TELECOM ParisTech

Institut Mines-Telecom

Multi-Variate High-Order Attacks of Shuffled Tables Recomputation

Nicolas BRUNEAU[1,2], Sylvain GUILLEY[1,3], Zakaria NAJM[1], Yannick TEGLIA[2],

[1] TELECOM-ParisTech, Crypto Group, Paris, FRANCE
[2] STMicroelectronics, AST division, Rousset, FRANCE
[3] Secure-IC S.A.S., Rennes, FRANCE

life.augmented

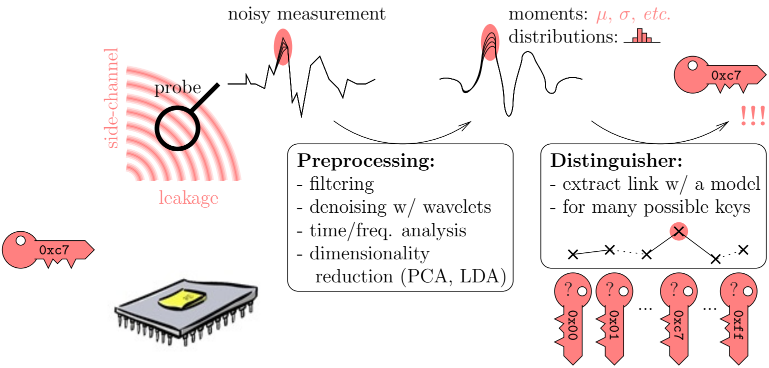
STMicroelectronics

CHES 2015 Saint-Malo , France

Outline

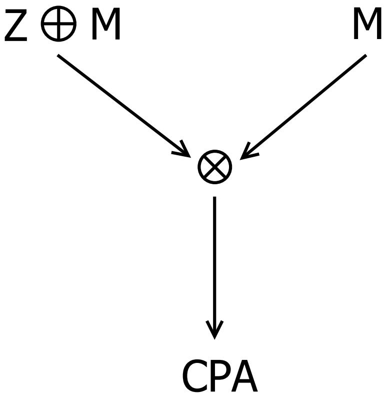
Outline

Side-Channel Analysis on Embedded Systems [GMN+11]



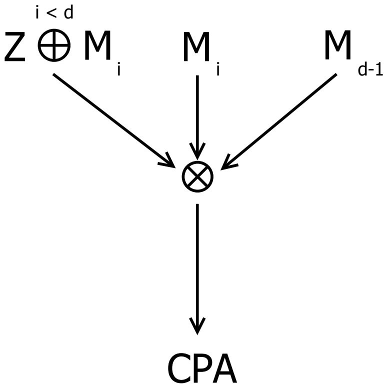
Masking scheme and attacks.

Figure : First order masking scheme and second order attack



Masking scheme and attacks.

Figure : High order masking scheme and high order attack



Masking schemes implementation

Implementation:

- Algebraic methods [BGK04, RP10].
- Global Look-up Table [PR07, SVCO+10] method.
- Table recomputation. For second order masking schemes [CJRR99, Mes00, AG01] and high order masking scheme [Cor14].

Table recomputation Algorithm

input : t, one byte of plaintext, and k, one byte of key
output: The application of AddRoundKey and SubBytes on t, i.e., $S(t \oplus k)$

$m \leftarrow_{\mathcal{R}} \mathbb{F}_2^n$, $m' \leftarrow_{\mathcal{R}} \mathbb{F}_2^n$ `// Draw of random input and output masks ;`
for $\omega \in \{0, 1, \ldots, 2^n - 1\}$ **do** `// Sbox masking`
  $z \leftarrow \omega \oplus m$ `// Masked input ;`
  $z' \leftarrow S[\omega] \oplus m'$ `// Masked output ;`
  $S'[z] \leftarrow z'$ `// Creating the masked Sbox entry ;`
end
$t \leftarrow t \oplus m$ `// Plaintext masking ;`
$t \leftarrow t \oplus k$ `// Masked AddRoundKey ;`
$t \leftarrow S'[t]$ `// Masked SubBytes ;`
$t \leftarrow t \oplus m'$ `// Demasking ;`
return t

Table recomputation Algorithm

input : t, one byte of plaintext, and k, one byte of key
output: The application of AddRoundKey and SubBytes on t, i.e., $S(t \oplus k)$

```
m ←_R F_2^n, m' ←_R F_2^n // Draw of random input and output masks ;
for ω ∈ {0, 1, ..., 2^n − 1} do // Sbox masking
    z ← ω ⊕ m // Masked input ;
    z' ← S[ω] ⊕ m' // Masked output ;
    S'[z] ← z' // Creating the masked Sbox entry ;
end
t ← t ⊕ m // Plaintext masking ;
t ← t ⊕ k // Masked AddRoundKey ;
t ← S'[t] // Masked SubBytes ;
t ← t ⊕ m' // Demasking ;
return t
```

Usual 2-variate 2nd-order attack on the S-Box input
Let us call this attack 2O-CPA

Table recomputation Algorithm

input : t, one byte of plaintext, and k, one byte of key
output: The application of AddRoundKey and SubBytes on t, i.e., $S(t \oplus k)$

$m \leftarrow_{\mathcal{R}} \mathbb{F}_2^n$, $m' \leftarrow_{\mathcal{R}} \mathbb{F}_2^n$ // Draw of random input and output masks ;
for $\omega \in \{0, 1, \ldots, 2^n - 1\}$ **do** // Sbox masking
    $z \leftarrow \omega \oplus m$ // Masked input ;
    $z' \leftarrow S[\omega] \oplus m'$ // Masked output ;
    $S'[z] \leftarrow z'$ // Creating the masked Sbox entry ;
end
$t \leftarrow t \oplus m$ // Plaintext masking ;
$t \leftarrow t \oplus k$ // Masked AddRoundKey ;
$t \leftarrow$ $S'[t]$ // Masked SubBytes ;
$t \leftarrow t \oplus$ m' // Demasking ;
return t

Usual 2 -variate 2nd -order attack on the S-Box output

Table recomputation Algorithm

input : t, one byte of plaintext, and k, one byte of key
output: The application of AddRoundKey and SubBytes on t, i.e., $S(t \oplus k)$

$m \leftarrow_{\mathcal{R}} \mathbb{F}_2^n$, $m' \leftarrow_{\mathcal{R}} \mathbb{F}_2^n$ `// Draw of random input and output masks ;`
for $\omega \in \{0, 1, \ldots, 2^n - 1\}$ **do** `// Sbox masking`
  $z \leftarrow \omega \oplus m$ `// Masked input ;`
  $z' \leftarrow S[\omega] \oplus m'$ `// Masked output ;`
  $S'[z] \leftarrow z'$ `// Creating the masked Sbox entry ;`
end
$t \leftarrow t \oplus m$ `// Plaintext masking ;`
$t \leftarrow t \oplus k$ `// Masked AddRoundKey ;`
$t \leftarrow S'[t]$ `// Masked SubBytes ;`
$t \leftarrow t \oplus m'$ `// Demasking ;`
return t

2-stage CPA attack [PdHL09, TWO13]

- Perform a horizontal CPA to recover the mask,
- Perform an vertical first order CPA (knowing the mask).

Table recomputation Algorithm

input : t, one byte of plaintext, and k, one byte of key
output: The application of AddRoundKey and SubBytes on t, i.e., $S(t \oplus k)$

$m \leftarrow_{\mathcal{R}} \mathbb{F}_2^n$, $m' \leftarrow_{\mathcal{R}} \mathbb{F}_2^n$ `// Draw of random input and output masks ;`
for $\omega \in \{0, 1, \ldots, 2^n - 1\}$ **do** `// Sbox masking`
    $z \leftarrow \omega \oplus m$ `// Masked input ;`
    $z' \leftarrow S[\omega] \oplus m'$ `// Masked output ;`
    $S'[z] \leftarrow z'$ `// Creating the masked Sbox entry ;`
end
$t \leftarrow t \oplus m$ `// Plaintext masking ;`
$t \leftarrow t \oplus k$ `// Masked AddRoundKey ;`
$t \leftarrow S'[t]$ `// Masked SubBytes ;`
$t \leftarrow t \oplus m'$ `// Demasking ;`
return t

Improved $(2^n + 1)$-variate 2nd-order attack on the input [BGHR14]

Table recomputation Algorithm

input : t, one byte of plaintext, and k, one byte of key
output: The application of AddRoundKey and SubBytes on t, i.e., $S(t \oplus k)$

```
m ←_R F_2^n, m' ←_R F_2^n // Draw of random input and output masks ;
for ω ∈ {0, 1, ..., 2^n − 1} do // Sbox masking
    z ← ω ⊕ m // Masked input ;
    z' ← S[ω] ⊕ m' // Masked output ;
    S'[z] ← z' // Creating the masked Sbox entry ;
end
t ← t ⊕ m // Plaintext masking ;
t ← t ⊕ k // Masked AddRoundKey ;
t ← S'[t] // Masked SubBytes ;
t ← t ⊕ m' // Demasking ;
return t
```

2-stage CPA attack

- Perform a horizontal CPA to recover the mask,
- Perform an vertical first order CPA (knowing the mask).

Table recomputation Algorithm

input : t, one byte of plaintext, and k, one byte of key
output: The application of AddRoundKey and SubBytes on t, i.e., $S(t \oplus k)$

$m \leftarrow_{\mathcal{R}} \mathbb{F}_2^n$, $m' \leftarrow_{\mathcal{R}} \mathbb{F}_2^n$ `// Draw of random input and output masks ;`
for $\omega \in \{0, 1, \ldots, 2^n - 1\}$ **do** `// Sbox masking`
  $z \leftarrow \omega \oplus m$ `// Masked input ;`
  $z' \leftarrow S[\omega] \oplus m'$ `// Masked output ;`
  $S'[z] \leftarrow z'$ `// Creating the masked Sbox entry ;`
end
$t \leftarrow t \oplus m$ `// Plaintext masking ;`
$t \leftarrow t \oplus k$ `// Masked AddRoundKey ;`
$t \leftarrow S'[t]$ `// Masked SubBytes ;`
$t \leftarrow t \oplus m'$ `// Demasking ;`
return t

$(2^n + 1)$-variate 2nd-order attack on the output

Classical countermeasure

Make the index of the loop unknown ➔ shuffle the recomputation.

Use random permutation:

- Random start index,
- LFSR,
- ...

Let us denote this permutation by φ

Outline

Protected table recomputation algorithm

input : t, one byte of plaintext, and k, one byte of key
output: The application of AddRoundKey and SubBytes on t

$m \leftarrow_{\mathcal{R}} \mathbb{F}_2^n$, $m' \leftarrow_{\mathcal{R}} \mathbb{F}_2^n$ `// Draw of random input and output masks` ;
$\varphi \leftarrow_{\mathcal{R}} \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ `// Draw of random permutation of` $\mathbb{F}_2^n$;
for $\varphi(\omega) \in \{\varphi(0), \varphi(1), \ldots, \varphi(2^n - 1)\}$ **do** `// S-box masking`
  $z \leftarrow \varphi(\omega) \oplus m$ `// Masked input` ;
  $z' \leftarrow S[\varphi(\omega)] \oplus m'$ `// Masked output` ;
  $S'[z] = z'$ `// Creating the masked S-box entry` ;
end
$t \leftarrow t \oplus m$ `// Plaintext masking` ;
$t \leftarrow t \oplus k$ `// Masked AddRoundKey` ;
$t \leftarrow S'[t]$ `// Masked SubBytes` ;
$t \leftarrow t \oplus m'$ `// Demasking` ;
return t

Protected table recomputation algorithm

input : t, one byte of plaintext, and k, one byte of key
output: The application of AddRoundKey and SubBytes on t

$m \leftarrow_{\mathcal{R}} \mathbb{F}_2^n$, $m' \leftarrow_{\mathcal{R}} \mathbb{F}_2^n$ `// Draw of random input and output masks` ;
$\varphi \leftarrow_{\mathcal{R}} \mathbb{F}_2^n \to \mathbb{F}_2^n$ `// Draw of random permutation of` $\mathbb{F}_2^n$;
for $\varphi(\omega) \in \{\varphi(0), \varphi(1), \ldots, \varphi(2^n - 1)\}$ **do** `// S-box masking`
    $z \leftarrow \varphi(\omega) \oplus m$ `// Masked input` ;
    $z' \leftarrow S[\varphi(\omega)] \oplus m'$ `// Masked output` ;
    $S'[z] = z'$ `// Creating the masked S-box entry` ;
end
$t \leftarrow t \oplus m$ `// Plaintext masking` ;
$t \leftarrow t \oplus k$ `// Masked AddRoundKey` ;
$t \leftarrow S'[t]$ `// Masked SubBytes` ;
$t \leftarrow t \oplus m'$ `// Demasking` ;
return t

New attack

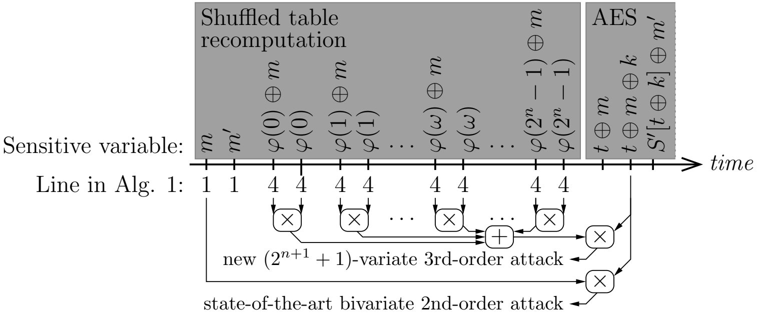
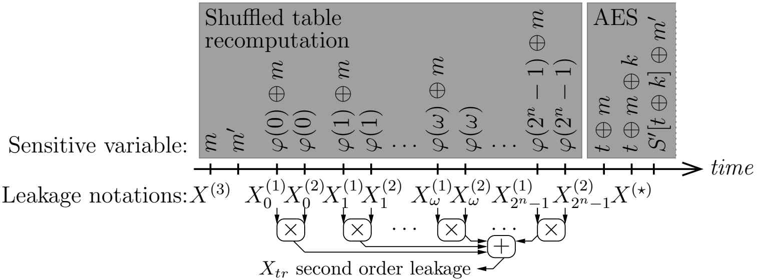


Figure : State-of-the-art attack and new attack

Notations



The combination function C_{tr} is given by:

$$C_{tr}: \begin{array}{ccc} \mathbb{R}^{2^{n+1}} \times \mathbb{R} & \longrightarrow & \mathbb{R} \\ \left(\left(X_\omega^{(1)}, X_\omega^{(2)}\right)_\omega, X^\star\right) & \longmapsto & \left(-2 \times \frac{1}{2^n} \sum_{\omega=0}^{2^n-1} X_\omega^{(1)} \times X_\omega^{(2)}\right) \times X^\star \end{array}$$

A new attack

The MultiVariate Attack exploiting the leakage of the table recomputation is given by the function :

$$\mathrm{MVA}_{tr}: \begin{array}{ccc} \mathbb{R}^{2^{n+1}} \times \mathbb{R} \times \mathbb{R} & \longrightarrow & \mathbb{F}_2^n \\ \left(\left(X_\omega^{(1)}, X_\omega^{(2)}\right)_\omega, X^\star, Y\right) & \longmapsto & \underset{K \in \mathbb{F}_2^n}{\mathrm{argmax}}\, \rho\left[C_{tr}\left(\left(\left(X_\omega^{(1)}, X_\omega^{(2)}\right)_\omega, X^\star\right), Y\right]\end{array} .$$

The MVA_{tr} is sound.

Main Theorem

Theorem

The SNR *of the "second-order leakage" is greater than the* SNR *of the leakage of the mask if and only if*

$$\sigma^2 \leqslant 2^{n-2} - \frac{n}{2} \ ,$$

where σ *denotes the standard deviation of the Gaussian noise.*

Corollary

The SR *of the* MVA_{tr} *is greater than the* SR *of the 2O-CPA if and only if*

$$\sigma^2 \leqslant 2^{n-2} - \frac{n}{2} \ .$$

An example

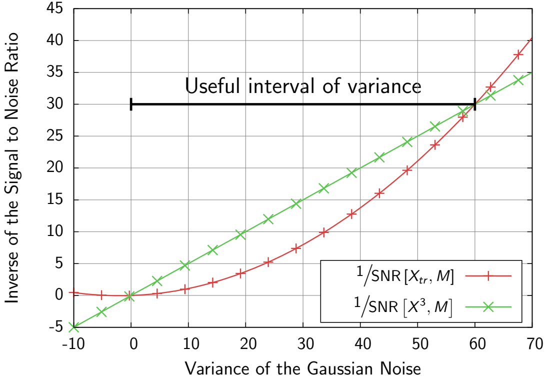
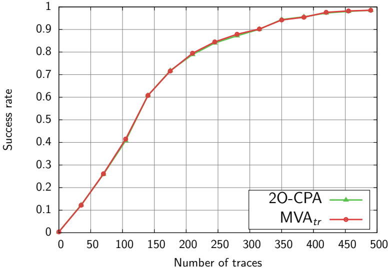
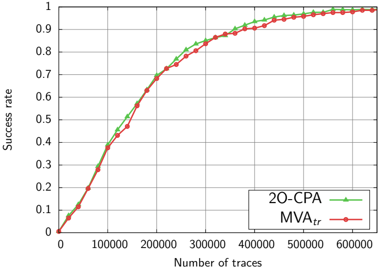


Figure : SNR of the second order leakage versus SNR of the mask.

Empirical validation



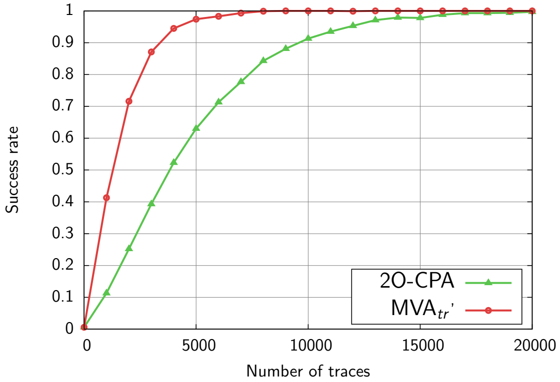
(a) σ^2=0



(b) σ^2=60

Figure : Comparison between 2O-CPA and MVA$_{tr}$.

Empirical Results



(a) $\sigma^2=9$

Figure : Comparison between 2O-CPA and MVA$_{tr}$.

Algorithm of the Coron's masking scheme

input : $x_1, \ldots, x_d$, such that $x = x_1 \oplus \ldots \oplus x_d$
output: $y_1, \ldots, y_d$, such that $y = y_1 \oplus \ldots \oplus y_d = S(x)$

for $\omega \in \mathbb{F}_2^n$ **do**
    $T(\omega) \leftarrow (S(\omega), 0, \ldots, 0) \in \left(\mathbb{F}_2^n\right)^d$ // $\oplus(T(\omega)) = S(u)$
end
for $i = 1$ *to* $i = d - 1$ **do** // $\oplus(T(\varphi(\omega))) = S(\varphi(\omega) \oplus x_1, \ldots, \oplus x_{d-1})\, \forall \omega \in \mathbb{F}_2^n$
    for $\omega \in \mathbb{F}_2^n$ **do**
        for $j = 1$ **to** d **do**
            $T'(\varphi(\omega))[j] \leftarrow T(\varphi(\omega) \oplus x_i)[j]$ // $T'(\varphi(\omega)) \leftarrow T(\varphi(\omega) \oplus x_i)$
        end
    end
    for $\omega \in \mathbb{F}_2^n$ **do**
        $T(\varphi(\omega)) \leftarrow \mathrm{RefreshMasks}(T(\varphi(\omega)))$
        // $\oplus(T(\varphi(\omega))) = S(\varphi(\omega) \oplus x_1, \ldots, \oplus x_i)$
    end
end
$(y_1, \ldots, y_d) \leftarrow \mathrm{RefreshMasks}(T(x_n))$ // $\oplus(T(x_d)) = S(x)$
return $y_1, \ldots, y_n$

Algorithm of the Coron's masking scheme

input : $x_1, \ldots, x_d$, such that $x = x_1 \oplus \ldots \oplus x_d$
output: $y_1, \ldots, y_d$, such that $y = y_1 \oplus \ldots \oplus y_d = S(x)$

for $\omega \in \mathbb{F}_2^n$ **do**
    $T(\omega) \leftarrow (S(\omega), 0, \ldots, 0) \in (\mathbb{F}_2^n)^d$ // $\oplus(T(\omega)) = S(u)$
end
for $i = 1$ *to* $i = d - 1$ **do** // $\oplus(T(\varphi(\omega))) = S(\varphi(\omega) \oplus x_1, \ldots, \oplus x_{d-1}) \forall \omega \in \mathbb{F}_2^n$
    for $\omega \in \mathbb{F}_2^n$ **do**
        for $j = 1$ **to** d **do**
            $T'\left(\varphi(\omega)\right)[j] \leftarrow T\left(\varphi(\omega) \oplus x_i\right)[j]$ // $T'(\varphi(\omega)) \leftarrow T(\varphi(\omega) \oplus x_i)$
        end
    end
    for $\omega \in \mathbb{F}_2^n$ **do**
        $T(\varphi(\omega)) \leftarrow \text{RefreshMasks}(T(\varphi(\omega)))$
        // $\oplus(T(\varphi(\omega))) = S(\varphi(\omega) \oplus x_1, \ldots, \oplus x_i)$
    end
end
$(y_1, \ldots, y_d) \leftarrow \text{RefreshMasks}(T(x_n))$ // $\oplus(T(x_d)) = S(x)$
return $y_1, \ldots, y_n$

Theorem

Theorem

The SNR *of the "second-order leakage" is greater than the* SNR *of the leakage of the mask if and only if*

$$\sigma^2 \leqslant d \times 2^{n-2} - \frac{n}{2} \quad , \qquad (1)$$

where σ denotes the standard deviation of the Gaussian noise.

Corollary

The SR *of the Multi-Variate attack is greater than the* SR *of the dO-CPA if and only if*

$$\sigma^2 \leqslant d \times 2^{n-2} - \frac{n}{2} \quad .$$

Theoretical evaluation

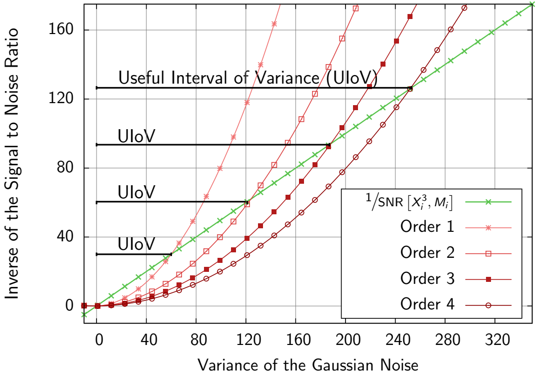
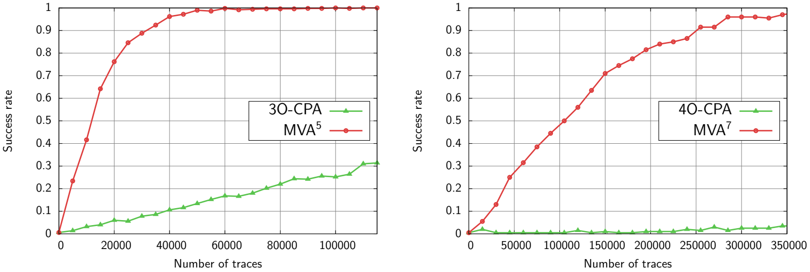


Figure : Interval of variance

Empirical results

The Multi-Variate Attack is a $2 \times (d-1)+1$ order attack.



(a) σ^2=9 (b) σ^2=9

Figure : Comparison between the {3,4}O-CPA and the MVA^{5,7}

Affine leakage function

A leakage function is said affine if this function is a weighted sum of the bits of the leaking value.
The affine leakage of a variable Z is:

$$\alpha \cdot (Z)_{i \leq n}$$

Where:

- α is the vector of weight with $\|\alpha\|_2^2 = n$.
- $(Z)_{i \leq n}$ is the vector of bit of Z
- $\cdot$ is the inner product.

Main theorem

Theorem

The SNR *of the "second-order leakage" is greater than the* SNR *of the leakage of the mask if and only if*

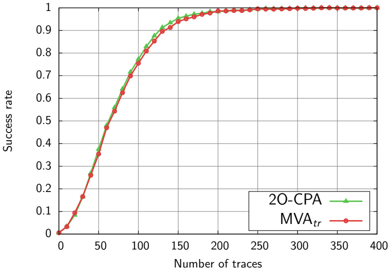
$$\sigma^2 \leqslant \|\alpha\|_4^4 \times \frac{2^{n-2}}{n} - \frac{n}{2} \; .$$

where σ denotes the standard deviation of the Gaussian noise.

Corollary

The worst case for the MVA_{tr} compared to the 2O-CPA is the Hamming Weight model.

Empirical results

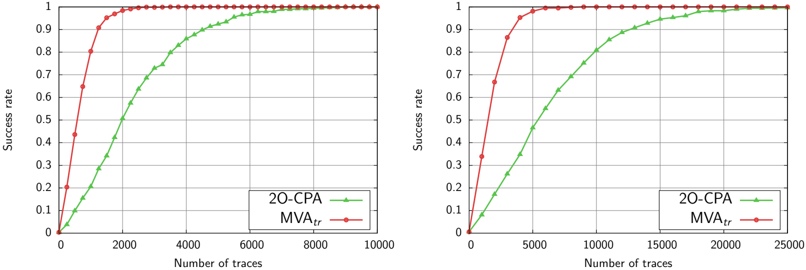


(a) σ^2=0 (b) σ^2=111

Figure : Comparison between 2O-CPA and MVA$_{tr}$ for $\varepsilon = 0.9$.

Let us assume α such as $\alpha_i^2 = 1 \pm \varepsilon$

Empirical results

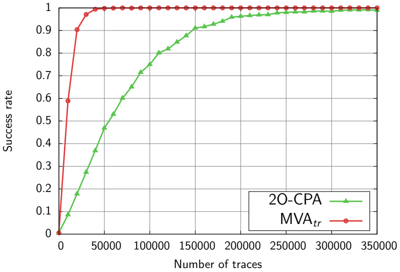


(a) $\sigma^2 = 9$ (b) $\sigma^2 = 16$

Figure : Comparison between 2O-CPA and MVA_{tr} for $\varepsilon = 0.9$.

Empirical results

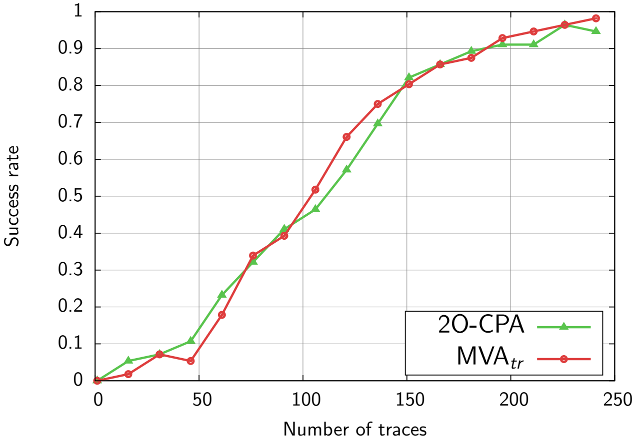
$$\alpha = \left(\sqrt{8}, 0, 0, 0, 0, 0, 0, 0\right).$$



(a) σ^2=49 (b) σ^2=64

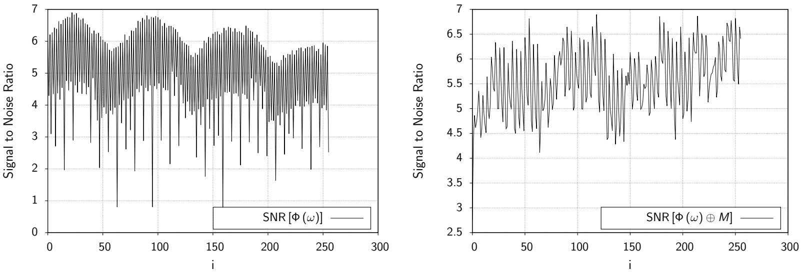
Figure : Comparison between the 2O-CPA and the MVA$_{tr}$ in case of one bit model in presence of High Gaussian noise.

Attack on real traces ATMega163



(a) Comparison of the attacks

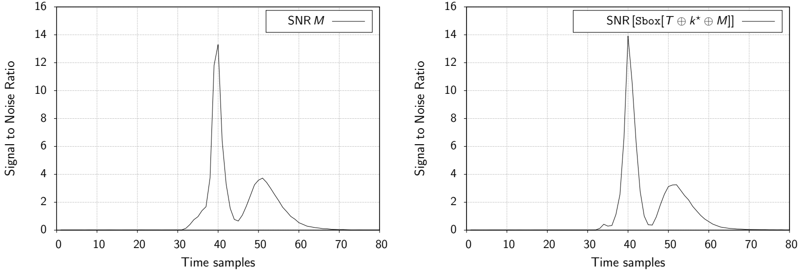
Analysis of the SNR



(a) SNR of the random index (b) SNR of the mask index

Figure : SNR of the sensitive points in the table recomputation.

Analysis of the SNR

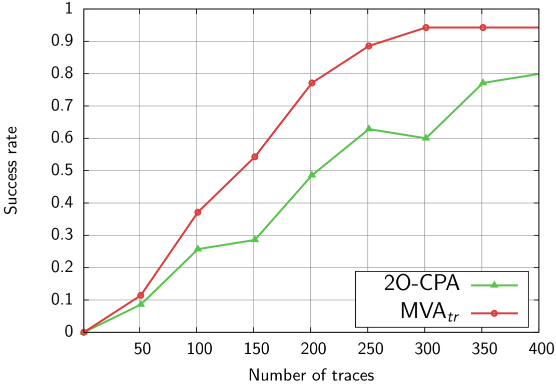


(a) SNR of the random index (b) SNR of the mask index

Figure : SNR of the sensitive points in the table recomputation.

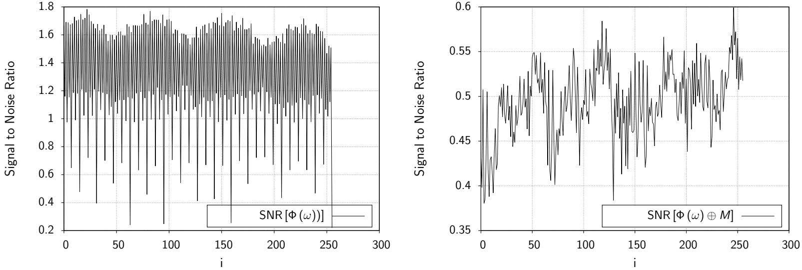
Empirical results

Addition of extrinsic independent Gaussian noise on the traces.



(a) Comparison of the attacks

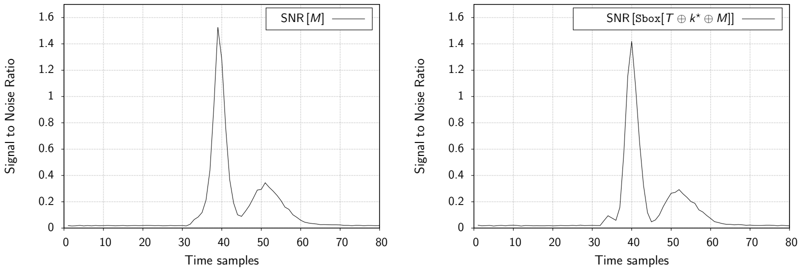
Analysis of the SNR



(a) SNR of the random index (b) SNR of the mask index

Figure : SNR of the sensitive points in the table recomputation.

Analysis of the SNR



(a) SNR of the random index

(b) SNR of the mask index

Figure : SNR of the sensitive points in the table recomputation.

Outline

Conclusion

Results

We have presented a high order attack better than the attack of minimal order in:

- Different leakage model,
- Real traces.

Perspective

Find the optimal combination to combine attack at different orders.

Thanks for your attention.

[AG01] Mehdi-Laurent Akkar and Christophe Giraud.
An Implementation of DES and AES Secure against Some Attacks.
In LNCS, editor, *Proceedings of CHES'01*, volume 2162 of *LNCS*, pages 309–318. Springer, May 2001.
Paris, France.

[BGHR14] Nicolas Bruneau, Sylvain Guilley, Annelie Heuser, and Olivier Rioul.
Masks Will Fall Off: Higher-Order Optimal Distinguishers.
In *ASIACRYPT*, volume 8874 of *LNCS*, pages 344–365. Springer, December 2014.
P. Sarkar and T. Iwata (Eds.): ASIACRYPT 2014, PART II.

[BGK04] Johannes Blömer, Jorge Guajardo, and Volker Krummel.
Provably Secure Masking of AES.
In Helena Handschuh and M. Anwar Hasan, editors, *Selected Areas in Cryptography*, volume 3357 of *Lecture Notes in Computer Science*, pages 69–83. Springer, 2004.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi.
Towards Sound Approaches to Counteract Power-Analysis Attacks.
In *CRYPTO*, volume 1666 of *LNCS*. Springer, August 15-19 1999. Santa Barbara, CA, USA. ISBN: 3-540-66347-9.

[Cor13] Jean-Sébastien Coron.
HTable countermeasure against side-channel attacks, 2013.
https://github.com/coron/htable.

[Cor14] Jean-Sébastien Coron.
Higher Order Masking of Look-Up Tables.
In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT*, volume 8441 of *Lecture Notes in Computer Science*, pages 441–458. Springer, 2014.

[GMN+11] Sylvain Guilley, Olivier Meynard, Maxime Nassar, Guillaume Duc, Philippe Hoogvorst, Houssem Maghrebi, Aziz Elaabid, Shivam Bhasin, Youssef Souissi, Nicolas Debande, Laurent Sauvage, and Jean-Luc Danger.
Vade Mecum on Side-Channels Attacks and Countermeasures for the Designer and the Evaluator.
In *DTIS (Design & Technologies of Integrated Systems)*, IEEE. IEEE, March 6-8 2011.
Athens, Greece. DOI: 10.1109/DTIS.2011.5941419 ; Online version:
`http://hal.archives-ouvertes.fr/hal-00579020/en/`.

[Mes00] Thomas S. Messerges.
Securing the AES Finalists Against Power Analysis Attacks.
In *Fast Software Encryption'00*, pages 150–164. Springer-Verlag, April 2000.
New York.

[PdHL09] Jing Pan, J. I. den Hartog, and Jiqiang Lu.
You cannot hide behind the mask: Power analysis on a provably secure *s*-box implementation.
In Heung Youl Youm and Moti Yung, editors, *WISA*, volume 5932 of *Lecture Notes in Computer Science*, pages 178–192. Springer, 2009.

[PR07] Emmanuel Prouff and Matthieu Rivain.
A Generic Method for Secure SBox Implementation.
In Sehun Kim, Moti Yung, and Hyung-Woo Lee, editors, *WISA*, volume 4867 of *Lecture Notes in Computer Science*, pages 227–244. Springer, 2007.

[RP10] Matthieu Rivain and Emmanuel Prouff.
Provably Secure Higher-Order Masking of AES.
In Stefan Mangard and François-Xavier Standaert, editors, *CHES*, volume 6225 of *LNCS*, pages 413–427. Springer, 2010.

[SVCO+10] François-Xavier Standaert, Nicolas Veyrat-Charvillon, Elisabeth Oswald, Benedikt Gierlichs, Marcel Medwed, Markus Kasper, and Stefan Mangard.
The World is Not Enough: Another Look on Second-Order DPA.
In *ASIACRYPT*, volume 6477 of *LNCS*, pages 112–129. Springer, December 5-9 2010.
Singapore.
http://www.dice.ucl.ac.be/~fstandae/PUBLIS/88.pdf.

[TWO13] Michael Tunstall, Carolyn Whitnall, and Elisabeth Oswald.
Masking Tables - An Underestimated Security Risk.
IACR Cryptology ePrint Archive, 2013:735, 2013.

Table Recomputation Code [Cor13]

```
for(i=0;i<(n-1);i++)
{
   for(k=0;k<K;k++)
      for(j=0;j<n;j++)
         Tp[k][j]}=T[k ^ a[i]][j];


   for(k=0;k<K;k++)
   {
      for(j=0;j<n;j++)
         T[k][j]=Tp[k][j];
      refresh(T[k],n);
   }
}
```