Reducing the Number of Non-linear Multiplications in Masking Schemes

Srinivas Vivek

University of Bristol, UK

Jürgen Pulkus
Giesecke & Devrient, Munich, Germany

August 19th, 2016
CHES 2016

Outline

Motivation

Our Contribution
- Improved CRV Method
- Further Improvement Using Bigger Fields
- Non-Linear Complexity: Generalised Lower Bounds

Conclusion

Motivation

Masking

Masking is a popular countermeasure against *DPA-like* side-channel attacks.

Well suited to protect block cipher implementations.

In (additive) masking, each sensitive variable is secret shared.

- Let $x \in \mathbb{F}_{2^n}$, then $x = x_0 + x_1 + \ldots + x_v$.

Security offered has been relatively well analysed

- w.r.t. *probing* leakage model [*ISW03*] and *noisy* leakage model [*CJJR99, RP13*].
- Loosely speaking, SCA complexity is exponential w.r.t. v.

[*ISW03*] Y. Ishai, A. Sahai, D. Wagner. *Private circuits: Securing hardware against probing attacks.* CRYPTO 2003.
[*CJRR99*] S. Chari, C.S. Jutla, J.R. Rao, P. Rohatgi. *Towards sound approaches to counteract power-analysis attacks.* CRYPTO 1999.
[*RP10*] M. Rivain, E. Prouff. *Provably secure higher-order masking of AES.* CHES 2010.

Masking

Masking is a popular countermeasure against *DPA-like* side-channel attacks.

Well suited to protect block cipher implementations.

In (additive) masking, each sensitive variable is secret shared.

- Let $x \in \mathbb{F}_{2^n}$, then $x = x_0 + x_1 + \ldots + x_v$.

Security offered has been relatively well analysed

- w.r.t. *probing* leakage model [*ISW03*] and *noisy* leakage model [*CJJR99, RP13*].
- Loosely speaking, SCA complexity is exponential w.r.t. v.

[*ISW03*] Y. Ishai, A. Sahai, D. Wagner. *Private circuits: Securing hardware against probing attacks*. CRYPTO 2003.
[*CJRR99*] S. Chari, C.S. Jutla, J.R. Rao, P. Rohatgi. *Towards sound approaches to counteract power-analysis attacks*. CRYPTO 1999.
[*RP10*] M. Rivain, E. Prouff. *Provably secure higher-order masking of AES*. CHES 2010.

Higher-Order Masking

Linear/Affine functions are straightforward to compute in presence of shares.

- Time and randomness complexity are both linear in the number of shares.

Main challenge is to securely compute *non-linear* functions.

- Various H-O masking schemes differ mainly in how these functions are evaluated.
- For block ciphers, this reduces to securing their S-boxes.

Higher-Order Masking

Linear/Affine functions are straightforward to compute in presence of shares.

- Time and randomness complexity are both linear in the number of shares.

Main challenge is to securely compute *non-linear* functions.

- Various H-O masking schemes differ mainly in how these functions are evaluated.
- For block ciphers, this reduces to securing their S-boxes.

CGPQR H-O Masking Scheme

Proposed in [*CGPQR12*].

- Based on [*ISW03, RP10*].
- Guarantees t-th order security in the probing leakage model when $v \geq 2t$.
- Suited well for software implementations.

A d-to-r-bit S-box S ($d \geq r$) is represented by a polynomial $\mathcal{P}(x) \in \mathbb{F}_{2^d}[x]$.

- Securely evaluating S reduces to evaluating $\mathcal{P}(x)$ when x is given as a secret-shared input.

CGPQR H-O Masking Scheme

Proposed in [*CGPQR12*].

- Based on [*ISW03, RP10*].
- Guarantees t-th order security in the probing leakage model when $v \geq 2t$.
- Suited well for software implementations.

A d-to-r-bit S-box S ($d \geq r$) is represented by a polynomial $\mathcal{P}(x) \in \mathbb{F}_{2^d}[x]$.

- Securely evaluating S reduces to evaluating $\mathcal{P}(x)$ when x is given as a secret-shared input.

CGPQR H-O Masking Scheme

Note that (polynomial) addition, multiplication by a scalar, (polynomial) squaring operations are $\mathbb{F}_2$-linear.

- Cheap: $\mathcal{O}(v)$ time and randomness.

Cost mainly determined by the *Non-Linear Multiplications* (NLMs).

- That are secured using a technique from [*ISW03, RP10*].
- Expensive: $\mathcal{O}(v^2)$ time and randomness.

Already there are several works improving the CGPQR scheme: [*R*V*13, CR*V*14, CGPZ16 (next talk)*] and [*GPS14, CPRR15*].

[*RV13*] A.Roy, S. Vivek. *Analysis and improvement of the generic higher-order masking scheme of FSE 2012.* CHES 2013.
[*CRV14*] J.-S. Coron, A. Roy, and S. Vivek. *Fast evaluation of polynomials over binary finite fields and application to side-channel countermeasures.* CHES 2014 & JCEN 2015.
[*GPS14*] V. Grosso, E. Prouff, F.-X. Standaert. *Efficient masked S-boxes processing - A step forward.* AFRICACRYPT 2014.
[*CPRR15*] C. Carlet, E. Prouff, M. Rivain, T. Roche. *Algebraic decomposition for probing security.* CRYPTO 2015.

CGPQR H-O Masking Scheme

Note that (polynomial) addition, multiplication by a scalar, (polynomial) squaring operations are $\mathbb{F}_2$-linear.

- Cheap: $\mathcal{O}(v)$ time and randomness.

Cost mainly determined by the *Non-Linear Multiplications* (NLMs).

- That are secured using a technique from [*ISW03, RP10*].
- Expensive: $\mathcal{O}(v^2)$ time and randomness.

Already there are several works improving the CGPQR scheme: [*RV13, CRV14, CGPZ16 (next talk)*] and [*GPS14, CPRR15*].

[*RV13*] A.Roy, S. Vivek. *Analysis and improvement of the generic higher-order masking scheme of FSE 2012.* CHES 2013.
[*CRV14*] J.-S. Coron, A. Roy, and S. Vivek. *Fast evaluation of polynomials over binary finite fields and application to side-channel countermeasures.* CHES 2014 & JCEN 2015.
[*GPS14*] V. Grosso, E. Prouff, F.-X. Standaert. *Efficient masked S-boxes processing - A step forward.* AFRICACRYPT 2014.
[*CPRR15*] C. Carlet, E. Prouff, M. Rivain, T. Roche. *Algebraic decomposition for probing security.* CRYPTO 2015.

CGPQR H-O Masking Scheme

Note that (polynomial) addition, multiplication by a scalar, (polynomial) squaring operations are $\mathbb{F}_2$-linear.

- Cheap: $\mathcal{O}(v)$ time and randomness.

Cost mainly determined by the *Non-Linear Multiplications* (NLMs).

- That are secured using a technique from [*ISW03, RP10*].
- Expensive: $\mathcal{O}(v^2)$ time and randomness.

Already there are several works improving the CGPQR scheme: [*R*V*13, CR*V*14, CGPZ16 (next talk)*] and [*GPS14, CPRR15*].

[*RV13*] A.Roy, S. Vivek. *Analysis and improvement of the generic higher-order masking scheme of FSE 2012*. CHES 2013.
[*CRV14*] J.-S. Coron, A. Roy, and S. Vivek. *Fast evaluation of polynomials over binary finite fields and application to side-channel countermeasures*. CHES 2014 & JCEN 2015.
[*GPS14*] V. Grosso, E. Prouff, F.-X. Standaert. *Efficient masked S-boxes processing - A step forward*. AFRICACRYPT 2014.
[*CPRR15*] C. Carlet, E. Prouff, M. Rivain, T. Roche. *Algebraic decomposition for probing security*. CRYPTO 2015.

Evaluating Polynomials over $\mathbb{F}_{2^d}$

Cost analysis of the CGPQR scheme reduces to the following problem.

- To evaluate any polynomial $P(x) \in \mathbb{F}_{2^d}[x]$, given x.
- **Count**: non-linear (polynomial) multiplications.
- **Ignore**: (polynomial) addition, scalar multiplication, (polynomial) squaring operations
  - Equivalent to ignoring the cost of $\mathbb{F}_2$-affine functions over $\mathbb{F}_{2^d}$.

Polynomial evaluation methods.

- **Knuth-Eve / Parity-Split Method** [*K62, E64, CGPQR12*].
  - (Proven) worst-case complexity: $1.5 \cdot \sqrt{2^d}$ NLMs.
- **CRV Method** [*CR*V*14*].
  - (Heuristic) worst-case complexity: $\approx 2 \cdot \sqrt{\frac{2^d}{d}}$ NLMs.
  - Lower bound: $\approx \sqrt{\frac{2^d}{d}}$ NLMs.

[*K62*] D.E. Knuth. *Evaluation of polynomials by computer.* CACM 1962.
[*E64*] J. Eve. *The Evaluation of Polynomials.* Numerische Mathematik 1964.

Evaluating Polynomials over $\mathbb{F}_{2^d}$

Cost analysis of the CGPQR scheme reduces to the following problem.

- To evaluate any polynomial $P(x) \in \mathbb{F}_{2^d}[x]$, given x.
- **Count**: non-linear (polynomial) multiplications.
- **Ignore**: (polynomial) addition, scalar multiplication, (polynomial) squaring operations
  - Equivalent to ignoring the cost of $\mathbb{F}_2$-affine functions over $\mathbb{F}_{2^d}$.

Polynomial evaluation methods.

- **Knuth-Eve / Parity-Split Method** [*K62, E64, CGPQR12*].
  - (Proven) worst-case complexity: $1.5 \cdot \sqrt{2^d}$ NLMs.
- **CRV Method** [*CR*V*14*].
  - (Heuristic) worst-case complexity: $\approx 2 \cdot \sqrt{\frac{2^d}{d}}$ NLMs.
  - Lower bound: $\approx \sqrt{\frac{2^d}{d}}$ NLMs.

[*K62*] D.E. Knuth. *Evaluation of polynomials by computer*. CACM 1962.
[*E64*] J. Eve. *The Evaluation of Polynomials*. Numerische Mathematik 1964.

Non-linear Complexity of S-boxes: State-of-the-Art

(d, r)	(4,4)	(5,5)	(6,4)	(6,6)	(7,7)	(8,8)
Cyclotomic-Class method [*CGPQR12*]	3	5	11	11	17	33
Parity-Split method [*CGPQR12*]	4	6	10	10	14	22
CRV method [*CR*V*14*]	2	4	4	5	7	10
Lower bounds (over $\mathbb{F}_{2^d}$) [*R*V*13*, **This Work**]	2	2	3	3	3	3

Table: Worst-case complexity in terms of NLMs of *previous* methods.

Our Contribution

Improved CRV Method

CRV Method: Recall

Input: d-to-r-bit S-box S

Output: A sequence of polynomials that eventually evaluates S.

Step 0: Naturally encode $\{0,1\}^d$ and $\{0,1\}^r$ in $\mathbb{F}_{2^d}$.

Step 1: Pre-compute a set of monomials $x^L = \{x^i \mid i \in L\}$

- *Closed* w.r.t. squaring.
- $x^L \cdot x^L$ must include all monomials in $\mathbb{F}_{2^d}[x]/(x^{2^d} - x)$.

CRV Method: Recall

Input: d-to-r-bit S-box S

Output: A sequence of polynomials that eventually evaluates S.

Step 0: Naturally encode $\{0,1\}^d$ and $\{0,1\}^r$ in $\mathbb{F}_{2^d}$.

Step 1: Pre-compute a set of monomials $x^L = \{x^i \mid i \in L\}$

- *Closed* w.r.t. squaring.
- $x^L \cdot x^L$ must include all monomials in $\mathbb{F}_{2^d}[x]/(x^{2^d} - x)$.

CRV Method: Recall

Input: d-to-r-bit S-box S

Output: A sequence of polynomials that eventually evaluates S.

Step 0: Naturally encode $\{0,1\}^d$ and $\{0,1\}^r$ in $\mathbb{F}_{2^d}$.

Step 1: Pre-compute a set of monomials $x^L = \{x^i \mid i \in L\}$

- *Closed* w.r.t. squaring.
- $x^L \cdot x^L$ must include all monomials in $\mathbb{F}_{2^d}[x]/(x^{2^d} - x)$.

CRV Method: Recall

Step 2: Find decomposition of the form

$$P_S(x) = \sum_{i=1}^{t-1} p_i(x) \cdot q_i(x) + p_t(x) \pmod{X^{2^d} - X},$$

where $p_i(x), q_i(x) \in \mathcal{T}(x^L)$. By

- Choosing random polynomials $q_i(x) \xleftarrow{\$} \mathcal{T}(x^L)$.
- Set up an $\mathbb{F}_2$-linear system of equations
  - By evaluating the above relation at each input.
  - Obtaining one equation for each output bit of S.
  - Note that $d - r$ output bits of $P_S(x)$ are discarded.
- Solve for the unknown bits of the coefficients of $p_i(x)$.

Our Method

Very similar to the CRV method.

- Mainly Step 0 and Step 1 are modified.

Step 0: Naturally encode $\{0,1\}^d$ and $\{0,1\}^r$ in $\mathbb{F}_{2^n}$.

- Need $d, r \leq n$.

Step 1: Pre-compute a set of monomials $x^L = \{x^i \mid i \in L\}$

- *Closed* w.r.t. squaring.
- *Heuristic*: $x^L \cdot x^L$ must yield a decomposition in the Step 2 below.
  - This condition leads to a lower bound on $|L|$.

Step 2: Same as in the CRV method but now working over $\mathbb{F}_{2^n}$.

Our Method

Very similar to the CRV method.

- Mainly Step 0 and Step 1 are modified.

Step 0: Naturally encode $\{0,1\}^d$ and $\{0,1\}^r$ in $\mathbb{F}_{2^n}$.

- Need $d, r \leq n$.

Step 1: Pre-compute a set of monomials $x^L = \{x^i \mid i \in L\}$

- *Closed* w.r.t. squaring.
- *Heuristic*: $x^L \cdot x^L$ must yield a decomposition in the Step 2 below.
  - This condition leads to a lower bound on $|L|$.

Step 2: Same as in the CRV method but now working over $\mathbb{F}_{2^n}$.

Our Method

Very similar to the CRV method.

- Mainly Step 0 and Step 1 are modified.

Step 0: Naturally encode $\{0,1\}^d$ and $\{0,1\}^r$ in $\mathbb{F}_{2^n}$.

- Need $d, r \leq n$.

Step 1: Pre-compute a set of monomials $x^L = \{x^i \mid i \in L\}$

- *Closed* w.r.t. squaring.
- *Heuristic*: $x^L \cdot x^L$ must yield a decomposition in the Step 2 below.
  - This condition leads to a lower bound on $|L|$.

Step 2: Same as in the CRV method but now working over $\mathbb{F}_{2^n}$.

Our Method

Very similar to the CRV method.

- Mainly Step 0 and Step 1 are modified.

Step 0: Naturally encode $\{0,1\}^d$ and $\{0,1\}^r$ in $\mathbb{F}_{2^n}$.

- Need $d, r \leq n$.

Step 1: Pre-compute a set of monomials $x^L = \{x^i \mid i \in L\}$

- *Closed* w.r.t. squaring.
- *Heuristic*: $x^L \cdot x^L$ must yield a decomposition in the Step 2 below.
  - This condition leads to a lower bound on $|L|$.

Step 2: Same as in the CRV method but now working over $\mathbb{F}_{2^n}$.

Our Method: Analysis

Total number of NLMs $M_{d,r,n} \approx |L|/n + t - 1$.

- Bigger field means longer cyclotomic classes.

As in the CRV method, to choose parameters L and t, we use

- *Heuristic*: we get full ranked matrix in Step 2 if $|L| \cdot t \cdot n \geq r \cdot 2^d$.

We *heuristically* show that

$$M_{d,r,n} \approx \sqrt{\frac{2^d}{d} + \frac{r \cdot \sqrt{d \cdot 2^d}}{n^2}}.$$

Hence $M_{d,r,\infty} \approx \sqrt{\frac{2^d}{d}}$.

- Note: CRV method needs $\approx 2 \cdot \sqrt{\frac{2^d}{d}}$ NLMs.

Our Method: Analysis

Total number of NLMs $M_{d,r,n} \approx |L|/n + t - 1$.

- Bigger field means longer cyclotomic classes.

As in the CRV method, to choose parameters L and t, we use

- *Heuristic*: we get full ranked matrix in Step 2 if $|L| \cdot t \cdot n \geq r \cdot 2^d$.

We *heuristically* show that

$$M_{d,r,n} \approx \sqrt{\frac{2^d}{d} + \frac{r \cdot \sqrt{d \cdot 2^d}}{n^2}}.$$

Hence $M_{d,r,\infty} \approx \sqrt{\frac{2^d}{d}}$.

- Note: CRV method needs $\approx 2 \cdot \sqrt{\frac{2^d}{d}}$ NLMs.

Our Method: Analysis

Total number of NLMs $M_{d,r,n} \approx |L|/n + t - 1$.

- Bigger field means longer cyclotomic classes.

As in the CRV method, to choose parameters L and t, we use

- *Heuristic*: we get full ranked matrix in Step 2 if $|L| \cdot t \cdot n \geq r \cdot 2^d$.

We *heuristically* show that

$$M_{d,r,n} \approx \sqrt{\frac{2^d}{d}} + \frac{r \cdot \sqrt{d \cdot 2^d}}{n^2}.$$

Hence $M_{d,r,\infty} \approx \sqrt{\frac{2^d}{d}}$.

- Note: CRV method needs $\approx 2 \cdot \sqrt{\frac{2^d}{d}}$ NLMs.

Non-linear Complexity of S-boxes: Comparison

(d, r)	(4,4)	(5,5)	(6,4)	(6,6)	(7,7)	(8,8)
Cyclotomic-Class method [*CGPQR12*]	3	5	11	11	17	33
Parity-Split method [*CGPQR12*]	4	6	10	10	14	22
CRV method [*CR*V*14*]	2	4	4	5	7	10
Our method (over $\mathbb{F}_{2^8}$)	**2**	**3**	**3**	**4**	**6**	**10**
Our method (over $\mathbb{F}_{2^{16}}$)	**2**	**3**	**3**	**3**	**4**	**6**
Lower bounds (over $\mathbb{F}_{2^n}$) [*R*V*13*, **This Work**]	2	2	3	3	3	3

Table: Comparison of worst-case complexity in terms of NLMs.

Masked Implementation of DES

DES uses eight 6-to-4-bit S-boxes.

Pre-compute $x^L = x^{C_0^8} \cup x^{C_1^8} \cup x^{C_3^8} \cup x^{C_7^8} \in \mathbb{F}_{2^8}[x]/(x^{2^8} - x)$.

Obtain the decomposition: $P(x) = p_1(x) \cdot q_1(x) + p_2 \pmod{X^{2^8} - X}$

- $p_1(x), q_1(x), p_2(x) \in \mathcal{T}(x^L)$.

We performed a proof-of-concept software implementation of masked DES in C.

- Used code from `https://github.com/coron/htable/`.
- Ran experiments on a DELL Laptop but manipulated only bytes.
- Tabulated linear functions in ROM for efficiency.

Masked Implementation of DES

DES uses eight 6-to-4-bit S-boxes.

Pre-compute $x^L = x^{C_0^8} \cup x^{C_1^8} \cup x^{C_3^8} \cup x^{C_7^8} \in \mathbb{F}_{2^8}[x]/(x^{2^8} - x)$.

Obtain the decomposition: $P(x) = p_1(x) \cdot q_1(x) + p_2 \pmod{X^{2^8} - X}$

- $p_1(x), q_1(x), p_2(x) \in \mathcal{T}(x^L)$.

We performed a proof-of-concept software implementation of masked DES in C.

- Used code from `https://github.com/coron/htable/`.
- Ran experiments on a DELL Laptop but manipulated only bytes.
- Tabulated linear functions in ROM for efficiency.

Masked Implementation of DES

DES uses eight 6-to-4-bit S-boxes.

Pre-compute $x^L = x^{C_0^8} \cup x^{C_1^8} \cup x^{C_3^8} \cup x^{C_7^8} \in \mathbb{F}_{2^8}[x]/(x^{2^8} - x)$.

Obtain the decomposition: $P(x) = p_1(x) \cdot q_1(x) + p_2 \pmod{X^{2^8} - X}$

- $p_1(x), q_1(x), p_2(x) \in \mathcal{T}(x^L)$.

We performed a proof-of-concept software implementation of masked DES in C.

- Used code from `https://github.com/coron/htable/`.
- Ran experiments on a DELL Laptop but manipulated only bytes.
- Tabulated linear functions in ROM for efficiency.

Masked Implementation of DES

DES uses eight 6-to-4-bit S-boxes.

Pre-compute $x^L = x^{C_0^8} \cup x^{C_1^8} \cup x^{C_3^8} \cup x^{C_7^8} \in \mathbb{F}_{2^8}[x]/(x^{2^8} - x)$.

Obtain the decomposition: $P(x) = p_1(x) \cdot q_1(x) + p_2 \pmod{X^{2^8} - X}$

- $p_1(x), q_1(x), p_2(x) \in \mathcal{T}(x^L)$.

We performed a proof-of-concept software implementation of masked DES in C.

- Used code from `https://github.com/coron/htable/`.
- Ran experiments on a DELL Laptop but manipulated only bytes.
- Tabulated linear functions in ROM for efficiency.

Masked Implementation of DES: Comparison

Method	t	$v+1$	Rand $\times 10^3$	RAM (bytes)	Time (ms)	OF
Unprotected					0.005	1
CGPQR+RV	1	3	2752	72	0.290	58
CGPQR+CRV	1	3	1600	40	0.093	18
CGPQR+This Work	1	3	1216	34	0.068	**13**
CGPQR+RV	2	5	9152	118	0.538	107
CGPQR+CRV	2	5	5312	64	0.175	35
CGPQR+This Work	2	5	4032	54	0.133	**26**
CGPQR+RV	3	7	19200	164	0.824	164
CGPQR+CRV	3	7	11136	88	0.293	58
CGPQR+This Work	3	7	8448	74	0.214	**42**
CGPQR+RV	4	9	32896	210	1.188	237
CGPQR+CRV	4	9	19072	112	0.455	91
CGPQR+This Work	4	9	14464	94	0.323	**64**

Further Improvement Using Bigger Fields

Improved Upper Bounds

Worst-case upper bound on the non-linear complexity of d-to-r-bit S-boxes.

- Even after our improvement to the CRV method, the upper bound is still $\mathcal{O}\left(\sqrt{\frac{2^d}{d}}\right)$ NLMs.
- Using a different technique, we "improve" the upper bound to $\lceil log_2 d \rceil$ NLMs. This bound is optimal.

Main idea is

- We can pack several independent multiplications over a smaller field in a multiplication over a suitable extension field.
- Then individual products can be "extracted" for free using linear projections.

We argument based on algebraic degrees to prove optimality of U.B.

Improved Upper Bounds

Worst-case upper bound on the non-linear complexity of d-to-r-bit S-boxes.

- Even after our improvement to the CRV method, the upper bound is still $\mathcal{O}\left(\sqrt{\frac{2^d}{d}}\right)$ NLMs.
- Using a different technique, we “improve” the upper bound to $\lceil log_2 d \rceil$ NLMs. This bound is optimal.

Main idea is

- We can pack several independent multiplications over a smaller field in a multiplication over a suitable extension field.
- Then individual products can be "extracted" for free using linear projections.

We argument based on algebraic degrees to prove optimality of U.B.

Improved Upper Bounds: AES case

Applying the preceding technique to the case of AES S-box

- We can evaluate ($x^{254} \in \mathbb{F}_{2^8}[x]$) using only 3 NLMs over $\mathbb{F}_{2^{16}}[x]$.
- Previously it needed 4 NLMs over $\mathbb{F}_{2^8}[x]$.

Method

- Identify $\mathbb{F}_{2^8}$ with a subfield of $\mathbb{F}_{2^{16}}$.
- Compute x^3.
- Compute $(x^2 + z \cdot x^3) \cdot (x^3)^4$, where $z \in \mathbb{F}_{2^{16}} \setminus \mathbb{F}_{2^8}$.
- $\mathbb{F}_2$-linearly extract the functions $X \mapsto X^{14}$ and $X \mapsto X^{15}$ over $\mathbb{F}_{2^8}$.
- Finally, compute $x^{254} = x^{14} \cdot (x^{15})^{16}$.
- The above sequence of operations is motivated by [*GHS12*].

[*GHS12*] C. Gentry, S. Halevi, N.P. Smart. *Homomorphic evaluation of the AES circuit.* CRYPTO 2012 & ePrint 2012/99.

Improved Upper Bounds: AES case

Applying the preceding technique to the case of AES S-box

- We can evaluate ($x^{254} \in \mathbb{F}_{2^8}[x]$) using only 3 NLMs over $\mathbb{F}_{2^{16}}[x]$.
- Previously it needed 4 NLMs over $\mathbb{F}_{2^8}[x]$.

Method

- Identify $\mathbb{F}_{2^8}$ with a subfield of $\mathbb{F}_{2^{16}}$.
- Compute x^3.
- Compute $(x^2 + z \cdot x^3) \cdot (x^3)^4$, where $z \in \mathbb{F}_{2^{16}} \setminus \mathbb{F}_{2^8}$.
- $\mathbb{F}_2$-linearly extract the functions $X \mapsto X^{14}$ and $X \mapsto X^{15}$ over $\mathbb{F}_{2^8}$.
- Finally, compute $x^{254} = x^{14} \cdot (x^{15})^{16}$.
- The above sequence of operations is motivated by [*GHS12*].

[*GHS12*] C. Gentry, S. Halevi, N.P. Smart. *Homomorphic evaluation of the AES circuit*. CRYPTO 2012 & ePrint 2012/99.

Non-Linear Complexity: Generalised Lower Bounds

Generalised Lower Bounds

Worst-case lower bound on the non-linear complexity of d-to-r-bit S-boxes.

- Previous best bound [*CR*V*14*]: $\sqrt{\frac{2^d}{d}} - 2$ NLMs.
  - But this bound holds only for $d = r$ and over $\mathbb{F}_{2^d}$.

We generalise the [*CR*V*14*] bound to any chosen field $\mathbb{F}_{2^n}$.

- **New lower bound**: $\frac{\sqrt{r(2^d-1-d)+(d+\frac{r-n}{2})^2}-(d+\frac{r-n}{2})}{n}$ NLMs.
  - As in [*CR*V*14*], we use counting-based arguments.
  - Additionally, we use the fact that projections are linear functions.

Generalised Lower Bounds

Worst-case lower bound on the non-linear complexity of d-to-r-bit S-boxes.

- Previous best bound [CRV14]: $\sqrt{\frac{2^d}{d}} - 2$ NLMs.
  - But this bound holds only for $d = r$ and over $\mathbb{F}_{2^d}$.

We generalise the [CRV14] bound to any chosen field $\mathbb{F}_{2^n}$.

- **New lower bound**: $\frac{\sqrt{r(2^d-1-d)+(d+\frac{r-n}{2})^2}-(d+\frac{r-n}{2})}{n}$ NLMs.
  - As in [CRV14], we use counting-based arguments.
  - Additionally, we use the fact that projections are linear functions.

Conclusion

Conclusion

We *improve* the [*CRV14*] method for evaluating S-box polynomials.

- Main idea is to work over fields **bigger** than $\mathbb{F}_{2^d}$ for a d-to-r-bit S-box.
- **Reduced** the non-linear complexity for many S-boxes.
  - **DES** S-boxes now need only **3** NLMs over $\mathbb{F}_{2^8}$.
  - Improvement in the running time of masked DES by around 25%.

"Improved" **upper** bound on the complexity of d-to-r-bit S-boxes

- **New**: $\lceil log_2 d \rceil$ NLMs. Previous: $\mathcal{O}\left(\sqrt{\frac{2^d}{d}}\right)$ NLMs.
- Comes at the cost of working in arbitrarily large fields.
- **AES** S-boxes now need only **3** NLMs over $\mathbb{F}_{2^{16}}$.

Generalised previous **lower** bound results to arbitrary binary finite fields.

Conclusion

We *improve* the [CRV14] method for evaluating S-box polynomials.

- Main idea is to work over fields **bigger** than $\mathbb{F}_{2^d}$ for a d-to-r-bit S-box.
- **Reduced** the non-linear complexity for many S-boxes.
  - **DES** S-boxes now need only **3** NLMs over $\mathbb{F}_{2^8}$.
  - Improvement in the running time of masked DES by around 25%.

"Improved" **upper** bound on the complexity of d-to-r-bit S-boxes

- **New**: $\lceil log_2 d \rceil$ NLMs. Previous: $\mathcal{O}\left(\sqrt{\frac{2^d}{d}}\right)$ NLMs.
- Comes at the cost of working in arbitrarily large fields.
- **AES** S-boxes now need only **3** NLMs over $\mathbb{F}_{2^{16}}$.

Generalised previous **lower** bound results to arbitrary binary finite fields.

Any Questions?