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

Srinivas Vivek

University of Luxembourg

August 11, 2014

Outline

- Motivation: application to higher-order masking of block ciphers.
- Evaluation of powers.
- Evaluation of generic polynomials.
- Future work.

Motivation: Masking

- Masking: a practical and 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 addition over $\mathbb{F}_{2^n}$.
- Masking Order, d = number of shares - 1.
- E.g.: first-order masking $\Rightarrow x = x_0 \perp x_1$.
- *Soundness*: attacks become exponentially hard to mount w.r.t. d.

Higher-Order Masking

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

- This scheme is based on the probing circuit model by (ISW, CRYPTO 2003) and later extended by Prouff and Rivain (CHES 2010).
- Provides t^{th}order security when $d \geq 2t + 1$.
- Advantages:
  - More efficient than [PR11], comparable to [Coron14].
  - Smaller memory and randomness requirement than [Coron14].
- Recent works: [CPRR, FSE 2013], [RV, CHES 2013], [CRV, CHES 2014].

CGPQR Scheme (Cont'd)

- In evaluating block ciphers, the only **non-($\mathbb{F}_2$-) linear** operations are the S-box computations.
- The main challenge for masking lies in the masking of S-boxes.
- Reason: $\mathbb{F}_2$-linear/-affine functions are easy to mask.
  - $f_{lin}(x) = f_{lin}(x_0 + \cdots + x_d) = f_{lin}(x_0) + \cdots + f_{lin}(x_d)$
  - $f_{aff}(x) = f_{aff}(x_0 + \cdots + x_d) = f_{aff}(x_0) + \cdots + f_{aff}(x_d) + c^*$
- 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 ($m \leq n$) is a function $SB : \{0,1\}^n \to \{0,1\}^m$.
- n-bit and m-bit strings can be naturally identified with elements of $\mathbb{F}_{2^n}$.
- Hence SB 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 now reduces to securely evaluating the corresponding polynomial with shares.

CGPQR Scheme (Cont'd)

- Task is to evaluate $P(x)$ on (shared) input $(x_0, \cdots, x_d)$.
- Note that any polynomial $P(x) \in \mathbb{F}_{2^n}[x]$ can be evaluated with a sequence of:
  - **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.
  - NLMs with shares are expensive: $O(d^2)$ time and randomness.

NLM with Shares

- To compute $c = a \cdot b$
  - Input: $(a_0, \cdots, a_d)$ and $(b_0, \cdots, b_d)$.
  - Output: $(c_0, \cdots, c_d)$
- Method [ISW03, RP10]:
  - For each $0 \leq i < j \leq d$, pick random $r_{i,j} \xleftarrow{\$} \mathbb{F}_{2^n}$.
  - For each $0 \leq i < j \leq d$, compute $r_{j,i} = (r_{i,j} + a_i b_j) + a_j b_i$.
  - For each $0 \leq i \leq d$, compute $c_i = a_i b_i + \sum_{j \neq i} r_{i,j}$.

$\mathbb{F}_{2^n}$-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**: $q(x) \times r(x)$
- Remark: more accurate cost models are possible, see [GPS, AFRICACRYPT 2014].

Evaluating Powers

- Consider evaluating a power x^{α}.
- A method to evaluate x^{α} (using only multiplications) is a sequence:
  - $x = x^{i_0}, x^{i_1}, x^{i_2}, \ldots, x^{i_l} = x^{\alpha}$, where
  - $\forall k,\ x^{i_k} = x^{i_a} \cdot x^{i_a}$, for some $0 \leq a, b < k$.
- The above in turn yields a sequence/chain of exponents:
  - $i_0 = 1,\ i_1,\ i_2,\ \ldots,\ i_l = \alpha$, where
  - $\forall k,\ i_k = i_a + i_b \pmod{2^n - 1}$, for some $0 \leq a, b < k$.
- Need to count only non-doubling steps.

Evaluating Powers: Cyclotomic Class

- Note that at step k, we can compute a set of elements C_{i_k}:
  - $C_{i_k} = \left\{ i_k \cdot 2^j \pmod{2^n - 1} \ : \ j = 0, 1, \ldots, n-1 \right\}$,
  - called the "**Cyclotomic Class of i_k (w.r.t. n)**" [CGQPR12].
- The above sequence for α can be succinctly written using CCs:
  - $C_{i_0} = C_1, \ C_{i_1}, \ C_{i_2}, \ \ldots, \ C_{i_l} = C_\alpha$, where
  - $\exists \ \beta_k \in C_{i_k}, \ \beta_a \in C_{i_a}, \ \beta_b \in C_{i_b}$, s.t. $\beta_k \equiv \beta_a + \beta_b \pmod{2^n - 1}$.
- Terminology: **CC addition chain for α (w.r.t. n)** [RV13].

Example

- Consider $n = 4$, i.e., $\mathbb{F}_{2^4}$.
- All the cyclotomic classes:
  - $C_0 = \{0\}$, $C_1 = \{1, 2, 4, 8\}$, $C_3 = \{3, 6, 12, 9\}$, $C_5 = \{5, 10\}$, $C_7 = \{7, 14, 13, 11\}$.
- In general, $|C_\alpha| \,|\, n$, and $\uplus C_\alpha = \mathbb{F}_{2^n}$.
- E.g.: CC addition chain for 14 (w.r.t. 4)
  - $\langle C_1,\ C_3,\ C_7 = C_{14} \rangle$
  - Expanded sequence: $\langle 1,\ 2,\ 4,\ 3,\ 7,\ 14 \rangle$.

CC Addition Chain

- Variant of the well known addition chain [Knuth, TAoCp V.2], q-addition chain for $\mathbb{F}_{q^n}$ [von zur Gathen and Nocker, C.C. 1997].
- Main difference with 2-addition chain for $\mathbb{F}_{2^n}$: use of the relation $x^{2^n} = x$.
- Use of this relation was already observed in [von zur Gathen 1991].
- But the notion was not formalized until [CGQPR12].
- CC-addition chain is a better cost model than 2-addition chain for $\mathbb{F}_{2^n}$ when an **Optimal Normal Basis** representation for the field is used.
- **ONB**: $(\beta_0, \beta_1, \ldots \beta_{n-1})$, where $\beta_i = \beta_0^{2^i}$.

CC Addition Chain: Properties

- Notation: $m_n(\alpha)$ - shortest CC- addition chain for α (w.r.t. n).
- Binary method provides an upper bound: $m_n(\alpha) \leq \mathrm{HW}(\alpha) - 1$.
- In fact, any addition chain for α gives an upper bound for any n.
- Best U. bound [Brauer 1939, RV13]: $m_n(\alpha) \leq \frac{\log_2 \alpha}{\log_2 \log_2 \alpha}(1 + o(1))$.
- Lower bound [RV13]: $m_n(\alpha) \geq \lceil \log_2(\mathrm{HW}(\alpha)) \rceil$.
  - Idea: CC is HW invariant.

CC Addition Chain: Properties (cont'd)

- Is $m_n(2^n - 2) \geq m_n(\alpha)$, $\forall\ 1 \leq \alpha \leq 2^n - 2$?
  - No. E.g., $m_9(510) = 3 < m_9(508) = 4$.
  - Open: are there infinitely many such n?
  - if $n = 2^t + 1$, then $m_n(2^n - 2) = t$ [RV13].
- Is $m_n(\alpha) = m_{n'}(\alpha)$ for $n \neq n'$?
  - No. E.g., $m_5(23) = 2$ but $m_6(23) = 3$.
  - E.g., $m_7(83) = 3$ but $m_9(83) = 2$.

CC Addition Chain: Properties (cont'd)

- Monotonicity [RV13]: if $n \mid n'$, then $m_n(\alpha) \leq m_{n'}(\alpha)$.
  - Idea: transform a CC addition chain w.r.t. n to one w.r.t. n', and vice-versa.
  - proof uses the fact that $(2^n - 1) \mid \left(2^{n'} - 1\right)$ if $n \mid n'$,
- $m_8(254) = 4$.
  - $\langle C_1,\ C_5,\ C_{25},\ C_{125},\ C_{127} = C_{254} \rangle$.
  - $x^{254} \in \mathbb{F}_{2^8}[x]$ represents the non-linear part of AES S-box.

Evaluation of Generic Polynomials

- To evaluate powers, we restricted the operations to multiplications only.
  - will this result in an optimal method?
- But to evaluate a generic polynomial we need the other operations as well:
  - additions and scalar multiplications.
  - But, ignored in the total cost

Naïve Methods

- Goal: to evaluate $P(x) \in \mathbb{F}_{2^n}[x]$, given $x \in \mathbb{F}_{2^n}$.

1. Compute x^2, x^3, ..., $x^{2^n-1=N}$, then $\beta_i \cdot x^i \ \forall i$, finally $\sum_{i=0}^{N} \beta_i x^i$.
    1. cost: $N-1$ **NLMs** (N SMs, N ADDs)
2. **Horner's rule**: $((\beta_N \cdot x + \beta_{N-1}) * x + \beta_{N-2}) + \ldots\ldots\ldots + \beta_0$
    1. cost: $N-1$ **NLMs** (1 SM, N ADDs)

- Can we exploit the cost model better?

Cyclotomic Class Method

- Proposed in [CGQPR12].
- Very similar to method of computing all the individual powers except for the order.
  - once x^i is computed, then every x^α ($\alpha \in C_i$) is computed for free by only squarings.
- Worst case complexity: $1+$ total number of CC classes w.r.t. n.
- Lower bound on the number of CC classes: $2^n/n$.
- If n is a prime, then the exact number of CC classes is $(2^n-1)/n$.

Cyclotomic Class Method (cont'd)

n (S-box size)	4	5	6	7	8	9	10
# NLMs	3	5	11	17	33	52	105

Table: CC method: Worst-case complexity

Parity-Split Method

- Also proposed in [CGQPR12].
- Write $P(x) = P_{1,1}(x^2) + x \cdot P_{1,2}(x^2)$.
  - $\deg(P_{1,1}), \deg(P_{1,2}) \leq N/2$, $(N = 2^n - 1)$.
- Split recursively:
  - E.g.: $P_{1,1}(y) = P_{2,1}(y^2) + y \cdot P_{2,2}(y^2)$.
  - $\deg(P_{2,1}), \deg(P_{2,2}) \leq N/4$.
- Optimal depth: $\lfloor n/2 \rfloor$.

Parity-Split Method (cont'd)

- Worst-case complexity (in terms of NLMs):
  - if n is even, $\approx 1.5 \cdot \sqrt{2^n}$.
  - if n is odd, $\approx \sqrt{2} \cdot \sqrt{2^n}$.
- Remark: Both the CC and the PS method do not exploit (to a large extent) specific properties of the coefficients of the polynomials.

Parity-Split Method (cont'd)

n (S-box size)	4	5	6	7	8	9	10
# NLMs	4	6	10	14	22	30	46

Table: PS method: Worst-case complexity

$\mathbb{F}_{2^n}$-Polynomial Chain

Definition

[RV13] An $\mathbb{F}_{2^n}$-*polynomial chain* C for a polynomial $P(x) \in \mathbb{F}_{2^n}[x]$ is

defined as

$$\lambda_{-1} = 1,\ \lambda_0 = x,\ \ldots\ ,\ \lambda_r = P(x),$$

where

$$\lambda_i = \begin{cases} \lambda_j + \lambda_k & -1 \leq j, k < i, \\ \lambda_j \times \lambda_k & -1 \leq j, k < i, \\ \gamma_i \cdot \lambda_j & -1 \leq j < i,\ \gamma_i \text{ is a scalar.} \\ \lambda_j^2 & -1 \leq j < i. \end{cases}$$

$\mathbb{F}_{2^n}$-Polynomial Chain (cont'd)

- γ_i - arbitrary function of the coefficients.
- This model is relevant when the same polynomial is evaluated multiple times.
- Definition: **Non-linear Complexity** of $P(x)$, $\mathcal{M}(P(x))$: least number of NLMs.
- Also called **"Masking Complexity"** in the context of masking S-boxes.

$\mathbb{F}_{2^n}$-Polynomial Chain: Properties

Fact

[RV13] Consider $P(x) := \sum_{i=0}^{2^n-1} \beta_i x^i$. *Then*

$$\mathcal{M}(P(x)) \geq \max_{\substack{0<i<2^n-1 \\ \beta_i \neq 0}} m_n(i).$$

- This fact implies that to evaluate powers, it suffices to consider only squarings and NLMs.
- Not true for the cost model of [GPS14].

$\mathbb{F}_{2^n}$-Polynomial Chain: Properties (cont'd)

- **Well-definedness** of non-linear complexity: **invariant** under affine transformations.
  - this holds also w.r.t. field representations.
- Hence the masking complexity of the entire AES S-box is the same as that of $x^{254} \in \mathbb{F}_{2^8}[x]$.
- Lower bound for S-boxes: PRESENT:2, DES: 3^*, AES: 4 NLMs.

Generic Lower Bounds

Theorem

[CRV14] There exists a polynomial $P(x) \in \mathbb{F}_{2^n}[x]$ *such that*

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

- Significant improvement over the $\lceil \log_2(n-1) \rceil$ bound from [RV13].
- Proof based on a counting argument, similar to [Paterson & Stockmeyer, SIAM J. Comp., 1973].
- **Proof**:
  - Any chain that uses r NLMs ($r \geq 0$) can be described as a polynomial sequence: $z_{-1}, z_0, \ldots, z_r$.

Generic Lower Bounds (cont'd)

- **Proof** (cont'd):
  - $z_{-1} = 1$,
  - $z_0 = x$,

$$z_k = \left(\beta_{k,-1} + \sum_{i=0}^{k-1}\sum_{j=0}^{n-1}\beta_{k,i,j}\, z_i^{2^j}\right) \cdot \left(\beta'_{k,-1} + \sum_{i=0}^{k-1}\sum_{j=0}^{n-1}\beta'_{k,i,j}\, z_i^{2^j}\right) \pmod{x^{2^n} + x}.$$

Generic Lower Bounds (cont'd)

- **Proof** (cont'd):
  - Count all the parameters .
  - This count should be at least as large as the total count of the polynomials of degree at most $2^n - 1$ in $\mathbb{F}_{2^n}[x]$, i.e.,$(2^n)^{2^n}$.
  - We get $r \geq \sqrt{\frac{2^n}{n}} - 2$.

Generic Lower Bounds (cont'd)

n	4	5	6	7	8	9	10	11	12
[RV13]	2	2	3	3	4	4	4	4	4
[CRV14]	0	1	2	3	4	6	9	12	17

Table: Lower bounds for non-linear complexity.

Divide and Conquer Approach

- Originally proposed in [PS73].
  - Most suited for polynomials of specific degrees.
  - Degree $N \approx \sqrt{N}\left(2^i - 1\right)$.
  - Best case: $\approx \sqrt{N}$ NLMs.
- Suitably adapted this technique for various block ciphers in [RV13].

Divide and Conquer Approach (cont'd)

- Idea:
  - To pre-compute certain monomials.
  - Then to decompose the given polynomial in terms of polynomials having monomials from the pre-computed set.
    - ⋆ Decomposition follows a pre-defined recursive procedure.
- **DES** S-boxes: 7 NLMs (instead of 10).
- **CAMELLIA/CLEFIA** S-boxes : ≤ 16 NLMs (instead of 22).

Roy-Vivek Method for DES

- Compute x, x^2, ..., x^9.
- Compute x^{18}, x^{36}.

$$P_{DES}(x) = (x^{36} + c(x)) * \Big(((x^{18} + c_1(x)) * q_1(x)) + (x^9 + s_1(x))\Big) + \Big((x^{18} + c_2(x)) * q_2(x) + (x^9 + s_2(x))\Big).$$

CRV Method

- Proposed in [CRV14].
- (Heuristic) worst-case complexity: $\approx 2 \cdot \sqrt{\frac{2^n}{n}}$.
  - Asymptotically optimal.
- **Method**:
  - Consider a collection $\mathcal{S}$ of ℓ cyclotomic classes w.r.t. n:
  - Let $L = \bigcup_{C_i \in \mathcal{S}} C_i$.
  - Pre-compute a set of monomials, x^L, with exponents in L.

CRV Method (cont'd)

- **Method** (cont'd):
  - Generate $t-1$ random polynomials $q_i(x) \xleftarrow{\$} \mathcal{P}(x^L)$.
  - Find t polynomials $p_i(x) \in \mathcal{P}(x^L)$ such that

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

  - Solve a linear system for the unknown coefficients - Lagrange interpolation technique.
  - Heuristic: full rank if $t \cdot |L| \geq 2^n$.

CRV Method: Analysis

- **Analysis**:
  - The set x^L can be computed using $\ell - 2$ NLMs .
  - Combining step: $t - 1$ NLMs.
  - Total: $N_{mult} = \ell + t - 3$.
  - 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.

CRV Method: Analysis (cont'd)

n (S-box size)	4	5	6	7	8	9	10
# NLMs	2	4	5	7	10	14	19

Table: CRV Method: Worst-case complexity

- Can we take advantage of the fact that $n > m$ for an (n, m)-bit S-box.
- Note that the lower bound of 3 for DES is proven for a 6-to-6 bit mapping with two leading zeroes.

CRV Method: DES

- DES S-boxes map 6 bits to 4 bits.
- If we ignore the leading two bits, then 2^{128} possible representations.
- Choose $L = C_0 \cup C_1 \cup C_3 \cup C_7$, $t = 3$, and $q_1(x), q_2(x) \xleftarrow{\$} \mathcal{P}(x^L)$.
- To 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).

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
Coron-Roy-Vivek [CRV14]	**2**	**4**	**5**	**7**	**10**	**14**	**19**

Table: Counting non-linear multiplications

Application to S-boxes

	S-box				
Method	DES	PRESENT	SERPENT	CAMELLIA	CLEFIA
Parity-Split [CGPQR12]	10	3	3	22	22
Roy-Vivek [RV13]	7	3	3	15	15,16
Coron-Roy-Vivek [CRV14]	**4**	**2**	**2**	**10**	**10**

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

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
Coron-Roy-Vivek [CRV14]	0.250	0.417	0.603	0.819	1.051	1.312

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

Future Work

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

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

   - No. of variables $\approx$ Equations.

3. Improve concrete lower/upper bounds.
   1. Evaluate DES with only 3 NLMs.
4. Investigate further the cost model of [GPS14].

Type-II vs. Type-III NLM

- [GPS14] differentiates between two types of NLMs: Type-II and Type-III NLMs.
- Type-II NLM: $y \cdot g(y)$, $g()$ is $\mathbb{F}_2$-affine.
  - E.g.: $y \cdot y^2$, $y \cdot y^4$, $y \cdot y^8$
  - E.g.: $y \cdot \left(\sum_{i=0}^{n-1} a_i \, y^{2^i} + a_{-1} \right)$.
- Cost of Type-III NLM $\approx 2 \cdot$ Type-II NLM:
  - when Type-II NLM is implemented as in [CPRR13].
  - medium sized S-boxes.

Type-II vs. Type-III NLM (cont'd)

- AES S-box: 3 Type-II and 1 Type-III NLMs [GPS14].
  - $\langle x, x^5, x^{25}, x^{125}, x^{127}, x^{254} \rangle$.
- AES can be evaluated using 9 Type-II NLMs [Remark 3, GPS14] :
  - Also uses additions.
  - Can be extended to any polynomial: proof by induction.
- DES S-boxes: 1 Type-II and 3 Type- III NLMs [CRV14].

Future Work (cont'd)

- DES S-boxes: 4 Type-II and 1 Type- III NLMs:
  - $L = C_0 \cup C_1 \cup C_3 \cup C_5 \cup C_9 \cup C_7$
  - $p_1(x) \cdot q_1(x) + p_2(x)$
- Open: Evaluate DES S-boxes with only 5 Type-II NLMs?

Thank You!

References

- Donald E. Knuth. **The Art of Computer Programming, Volume II: Seminumerical Algorithms**, 3rd Edition.
- Jean-Sébastien Coron, Arnab Roy, Srinivas Vivek. **Fast Evaluation of Polynomials over Binary Finite Fields and Application to Side-channel Countermeasures**. CHES 2014, To appear.
- Vincent Grosso, Emmanuel Prouff, François-Xavier Standaert. **Efficient Masked S-boxes Processing, a Step Forward**. AFRICACRYPT 2014.

References (cont'd)

- Arnab Roy, Srinivas Vivek. **Analysis and improvement of the generic higher-order masking scheme of FSE 2012**. CHES 2013.
- Claude Carlet, Louis Goubin, Emmanuel Prouff, Michaël Quisquater, Matthieu Rivain. **Higher-order masking schemes for S-Boxes**. FSE 2012.
- Mike Paterson, Larry J. Stockmeyer. **On the number of nonscalar multiplications necessary to evaluate polynomials**. SIAM J. Computing, 1973.