Institut
Mines-Telecom

STMicroelectronics

Taylor Expansion of Maximum Likelihood Attacks

Nicolas Bruneau[1,2], Sylvain Guilley[1,3], Annelie Heuser[1], Olivier Rioul[1], François-Xavier Standaert[4], Yannick Teglia[5]

[1] Télécom-ParisTech, Crypto & ComNum Group, Paris, FRANCE
[2] STMicroelectronics, AST division, Rousset, FRANCE
[3] Secure-IC S.A.S., Rennes, FRANCE
[4] Université Catholique de Louvain, Louvain-la-Neuve, BELGIQUE
[5] Gemalto, La Ciotat, FRANCE

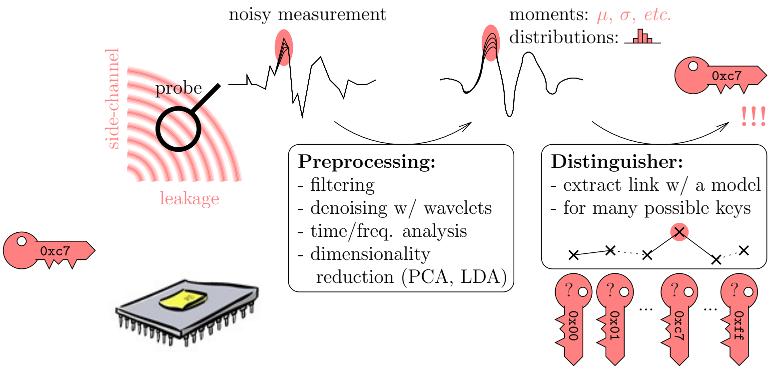
ASIACRYPT 2016 — Hanoi, Vietnam

Nicolas Bruneau, Sylvain Guilley,
Annelie Heuser, Olivier Rioul,
François-Xavier Standaert, Yannick Teglia

Outline

Outline

Side-Channel Analysis on Embedded Systems



$(\Omega - 1)$th-Order Masking: Principle

Aim

The sensitive variable Z is randomly split into Ω shares:
$\Rightarrow$ need random masks M_i , $0 < i < \Omega$

$$Z$$

$$Z \perp M_1 \perp \ldots \perp M_{\Omega-1} \qquad M_1 \qquad \cdots \qquad M_{\Omega-1}$$

$(\Omega - 1)$th-Order Masking: Principle

Aim

The sensitive variable Z is randomly split into Ω shares:
$\Rightarrow$ need random masks M_i , $0 < i < \Omega$

$$
\begin{tikzcd}
 & Z \arrow[dl] \arrow[d] \arrow[dr] \arrow[drr] & & \\
Z \perp M_1 \perp \ldots \perp M_{\Omega-1} & M_1 & \cdots & M_{\Omega-1}
\end{tikzcd}
$$

$(\Omega - 1)$th-Order Masking: Principle

Aim

The sensitive variable Z is randomly split into Ω shares:
$\Rightarrow$ need random masks M_i, $0 < i < \Omega$

$$Z$$

$$Z \perp M_1 \perp \ldots \perp M_{\Omega-1} \qquad M_1 \qquad \cdots \qquad M_{\Omega-1}$$

Consequence

Increases the minimum key-dependent statistical moment.

Shuffling: Principle

Aim

Randomize the order of execution
$\Rightarrow$ need a random permutation π

$\boxed{Z_1}$

Shuffling: Principle

Aim

Randomize the order of execution
$\Rightarrow$ need a random permutation π

$$Z_1 \longrightarrow Z_2$$

Shuffling: Principle

Aim

Randomize the order of execution
$\Rightarrow$ need a random permutation π

$$Z_1 \longrightarrow Z_2 \longrightarrow Z_3$$

Shuffling: Principle

Aim

Randomize the order of execution

$\Rightarrow$ need a random permutation π

$$Z_1 \longrightarrow Z_2 \longrightarrow Z_3 \longrightarrow Z_4$$

Shuffling: Principle

Aim

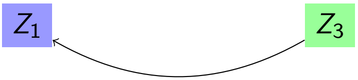
Randomize the order of execution
$\Rightarrow$ need a random permutation π

Z_3

Shuffling: Principle

Aim

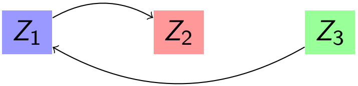
Randomize the order of execution
$\Rightarrow$ need a random permutation π



Shuffling: Principle

Aim

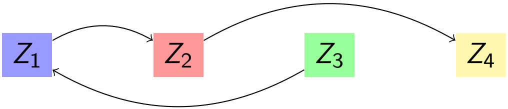
Randomize the order of execution
$\Rightarrow$ need a random permutation π



Shuffling: Principle

Aim

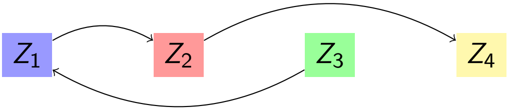
Randomize the order of execution
$\Rightarrow$ need a random permutation π



Shuffling: Principle

Aim

Randomize the order of execution
$\Rightarrow$ need a random permutation π



Consequences

The attacks are applied on the sum of the variables $\Rightarrow$ increases the algorithmic noise.

Protection Parameters

The security level of the protections depends on these parameters:

Masking

- Ω: the number of shares ($\Omega - 1$ masks);
- O: the order (i.e. the minimal key-dependent statistical moment).

Perfect masking scheme $\Leftrightarrow O = \Omega$.

Shuffling

- Π the size of the permutation.

Template Attacks

Template attacks are the most powerful in a information-theoretic sense [Chari et al., 2002].

Offline Profiling

The leakage model is learned:

- non-parametric methods (e.g. histogram, kernel methods...);
- parametric methods (e.g. mixture models).

Online Attack

Recover the key using the models by applying a maximum likelihood (ML) attack.

Parametric or Non-Parametric ?

Parametric

The only random part is the noise with known distribution.

- easy to estimate;
- shuffle and mask are known;
- many templates are learned.

Non-Parametric

Shuffle and masks are part of the noise.

- can be hard to estimate $\Rightarrow$ curse of dimensionality;
- shuffle and mask are unknown.

Parametric or Non-Parametric ?

Parametric

The only random part is the noise with known distribution.

- easy to estimate;
- shuffle and mask are known;
- many templates are learned.

Non-Parametric

Shuffle and masks are part of the noise.

- can be hard to estimate $\Rightarrow$ curse of dimensionality;
- shuffle and mask are unknown.

Notations for the Online attack

The attacks are applied on:

- Q queries (i.e. the traces).
- D dimension (i.e. the number of leakage samples);

A leakage measurement is $X = y(t, k^*, R) + N$ where:

- $y(t, k^*, R)$ is the deterministic part of the model;
- the secret key k^* and the plaintext t are n-bit words;
- R is the random countermeasure;
- N is a random Gaussian noise of variance σ^2.

Maximum Likelihood Attacks

Theorem (Maximum Likelihood [Bruneau et al., 2014])

When the model is known the optimal distinguisher (OPT) consists in maximizing the sum over all traces $q = 1, \ldots, Q$ of the log-likelihood:

$$\mathrm{LL} = \sum_{q=1}^{Q} \log \mathbb{E} \exp \frac{-\|x^{(q)} - y(t^{(q)}, k, R)\|^2}{2\sigma^2} \quad ,$$

where expectation $\mathbb{E}$ is applied to the random variable $R \in \mathcal{R}$ and $\|\cdot\|$ is the Euclidean norm on $\mathbb{R}^D$.

For convenience we let $\gamma = \frac{1}{2\sigma^2}$ be the SNR parameter.

Complexity in presence of Masking and Shuffling

$$\mathcal{O}\left(Q \cdot D \cdot (2^n)^{\Omega-1} \cdot \Pi! \right)$$

- number of traces
- dimension of the attack
- number of possible share values
- number of possible permutations

Complexity in presence of Masking and Shuffling

$$\mathcal{O}\left(Q \cdot D \cdot (2^n)^{\Omega-1} \cdot \Pi! \right)$$

- number of traces
- dimension of the attack
- number of possible share values
- number of possible permutations

Complexity in presence of Masking and Shuffling

$$\mathcal{O}\left(Q \cdot D \cdot (2^n)^{\Omega-1} \cdot \Pi! \right)$$

- number of traces
- dimension of the attack
- number of possible share values
- number of possible permutations

Complexity in presence of Masking and Shuffling

$$\mathcal{O}\left(Q \cdot D \cdot (2^n)^{\Omega-1} \cdot \Pi! \right)$$

- number of traces
- dimension of the attack
- number of possible share values
- number of possible permutations

Complexity in presence of Masking and Shuffling

$$\mathcal{O}\left(Q \cdot D \cdot (2^n)^{\Omega - 1} \cdot \Pi! \right)$$

- number of traces
- dimension of the attack
- number of possible share values
- number of possible permutations

Not computable for large Π !

Outline

Taylor Expansion of Optimal Attacks in Gaussian Noise

The optimal attack consists in maximizing the sum over all traces $q = 1, \ldots, Q$ of the log-likelihood:

$$\mathrm{LL} = \sum_{q=1}^{Q} \log \mathbb{E} \exp \frac{-\|x^{(q)} - y(t^{(q)}, k, R)\|^2}{2\sigma^2} \quad .$$

It can be rewritten using the cumulant generating function:

$$\mathrm{LL} = \sum_{q=1}^{Q} \sum_{\ell=1}^{+\infty} \frac{\kappa_\ell^{(q)}}{\ell!} (-\gamma)^\ell \quad ,$$

where $\kappa_\ell^{(q)}$ is the ℓth-order cumulant of $\|x^{(q)} - y(t^{(q)}, k, R)\|^2$.

High order Cumulants

The ℓth-order cumulant of $\|x - y(t,k,R)\|^2$ is given by:

$$\kappa_\ell = \mu_\ell - \sum_{\ell'=1}^{\ell-1} \binom{\ell-1}{\ell'-1} \kappa_{\ell'} \mu_{\ell-\ell'} \qquad (\ell \geq 1),$$

where μ_ℓ is the corresponding moment:

$$\mu_\ell = \mathbb{E}_R\left(\|x - y(t,k,R)\|^{2\ell}\right) \ .$$

Rounded Optimal Attack

Rounded Optimal Attack (ROPT_L)

The rounded optimal *Lth-degree attack* consists in maximizing the sum over all traces of the Lth-order Taylor expansion LL_L in the SNR of the log-likelihood :

$$\mathrm{LL}_L = \sum_{q=1}^{Q} \sum_{\ell=1}^{L} (-1)^\ell \kappa_\ell^{(q)} \frac{\gamma^\ell}{\ell!} \quad ,$$

and we have

$$\boxed{\mathrm{LL} = \mathrm{LL}_L + o(\gamma^L)} \quad .$$

Complexity Gain

- number of possible share values
- number of traces

$$\mathcal{O}\left(Q \cdot L \cdot \binom{D+L-1}{L} \cdot 2^{(\Omega-1)n} \cdot \binom{\Pi}{\min(\lceil \frac{\Pi}{2} \rceil, L)} \right)$$

- Factorial terms
  - dimension of the attack
  - degree of the Taylor Expansion
  - size of the permutation

Complexity Gain

- number of possible share values
- number of traces

$$\mathcal{O}\left(Q \cdot L \cdot \binom{D+L-1}{L} \cdot 2^{(\Omega-1)n} \cdot \binom{\Pi}{\min(\lceil \frac{\Pi}{2} \rceil, L)} \right)$$

- Factorial terms
  - dimension of the attack
  - degree of the Taylor Expansion
  - size of the permutation

Complexity Gain

- number of possible share values
- number of traces

$$\mathcal{O}\left(\boxed{Q} \cdot \boxed{L \cdot \binom{D+L-1}{L}} \cdot \boxed{2^{(\Omega-1)n}} \cdot \boxed{\binom{\Pi}{\min(\lceil \frac{\Pi}{2} \rceil, L)}} \right)$$

- Factorial terms
  - dimension of the attack
  - degree of the Taylor Expansion
  - size of the permutation

Complexity Gain

- number of possible share values
- number of traces

$$\mathcal{O}\left(\boxed{Q} \cdot \boxed{L \cdot \binom{D+L-1}{L}} \cdot \boxed{2^{(\Omega-1)n}} \cdot \boxed{\binom{\Pi}{\min(\lceil \frac{\Pi}{2} \rceil, L)}} \right)$$

- Factorial terms
  - dimension of the attack
  - degree of the Taylor Expansion
  - size of the permutation

Complexity Gain

- number of possible share values
- number of traces

$$\mathcal{O}\left(Q \cdot L \cdot \binom{D+L-1}{L} \cdot 2^{(\Omega-1)n} \cdot \binom{\Pi}{\min(\lceil \frac{\Pi}{2} \rceil, L)} \right)$$

- Factorial terms
  - dimension of the attack
  - degree of the Taylor Expansion
  - size of the permutation

Complexity Gain

- number of possible share values
- number of traces

$$\mathcal{O}\left(Q \cdot L \cdot \binom{D+L-1}{L} \cdot 2^{(\Omega-1)n} \cdot \binom{\Pi}{\min(\lceil \frac{\Pi}{2} \rceil, L)} \right)$$

- Factorial terms
  - dimension of the attack
  - degree of the Taylor Expansion
  - size of the permutation

Complexity Gain

- number of possible share values
- number of traces

$$\mathcal{O}\left(\boxed{Q} \cdot \boxed{L \cdot \binom{D+L-1}{L}} \cdot \boxed{2^{(\Omega-1)n}} \cdot \boxed{\binom{\Pi}{\min(\lceil \frac{\Pi}{2} \rceil, L)}} \right)$$

- Factorial terms
  - dimension of the attack
  - degree of the Taylor Expansion
  - size of the permutation

Reduces to small constants when $L \ll D$

Outline

Implementation of Masking Schemes

In masking schemes, while the implementation of the linear parts is obvious, that of the non linear parts is more difficult.

- algebraic methods [Blömer et al., 2004];
- global look-up table method [Prouff and Rivain, 2007];
- table recomputation methods which precompute a masked S-box stored in a table [Chari et al., 1999].

In [Coron, 2014] a table recomputation scheme secure at order $\Omega - 1$ was presented.

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;
- 2-stage CPA attack [Pan et al., 2009];
- improved $(2^n + 1)$-variate 2nd-order attack on the input [Bruneau et al., 2014].

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

Make the index of the loop unknown, use some random permutation φ.

Leakages

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

Leakages

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

- second-order Correlation Power Analysis 2O-CPA;
- OPTimal distinguisher OPT;
  - Rounded OPTimal Distinguisher ROPT_2, ROPT_4

Bi-Variate Attacks

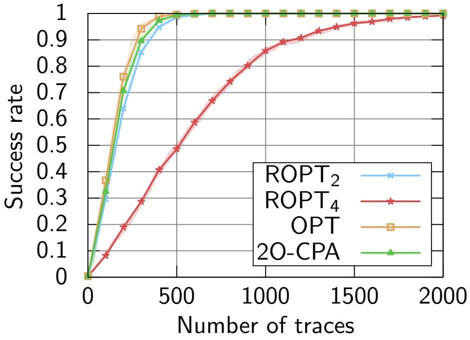


Figure: $\sigma = 1$

Bi-Variate Attacks

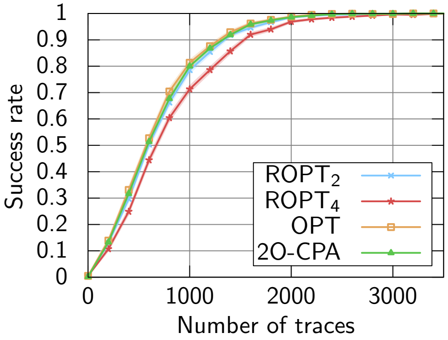


Figure: $\sigma = 2$

Leakages, with Table Recomputation

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

Leakages, with Table Recomputation

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

- optimal distinguisher NOT computable due to the term 2^n!

Leakages, with Table Recomputation

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 $\varphi(\omega) \in \{\varphi(0), \varphi(1), \dots, \varphi(2^n - 1)\}$ **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

- third order attack MVA_{TR} [Bruneau et al., 2015];
- Rounded Optimal Distinguisher ROPT_3.

Complexity of the Case Study

Attack	Time (seconds)	Computational Complexity
2O-CPA	39	$\mathcal{O}(Q)$
MVA_{TR}	130	$\mathcal{O}(Q \cdot 2^n)$
ROPT_3	2495	$\mathcal{O}\left(Q \cdot 2^{2n}\right)$
OPT_{2O}	9473	$\mathcal{O}(Q \cdot 2^n)$
OPT	Not computable	$\mathcal{O}\left(Q \cdot 2^n \cdot 2^n! \cdot \left(2^{n+1} + 2\right)\right)$

The time of execution have been computed on a Intel Xeon X5660.

$(2^{n+1}+2)$-Variate Attacks on Shuffled Table Recomputation

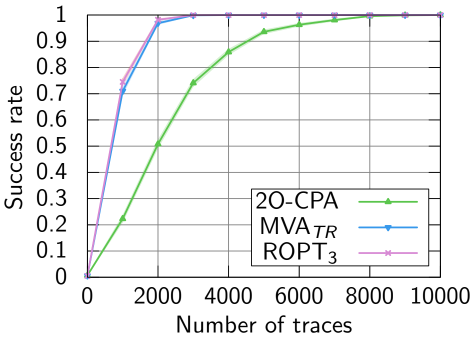


Figure: $\sigma = 3$

$(2^{n+1}+2)$-Variate Attacks on Shuffled Table Recomputation

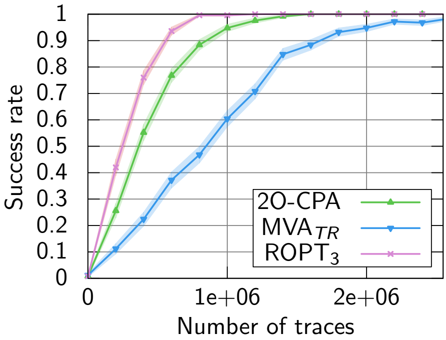


Figure: $\sigma = 12$

$(2^{n+1}+2)$-Variate Attacks on Shuffled Table Recomputation

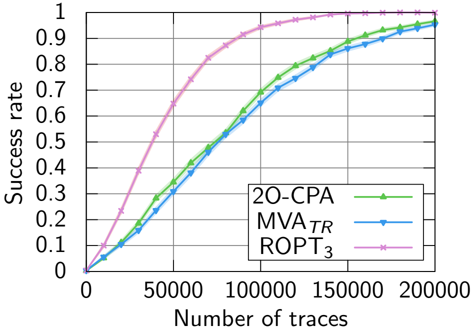


Figure: $\sigma = 8$

$(2^{n+1}+2)$-Variate Attacks on Shuffled Table Recomputation

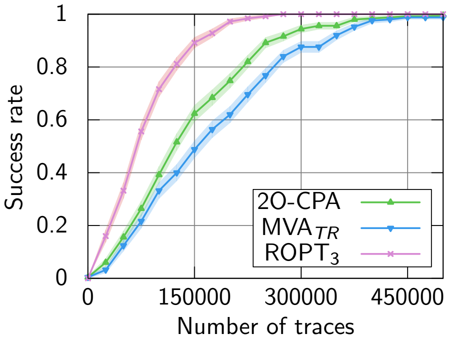


Figure: $\sigma = 9$

$(2^{n+1}+2)$-Variate Attacks on Shuffled Table Recomputation

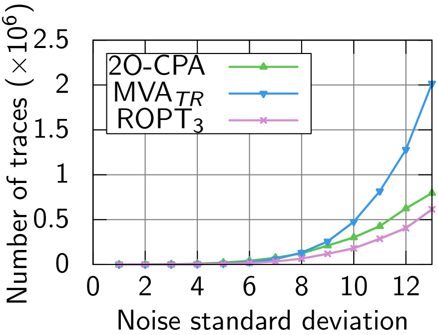


Figure: Number of traces to reach 80% of success

Conclusion

Results

We have presented a practical, truncated version of the theoretical, optimal distinguisher:

- becomes efficient;
- remains effective.

Perspective

How to quantify the accuracy of the approximation?

Conclusion

Results

We have presented a practical, truncated version of the theoretical, optimal distinguisher:

- becomes efficient;
- remains effective.

Perspective

How to choose the degree of the Taylor Expansion?

Thank you for your attention.

[Blömer et al., 2004] Blömer, J., Guajardo, J., and Krummel, V. (2004).
Provably Secure Masking of AES.
In Handschuh, H. and Hasan, M. A., editors, *Selected Areas in Cryptography*, volume 3357 of *Lecture Notes in Computer Science*, pages 69–83. Springer.

[Bruneau et al., 2014] Bruneau, N., Guilley, S., Heuser, A., and Rioul, O. (2014).
Masks Will Fall Off: Higher-Order Optimal Distinguishers.
In *ASIACRYPT*, volume 8874 of *LNCS*, pages 344–365. Springer.
P. Sarkar and T. Iwata (Eds.): ASIACRYPT 2014, PART II.

[Bruneau et al., 2015] Bruneau, N., Guilley, S., Najm, Z., and Teglia, Y. (2015).
Multi-variate high-order attacks of shuffled tables recomputation.
In Güneysu, T. and Handschuh, H., editors, *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, volume 9293 of *Lecture Notes in Computer Science*, pages 475–494. Springer.

[Chari et al., 1999] Chari, S., Jutla, C. S., Rao, J. R., and Rohatgi, P. (1999).
Towards Sound Approaches to Counteract Power-Analysis Attacks.
In *CRYPTO*, volume 1666 of *LNCS*. Springer.
Santa Barbara, CA, USA. ISBN: 3-540-66347-9.

[Chari et al., 2002] Chari, S., Rao, J. R., and Rohatgi, P. (2002).
Template Attacks.
In *CHES*, volume 2523 of *LNCS*, pages 13–28. Springer.
San Francisco Bay (Redwood City), USA.

[Coron, 2014] Coron, J.-S. (2014).
Higher Order Masking of Look-Up Tables.
In Nguyen, P. Q. and Oswald, E., editors, *EUROCRYPT*, volume 8441 of *Lecture Notes in Computer Science*, pages 441–458. Springer.

[Pan et al., 2009] Pan, J., den Hartog, J. I., and Lu, J. (2009).
You cannot hide behind the mask: Power analysis on a provably secure *S*-box implementation.
In Youm, H. Y. and Yung, M., editors, *Information Security Applications, 10th International Workshop, WISA 2009, Busan, Korea, August 25-27, 2009, Revised Selected Papers*, volume 5932 of *Lecture Notes in Computer Science*, pages 178–192. Springer.

[Prouff and Rivain, 2007] Prouff, E. and Rivain, M. (2007).
A Generic Method for Secure SBox Implementation.
In Kim, S., Yung, M., and Lee, H.-W., editors, *WISA*, volume 4867 of *Lecture Notes in Computer Science*, pages 227–244. Springer.