Fast Evaluation of Polynomials over Binary Finite Fields and Application to Side-channel Countermeasures

Jean-Sébastien Coron[1], Arnab Roy[1,2], **Srinivas Vivek**[1]

[1]**University of Luxembourg**

[2]DTU, Denmark

September 25, 2014

Outline

- Background & previous work
- Our results:
  - new polynomial evaluation algorithm
  - improved generic lower bound
- Future work

Outline

- Background & previous work

Motivation: Masking

- **Masking**: effective countermeasure for *block ciphers* against *DPA attacks.*

Motivation: Masking

- **Masking**: effective countermeasure for *block ciphers* against *DPA attacks.*
- Approach: to *split* (secret share) every *sensitive* variable x.
  - $x = x_0 \perp x_1 \perp \ldots \perp x_d$.
  - $\perp$: $\oplus$, or $+$ over $\mathbb{F}_{2^n}$.
  - **Masking Orde**r: d.
  - **Order of security**: $t \leq d$.

Motivation: Masking

- **Masking**: effective countermeasure for *block ciphers* against *DPA attacks.*
- Approach: to *split* (secret share) every *sensitive* variable x.
  - $x = x_0 \perp x_1 \perp \ldots \perp x_d$.
  - $\perp$: $\oplus$, or $+$ over $\mathbb{F}_{2^n}$.
  - **Masking Orde**r: d.
  - **Order of security**: $t \leq d$.
- **Soundness**: attack complexity is exponential w.r.t. t.

Higher-Order Masking

- Higher-order attacks are feasible [Messerges, CHES 2000].

Higher-Order Masking

- Higher-order attacks are feasible [Messerges, CHES 2000].
- Both customized and generic countermeasures exists.

Higher-Order Masking

- Higher-order attacks are feasible [Messerges, CHES 2000].
- Both customized and generic countermeasures exists.
- Generic higher-order masking schemes:
  - arbitrary block ciphers (S-boxes).
  - arbitrary masking order (i.e., shares).

Generic Higher-Order Masking Schemes

1. Prouff and Roche scheme (CHES 2011)
   - based on MPC techniques.

Generic Higher-Order Masking Schemes

1. Prouff and Roche scheme (CHES 2011)
   - based on MPC techniques.
2. CGPQR scheme by *Carlet et al.* (FSE 2012)
   - based on polynomial representation of S-boxes.

Generic Higher-Order Masking Schemes

1. Prouff and Roche scheme (CHES 2011)
   - based on MPC techniques.
2. CGPQR scheme by *Carlet et al.* (FSE 2012)
   - based on polynomial representation of S-boxes.
3. Table recompuation method by *Coron* (EUROCRYPT 2014)
   - based on randomized masking tables.

Generic Higher-Order Masking Schemes

1. Prouff and Roche scheme (CHES 2011)
   - based on MPC techniques.
2. CGPQR scheme by *Carlet et al.* (FSE 2012)
   - based on polynomial representation of S-boxes.
3. Table recompuation method by *Coron* (EUROCRYPT 2014)
   - based on randomized masking tables.

- Other specialized higher-order schemes:
  - GPQ scheme by *Genelle et al.* (CHES 2011): mainly for AES.

CGPQR H-O Masking Scheme

- Based on the probing circuit model by [ISW, CRYPTO 2003] and later extended by [PR, CHES 2010].

CGPQR H-O Masking Scheme

- Based on the probing circuit model by [ISW, CRYPTO 2003] and later extended by [PR, CHES 2010].
- Provides t^{th}order security when $d \geq 2t$.

CGPQR H-O Masking Scheme

- Based on the probing circuit model by [ISW, CRYPTO 2003] and later extended by [PR, CHES 2010].
- Provides t^{th}order security when $d \geq 2t$.
- Advantages:
  - More efficient than [PR11], comparable to [Coron14].
  - Smaller memory and randomness requirement than [Coron14].

CGPQR H-O Masking Scheme

- Based on the probing circuit model by [ISW, CRYPTO 2003] and later extended by [PR, CHES 2010].
- Provides t^{th}order security when $d \geq 2t$.
- Advantages:
  - More efficient than [PR11], comparable to [Coron14].
  - Smaller memory and randomness requirement than [Coron14].
- Recent works: [CPRR, FSE 2013], [RV, CHES 2013].

CGPQR Scheme (Cont'd)

- Main challenge for masking block ciphers: **masking of S-boxes**.

CGPQR Scheme (Cont'd)

- Main challenge for masking block ciphers: **masking of S-boxes**.
- Reason: $\mathbb{F}_2$-linear/-affine functions are easy to mask:
  - $f_{lin}(x) = f_{lin}(x_0 + \cdots + x_d) = f_{lin}(x_0) + \cdots + f_{lin}(x_d)$.

CGPQR Scheme (Cont'd)

- Main challenge for masking block ciphers: **masking of S-boxes**.
- Reason: $\mathbb{F}_2$-linear/-affine functions are easy to mask:
  - $f_{lin}(x) = f_{lin}(x_0 + \cdots + x_d) = f_{lin}(x_0) + \cdots + f_{lin}(x_d)$.
- Squaring is $\mathbb{F}_2$-linear in $\mathbb{F}_{2^n}$: $(a+b)^2 = a^2 + b^2$.

CGPQR Scheme (Cont'd)

- An (n, m)-S-box ($m \leq n$) can be identified with $f : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$.

CGPQR Scheme (Cont'd)

- An (n, m)-S-box ($m \leq n$) can be identified with $f : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$.
- By Lagrange interpolation,
  - $f(\cdot)$ can be (uniquely) represented by $P(x) \in \mathbb{F}_{2^n}[x]$, $\deg(P(x)) \leq 2^n - 1$.

CGPQR Scheme (Cont'd)

- An (n, m)-S-box ($m \leq n$) can be identified with $f : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$.
- By Lagrange interpolation,
  - $f(\cdot)$ can be (uniquely) represented by $P(x) \in \mathbb{F}_{2^n}[x]$, $\deg(P(x)) \leq 2^n - 1$.
- Masking an S-box $\Longrightarrow$ securely evaluating the corresponding polynomial with shares.

CGPQR Scheme (Cont'd)

- Task is to evaluate $P(x)$ on (shared) input $(x_0, \cdots, x_d)$.

CGPQR Scheme (Cont'd)

- Task is to evaluate $P(x)$ on (shared) input $(x_0, \cdots, x_d)$.
- To evaluate any polynomial $P(x) \in \mathbb{F}_{2^n}[x]$, we need:
  - **Linear operations**: (polynomial) addition, multiplication by a scalar, (polynomial) squaring.
  - **Non-Linear Multiplications** (NLMs).

CGPQR Scheme (Cont'd)

- Task is to evaluate $P(x)$ on (shared) input $(x_0, \cdots, x_d)$.
- To evaluate any polynomial $P(x) \in \mathbb{F}_{2^n}[x]$, we need:
  - **Linear operations**: (polynomial) addition, multiplication by a scalar, (polynomial) squaring.
  - **Non-Linear Multiplications** (NLMs).
- Each step above to be performed securely on the shares:
  - Linear operations with shares are cheap: $O(d)$ time and randomness.
  - NLMs with shares are expensive: $O(d^2)$ time and randomness.

$\mathbb{F}_{2^n}$-Polynomial Evaluation: Cost Model

- To evaluate any polynomial $P(x) \in \mathbb{F}_{2^n}[x]$, given x.

$\mathbb{F}_{2^n}$-Polynomial Evaluation: Cost Model

- To evaluate any polynomial $P(x) \in \mathbb{F}_{2^n}[x]$, given x.
- **Ignore**: (polynomial) additions, scalar multiplications, (polynomial) squarings.

$\mathbb{F}_{2^n}$-Polynomial Evaluation: Cost Model

- To evaluate any polynomial $P(x) \in \mathbb{F}_{2^n}[x]$, given x.
- **Ignore**: (polynomial) additions, scalar multiplications, (polynomial) squarings.
- **Count**: non-linear (polynomial) multiplications.

$\mathbb{F}_{2^n}$-Polynomial Evaluation: Cost Model

- To evaluate any polynomial $P(x) \in \mathbb{F}_{2^n}[x]$, given x.
- **Ignore**: (polynomial) additions, scalar multiplications, (polynomial) squarings.
- **Count**: non-linear (polynomial) multiplications.
- Example: Consider $q(x) \neq r(x) \in \mathbb{F}_{2^n}[x]$, $c \in \mathbb{F}_{2^n}$,
  - **ignore**: $q(x) + r(x)$, $c \cdot q(x)$, $(q(x))^2$
  - **count**: $q(x) \times r(x)$

Previous Evaluation Methods

1. **Cyclotomic Class Method** [CGQPR12],
   - worst-case complexity: atleast $2^n/n$ NLMs.

Previous Evaluation Methods

1. **Cyclotomic Class Method** [CGQPR12],
   - worst-case complexity: atleast $2^n/n$ NLMs.
2. **Parity-Split Method** [CGQPR12],
   - worst-case complexity: $1.5 \cdot \sqrt{2^n}$ NLMs.

Previous Evaluation Methods

1. **Cyclotomic Class Method** [CGQPR12],
   - worst-case complexity: atleast $2^n/n$ NLMs.
2. **Parity-Split Method** [CGQPR12],
   - worst-case complexity: $1.5 \cdot \sqrt{2^n}$ NLMs.
3. **Divide-and-Conquer Method** [PS73, RV13],
   - *non-generic*: degree $2^n = N \approx \sqrt{N}\left(2^i - 1\right)$.
   - complexity:$\approx \sqrt{2^n}$ NLMs.

Outline

- Our results

Our Results

- New polynomial evaluation algorithm (over $\mathbb{F}_{2^n}$):
  - (Heuristic) worst-case complexity: $\approx 2 \cdot \sqrt{\frac{2^n}{n}}$ NLMs.
  - Previous best: $O\left(\sqrt{2^n}\right)$ NLMs.

Our Results

- New polynomial evaluation algorithm (over $\mathbb{F}_{2^n}$):
  - (Heuristic) worst-case complexity: $\approx 2 \cdot \sqrt{\frac{2^n}{n}}$ NLMs.
  - Previous best: $O\left(\sqrt{2^n}\right)$ NLMs.
- New generic lower bound on evaluation complexity:
  - Lower bound: $\approx \sqrt{\frac{2^n}{n}}$ NLMs.
  - Previous best: $\approx \log_2 n$ NLMs.

Comparison of Generic Methods

n	4	5	6	7	8	9	10
Cyclotomic-Class method [CGPQR12]	3	5	11	17	33	53	105
Parity-Split method [CGPQR12]	4	6	10	14	22	30	46
This work	**2**	**4**	**5**	**7**	**10**	**14**	**19**

Table: Counting non-linear multiplications

Application to S-boxes

	S-box				
Method	DES	PRESENT	SERPENT	CAMELLIA	CLEFIA
Cyclo.-Class method [CGPQR12]	11	3	3	33	33
Parity-Split [CGPQR12]	10	4	4	22	22
Roy-Vivek [RV13]	7	3	3	15	15,16
This work	**4**	**2**	**2**	**10**	**10**

Table: Number of NLMs required for the CGPQR masking scheme.

Our Results: Evaluation Method

1. Precompute a "closed" set $x^L = \left\{x^i \mid i \in L\right\}$ of monomials,
   - "closed" w.r.t. squaring.

Our Results: Evaluation Method

1. Precompute a "closed" set $x^L = \left\{x^i \mid i \in L\right\}$ of monomials,
   - "closed" w.r.t. squaring.
2. Generate $t-1$ random polynomials $q_i(x) \xleftarrow{\$} \mathcal{P}(x^L)$.

Our Results: Evaluation Method

1. Precompute a "closed" set $x^L = \{x^i \mid i \in L\}$ of monomials,
   - "closed" w.r.t. squaring.
2. Generate $t-1$ random polynomials $q_i(x) \xleftarrow{\$} \mathcal{P}(x^L)$.
3. Find t polynomials $p_i(x) \in \mathcal{P}(x^L)$ such that

$$P(x) = \sum_{i=1}^{t-1} p_i(x) \cdot q_i(x) + p_t(x).$$

Our Results: Evaluation Method

1. Precompute a "closed" set $x^L = \{x^i \mid i \in L\}$ of monomials,
   - "closed" w.r.t. squaring.
2. Generate $t-1$ random polynomials $q_i(x) \xleftarrow{\$} \mathcal{P}(x^L)$.
3. Find t polynomials $p_i(x) \in \mathcal{P}(x^L)$ such that

$$P(x) = \sum_{i=1}^{t-1} p_i(x) \cdot q_i(x) + p_t(x).$$

4. Solve a linear system for the unknown coefficients,
   - similar to the Lagrange interpolation technique.

Evaluation Method: Analysis

- *Heuristic*: full rank if $t \cdot |L| \geq 2^n$.

Evaluation Method: Analysis

- *Heuristic*: full rank if $t \cdot |L| \geq 2^n$.
- Total NLMs: $N_{mult} \approx \ell + t$.

Evaluation Method: Analysis

- *Heuristic*: full rank if $t \cdot |L| \geq 2^n$.
- Total NLMs: $N_{mult} \approx \ell + t$.
- Optimal values: $t \approx \ell \approx \sqrt{\frac{2^n}{n}}$.
  - $N_{mult} \approx 2 \cdot \sqrt{\frac{2^n}{n}}$.

Evaluation Method: Analysis

- *Heuristic*: full rank if $t \cdot |L| \geq 2^n$.
- Total NLMs: $N_{mult} \approx \ell + t$.
- Optimal values: $t \approx \ell \approx \sqrt{\frac{2^n}{n}}$.
  - $N_{mult} \approx 2 \cdot \sqrt{\frac{2^n}{n}}$.
- **Open problem**: existence of L, and condition for full rank.

Evaluation Method: Optimization

- Example: DES $(6, 4)$-bit S-boxes.
  - Ignore leading two bits $\implies 2^{128}$ possible representations.

Evaluation Method: Optimization

- Example: DES $(6, 4)$-bit S-boxes.
  - Ignore leading two bits $\implies 2^{128}$ possible representations.
  - Choose $L = C_0 \cup C_1 \cup C_3 \cup C_7$, and $q_1(x), q_2(x) \xleftarrow{\$} \mathcal{P}(x^L)$.

Evaluation Method: Optimization

- Example: DES $(6, 4)$-bit S-boxes.
  - Ignore leading two bits $\Longrightarrow 2^{128}$ possible representations.
  - Choose $L = C_0 \cup C_1 \cup C_3 \cup C_7$, and $q_1(x), q_2(x) \xleftarrow{\$} \mathcal{P}(x^L)$.
  - Find the decomposition: $P(x) = p_1(x) \cdot q_1(x) + p_2(x) \cdot q_2(x) + p_3(x)$.

Evaluation Method: Optimization

- Example: DES $(6, 4)$-bit S-boxes.
  - Ignore leading two bits $\Longrightarrow 2^{128}$ possible representations.
  - Choose $L = C_0 \cup C_1 \cup C_3 \cup C_7$, and $q_1(x), q_2(x) \xleftarrow{\$} \mathcal{P}(x^L)$.
  - Find the decomposition: $P(x) = p_1(x) \cdot q_1(x) + p_2(x) \cdot q_2(x) + p_3(x)$.
  - For each $x_j \in \mathbb{F}_{2^6}$, we get 4 equations over $\mathbb{F}_2$.

Evaluation Method: Optimization

- Example: DES $(6, 4)$-bit S-boxes.
  - Ignore leading two bits $\implies 2^{128}$ possible representations.
  - Choose $L = C_0 \cup C_1 \cup C_3 \cup C_7$, and $q_1(x), q_2(x) \xleftarrow{\$} \mathcal{P}(x^L)$.
  - Find the decomposition: $P(x) = p_1(x) \cdot q_1(x) + p_2(x) \cdot q_2(x) + p_3(x)$.
  - For each $x_j \in \mathbb{F}_{2^6}$, we get 4 equations over $\mathbb{F}_2$.
  - Resulting matrix needs to have rank 256 only (not $384 = 6 \times 64$).

Evaluation Method: Optimization

- Example: DES $(6, 4)$-bit S-boxes.
  - Ignore leading two bits $\implies 2^{128}$ possible representations.
  - Choose $L = C_0 \cup C_1 \cup C_3 \cup C_7$, and $q_1(x), q_2(x) \xleftarrow{\$} \mathcal{P}(x^L)$.
  - Find the decomposition: $P(x) = p_1(x) \cdot q_1(x) + p_2(x) \cdot q_2(x) + p_3(x)$.
  - For each $x_j \in \mathbb{F}_{2^6}$, we get 4 equations over $\mathbb{F}_2$.
  - Resulting matrix needs to have rank 256 only (not $384 = 6 \times 64$).
  - Need only 4 NLMs (instead of 5 NLMs).

Implementation for DES

	No. of shares					
Method	3	5	7	9	11	13
Roy-Vivek [RV13]	0.193	0.347	0.533	0.765	1.040	1.349
Table Recomputation [Coron14]	0.096	0.221	0.413	0.597	0.893	1.409
This work	**0.250**	**0.417**	**0.603**	**0.819**	**1.051**	**1.312**

Table: Implementation in C on Intel Core i7. Execution time in ms.

Our Results: Generic Lower Bounds

Theorem

There exists a polynomial $P(x) \in \mathbb{F}_{2^n}[x]$ *such that*

$$\mathrm{NLM}(P(x)) \geq \sqrt{\frac{2^n}{n}} - 2.$$

Our Results: Generic Lower Bounds

Theorem

There exists a polynomial $P(x) \in \mathbb{F}_{2^n}[x]$ *such that*

$$\mathrm{NLM}(P(x)) \geq \sqrt{\frac{2^n}{n}} - 2.$$

- Significant improvement over $\lceil \log_2 (n-1) \rceil$ bound [RV13].

Our Results: Generic Lower Bounds

Theorem

There exists a polynomial $P(x) \in \mathbb{F}_{2^n}[x]$ *such that*

$$\mathrm{NLM}(P(x)) \geq \sqrt{\frac{2^n}{n}} - 2.$$

- Significant improvement over $\lceil \log_2 (n-1) \rceil$ bound [RV13].
- Proof based on a counting argument, similar to [PS73].
  - No. of possible polynomials using r NLMs $\geq (2^n)^{2^n}$.

Generic Lower Bounds: Comparison

n	4	5	6	7	8	9	10	11	12
[RV13]	2	2	3	3	4	4	4	4	4
This work	**0**	**1**	**2**	**3**	**4**	**6**	**9**	**12**	**17**

Table: Lower bounds for non-linear complexity.

Future Work

1. Rigorously prove the complexity of the new evaluation method.

Future Work

1. Rigorously prove the complexity of the new evaluation method.
2. Solve multivariate quadratic system to obtain

$$P(x) = \sum_{i=1}^{(t-1)/2} p_i(x) \cdot q_i(x) + p_t(x).$$

Future Work

1. Rigorously prove the complexity of the new evaluation method.
2. Solve multivariate quadratic system to obtain
$$P(x) = \sum_{i=1}^{(t-1)/2} p_i(x) \cdot q_i(x) + p_t(x).$$
3. Improve concrete lower/upper complexity bounds.
    1. Evaluate DES with only 3 NLMs.

Future Work

1. Rigorously prove the complexity of the new evaluation method.
2. Solve multivariate quadratic system to obtain
$$P(x) = \sum_{i=1}^{(t-1)/2} p_i(x) \cdot q_i(x) + p_t(x).$$
3. Improve concrete lower/upper complexity bounds.
    1. Evaluate DES with only 3 NLMs.
4. Investigate further the cost model of [GPS, AFRICACRYPT 2014].

Thank You!

&

Questions?