{R})$ of Z.
Output: the $(2n, d)$-sharing $(\mathbf{L}^\star, \mathbf{R}^\star)$ such that $\langle \mathbf{L}^\star, \mathbf{R}^\star \rangle = \langle \mathbf{L}, \mathbf{R} \rangle$.

```
/* Refresh Masks                                              */
L* ← (randNonZero())^n;
for i = 1 to n do
    A_i ← L_i ⊕ L*_i;
X ← ⟨A, R⟩;
T ← IPHalfMask(X, L*);
R* ← R ⊕ T;
return (L*, R*);
```

Algorithm RefreshMasks

$\langle \mathbf{L}, \mathbf{R} \rangle$ denotes the scalar product.

Input : the $(2n, d)$-sharing $(\mathbf{L}, \mathbf{R})$ of Z.
Output: the $(2n, d)$-sharing $(\mathbf{L}^\star, \mathbf{R}^\star)$ such that $\langle \mathbf{L}^\star, \mathbf{R}^\star \rangle = \langle \mathbf{L}, \mathbf{R} \rangle$.

```
/* Refresh Masks                                  */
L* ← (randNonZero())^n;
for i = 1 to n do
    A_i ← L_i ⊕ L*_i;
X ← ⟨A, R⟩;
T ← IPHalfMask(X, L*);
R* ← R ⊕ T;
return (L*, R*);
```

Algorithm `RefreshMasks`

$\langle \mathbf{L}, \mathbf{R} \rangle$ denotes the scalar product.

Input : the $(2n, d)$-sharing $(\mathbf{L}, \mathbf{R})$ of Z.
Output: the $(2n, d)$-sharing $(\mathbf{L}^\star, \mathbf{R}^\star)$ such that $\langle \mathbf{L}^\star, \mathbf{R}^\star \rangle = \langle \mathbf{L}, \mathbf{R} \rangle$.

```
/* Refresh Masks                                          */
L* ← (randNonZero())^n;
for i = 1 to n do
    A_i ← L_i ⊕ L*_i;
X ← ⟨A, R⟩;
T ← IPHalfMask(X, L*);
R* ← R ⊕ T;
return (L*, R*);
```

For $n = 2$,

$$
\begin{aligned}
Z &= L_1 R_1 \oplus L_2 R_2 \\
X &= (L_1 \oplus L_1^\star) R_1 \oplus (L_2 \oplus L_2^\star) R_2
\end{aligned}
$$

Algorithm RefreshMasks

$\langle \mathbf{L}, \mathbf{R} \rangle$ denotes the scalar product.

Input : the $(2n, d)$-sharing $(\mathbf{L}, \mathbf{R})$ of Z.
Output: the $(2n, d)$-sharing $(\mathbf{L}^\star, \mathbf{R}^\star)$ such that $\langle \mathbf{L}^\star, \mathbf{R}^\star \rangle = \langle \mathbf{L}, \mathbf{R} \rangle$.

/* Refresh Masks */
$\mathbf{L}^\star \leftarrow (\texttt{randNonZero()})^n$;
for $i = 1$ **to** n **do**
  $A_i \leftarrow L_i \oplus L_i^\star$;
$X \leftarrow \langle \mathbf{A}, \mathbf{R} \rangle$;
$\mathbf{T} \leftarrow \texttt{IPHalfMask}(X, \mathbf{L}^\star)$;
$\mathbf{R}^\star \leftarrow \mathbf{R} \oplus \mathbf{T}$;
return $(\mathbf{L}^\star, \mathbf{R}^\star)$;

For $n = 2$,

$$\begin{aligned} Z &= L_1 R_1 \oplus L_2 R_2 \\ X &= (L_1 \oplus L_1^\star) R_1 \oplus (L_2 \oplus L_2^\star) R_2 \end{aligned}$$

Algorithm RefreshMasks

$\langle \mathbf{L}, \mathbf{R} \rangle$ denotes the scalar product.

Input : the $(2n, d)$-sharing $(\mathbf{L}, \mathbf{R})$ of Z.
Output: the $(2n, d)$-sharing $(\mathbf{L}^\star, \mathbf{R}^\star)$ such that $\langle \mathbf{L}^\star, \mathbf{R}^\star \rangle = \langle \mathbf{L}, \mathbf{R} \rangle$.

/* Refresh Masks */
$\mathbf{L}^\star \leftarrow (\texttt{randNonZero()})^n$;
for $i = 1$ **to** n **do**
  $A_i \leftarrow L_i \oplus L_i^\star$;
$X \leftarrow \langle \mathbf{A}, \mathbf{R} \rangle$;
$\mathbf{T} \leftarrow \texttt{IPHalfMask}(X, \mathbf{L}^\star)$;
$\mathbf{R}^\star \leftarrow \mathbf{R} \oplus \mathbf{T}$;
return $(\mathbf{L}^\star, \mathbf{R}^\star)$;

For $n = 2$,

$$\begin{aligned} Z &= L_1 R_1 \oplus L_2 R_2 \\ X &= (L_1 \oplus L_1^\star) R_1 \oplus (L_2 \oplus L_2^\star) R_2 \end{aligned}$$

A 1^{st}-order Flaw for any d

$$\Pr[X = x \mid Z = 0] = \begin{cases} \frac{1}{2^\ell} + \frac{1}{2^\ell(2^\ell - 1)^{n-2}} & \text{if } x = 0 \\ \frac{1}{2^\ell} - \frac{1}{2^\ell(2^\ell - 1)^{n-1}} & \text{if } x \neq 0 \end{cases}$$

and

$$\Pr[X = x \mid Z = z] = \begin{cases} \frac{1}{2^\ell} - \frac{1}{2^\ell(2^\ell - 1)^{n-1}} & \text{if } x = z \\ \frac{1}{2^\ell} + \frac{1}{2^\ell(2^\ell - 1)^{n}} & \text{if } x \neq z \end{cases},$$

if $z \neq 0$.

A 1^{st}-order Flaw for any d

$$\Pr[X = x \mid Z = 0] = \begin{cases} \frac{1}{2^\ell} + \frac{1}{2^\ell(2^\ell-1)^{n-2}} & \text{if } x = 0 \\ \frac{1}{2^\ell} - \frac{1}{2^\ell(2^\ell-1)^{n-1}} & \text{if } x \neq 0 \end{cases}$$

and

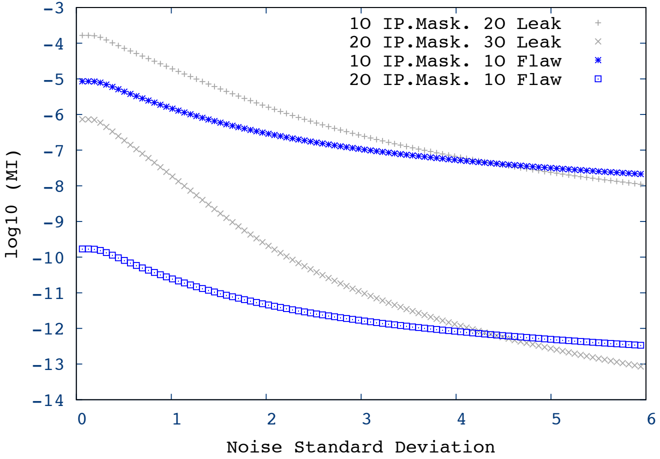
$$\Pr[X = x \mid Z = z] = \begin{cases} \frac{1}{2^\ell} - \frac{1}{2^\ell(2^\ell-1)^{n-1}} & \text{if } x = z \\ \frac{1}{2^\ell} + \frac{1}{2^\ell(2^\ell-1)^{n}} & \text{if } x \neq z \end{cases},$$

if $z \neq 0$.

$$\mathcal{I}(\mathcal{O}(X); Z) \neq 0$$

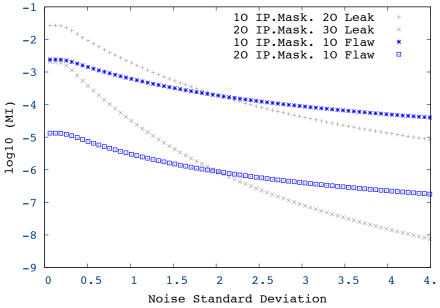
Information Leaked by the 1st-order Flaw

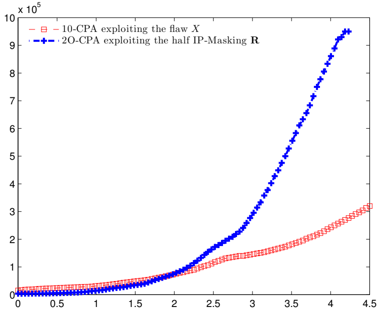
8-bit variables



Information Leaked by the 1st-order Flaw

4-bit variables





A security flaw in Balasch *et al.* scheme

- 1st-order flaw (exponential decay *w.r.t.* the mask order)
  - ↪ in practice much easier to mount than a dth-order attack.
  - ↪ noise addition techniques won't help that much.
- proof in the continuous bounded-range leakage model is still standing
  - ↪ ways of improving the $n \geq 130$ bound ?

A security flaw in Balasch *et al.* scheme

- 1st-order flaw (exponential decay *w.r.t.* the mask order)
  - ↪ in practice much easier to mount than a dth-order attack.
  - ↪ noise addition techniques won't help that much.
- proof in the continuous bounded-range leakage model is still standing
  - ↪ ways of improving the $n \geq 130$ bound ?

A security flaw in Balasch *et al.* scheme

- 1st-order flaw (exponential decay *w.r.t.* the mask order)
  - $\hookrightarrow$ in practice much easier to mount than a dth-order attack.
  - $\hookrightarrow$ noise addition techniques won't help that much.
- proof in the continuous bounded-range leakage model is still standing
  - $\hookrightarrow$ ways of improving the $n \geq 130$ bound ?

A security flaw in Balasch *et al.* scheme

- 1st-order flaw (exponential decay *w.r.t.* the mask order)
  - $\hookrightarrow$ in practice much easier to mount than a dth-order attack.
  - $\hookrightarrow$ noise addition techniques won't help that much.
- proof in the continuous bounded-range leakage model is still standing
  - $\hookrightarrow$ ways of improving the $n \geq 130$ bound ?

IP-Masking Scheme *w.r.t.* to recent results in leakage resilience proofs

- ProufRivain, EUROCRYPT 2013
- security proofs in continuous leakage model

  practical noisy leakage models
- Boolean masking (Ishai *et al.* scheme)
- improvements and link with probing security

  DucDziembowskiFaust, to appear EUROCRYPT 2014

THE MYTH OF GENERIC DPA... AND THE MAGIC OF LEARNING

Carolyn Whitnall[1], Elisabeth Oswald[1], François-Xavier Standaert[2]

[1]Department of Computer Science, University of Bristol
[2]UCL Crypto Group, Université catholique de Louvain

carolyn.whitnall@bris.ac.uk

26th February 2014

Outline

The 'myth'…

- What is 'generic' DPA? – rethinking the role of the power model
- Does 'generic' DPA work? – only in special cases, it turns out

The 'magic'…

- Where do we go from here? – linear regression-based methods as an interesting avenue for generic-emulating DPA
- Does our proposed technique work? – some experimental results

Outline

The 'myth'…

- What is 'generic' DPA? – rethinking the role of the power model
- Does 'generic' DPA work? – only in special cases, it turns out

The 'magic'…

- Where do we go from here? – linear regression-based methods as an interesting avenue for generic-emulating DPA
- Does our proposed technique work? – some experimental results

What is 'generic' DPA?

Intuitive idea

A strategy to exploit the data-dependent leakage of a device without any prior knowledge of the functional form of that leakage.

Typical approach

Use distinguishing statistics which require *few distributional assumptions*:

- Mutual information [Gierlichs et al. CHES '08];
- Kolmogorov–Smirnov test statistic [Veyrat-Charvillon et al. CHES '09];
- Cramér–von Mises [Veyrat-Charvillon et al. CHES '09];
- Copulas [Veyrat-Charvillon et al. CRYPTO '11] ...

But this approach does *not* automatically constitute'generic' DPA:

- Often paired with a power model such as Hamming weight;
- Use of 'arbitrary' power models (e.g. 7 LSB) only works if a reasonable leakage approximation is 'accidentally' achieved [Whitnall et al. JCEN '11].

What is 'generic' DPA?

Intuitive idea

A strategy to exploit the data-dependent leakage of a device without any prior knowledge of the functional form of that leakage.

Typical approach

Use distinguishing statistics which require *few distributional assumptions*:

- Mutual information [Gierlichs et al. CHES '08];
- Kolmogorov–Smirnov test statistic [Veyrat-Charvillon et al. CHES '09];
- Cramér–von Mises [Veyrat-Charvillon et al. CHES '09];
- Copulas [Veyrat-Charvillon et al. CRYPTO '11] ...

But this approach does *not* automatically constitute'generic' DPA:

- Often paired with a power model such as Hamming weight;
- Use of 'arbitrary' power models (e.g. 7 LSB) only works if a reasonable leakage approximation is 'accidentally' achieved [Whitnall et al. JCEN '11].

What is 'generic' DPA?

Intuitive idea

A strategy to exploit the data-dependent leakage of a device without any prior knowledge of the functional form of that leakage.

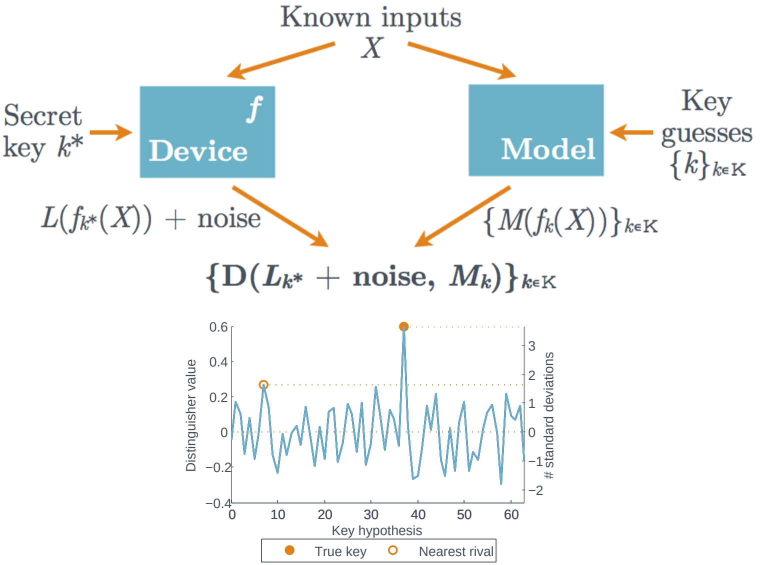
Typical approach

Use distinguishing statistics which require *few distributional assumptions*:

- Mutual information [Gierlichs et al. CHES '08];
- Kolmogorov–Smirnov test statistic [Veyrat-Charvillon et al. CHES '09];
- Cramér–von Mises [Veyrat-Charvillon et al. CHES '09];
- Copulas [Veyrat-Charvillon et al. CRYPTO '11] . . .

But this approach does *not* automatically constitute'generic' DPA:

- Often paired with a power model such as Hamming weight;
- Use of 'arbitrary' power models (e.g. 7 LSB) only works if a reasonable leakage approximation is 'accidentally' achieved [Whitnall et al. JCEN '11].



Known inputs
X
Secret key k*
f
Device
Model
Key guesses {k}k∈K
L(fk*(X)) + noise
{M(fk(X))}k∈K
{D(Lk* + noise, Mk)}k∈K
0.6
0.4
0.2
0
−0.2
−0.4
Distinguisher value
3
2
1
0
−1
−2
standard deviations
0
10
20
30
40
50
60
Key hypothesis
True key
Nearest rival

What is 'generic' DPA?

Determined by the power model, not the distinguishing statistic!

Classifying power models according to Stevens' levels of measurement

Suppose M is the power model for leakage function L. Then...

- Direct approximation $M \approx L$ (c.f. the 'ratio scale'), as exploited by profiled attacks (e.g. Bayesian templates and stochastic profiling).
- Proportional approximation $M \approx \alpha L$ (c.f. the 'interval scale'). Suitable for use with (e.g.) Pearson's correlation coefficient.
- Ordinal approximation $\{z|M(z) < M(z')\} \approx \{z|L(z) < L(z')\} \; \forall z' \in \mathcal{Z}$ (c.f. the 'ordinal scale'). Suitable for use with (e.g.) Spearman's rank correlation coefficient.
- Nominal approximation $\{z|M(z) = M(z')\} \approx \{z|L(z) = L(z')\} \; \forall z' \in \mathcal{Z}$ (c.f. the 'nominal scale'). Appropriate statistics correspond to the 'partition-based' distinguishers of Standaert et al. (ISISC '08), e.g. MI.

What is 'generic' DPA?

Determined by the power model, not the distinguishing statistic!

Classifying power models according to Stevens' levels of measurement

Suppose M is the power model for leakage function L. Then...

- Direct approximation $M \approx L$ (c.f. the 'ratio scale'), as exploited by profiled attacks (e.g. Bayesian templates and stochastic profiling).
- Proportional approximation $M \approx \alpha L$ (c.f. the 'interval scale'). Suitable for use with (e.g.) Pearson's correlation coefficient.
- Ordinal approximation $\{z|M(z) < M(z')\} \approx \{z|L(z) < L(z')\} \; \forall z' \in \mathcal{Z}$ (c.f. the 'ordinal scale'). Suitable for use with (e.g.) Spearman's rank correlation coefficient.
- Nominal approximation $\{z|M(z) = M(z')\} \approx \{z|L(z) = L(z')\} \; \forall z' \in \mathcal{Z}$ (c.f. the 'nominal scale'). Appropriate statistics correspond to the 'partition-based' distinguishers of Standaert et al. (ISISC '08), e.g. MI.

What is 'generic' DPA?

Determined by the power model, not the distinguishing statistic!

Classifying power models according to Stevens' levels of measurement

Suppose M is the power model for leakage function L. Then…

- Direct approximation $M \approx L$ (c.f. the 'ratio scale'), as exploited by profiled attacks (e.g. Bayesian templates and stochastic profiling).
- Proportional approximation $M \approx \alpha L$ (c.f. the 'interval scale'). Suitable for use with (e.g.) Pearson's correlation coefficient.
- Ordinal approximation $\{z|M(z) < M(z')\} \approx \{z|L(z) < L(z')\}\ \forall z' \in \mathcal{Z}$ (c.f. the 'ordinal scale'). Suitable for use with (e.g.) Spearman's rank correlation coefficient.
- Nominal approximation $\{z|M(z) = M(z')\} \approx \{z|L(z) = L(z')\}\ \forall z' \in \mathcal{Z}$ (c.f. the 'nominal scale'). Appropriate statistics correspond to the 'partition-based' distinguishers of Standaert et al. (ISISC '08), e.g. MI.

What is 'generic' DPA?

Determined by the power model, not the distinguishing statistic!

Classifying power models according to Stevens' levels of measurement

Suppose M is the power model for leakage function L. Then...

- Direct approximation $M \approx L$ (c.f. the 'ratio scale'), as exploited by profiled attacks (e.g. Bayesian templates and stochastic profiling).
- Proportional approximation $M \approx \alpha L$ (c.f. the 'interval scale'). Suitable for use with (e.g.) Pearson's correlation coefficient.
- Ordinal approximation $\{z|M(z) < M(z')\} \approx \{z|L(z) < L(z')\}\ \forall z' \in \mathcal{Z}$ (c.f. the 'ordinal scale'). Suitable for use with (e.g.) Spearman's rank correlation coefficient.
- Nominal approximation $\{z|M(z) = M(z')\} \approx \{z|L(z) = L(z')\}\ \forall z' \in \mathcal{Z}$ (c.f. the 'nominal scale'). Appropriate statistics correspond to the 'partition-based' distinguishers of Standaert et al. (ISISC '08), e.g. MI.

WHAT IS 'GENERIC' DPA?

Determined by the power model, not the distinguishing statistic!

CLASSIFYING POWER MODELS ACCORDING TO STEVENS' LEVELS OF MEASUREMENT

Suppose M is the power model for leakage function L. Then...

- Direct approximation $M \approx L$ (c.f. the 'ratio scale'), as exploited by profiled attacks (e.g. Bayesian templates and stochastic profiling).
- Proportional approximation $M \approx \alpha L$ (c.f. the 'interval scale'). Suitable for use with (e.g.) Pearson's correlation coefficient.
- Ordinal approximation $\{z|M(z) < M(z')\} \approx \{z|L(z) < L(z')\} \; \forall z' \in \mathcal{Z}$ (c.f. the 'ordinal scale'). Suitable for use with (e.g.) Spearman's rank correlation coefficient.
- Nominal approximation $\{z|M(z) = M(z')\} \approx \{z|L(z) = L(z')\} \; \forall z' \in \mathcal{Z}$ (c.f. the 'nominal scale'). Appropriate statistics correspond to the 'partition-based' distinguishers of Standaert et al. (ISISC '08), e.g. MI.

What is 'generic' DPA?

Determined by the power model, not the distinguishing statistic!

Classifying power models according to Stevens' levels of measurement

Suppose M is the power model for leakage function L. Then...

- Direct approximation $M \approx L$ (c.f. the 'ratio scale'), as exploited by profiled attacks (e.g. Bayesian templates and stochastic profiling).
- Proportional approximation $M \approx \alpha L$ (c.f. the 'interval scale'). Suitable for use with (e.g.) Pearson's correlation coefficient.
- Ordinal approximation $\{z|M(z) < M(z')\} \approx \{z|L(z) < L(z')\}\ \forall z' \in \mathcal{Z}$ (c.f. the 'ordinal scale'). Suitable for use with (e.g.) Spearman's rank correlation coefficient.
- Nominal approximation $\{z|M(z) = M(z')\} \approx \{z|L(z) = L(z')\}\ \forall z' \in \mathcal{Z}$ (c.f. the 'nominal scale'). Appropriate statistics correspond to the 'partition-based' distinguishers of Standaert et al. (ISISC '08), e.g. MI.

(STANDARD, UNIVARIATE) GENERIC DPA STRATEGY

=

GENERIC POWER MODEL

The nominal mapping to the equivalence classes induced by the target function F_k.

+

GENERIC-COMPATIBLE DISTINGUISHER

Any distinguishing statistic which operates on nominal scale measurements.

Does 'generic' DPA work?

A strategy 'works' (given enough data and a compatible distinguisher) if the power model approximation under the correct hypothesis is *strictly more accurate* than the approximation under any incorrect alternative.

1. For F injective: generic power model predictions under all hypotheses are *equally accurate*—no generic strategy works.
2. For F balanced and non-injective; k introduced by (XOR) key addition:
   1. If F is *affine* then no generic strategy is able to distinguish the correct key from any other.
   2. If $a \in \mathbb{F}_2^n$ is a linear structure of F then no generic strategy is able to distinguish between k^* and $k^* \oplus a$.
   3. If, for some $a \in \mathbb{F}_2^n$ we have that $D_aF(x)$ (the differential of F wrt a) depends on x *only via $F(x)$*, then no generic strategy is able to distinguish between k^* and $k^* \oplus a$

Scenarios 1 and 2.1 produce flat distinguishing vectors; 2.2 and 2.3 produce ghost peaks.

Does 'generic' DPA work?

A strategy 'works' (given enough data and a compatible distinguisher) if the power model approximation under the correct hypothesis is *strictly more accurate* than the approximation under any incorrect alternative.

1. For F injective: generic power model predictions under all hypotheses are *equally accurate*—no generic strategy works.
2. For F balanced and non-injective; k introduced by (XOR) key addition:
   1. If F is *affine* then no generic strategy is able to distinguish the correct key from any other.
   2. If $a \in \mathbb{F}_2^n$ is a linear structure of F then no generic strategy is able to distinguish between k^* and $k^* \oplus a$.
   3. If, for some $a \in \mathbb{F}_2^n$ we have that $D_aF(x)$ (the differential of F wrt a) depends on x *only via $F(x)$*, then no generic strategy is able to distinguish between k^* and $k^* \oplus a$

Scenarios 1 and 2.1 produce flat distinguishing vectors; 2.2 and 2.3 produce ghost peaks.

Does 'generic' DPA work?

A strategy 'works' (given enough data and a compatible distinguisher) if the power model approximation under the correct hypothesis is *strictly more accurate* than the approximation under any incorrect alternative.

1. For F injective: generic power model predictions under all hypotheses are *equally accurate*—no generic strategy works.
2. For F balanced and non-injective; k introduced by (XOR) key addition:
   1. If F is *affine* then no generic strategy is able to distinguish the correct key from any other.
   2. If $a \in \mathbb{F}_2^n$ is a linear structure of F then no generic strategy is able to distinguish between k^* and $k^* \oplus a$.
   3. If, for some $a \in \mathbb{F}_2^n$ we have that $D_aF(x)$ (the differential of F wrt a) depends on x *only via $F(x)$*, then no generic strategy is able to distinguish between k^* and $k^* \oplus a$

Scenarios 1 and 2.1 produce flat distinguishing vectors; 2.2 and 2.3 produce ghost peaks.

Does 'generic' DPA work?

A strategy 'works' (given enough data and a compatible distinguisher) if the power model approximation under the correct hypothesis is *strictly more accurate* than the approximation under any incorrect alternative.

1. For F injective: generic power model predictions under all hypotheses are *equally accurate*—no generic strategy works.
2. For F balanced and non-injective; k introduced by (XOR) key addition:
   1. If F is *affine* then no generic strategy is able to distinguish the correct key from any other.
   2. If $a \in \mathbb{F}_2^n$ is a linear structure of F then no generic strategy is able to distinguish between k^* and $k^* \oplus a$.
   3. If, for some $a \in \mathbb{F}_2^n$ we have that $D_aF(x)$ (the differential of F wrt a) depends on x *only via $F(x)$*, then no generic strategy is able to distinguish between k^* and $k^* \oplus a$

Scenarios 1 and 2.1 produce flat distinguishing vectors; 2.2 and 2.3 produce ghost peaks.

Does 'generic' DPA work?

A strategy 'works' (given enough data and a compatible distinguisher) if the power model approximation under the correct hypothesis is *strictly more accurate* than the approximation under any incorrect alternative.

1. For F injective: generic power model predictions under all hypotheses are *equally accurate*—no generic strategy works.
2. For F balanced and non-injective; k introduced by (XOR) key addition:
   1. If F is *affine* then no generic strategy is able to distinguish the correct key from any other.
   2. If $a \in \mathbb{F}_2^n$ is a linear structure of F then no generic strategy is able to distinguish between k^* and $k^* \oplus a$.
   3. If, for some $a \in \mathbb{F}_2^n$ we have that $D_aF(x)$ (the differential of F wrt a) depends on x *only via $F(x)$*, then no generic strategy is able to distinguish between k^* and $k^* \oplus a$

Scenarios 1 and 2.1 produce flat distinguishing vectors; 2.2 and 2.3 produce ghost peaks.

Does 'generic' DPA work?

Suppose F is a balanced, noninjective (n-m) function, with k introduced by (XOR) key-addition.

A *necessary condition* for a generic strategy to distinguish k^* from k is: $\exists x \in \mathbb{F}_2^n$ such that $\# D_{k^* \oplus k} F(F^{-1}[F(x)]) \neq 1$.

If L is injective then this becomes a *sufficient condition*.

S-box design goals of differential uniformity increase the chances of this condition being met for a given XOR difference from the correct key.

CRYPTANALYTIC RESILIENCE $\stackrel{\sim}{\Longleftrightarrow}$ SIDE-CHANNEL VULNERABILITY

OBSERVATION: Leakage function $L : \mathbb{F}_2^m \to \mathbb{R}$ can be expressed as a polynomial in function of the target bits.

- $L(z) = \sum_{u \in \mathbb{F}_2^m} \alpha_u z^u, \ \forall z \in \mathbb{F}_2^m$, where z^u denotes the monomial $\prod_{i=1}^m z_i^{u_i}$, with z_i the i^{th} bit of z.

ATTACK STRATEGY: Using prior knowledge about the contributing terms, estimate the model according to each key guess and pick the one which produces the 'best fit'.

- $\forall k \in \mathcal{K}$ compute the OLS coefficients for $L_{k^*}(X) + \varepsilon = \alpha_{\mathbf{0}} + \sum_{u \in \mathcal{U}} F_k(X)^u \alpha_u$, where $\mathcal{U} \subseteq \mathbb{F}_2^m \setminus \{\mathbf{0}\}$.
- If the R^2 'goodness-of-fit' measure is largest under the correct key guess then the attack has succeeded.

Linear regression-based DPA

OBSERVATION: Leakage function $L : \mathbb{F}_2^m \to \mathbb{R}$ can be expressed as a polynomial in function of the target bits.

- $L(z) = \sum_{u \in \mathbb{F}_2^m} \alpha_u z^u, \ \forall z \in \mathbb{F}_2^m$, where z^u denotes the monomial $\prod_{i=1}^m z_i^{u_i}$, with z_i the i^{th} bit of z.

ATTACK STRATEGY: Using prior knowledge about the contributing terms, estimate the model according to each key guess and pick the one which produces the 'best fit'.

- $\forall k \in \mathcal{K}$ compute the OLS coefficients for $L_{k^*}(X) + \varepsilon = \alpha_{\mathbf{0}} + \sum_{u \in \mathcal{U}} F_k(X)^u \alpha_u$, where $\mathcal{U} \subseteq \mathbb{F}_2^m \setminus \{\mathbf{0}\}$.
- If the R^2 'goodness-of-fit' measure is largest under the correct key guess then the attack has succeeded.

LR-based DPA as a 'generic strategy'

Including all polynomial terms (i.e. selecting $\mathcal{U} = \mathbb{F}_2^m \setminus \{\mathbf{0}\}$) equates to a 'generic strategy' (see paper).

Case 1 – noninjective (cryptographic) target: System of equations is over-determined and…

- Consistent (bar noise) under the correct guess $\longrightarrow$ good model fit;
- Inconsistent under any incorrect guess $\longrightarrow$ poor model fit.

I.e. the true key is distinguished.

Case 2 – injective target: Full-degree model is equally adequate to describe the leakage under any hypothesis…

- Goodness-of-fit scores produce a flat distinguishing vector, *but*
- Procedure returns additional information which may be exploited…

LR-based DPA as a 'generic strategy'

Including all polynomial terms (i.e. selecting $\mathcal{U} = \mathbb{F}_2^m \setminus \{\mathbf{0}\}$) equates to a 'generic strategy' (see paper).

Case 1 – noninjective (cryptographic) target: System of equations is over-determined and…

- Consistent (bar noise) under the correct guess $\longrightarrow$ good model fit;
- Inconsistent under any incorrect guess $\longrightarrow$ poor model fit.

I.e. the true key is distinguished.

Case 2 – injective target: Full-degree model is equally adequate to describe the leakage under any hypothesis…

- Goodness-of-fit scores produce a flat distinguishing vector, *but*
- Procedure returns additional information which may be exploited…

LR-based DPA as a 'generic strategy'

Including all polynomial terms (i.e. selecting $\mathcal{U} = \mathbb{F}_2^m \setminus \{\mathbf{0}\}$) equates to a 'generic strategy' (see paper).

Case 1 – noninjective (cryptographic) target: System of equations is over-determined and…

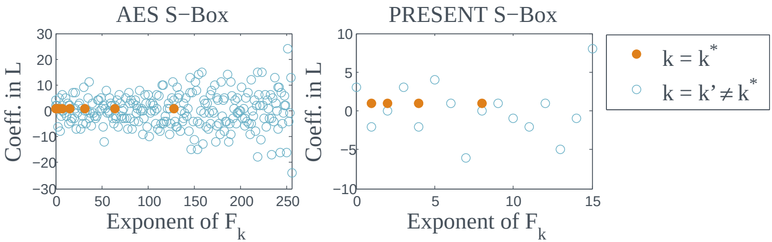
- Consistent (bar noise) under the correct guess $\longrightarrow$ good model fit;
- Inconsistent under any incorrect guess $\longrightarrow$ poor model fit.

I.e. the true key is distinguished.

Case 2 – injective target: Full-degree model is equally adequate to describe the leakage under any hypothesis…

- Goodness-of-fit scores produce a flat distinguishing vector, *but*
- Procedure returns additional information which may be exploited…

Coefficients from fitted LR models



- Under the correct key guess, coefficients on the fitted terms represent an expression for the leakage function L.
- Under an incorrect guess, the coefficients represent an expression for $L \circ f_k \circ f_{k^*}^{-1}$ – highly nonlinear by design of f.
- Assuming L is always 'simpler' than $L \circ f_k \circ f_{k^*}^{-1}$ this suggests a differentiating criteria.

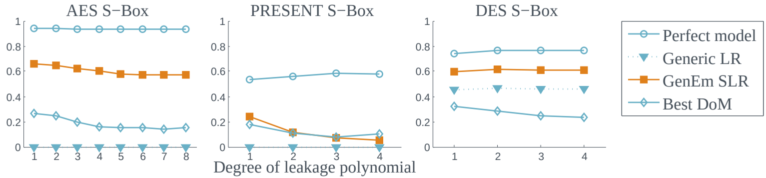
Stepwise regression

- Model building tool to ‘learn’ the correct model specification.
  - Iteratively adds and removes potential explanatory variables.
  - Favours variables with the most explanatory power.
- Our proposal: Provide the stepwise algorithm with the full set of polynomial terms $\mathcal{U} = \mathbb{F}_2^m$ and let it choose which to privilege.
  - Under incorrect guess, the explanatory power of the model terms is highly dispersed – contribution of any individual term decreases.
  - If there is sufficient loss in excluding these small contributions then we may be able to distinguish the correct key according to the resulting R^2 values.

Stepwise regression

- Model building tool to 'learn' the correct model specification.
    - Iteratively adds and removes potential explanatory variables.
    - Favours variables with the most explanatory power.
- Our proposal: Provide the stepwise algorithm with the full set of polynomial terms $\mathcal{U} = \mathbb{F}_2^m$ and let it choose which to privilege.
    - Under incorrect guess, the explanatory power of the model terms is highly dispersed – contribution of any individual term decreases.
    - If there is sufficient loss in excluding these small contributions then we may be able to distinguish the correct key according to the resulting R^2 values.

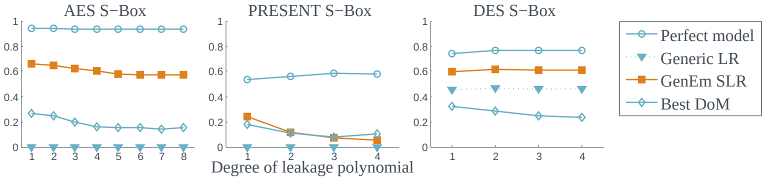
Does stepwise regression work?



Median asymptotic distinguishing margins for 500 randomly generated leakage functions as leakage degree increases…

- Stepwise regression is effective against all three targets, even for high degree leakage.
- Stepwise regression succeeds in the scenarios where 'generic' linear regression DPA fails, and achieves larger margins against the (noninjective) DES S-box.
- Stepwise regression improves on, or at least rivals, the 'best' difference-of-means (when all possible bits are considered).

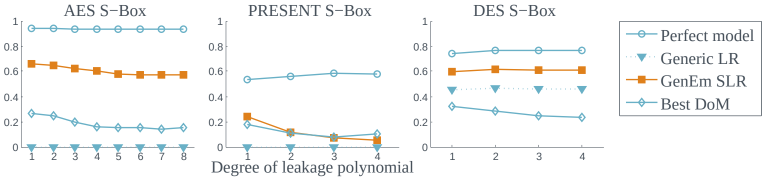
Does stepwise regression work?



Median asymptotic distinguishing margins for 500 randomly generated leakage functions as leakage degree increases…

- Stepwise regression is effective against all three targets, even for high degree leakage.
- Stepwise regression succeeds in the scenarios where ‘generic’ linear regression DPA fails, and achieves larger margins against the (noninjective) DES S-box.
- Stepwise regression improves on, or at least rivals, the ‘best’ difference-of-means (when all possible bits are considered).

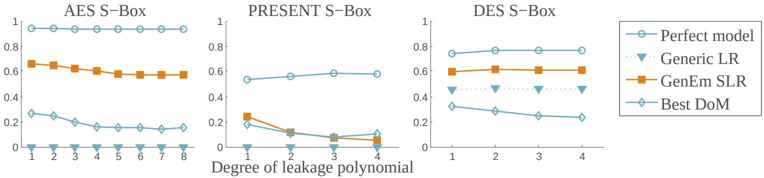
Does stepwise regression work?



Median asymptotic distinguishing margins for 500 randomly generated leakage functions as leakage degree increases…

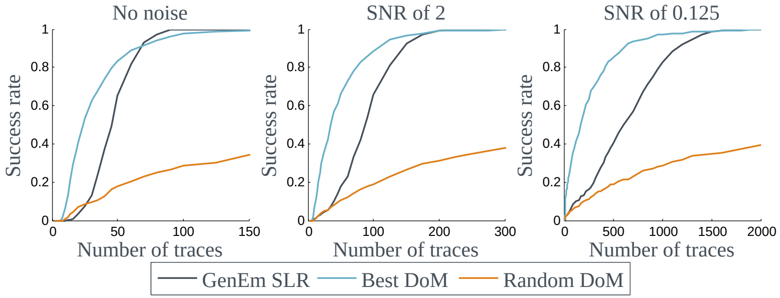
- Stepwise regression is effective against all three targets, even for high degree leakage.
- Stepwise regression succeeds in the scenarios where 'generic' linear regression DPA fails, and achieves larger margins against the (noninjective) DES S-box.
- Stepwise regression improves on, or at least rivals, the 'best' difference-of-means (when all possible bits are considered).

DOES STEPWISE REGRESSION WORK?



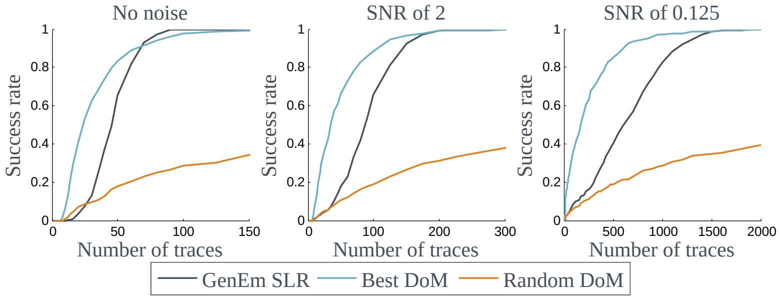
Median asymptotic distinguishing margins for 500 randomly generated leakage functions as leakage degree increases. . .

- Stepwise regression is effective against all three targets, even for high degree leakage.
- Stepwise regression succeeds in the scenarios where 'generic' linear regression DPA fails, and achieves larger margins against the (noninjective) DES S-box.
- Stepwise regression improves on, or at least rivals, the 'best' difference-of-means (when all possible bits are considered).



Success rates against high degree leakage of the AES S-box...

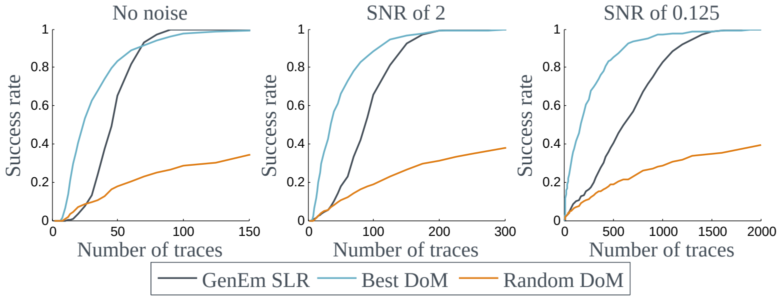
- Much higher success rates than DoM against a randomly selected bit.
- Lower success rates than the strongest DoM out of all 8 possible bits.
- SLR exploits the leaked information more comprehensively than DoM, but carries hefty estimation overheads:
  - SLR – up to 2^8 unknown coefficients to estimate per key hypothesis;
  - DoM – two means to estimate per key hypothesis.



Success rates against high degree leakage of the AES S-box...

- Much higher success rates than DoM against a randomly selected bit.
- Lower success rates than the strongest DoM out of all 8 possible bits.
- SLR exploits the leaked information more comprehensively than DoM, but carries hefty estimation overheads:
  - SLR – up to 2^8 unknown coefficients to estimate per key hypothesis;
  - DoM – two means to estimate per key hypothesis.

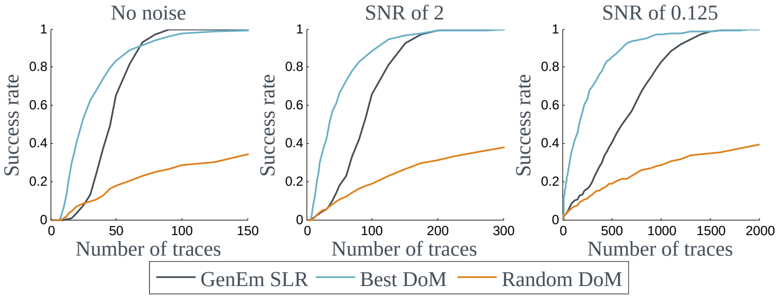
Comparison with DoM attacks



Success rates against high degree leakage of the AES S-box...

- Much higher success rates than DoM against a randomly selected bit.
- Lower success rates than the strongest DoM out of all 8 possible bits.
- SLR exploits the leaked information more comprehensively than DoM, but carries hefty estimation overheads:
  - SLR – up to 2^8 unknown coefficients to estimate per key hypothesis;
  - DoM – two means to estimate per key hypothesis.

Comparison with DoM attacks



Success rates against high degree leakage of the AES S-box...

- Much higher success rates than DoM against a randomly selected bit.
- Lower success rates than the strongest DoM out of all 8 possible bits.
- SLR exploits the leaked information more comprehensively than DoM, but carries hefty estimation overheads:
  - SLR – up to 2^8 unknown coefficients to estimate per key hypothesis;
  - DoM – two means to estimate per key hypothesis.

Conclusion

- The notion of 'generic DPA' should follow from the properties of the *power model* used.
- Such a definition facilitates conclusive statements about attack outcomes independent of the distinguishing statistic chosen.
  - Generic strategies *can* succeed against noninjective cryptographic functions.
  - They invariably fail against injective targets – no universally-applicable attacks exist.
- *'Generic-emulating'* DPA, relying only on 'non-device-specific intuition', can succeed against injective targets.
  - E.g. stepwise linear regression – rivals difference-of-means but is more costly to estimate.
  - Can we find other methodologies achieving a similar end? (...more efficiently?)

CONCLUSION

- The notion of 'generic DPA' should follow from the properties of the *power model* used.
- Such a definition facilitates conclusive statements about attack outcomes independent of the distinguishing statistic chosen.
  - Generic strategies *can* succeed against noninjective cryptographic functions.
  - They invariably fail against injective targets – no universally-applicable attacks exist.
- *'Generic-emulating'* DPA, relying only on 'non-device-specific intuition', can succeed against injective targets.
  - E.g. stepwise linear regression – rivals difference-of-means but is more costly to estimate.
  - Can we find other methodologies achieving a similar end? (…more efficiently?)

Conclusion

- The notion of ‘generic DPA’ should follow from the properties of the *power model* used.
- Such a definition facilitates conclusive statements about attack outcomes independent of the distinguishing statistic chosen.
  - Generic strategies *can* succeed against noninjective cryptographic functions.
  - They invariably fail against injective targets – no universally-applicable attacks exist.
- *‘Generic-emulating’* DPA, relying only on ‘non-device-specific intuition’, can succeed against injective targets.
  - E.g. stepwise linear regression – rivals difference-of-means but is more costly to estimate.
  - Can we find other methodologies achieving a similar end? (. . . more efficiently?)

Thank you for listening!

Any questions?

RSA CONFERENCE 2014
FEBRUARY 24 – 28 | MOSCONE CENTER | SAN FRANCISCO

Share.
Learn.
Secure.
Capitalizing on
Collective Intelligence

Hardware Implementation and Side-Channel Analysis of Lapin

SESSION ID: CRYP-W02

Lubos Gaspar[1], Gaëtan Leurent[1,2], François-Xavier Standaert[1]

[1] Crypto group, Université catholique de Louvain, Louvain-la-Neuve, Belgium
[2] Inria, EPI SECRET, Rocquencourt, France

lubos.gaspar@uclouvain.be, gaetan.leurent@inria.fr, fstandae@uclouvain.be

Riddle:

Do you know what does Lapin mean?

Do you know what does Lapin mean?

In French: Lapin = Rabbit

Do you know what does Lapin mean?

OR?

Learning **P**arity with **N**oise

L a P i N

Do you know what does Lapin mean?

OR?

Learning **P**arity with **N**oise

L a P i N

With something **random** in between

Outline

- Introduction
- Lapin protocol
- Implementation
- Performance evaluation
- Side-channel analysis
- Conclusion

Introduction to Lapin

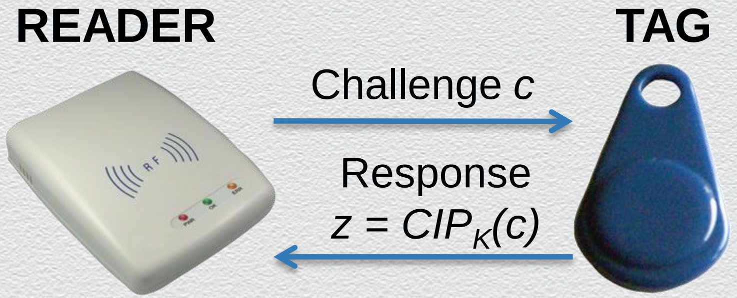
Light-weight Shared-key Authentication Protocols

- Lightweight shared-key authentication protocols are widely used

Example – wireless tags

Light-weight Shared-key Authentication Protocols

- Typical settings:
  1. Reader generates a challenge c
  2. Tag computes response $z = F_K(c)$
  3. Reader computes $z' = F_K(c)$
  4. Reader accepts the Tag if $z = z'$



Ideal Authentication Protocol

Considered conditions:

- Protocol properties:
  1. Provably secure
  2. Small amount of transferred data
  3. Minimum of rounds (i.e. 2)
  4. Fast response (low latency)
- Tag properties:
  1. Small footprint (in hardware)
  2. Small code size (in software)
  3. Low power consumption
  4. Low cost

Ideal Authentication Protocol

Considered conditions:

- Protocol properties:
  1. Provably secure
  2. Small amount of transferred data
  3. Minimum of rounds (i.e. 2)
  4. Fast response (low latency)

- Tag properties:
  1. Small footprint (in hardware)
  2. Small code size (in software)
  3. Low power consumption
  4. Low cost

Lapin protocol

Lapin[1]

- Based on the Learning Parity with Noise problem (LPN)
- Authentication scheme
- Non-deterministic (because of random errors)
- Defined on the ring $\mathrm{R} = \mathbb{F}_2[X]/f(X)$, deg($f$) = n

- **Lapin is provably secure based on the Ring-LPN problem**

[1] Lapin: an efficient authentication protocol based on Ring-LPN, S. Heyse, E. Kiltz, V. Lyubashevsky, Ch. Paar, K. Pietrzak, p. 346-365, FSE 2012

Lapin Protocol Description

Public parameters: R, $\pi:\{0,1\}^\lambda \to \mathrm{R}, \tau, \tau', \lambda$

Secret key: $K = (s, s') \in \mathrm{R}^2$

	Tag		**Reader**
①		$\xleftarrow{c}$	$c \xleftarrow{\$} \{0,1\}^\lambda$
②	$r \xleftarrow{\$} \mathrm{R}^*;\ e \xleftarrow{\$} \mathrm{Ber}_\tau^{\mathrm{R}} \in \mathrm{R}$		
③	$z := r \cdot (s \cdot \pi(c) + s') + e$	$\xrightarrow{(r,z)}$	
④			if $r \notin \mathrm{R}^*$ reject
⑤			$e' := z - r \cdot (s \cdot \pi(c) + s')$
⑥			if $HW(e') > n \cdot \tau'$ reject else accept

Masking Countermeasure

- **Objective:** decrease the correlation between the consumed power and the processed sensitive data
- **Implementation:** all sensitive variables must be split to shares and computations should be performed on each share separately (if possible)
- **Conditions** for effective masking:
  - The leakage of each share is independent from the others
  - Sufficient noise is present in the device
- **Example:**

$$\begin{aligned} h_1 &= q_1 \\ \dots \\ h_{d-1} &= q_{d-1} \\ h_d &= h \bigoplus \bigoplus_{i=1}^{d-1} q_i \end{aligned}$$

Masking of Lapin

1. Split sensitive variable *s*, *s'* and *e* into *d* shares

$$
\begin{aligned}
s &= s_1 \oplus s_2 \oplus \cdots \oplus s_d\,, \\
s' &= s'_1 \oplus s'_2 \oplus \cdots \oplus s'_d\,, \\
e &= e_1 \oplus e_2 \oplus \cdots \oplus e_d
\end{aligned}
$$

2. Derive a formula allowing to demask the output

$$
\begin{aligned}
z &= (\pi(c) \cdot s \oplus s') \cdot r \oplus e \\
&= [\pi(c) \cdot (s_1 \oplus \cdots \oplus s_d) \oplus (s'_1 \oplus \cdots \oplus s'_d)] \cdot r \oplus (e_1 \oplus \cdots \oplus e_d) \\
&= [(\pi(c) \cdot s_1 \oplus s'_1) \cdot r \oplus e_1] \oplus \cdots \oplus [(\pi(c) \cdot s_d \oplus s'_d) \cdot r \oplus e_d] \\
&= z_1 \oplus \cdots \oplus z_d
\end{aligned}
$$

- **Lapin is linear** = each share is computed **separately**

Implementation

Definition of constants

Constants are chosen as in the Lapin paper (CRT impl.):

- *n = deg(f(X)) = 621*
- *m = 5*
- *M factors of f(X) are:*
- $\tau = {}^1\!/_6$
- $\tau' = 0.29$
- $\lambda = 80$ bits

$$\begin{aligned}
f_1(X) &= X^{127} + X^8 + X^7 + X^3 + 1\\
f_2(X) &= X^{126} + X^9 + X^6 + X^5 + 1\\
f_3(X) &= X^{125} + X^9 + X^7 + X^4 + 1\\
f_4(X) &= X^{122} + X^7 + X^4 + X^3 + 1\\
f_5(X) &= X^{121} + X^8 + X^5 + X + 1
\end{aligned}$$

$\Rightarrow$ 128-bit datapath is suitable, since $deg(f_j(X)) < 128$

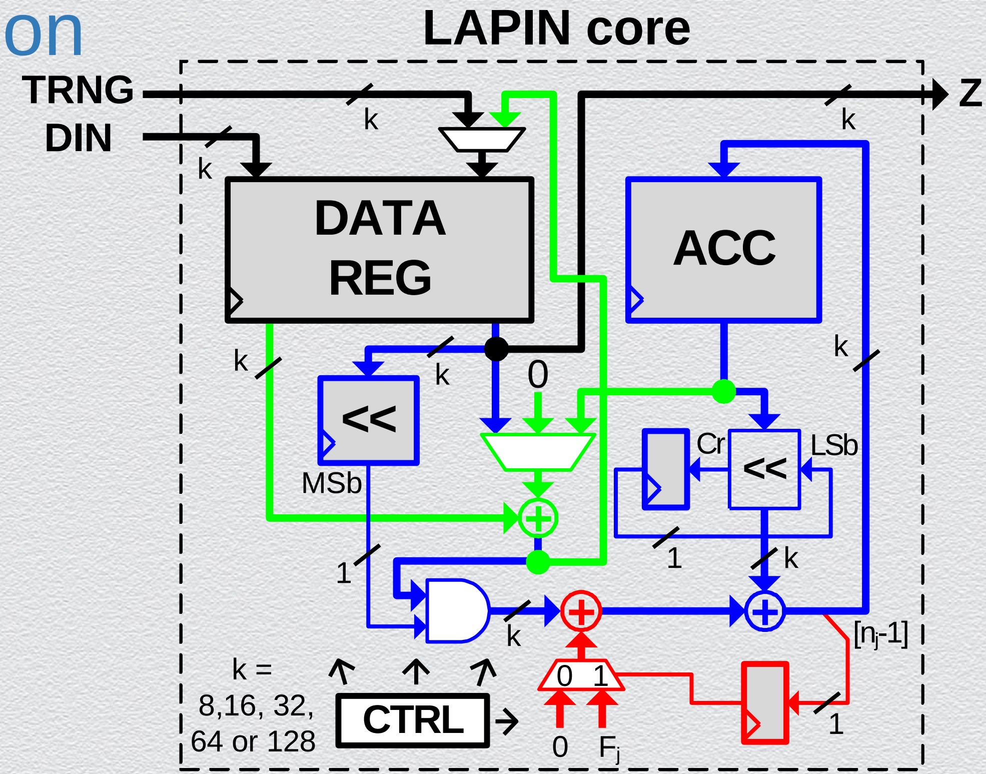
Polynomial multiplication & reduction

- We have implemented a **128-bit "school-book" polynomial multiplication unit** because:
  - It can be performed **in parallel with 1-bit reduction**
  - Its hardware implementation is **very small**
  - Its implementation can operate on **high frequencies**

- This **unit can be shared** for Lapin computations as well as error *e* transformation

Implementation description

- 8b to 128b datapath width
- Data registers in RAM
- Accumulator in RAM
- Carry register if k<128
- Shift register must not load sensitive data



Multiplication, reduction and addition

Performance evaluation

Cost evaluation & Timing results

- Lapin was synthesized for Xilinx Virtex 5 FPGA

Datapath (k)	Slices	BRAM		f_{max} (MHz)	Clock cycles		
		18kb	36kb		d = 1	d = 2	d = 3
8	213	2	0	125.3	20,977	41,969	62,961
16	232	2	0	127.5	10,489	20,985	31,481
32	311	1	1	127.2	5,245	10,493	15,741
64	330	0	3	130.2	2,623	5,247	7,871
128	451	0	6	140.3	1,332	2,664	3,996

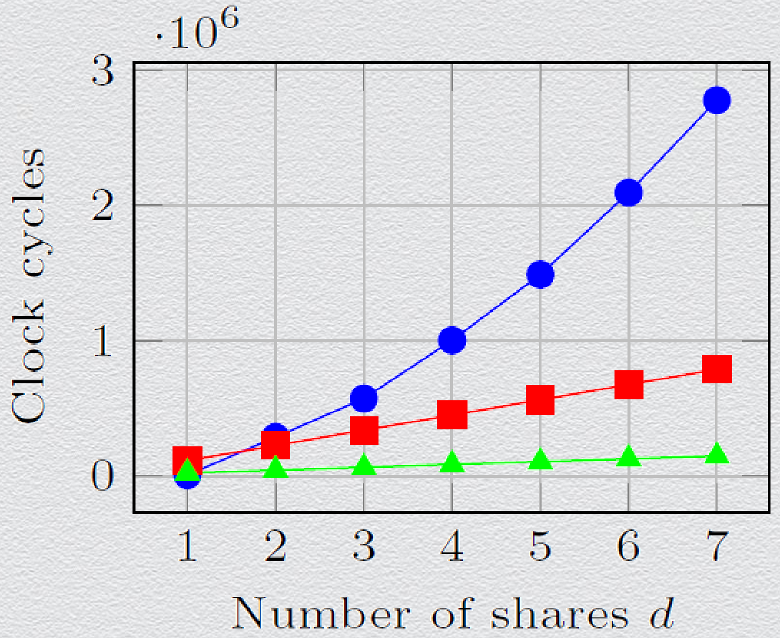
- d = 1: Lapin without masking

  d > 1: Masked Lapin – secure to (*d-1*) – order attacks

Comparison

d	AES softw.	Lapin[a] softw.	Lapin 8b hardw.
1	5,100	112,500	20,977
2	286,844	225,016	41,969
3	572,069	337,532	62,961
4	1,003,154	450,048	83,953
5	1,489,539	562,564	104,945
6	2,095,756	675,080	125,937
7	2,779,561	787,596	146,929

[a] For d>1 values are estimated



- By increasing *d*, number of clk. cycles grows **linearly for Lapin** and **quadratically for AES**

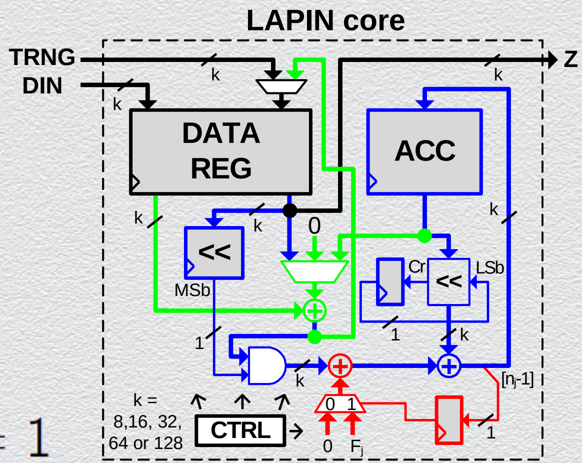
  ⇒ **much cheaper to increase Lapin security to higher-order SCA than that of AES**

RSA CONFERENCE 2014

FEBRUARY 24 – 28 | MOSCONE CENTER | SAN FRANCISCO

Resistance to side-channel attacks

Leakage Model



- **Target operation:** $s \cdot \pi(c)$, where π is zero padding
- **Assumption:** Accumulator leaks Hamming weight
- Accumulator is updated during the multiplication loop:

$$a_0 = 0 \qquad a_{i+1} \leftarrow \begin{cases} 2 \cdot a_i + s & \text{if } c[80-i] = 1 \\ 2 \cdot a_i & \text{otherwise} \end{cases}$$

- The value of *a* after few clock cycles of computation is a small multiple of the secret:

$$a_{80} = s \cdot c \qquad a_i = s \cdot \underbrace{\sum_{j=1}^{i} c[80-j]X^{i-j}}_{m_i(c)}$$

  - Device leaks **HW(a_i)**

Attack time points

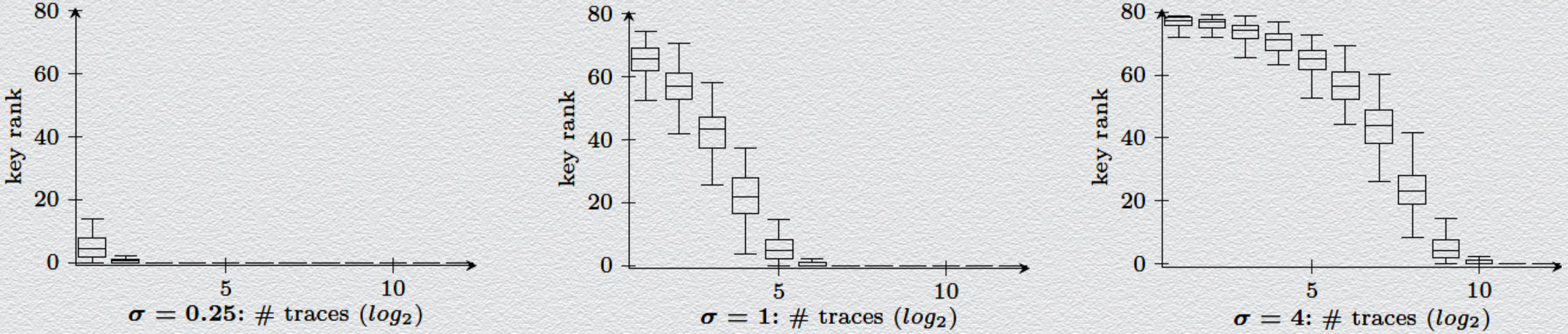
- Two equally efficient attack options:
  - Attack can target **several clock cycles in a single trace** with the **same challenge *c***

$\mathrm{HW}(a_i) = \mathrm{HW}(s \cdot m_i(c))$, for the same secret *c* and different values of *i*

  - Attack can target the **same clock cycle ι in several traces**, while **challenges are chosen** appropriately

$$m_\iota(c_j) = m_j(c)$$

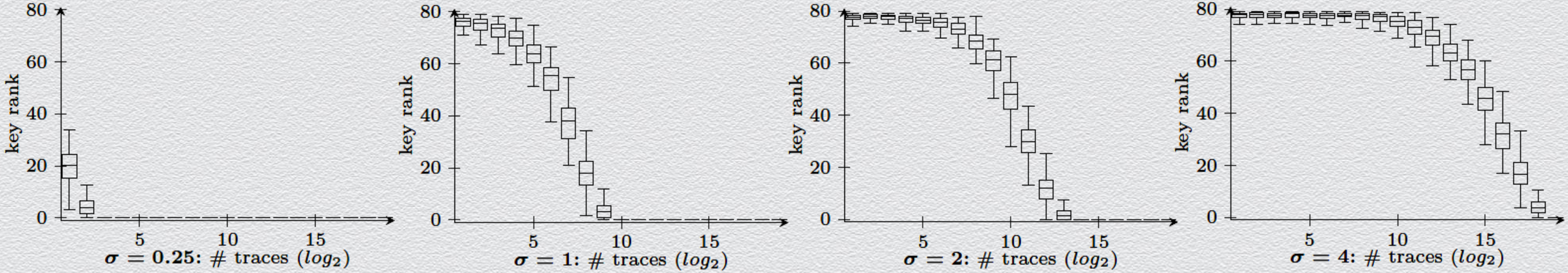
Collision-like Attack on Unprotected Lapin (d = 1)



- **Graphs:** Rank of the full key for *k = 128* using all clock cycles
- We can recover 80 key bits using about $2^6.\sigma^2$ traces for *k = 128*
- For k < 128 about $2^6.\sigma^2.128/k$ traces (128/k measurements are combined to get *HW(a)*)
- **Attack order:** 1st order bivariate (difference of 2 measures, information in average)

Collision-like Attack on Masked Lapin (e.g. d = 2)

- Distributions were used to mount a template attack for k = *128* using all clock cycles



- Data complexity increases roughly by σ^4 $\leftarrow$ typical for second order attacks
- **Attack order:** 2nd order 4-variate (4 measures combined pair-wise using difference, distributions are distinguished using covariance)

RSA CONFERENCE 2014

FEBRUARY 24 – 28 | MOSCONE CENTER | SAN FRANCISCO

Conclusions & perspectives

Conclusions

- **Lapin is linear → straightforward to mask**
- **First hardware implementation of Lapin**
  - Compact and very fast
  - Flexible datapath size (8-, 16-, 32-, 64- and 128-bit)
- **Advantages of Lapin over AES**
  - Smaller for large datapaths
  - High-order masking overhead increases linearly (quadratically for AES)
  - Shares are manipulated independently (independent leakage property)

Conclusions

- **Leakage model:** Hamming weight of accumulator
- Side-channel attacks against unprotected Lapin ($d = 1$)
  - Collision-like attack – 1st order bivariate attack
- Side-channel attack against masked Lapin ($d \geq 2$)
  - Collision-like attack – 2nd order 4-variate attack

Perspectives

- SCA using **Hamming distance model**
- Further study of the data-dependent algorithmic noise
- On-chip randomness generation is a problem => could it be solved using Learning With Rounding assumption?

Thank you for attention!

Protocol Classification

- Block-cipher based schemes
  - AES-based – may be too heavy for some appl.
  - Present-based - more suitable
- Schemes based on hardness of a mathematical problem
  - **L**earning **P**arity with **N**oise problem (LPN)
    - Hopper-Blum protocol (HB) and its variants (HB+, HB-MP, etc.)
    - **Lapin protocol**[1]
  - Others

[1] Lapin: an efficient authentication protocol based on Ring-LPN, S. Heyse, E. Kiltz, V. Lyubashevsky, Ch. Paar, K. Pietrzak, p. 346-365, FSE 2012

Protocol Classification

- Block-cipher based schemes
  - AES-based – may be too heavy for some appl.
  - Present-based - more suitable
- Schemes based on hardness of a mathematical problem
  - **L**earning **P**arity with **N**oise problem (LPN)
    - Hopper-Blum protocol (HB) and its variants (HB+, HB-MP, etc.)
    - **Lapin protocol**[1]
  - Others

[1] Lapin: an efficient authentication protocol based on Ring-LPN, S. Heyse, E. Kiltz, V. Lyubashevsky, Ch. Paar, K. Pietrzak, p. 346-365, FSE 2012

Protocol Classification

- Block-cipher based schemes
  - AES-based – may be too heavy for some appl.
  - Present-based - more suitable
- Schemes based on hardness of a mathematical problem
  - **L**earning **P**arity with **N**oise problem (LPN)
    - Hopper-Blum protocol (HB) and its variants (HB+, HB-MP, etc.)
    - **Lapin protocol**[1]
  - Others

[1] Lapin: an efficient authentication protocol based on Ring-LPN, S. Heyse, E. Kiltz, V. Lyubashevsky, Ch. Paar, K. Pietrzak, p. 346-365, FSE 2012

Learning Parity with Noise Problem (LPN)

- Given a set of samples *(A, t = A . s + e)* with a random error *e*, where $t, e \in \mathbb{F}_2^n$ and $A \in \mathbb{F}_2^{n \times n}$
- Find the secret $s \in \mathbb{F}_2^n$
- Solution:
  - if ***e = 0*** then Gaussian elimination can solve it → **no security!**
  - if ***e > 0*** then it may become an NP-Hard problem
    → **suitable for cryptography!**

Note: The error *e* is generated with the Bernoulli distribution with parameter τ.

$$\mathrm{HW}(e) \approx n\tau$$

Lapin Protocol Parameters

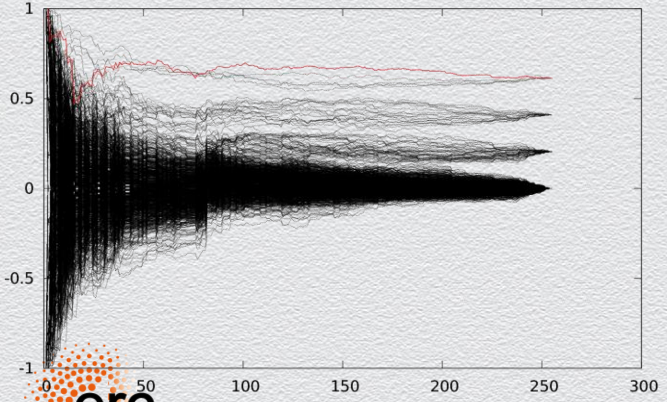
- 2-round protocol
- Public parameters:
  - R, *n* — ring $\mathsf{R} = \mathbb{F}_2[X]/f(X)$, deg(*f*) = *n*
  - λ — security level parameter (in bits)
  - π — mapping $\{0,1\}^\lambda \to \mathsf{R}$
  - $\tau \in (0, {}^1\!/\!_2)$ — parameter of Bernoulli distribution
  - $\tau' \in (\tau, {}^1\!/\!_2)$ — reader acceptance threshold
- Secret parameters:
  - $K = (s, s')$ — shared secret key, while $(s, s') \xleftarrow{\$} \mathsf{R}$

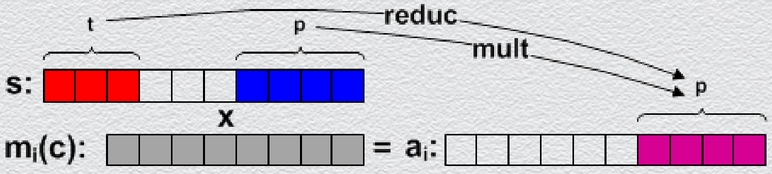
Ring-LPN Problem

- Ring Learning Parity with Noise (Ring-LPN) is an extension of LPN to rings
- The matrix A has a special structure. This way *A . s* is equivalent to the multiplication in the ring $\mathrm{R} = \mathbb{F}_2[X]/f(X)$
- **Lapin is provably secure based on the Ring-LPN problem**

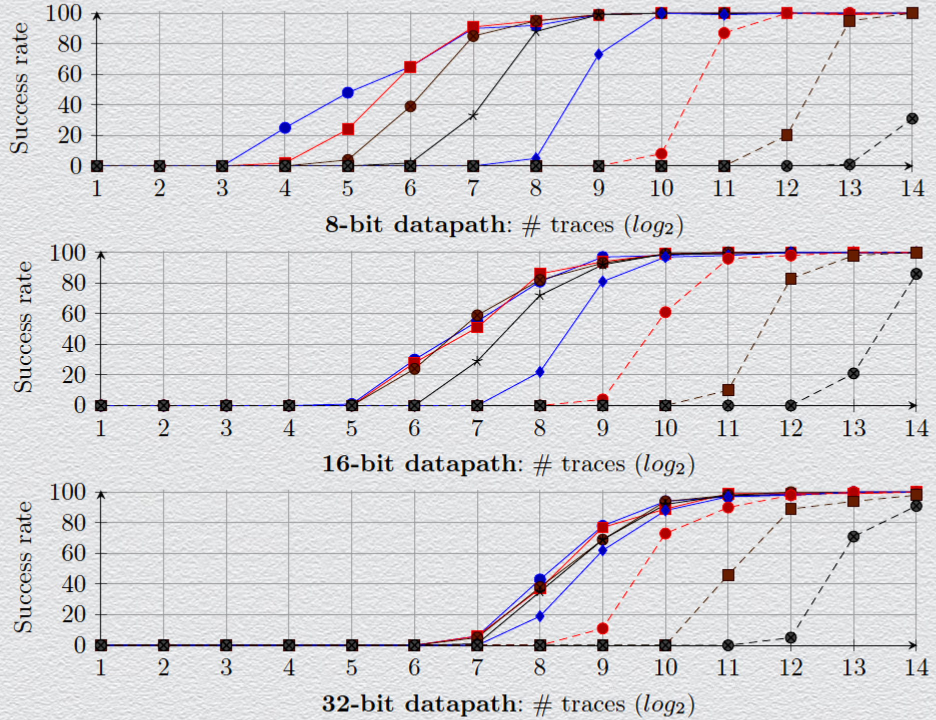
DPA-like Attack Against Unprotected Lapin (d = 1)

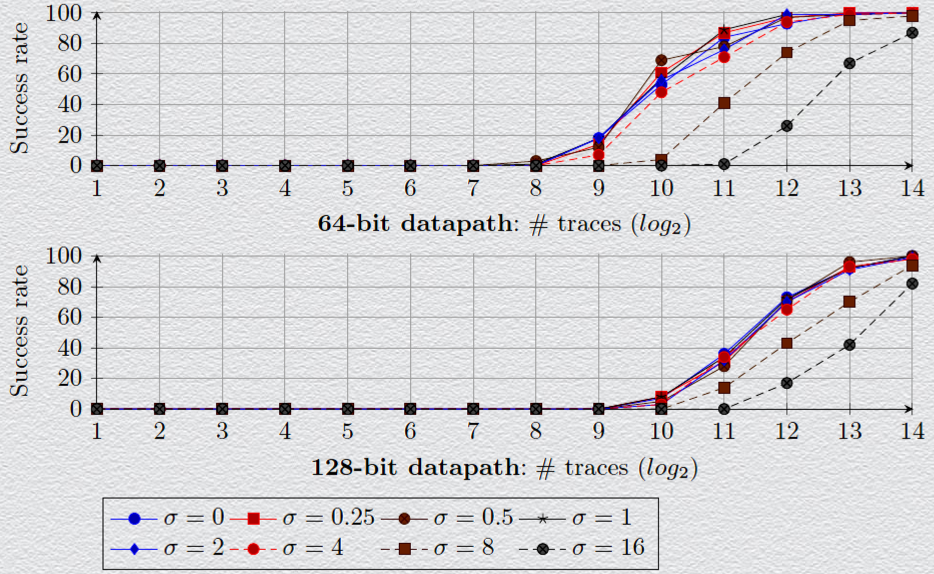
- Attack:
  - Predict some bits of $a_i = s \cdot m_i(c)$
  - If $deg(a_i) \leq t$ we can compute ***p* least significant bits** of a_i from the *p* least significant and *t* most significant bits of *s*.
  - Ex.: Correlation for *t* = *7* and *p* = *3*:





DPA-like Attack Against Unprotected Lapin (d = 1)





- Success rate for full-key recovery

Collision-like Attack on Unprotected Lapin (d = 1)

- **Approach:** Prediction of modular reduction impact on HW (i.e $\alpha \mapsto \alpha \cdot X$)
- **Assumption:** Accumulator contains a value α that will be rotated and reduced in the next clock cycle

$$\alpha \cdot X \bmod f = \begin{cases} (\alpha \lll 1) & \text{if } \mathrm{MSB}(\alpha) = 0 \\ (\alpha \lll 1) \oplus \bar{f} & \text{if } \mathrm{MSB}(\alpha) = 1 \end{cases}$$, where $\bar{f} = f \oplus X^{\deg(f)} \oplus 1$

- Since $\mathrm{HW}(\bar{f}) = 3$ the relations between HW of α and $\alpha \cdot X \bmod f$ is as follows:

$$\mathrm{HW}(\alpha \cdot X \bmod f) = \begin{cases} \mathrm{HW}(\alpha) & \text{if } \mathrm{MSB}(\alpha) = 0 \\ \mathrm{HW}(\alpha) + 3 & \text{if } \mathrm{MSB}(\alpha) = 1 \text{ and } \mathrm{HW}(\alpha \lll 1 \wedge \bar{f}) = 0 \\ \mathrm{HW}(\alpha) + 1 & \text{if } \mathrm{MSB}(\alpha) = 1 \text{ and } \mathrm{HW}(\alpha \lll 1 \wedge \bar{f}) = 1 \\ \mathrm{HW}(\alpha) - 1 & \text{if } \mathrm{MSB}(\alpha) = 1 \text{ and } \mathrm{HW}(\alpha \lll 1 \wedge \bar{f}) = 2 \\ \mathrm{HW}(\alpha) - 3 & \text{if } \mathrm{MSB}(\alpha) = 1 \text{ and } \mathrm{HW}(\alpha \lll 1 \wedge \bar{f}) = 3 \end{cases}$$

Collision-like Attack on Unprotected Lapin (d = 1)

- Therefore the distribution for of $\mathrm{HW}(\alpha \cdot X) - \mathrm{HW}(\alpha)$ for a random α is as follows:

$$\textbf{if } \mathbf{MSB}(\boldsymbol{\alpha}) = \mathbf{0}\textbf{:}\ \mathrm{HW}(\alpha \cdot X) - \mathrm{HW}(\alpha) = 0,$$

$$\textbf{if } \mathbf{MSB}(\boldsymbol{\alpha}) = \mathbf{1}\textbf{:}\ \mathrm{HW}(\alpha \cdot X) - \mathrm{HW}(\alpha) = \begin{cases} +3 & \text{with probability } 1/8 \\ +1 & \text{with probability } 3/8 \\ -1 & \text{with probability } 3/8 \\ -3 & \text{with probability } 1/8 \end{cases}$$

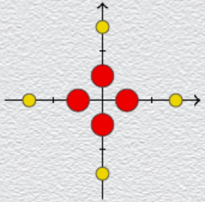
- This can be exploited using two chosen challenges $m_i(c) = m$ and $m_{i'}(c') = m \cdot X$
- Then we can recover $\mathrm{MSB}(m \cdot s)$ by comparing $\mathrm{HW}(m \cdot s)$ and $\mathrm{HW}(m \cdot X \cdot s)$
- **Result:** without noise 2 measures are sufficient to recover 1 key bit with probability 1
- **Advantage:** analysis of the full multiplier state and avoids algorithmic noise due to HW

Collision-like Attack on Masked Lapin (d > 1)

- We must combine leakages from all shares to get the key $s = \bigoplus_{j=1}^{d} s_j$
- We need to choose two challenges such that $m_i(c) = m$ and $m_{i'}(c') = m \cdot X$
- Then we can recover $\mathrm{MSB}(m \cdot s_j)$ by comparing $\mathrm{HW}(m \cdot s_j)$ and $\mathrm{HW}(m \cdot X \cdot s_j)$
- We study 2D distribution: $\left(\mathrm{HW}(\alpha_j \cdot X) - \mathrm{HW}(\alpha_j)\right)_{j=1}^{d},$ $\quad \text{with } \alpha = \bigoplus_{j=1}^{d} \alpha_j$

Collision-like Attack on Masked Lapin (e.g. d = 2)

$$\mathbf{MSB(\alpha)=1:}\quad
\begin{cases}
\mathrm{MSB}(\alpha_1)=0 \rightarrow \mathrm{HW}(\alpha_1\cdot X)-\mathrm{HW}(\alpha_1)=0\\
\mathrm{MSB}(\alpha_2)=1 \rightarrow \mathrm{HW}(\alpha_2\cdot X)-\mathrm{HW}(\alpha_2)\in\{-3,-1,+1,+3\}\\
\mathrm{MSB}(\alpha_1)=1 \rightarrow \mathrm{HW}(\alpha_1\cdot X)-\mathrm{HW}(\alpha_1)\in\{-3,-1,+1,+3\}\\
\mathrm{MSB}(\alpha_2)=0 \rightarrow \mathrm{HW}(\alpha_2\cdot X)-\mathrm{HW}(\alpha_2)=0
\end{cases}$$



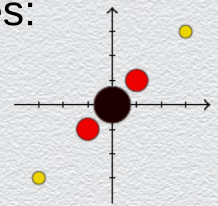
$\sigma(x,y)=0$

$$\mathbf{MSB(\alpha)=0:}\quad
\begin{cases}
\mathrm{MSB}(\alpha_1)=0 \rightarrow \mathrm{HW}(\alpha_1\cdot X)-\mathrm{HW}(\alpha_1)=0\\
\mathrm{MSB}(\alpha_2)=0 \rightarrow \mathrm{HW}(\alpha_2\cdot X)-\mathrm{HW}(\alpha_2)=0\\
\mathrm{MSB}(\alpha_1)=1 \rightarrow \mathrm{HW}(\alpha_1\cdot X)-\mathrm{HW}(\alpha_1)\in\{-3,-1,+1,+3\}\\
\mathrm{MSB}(\alpha_2)=1 \rightarrow \mathrm{HW}(\alpha_2\cdot X)-\mathrm{HW}(\alpha_2)\in\{-3,-1,+1,+3\}
\end{cases}$$

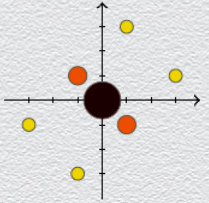
$\mathrm{HW}(\alpha \lll 1 \wedge \bar{f})\in\{0,1,2,3\}$

- Probabilities:

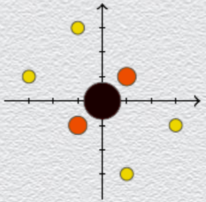
1/16
2/16
3/16
8/16



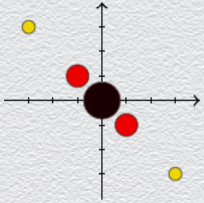
$\sigma(x,y)=6$
$\mathrm{HW}(\alpha\lll 1\wedge\bar{f})=0$



$\sigma(x,y)=2.25$
$\mathrm{HW}(\alpha\lll 1\wedge\bar{f})=1$



$\sigma(x,y)=-2.25$
$\mathrm{HW}(\alpha\lll 1\wedge\bar{f})=2$



$\sigma(x,y)=-6$
$\mathrm{HW}(\alpha\lll 1\wedge\bar{f})=3$