Mixing Additive and Multiplicative Masking for Probing Secure Polynomial Evaluation Methods

Axel Mathieu-Mahias and Michaël Quisquater

University of Versailles (UVSQ)

CHES’18 September
The Concept of Masking

- Side-channel analysis
  - Information leak through physical leakages
  - Data and physical leakages are dependent
The Concept of Masking

- Side-channel analysis
  - Information leak through physical leakages
  - Data and physical leakages are dependent

- The masking countermeasure
  1. Randomly split every variable into several shares
  2. Secure the processing through internal operations
The Concept of Masking

- Side-channel analysis
  - Information leak through physical leakages
  - Data and physical leakages are dependent

- The masking countermeasure
  1. Randomly split every variable into several shares
  2. Secure the processing through internal operations

- Higher-order masking
  - More than 2 shares
  - Sound countermeasure
The Probing Model [ISW03]

Inputs

$(x_1, \ldots, x_d)$

$(y_1, \ldots, y_d)$

Internals

$Sec-Op_1$

$Sec-Op_2$

$Sec-Op_3$

Outputs

$(z_1, \ldots, z_d)$

Adversary observations

$\Omega = (I_1, I_2, \ldots I_t)$
The Probing Model [ISW03]

Adversary observations
\[ \Omega = (I_1, I_2, \ldots I_t) \]

Is any set of \( t \) observations independent of sensitive variables?
The Probing Model [ISW03]

- Two security notions: \textbf{t-NI} and \textbf{t-SNI} [BBDFG15]
- \textbf{t-SNI} transformations can be composed safely
State of the Art of Masking S-boxes (Additive Masking)

- Split every variable $x$ into $d = t + 1$ shares such that
  $$x_1 \oplus x_2 \oplus \ldots \oplus x_d = x$$

- Processing of **linear transformations** : very efficient
- Processing of **multiplications** : much more expensive
State of the Art of Masking S-boxes (Additive Masking)

- Split every variable $x$ into $d = t + 1$ shares such that
  $$x_1 \oplus x_2 \oplus \ldots \oplus x_d = x$$

- Processing of **linear transformations** : very efficient
- Processing of **multiplications** : much more expensive

**AES** : [RP10]

$$S_{AES}(x) : x \mapsto x^{254} \text{ over } \mathbb{F}_{2^8}$$

**Generic case** : [CGPQQR12]

$$S(x) : x \mapsto \sum_{i=0}^{2^n-1} a_i x^i \text{ over } \mathbb{F}_{2^n}$$
State of the Art of Masking S-boxes

- Masking schemes in additive encoding
  
  FSE’12 : Carlet et al.
  CHES’13 : Roy and Vivek
  CHES’14 : Coron et al.
State of the Art of Masking S-boxes

- Masking schemes in additive encoding
  - FSE’12 : Carlet et al.
  - CHES’13 : Roy and Vivek
  - CHES’14 : Coron et al.

- Masking schemes in other encodings
  - CHES’11 : Prouff and Roche
  - CRYPTO’15 : Carlet et al.
  - EUROCRYPT’14 : Coron
  - EUROCRYPT’15 : Balasch et al.
  - CHES’16 : Goudarzi and Rivain
The use of several encodings simultaneously

GPQ : masking scheme for power functions \ [GPQ11] 
- Mixes additive and multiplicative masking
The use of several encodings simultaneously

**GPQ**: masking scheme for **power functions** [GPQ11]
- Mixes **additive** and **multiplicative** masking

**The idea**
- Linear transformations: efficient in additive masking
- Multiplications: efficient in multiplicative masking
The use of several encodings simultaneously

GPQ: masking scheme for power functions [GPQ11]
- Mixes additive and multiplicative masking

The idea
- Linear transformations: efficient in additive masking
- Multiplications: efficient in multiplicative masking

The scheme
- Secure processing of a Dirac function (Secure-dirac)
- Transformations to switch from additive into multiplicative masking (AMtoMM) and conversely (MMtoAM)
GPQ : Masking Scheme for Power Functions

\[ x \xrightarrow{\text{Sec-dirac}} (x + \delta(x))^\alpha \xrightarrow{\text{AMtoMM}} \]

\[ x^\alpha \xleftarrow{\text{MMtoAM}} \]
GPQ : Masking Scheme for Power Functions

Our first contribution

GPQ t-NI → GPQ t-SNI
### Our Issue and Our Proposals

**How to extend GPQ to evaluate polynomials?**
Our Issue and Our Proposals

How to extend GPQ to evaluate polynomials?

Our issues

- **Adding monomials**: not efficient in multiplicative masking
- **Converting every monomials** back in additive masking before adding them: not efficient
Our Issue and Our Proposals

How to extend GPQ to evaluate polynomials?

Our issues

- **Adding monomials**: not efficient in multiplicative masking
- **Converting every monomials** back in additive masking before adding them: not efficient

Our t-SNI proposals

1. One method based on the cyclotomic method [CGPQR12]
2. One method based on our first proposal and the CRV method [CRV14]
Our First Proposal: The Alternate Cyclotomic Method

Reminder of the Cyclotomic Method  [CGPQR12]

- The cyclotomic class of $\alpha$: $C_{\alpha} = \{\alpha \cdot 2^j \mod 2^n - 1; j < n\}$
The cyclotomic method

Our First Proposal: The Alternate Cyclotomic Method

Reminder of the Cyclotomic Method [CGPQR12]

- The cyclotomic class of $\alpha$: $C_\alpha = \{\alpha \cdot 2^j \mod 2^n - 1; j < n\}$
- Any n-bit S-box can be expressed as

$$S(x) = a_0 + \left( \sum_{i=1}^{q} L_i(x^{\alpha_i}) \right) + a_{2^n-1}x^{2^n-1}$$

where $L_i(x) = \sum_j a_{i,j}x^{2^j}$ and $q$ is the number of distinct cyclotomic classes
Our First Proposal: The Alternate Cyclotomic Method

Reminder of the Cyclotomic Method [CGPQR12]

- The cyclotomic class of $\alpha: C_\alpha = \{\alpha \cdot 2^j \mod 2^n - 1; j < n\}$
- Any n-bit S-box can be expressed as

$$S(x) = a_0 + \left( \sum_{i=1}^{q} L_i(x^{\alpha_i}) \right) + a_{2^n-1}x^{2^n-1}$$

where $L_i(x) = \sum_j a_{i,j}x^{2^j}$ and $q$ is the number of distinct cyclotomic classes

- Deriving the $x^{\alpha_i}$'s requires multiplications: expensive in additive masking.
The Alternate Cyclotomic Method

Our First Proposal: The Alternate Cyclotomic Method

\[
S(x) \rightarrow \text{Linear Processing} \rightarrow \text{Sec-dirac} \rightarrow \text{AMtoMM} \rightarrow (x + \delta(x))^{\alpha_1} \rightarrow \cdots \rightarrow (x + \delta(x))^{\alpha_q} \rightarrow \text{MMtoAM} \rightarrow \cdots \rightarrow \text{MMtoAM} \rightarrow L_1((x + \delta(x))^{\alpha_1}) \rightarrow \cdots \rightarrow L_q((x + \delta(x))^{\alpha_q})
\]

: In multiplicative masking
Our First Proposal: The Alternate Cyclotomic Method

The alternate cyclotomic method is \( t\)-SNI
The cyclotomic method vs The alternate cyclotomic method

Assembly Language Performances : 8-bit Architecture

Costs (in clock cycles) of evaluating S-boxes of size $4 \leq n \leq 8$ with the cyclotomic method and our proposal

<table>
<thead>
<tr>
<th>Method</th>
<th>Order</th>
<th>$n$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Our proposal</td>
<td>1</td>
<td>83</td>
</tr>
<tr>
<td>Original</td>
<td></td>
<td>132</td>
</tr>
<tr>
<td>Our proposal</td>
<td>2</td>
<td>276</td>
</tr>
<tr>
<td>Original</td>
<td></td>
<td>174</td>
</tr>
<tr>
<td>Our proposal</td>
<td>3</td>
<td>477</td>
</tr>
<tr>
<td>Original</td>
<td></td>
<td>293</td>
</tr>
</tbody>
</table>
Our Second Proposal: The Alternate CRV Method

Reminder of the original CRV Method \cite{CRV14}

- Express any n-bit S-box as

\[
S(x) = \sum_{i=1}^{k-1} p_i(x) \cdot q_i(x) + p_k(x)
\]

where monomials of \( p_i(x), q_i(x) \) belong to \( x^L \) with \( L \leftarrow \bigcup_{i=1}^{l} C_{\alpha_i} \)
The original CRV method

Our Second Proposal: The Alternate CRV Method

Reminder of the original CRV Method [CRV14]

Express any n-bit S-box as

\[ S(x) = \sum_{i=1}^{k-1} p_i(x) \cdot q_i(x) + p_k(x) \]

where monomials of \( p_i(x), q_i(x) \) belong to \( x^L \) with \( L \leftarrow \bigcup_{i=1}^{l} C_{\alpha_i} \)

Evaluation in two steps

1. Evaluating \( q_i(x), p_i(x) \) requires \( l - 2 \) multiplications
2. Evaluating \( S(x) \) requires \( k - 1 \) multiplications
Our Second Proposal: The Alternate CRV Method

Reminder of the original CRV Method \([\text{CRV14}]\)

- Express any n-bit S-box as

\[ S(x) = \sum_{i=1}^{k-1} p_i(x) \cdot q_i(x) + p_k(x) \]

where monomials of \(p_i(x), q_i(x)\) belong to \(x^L\) with \(L \leftarrow \bigcup_{i=1}^{l} C_{\alpha_i} \)

- Evaluation in two steps
  1. Evaluating \(q_i(x), p_i(x)\) requires \(l - 2\) multiplications
  2. Evaluating \(S(x)\) requires \(k - 1\) multiplications

- Remark: trade-off between \(l\) and \(k\)
Our alternate approach

Our Second Proposal: The Alternate CRV Method

\[
S(x) = \sum_{i=1}^{k-1} p_i(x) \cdot q_i(x) + p_k(x)
\]

Our evaluation method

1. Evaluating \(q_i(x), p_i(x)\) with our t-SNI alternate cyclotomic method
2. Evaluating \(S(x)\) in additive masking (unchanged)
Our alternate approach

Our Second Proposal: The Alternate CRV Method

Our evaluation method

1. Evaluating $q_i(x), p_i(x)$ with our t-SNI alternate cyclotomic method
2. Evaluating $S(x)$ in additive masking (unchanged)

Remarks

- **More choices** of cyclotomic classes to build $x^L$
- **Larger sets** $L \leftarrow \bigcup_{i=1}^{l} C_{\alpha_i}$ can be considered
- The alternate CRV method is **t-SNI**

\[
S(x) = \sum_{i=1}^{k-1} p_i(x) \cdot q_i(x) + p_k(x)
\]
Assembly Language Performances: 8-bit Architecture

Costs (in clock cycles) of evaluating S-boxes of size $4 \leq n \leq 8$ with the CRV method and our alternate proposal

<table>
<thead>
<tr>
<th>Method</th>
<th>Order</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our proposal</td>
<td>1</td>
<td>127</td>
<td>402</td>
<td>559</td>
<td>713</td>
<td>972</td>
</tr>
<tr>
<td>Original CRV</td>
<td>1</td>
<td>88</td>
<td>624</td>
<td>780</td>
<td>1092</td>
<td>1560</td>
</tr>
<tr>
<td>Our proposal</td>
<td>2</td>
<td>276</td>
<td>939</td>
<td>1296</td>
<td>1685</td>
<td>2300</td>
</tr>
<tr>
<td>Original CRV</td>
<td>2</td>
<td>204</td>
<td>1416</td>
<td>1770</td>
<td>2478</td>
<td>3540</td>
</tr>
<tr>
<td>Our proposal</td>
<td>3</td>
<td>477</td>
<td>1668</td>
<td>2305</td>
<td>3012</td>
<td>4117</td>
</tr>
<tr>
<td>Original CRV</td>
<td>3</td>
<td>368</td>
<td>2528</td>
<td>3160</td>
<td>4424</td>
<td>6320</td>
</tr>
</tbody>
</table>
Conclusion

1. GPQ t-NI → GPQ t-SNI
Conclusion

1. **GPQ t-NI → GPQ t-SNI**

2. The **Alternate cyclotomic method**
   - Extends GPQ to polynomial evaluations
   - Three times faster than the original method
   - Satisfies the t-SNI property
GPQ t-NI $\rightarrow$ GPQ t-SNI

The **Alternate cyclotomic method**
- Extends GPQ to polynomial evaluations
- Three times faster than the original method
- Satisfies the t-SNI property

The **Alternate CRV method**
- Uses Alternate cyclotomic for one evaluation step
- New sets of parameters can be derived
- Outperforms the original method in most scenarios
- Satisfies the t-SNI property
Thanks for your attention!