Analysis and Improvement of the Generic Higher-Order Masking Scheme of FSE 2012[⋆]

Arnab Roy and Srinivas Vivek

University of Luxembourg, Luxembourg
{arnab.roy,srinivasvivek.venkatesh}@uni.lu

Abstract. Masking is a well-known technique used to prevent block cipher implementations from side-channel attacks. Higher-order side channel attacks (e.g. higher-order DPA attack) on widely used block cipher like AES have motivated the design of efficient higher-order masking schemes. Indeed, it is known that as the masking order increases, the difficulty of side-channel attack increases exponentially. However, the main problem in higher-order masking is to design an efficient and secure technique for S-box computations in block cipher implementations. At FSE 2012, Carlet et al. proposed a generic masking scheme that can be applied to any S-box at any order. This is the first generic scheme for efficient software implementations. Analysis of the running time, or *masking complexity*, of this scheme is related to a variant of the well-known problem of efficient exponentiation (*addition chain*), and evaluation of polynomials.

In this paper we investigate optimal methods for exponentiation in $\mathbb{F}_{2^n}$ by studying a variant of addition chain, which we call *cyclotomic-class addition chain*, or *CC-addition chain*. Among several interesting properties, we prove lower bounds on min-length CC-addition chains. We define the notion of $\mathbb{F}_{2^n}$-polynomial chain, and use it to count the number of *non-linear* multiplications required while evaluating polynomials over $\mathbb{F}_{2^n}$. We also give a lower bound on the length of such a chain for any polynomial. As a consequence, we show that a lower bound for the masking complexity of DES S-boxes is three, and that of PRESENT S-box is two. We disprove a claim previously made by Carlet et al. regarding min-length CC-addition chains. Finally, we give a polynomial evaluation method, which results into an improved masking scheme (compared to the technique of Carlet et al.) for DES S-boxes. As an illustration we apply this method to several other S-boxes and show significant improvement for them.

Keywords: block cipher, S-box, masking complexity, addition chain, polynomial evaluation, side-channel attack.

1 Introduction

Side-channel attacks are considered to be an important class of cryptanalysis techniques in modern cryptography. These attacks exploit various types of phys-

[⋆] This paper has been accepted for publication at CHES 2013. This is the full version.

ical leakage of information including power consumption, running time, electromagnetic emission etc. during the execution of cryptographic algorithm on a target device [11]. In practice they are often more successful than the black-box cryptanalysis, and many such practical attacks were demonstrated against well-known ciphers. Hence it is a natural concern to protect a cryptosystem against these attacks.

Masking is a widely used technique to protect block cipher implementations from side-channel attacks. Goubin and Patarin proposed one such scheme for DES [7]. Many other techniques for both hardware and software implementation were later proposed, especially for AES (see [4] and references therein). Most of these schemes have masking order one and, as a result, they are only resilient against *first-order* side-channel attacks. However in the past years, higher-order side-channel attacks have been proposed against well-known ciphers like AES. Motivated by these attacks, several higher-order masking schemes have been proposed.

In a higher-order masking scheme each sensitive variable (e.g. variables involving secret keys) is randomly split into $d+1$ shares, where d is known as the *masking order*. Chari et al. [5] showed that the complexity of side-channel attacks increases exponentially with the masking order. However implementing a higher-order masking scheme will also affect the performance of the cryptographic algorithm. Hence an algorithm resilient to higher-order attacks aims at designing efficient masking techniques for block ciphers.

Higher-Order Masking: Although many masking techniques have been proposed in literature, there are only a few that deal with higher-order masking. Schramm and Paar [18] generalized the first-order table recomputation method given in [1,12]. Their method can be applied to protect any S-box, but a third-order attack was shown against this scheme by Coron et al. [6]. Rivain et al. also proposed a scheme with formal security proofs but their method only gives second-order security [15]. Ishai et al. [8] provided the first dth-order masking method that can be applied to any S-box, for arbitrary d. However, applying this technique for masking S-boxes in software becomes inefficient. Rivain and Prouff [16] presented an efficient technique for masking AES S-box for any order. Further Kim et al. [9] extended this scheme based on an approach of [17]. In FSE 2012, Carlet et al. [4] presented the first generic dth-order masking scheme, suitable for software implementation, that can be applied to any S-box. Currently, this is the only such generic scheme.

Masking, Polynomial Evaluation, and Addition Chains An $(n,\, m)$-S-box is a function from $\{0,1\}^n$ to $\{0,1\}^m$, where $m \leq n$. For most of the well-known ciphers, n is 4, 6 or 8. To design a generic masking scheme, Carlet et al. [4] consider a polynomial representation of an $(n,\, m)$-S-box over $\mathbb{F}_{2^n}$. The n-bit and m-bit strings are identified with elements of $\mathbb{F}_{2^n}$ in a natural way, if necessary, by appending m-bit strings with leading zeros. Such a polynomial can be easily computed from the S-box table by applying Lagrange interpolation method. The polynomial will be of the form $\sum_{i=0}^{2^n-1} a_i\, x^i$, where $a_i \in \mathbb{F}_{2^n}$. Hence the evaluation of an S-box reduces to evaluating the corresponding polynomial for some element in $\mathbb{F}_{2^n}$. Operations involved in this polynomial evaluation are: addition, multiplication by a scalar (from $\mathbb{F}_{2^n}$), squaring, and multiplications that are not squaring. Except the last one, all the above operations are affine in $\mathbb{F}_{2^n}$. In this masking scheme only the *non-linear multiplications* are significant. Because the dth-order masking of an affine operation requires $O(d)$ logical operations, whereas a non-linear multiplication requires $O(d^2)$ operations [4]. Hence the *masking complexity* of a S-box is defined as the minimum number of non-linear multiplications needed to evaluate its corresponding polynomial.

Efficient methods for polynomial evaluation is a well-studied area [10, Section 4.6.4]. Of particular interest is the evaluation of a power function (i.e. x^α), because of its simplicity. Not only are these functions of theoretical interest, there are also studies on the suitability of S-boxes based on power functions [13]. Formal analysis of the optimal methods to evaluate these powers has led to a detailed study of *addition chains* [21,10, Section 4.6.3]. The length of these chains correspond to the number of multiplications needed for the corresponding exponentiation. However, to analyze the number of non-linear multiplications required to evaluate an S-box, we need to investigate a variant of addition chain introduced in [4]. We call this variant as *cyclotomic-class addition chain*, or in short, *CC-addition chain* to distinguish it from the usual addition chain. Also, CC-addition chains more accurately model the cost of exponentiations in $\mathbb{F}_{2^n}$. This is because squaring is very efficient in $\mathbb{F}_{2^n}$, and we can also use the relation $x^{2^n} = x$ to our advantage.

Our Results

In this article we analyze and improve the generic higher order masking scheme proposed by Carlet et al. at FSE 2012 [4]. We start by establishing several interesting properties of CC-addition chain. We prove a lower bound on the min-length CC-addition chain of any integer, which turns out to be logarithmic in the Hamming weight of the integer. As a consequence, we disprove the claim in [4, pp. 373] saying that integers of the form $2^n - 2$ have the longest min-length CC-addition chain than any other lesser number. We give an elegant mathematical proof showing that the masking complexity of AES is at least four, which was previously established by the brute-force method in [4]. We also give a result on the monotonicity property of the min-length CC-additions of an integer.

We propose and define the notion of $\mathbb{F}_{2^n}$-polynomial chain. Although the notion of CC-addition chain helps to evaluate the masking complexity of power functions, in case of general polynomials the idea of $\mathbb{F}_{2^n}$-polynomial is more natural and useful. Such a notion is necessary to formally define and establish lower bounds on the masking complexity of an S-box. We prove a lower bound on the minimum number of non-linear multiplications required to evaluate a polynomial in $\mathbb{F}_{2^n}$. This lower bound is related to the min-length CC-addition chains of the integers present in the exponents of the polynomial. As a corollary we show that the masking complexity of DES (S-box) is at least three and that of PRESENT is at least two. Previously no such lower bounds were known. We prove that the notion of masking complexity is invariant of the way of representing the corresponding field. One can argue that the linearity of the field isomorphism reasoning given in [4] is incomplete.

Finally, we give a polynomial evaluation technique which improves the efficiency of generic higher-order masking of S-boxes. For DES this algorithm gives improvement over the previously proposed algorithm in [4] and automatically improves the upper bound on the masking complexity of DES S-boxes to 7, from 10. We apply this technique to other well-known ciphers to demonstrate the efficiency of this technique (c.f. Table 1). When applied to AES this technique gives the optimal masking complexity.

2 Results on Cyclotomic-Class Addition Chains

2.1 Definitions

Let $\mathbb{N}$ be the set of positive integers and $\mathbb{Z}$ be the set of integers. $\nu(n)$ refers to the number of bits that are one in the binary representation of n, i.e. the Hamming weight of n. For a binary string z in $\{0,1\}^*$, $\langle z \rangle_2$ denotes the binary representation of some non-negative integer. Let us recollect the standard notion of *addition chain*.

Definition 1. [Addition Chain [10, Section 4.6.3]] *An* addition chain *S for α ($\alpha \in \mathbb{N}$) is a sequence of integers*

$$a_0 = 1,\ a_1,\ a_{2,}\ \dots\ ,\ a_r = \alpha, \tag{1}$$

such that for every $i = 1, 2, \dots, r$, there exist some $0 \leq j, k < i$ such that

$$a_i = a_j + a_k.$$

The length of S, denoted by $L(S)$, is r.

Thus in an addition chain, any element in the sequence (except the first) must be a sum of some previous two elements. The length of a shortest addition chain for α is denoted by $l(\alpha)$. Formally,

$$l(\alpha) = \min\{L(S)\ :\ S \text{ is an addition chain for } \alpha\}\,. \tag{2}$$

Intuitively, $l(\alpha)$ represents the minimum number of "multiplications" needed to compute x^α from x (x is an element of a *monoid*).

The notion of "addition chain" has been generalized to *q-addition chain* ($q \in \mathbb{N}$) in [20]. In this generalization of the "usual" addition chains the multiple of an element by q can be computed in a single step. Note that an (usual) addition chain is a 2-addition chain.

The q-addition chains are more relevant than (2-)addition chains in the case of exponentiations in finite fields $\mathbb{F}_{q^n}$ of characteristic $q \neq 2$. In such a field it is possible to compute x^q very efficiently, often "free" [20].

In this work we study another variant of addition chain introduced in [4]. Before we describe the variant, let us first see the following definition.

Definition 2. [Cyclotomic Class [4]] *Let $n \in \mathbb{N}$ and $\alpha \in \{0, 1, \ldots, 2^n - 2\}$. The* cyclotomic class *of α (w.r.t. n), denoted by C_α, is defined as*

$$C_\alpha = \left\{\alpha \cdot 2^i \pmod{2^n - 1} \; : \; i = 0, 1, \ldots, n - 1\right\}.$$

The intuition for introducing the above definition comes from the following scenario. Let g be a generator of the multiplicative group $\mathbb{F}_{2^n}^\times$. Given $x = g^\alpha$, the set $\left\{x, x^2, x^4, x^8, \ldots,\right\}$ is the same as $\left\{g^i \,|\, i \in C_\alpha\right\}$. Note that $x^{2^n} = x$ in $\mathbb{F}_{2^n}^\times$. Since $2^n \equiv 1 \pmod{2^n - 1}$, therefore $|C_\alpha| \leq n$. It is easy to see that the relation R on set $\{0, 1, \ldots, 2^n - 2\}$, defined as $(\alpha, \beta) \in R$ iff $\beta \in C_\alpha$, is an equivalence relation. Hence the collection of cyclotomic classes forms a partition of the set $\{0, 1, \ldots, 2^n - 2\}$. Since $|C_\alpha| \leq n$, we obtain the following observation.

Remark 1. The number of cyclotomic classes w.r.t. n is at least $\frac{2^n - 1}{n}$.

In [4], the exact count of the number of cyclotomic classes (w.r.t. n) is given as $\sum\limits_{\delta | (2^n - 1)} \frac{\phi(\delta)}{\mu(\delta)}$, where ϕ is the Euler's totient function and $\mu(\delta)$ is the multiplicative order of 2 modulo δ. However, no lower bound on this expression was given there. The simple observation in Remark 1 shows that $\sum\limits_{\delta | (2^n - 1)} \frac{\phi(\delta)}{\mu(\delta)} \geq \frac{2^n - 1}{n}$.

A variant of addition chain proposed in [4] is the *cyclotomic-class addition chain*, in short, *CC-addition chain*.

Definition 3. [CC-Addition Chain [4]] *Let $n \in \mathbb{N}$, $\alpha \in \{1, 2, \ldots, 2^n - 2\}$, and $C = \{C_i \, : \, i = 0, 1, \;\; \ldots, 2^n - 2\}$ be the collection of cyclotomic classes w.r.t. n, A* cyclotomic-class addition chain *S_C of α (w.r.t. n) is a sequence of cyclotomic classes*

$$C_{a_0} = C_1, \; C_{a_1}, \; C_{a_{2,}} \; \ldots, \; C_{a_r} = C_\alpha, \tag{3}$$

such that for every $i = 1, 2, \ldots, r$, there exist some $0 \leq j, k < i$, $\beta_i \in C_{a_i}$, $\beta_j \in C_{a_j}$, and $\beta_k \in C_{a_k}$ such that

$$\beta_i \equiv \beta_j + \beta_k \pmod{2^n - 1}.$$

The length of S_C, denoted by $LC_n\left(S_C\right)$, is r.

Formally, a shortest CC-addition chain for α (w.r.t. n), denoted by $m_n(\alpha)$, is defined as

$$m_n(\alpha) = \min\left\{LC_n\left(S_C\right) \; : \; S_C \text{ is an addition chain for } \alpha \text{ (w.r.t. } n)\right\}. \tag{4}$$

The phrase "masking complexity of α" has been used in [4] to describe $m_n(\alpha)$. CC-addition chains describe a way to compute x^α from $x \in \mathbb{F}_{2^n}^\times$, where squaring operations are considered free and hence not counted. These sort of chains model the complexity of exponentiation in $\mathbb{F}_{2^n}$ more accurately than (2-)addition chains when squaring is implemented very efficiently using a special representation of field elements [20]. CC-addition chains also model exactly the number of *non-linear* multiplications required to mask S-boxes that are represented by *power functions* [4]. An important difference between q-addition chains, in particular

2-addition chains, and CC-addition chains is that the former is a sequence of positive integers while the latter is a sequence of classes. It is for this reason that we refer to the latter chain as "cyclotomic-class addition chain" and not just 2-addition chain as done in [4]. The notion of CC-addition chains can be extended in a natural way to $\mathbb{F}_{q^n}$ to obtain q-CC-addition chain, analogous to q-addition chain. Accordingly, the CC-addition chain in Definition 3 may also be referred to as 2-CC-addition chain. In this work, we restrict ourselves to (2-)CC-addition chains, particularly keeping applications to higher-order masking in mind.

Note that $m_n(\alpha)$ is not necessarily equal to the minimum number of non-doubling steps in all of addition chains for α, though $m_n(\alpha) \leq l(\alpha)$. That is, every CC-addition chain does not necessarily need to be derived from an addition chain by not explicitly writing the doubling steps. This is a consequence of the fact that there exist α, n_1 and n_2 such that $m_{n_1}(\alpha) \neq m_{n_2}(\alpha)$. For example, $m_5(23) = 2$ but $m_6(23) = 3$. We refer to the table of values for $m_n(\alpha)$ for $n \leq 11$ in [4].

Nevertheless, we can obtain upper bounds on the value of $m_n(\alpha)$ using previous results on addition chains in a straightforward way. Note that for a given value of α, $m_n(\alpha)$ is defined only for those n such that $\alpha \leq 2^n - 2$. Hence we require $n \geq \lceil \log_2 (\alpha + 2) \rceil$.

Upper bound for $\boldsymbol{m_n(\alpha)}$ A trivial upper bound $m_n(\alpha) \leq \nu(\alpha) - 1$ is obtained from the *binary method* [10, Section 4.6.3]. Let $\alpha = b_t 2^t + b_{t-1} 2^{t-1} + \ldots + b_1 2^1 + b_0$, where $t = \lfloor \log_2 \alpha \rfloor$, $b_i \in \{0, 1\}$ $\forall i = 1, \ldots, t$, and $b_t = 1$. An addition chain obtained from the binary method is as follows

$$b_t = 1,\ b_t 2,\ b_t 2 + b_{t-1},\ 2\left(b_t 2 + b_{t-1}\right),\ b_t 2^2 + b_{t-1} 2 + b_{t-2},\ \ldots\ ,\ \alpha.$$

The above addition chain yields a CC-addition chain for α (w.r.t. any $n \geq \lceil \log_2 (\alpha + 2) \rceil$). Hence the length of such a chain is $\nu(\alpha) - 1$. Note that we count only those additions that are not doublings.

An improved upper bound for $m_n(\alpha)$ is possible if we use the techniques of Brauer [3]. In [3], addition chains much shorter than those from the binary method have been constructed. This result on (2-)addition chains has also been extended to q-addition chains in [20]. See also [22,10, Section 4.6.3].

Brauer's method of constructing addition chains is a generalization of the binary method mentioned above. Instead of working in the base-2 expansion of α, we now work with base-2^k expansion ($k \in \mathbb{N}$). Let $z = 2^k$ and $\alpha = b_t z^t + b_{t-1} z^{t-1} + \ldots + b_1 z^1 + b_0$, where $t = \lfloor \log_z \alpha \rfloor$, $b_i \in \{0, 1, \ldots, z-1\}$ $\forall i = 0, 1, \ldots, t$, and $b_t \neq 0$. The corresponding addition chain is

$$\begin{aligned}
&1,\ 2,\ \ldots\ ,\ z-2,\ z-1,\\
&\quad b_t 2,\ b_t 4,\ \ldots\ ,\ b_t z,\ b_t z + b_{t-1},\\
&\qquad \left(b_t z + b_{t-1}\right) 2,\ \left(b_t z + b_{t-1}\right) 4,\ \ldots\ ,\ \left(b_t z + b_{t-1}\right) z,\ b_t z^2 + b_{t-1} z + b_{t-2},\\
&\qquad\qquad\qquad\qquad \ldots \qquad\qquad bz^t + b_{t-1} z^{t-1} + \ldots + b_1 z^1 + z_0.
\end{aligned}$$

The total length of the above addition chain is $z - 2 + t(k+1)$. The number of non-doubling steps is $(z-2)/2 + t = 2^{k-1} - 1 + \left\lfloor \frac{\log_2 \alpha}{k} \right\rfloor$, which is also the length of the corresponding CC-addition chain for α (w.r.t. any n). This value is minimized when $k \approx \log_2 \log_2 \alpha - 2 \log_2 \log_2 \log_2 \alpha$ and the corresponding value is about $\frac{\log_2 \alpha}{\log_2 \log_2 \alpha - 2 \log_2 \log_2 \log_2 \alpha} + \frac{\log_2 \alpha}{2(\log_2 \log_2 \alpha)^2} - 1$. Hence as $\alpha \to \infty$, we obtain

$$m_n(\alpha) \leq \frac{\log_2 \alpha}{\log_2 \log_2 \alpha} \left(1 + o(1)\right). \tag{5}$$

2.2 Lower bound

No non-trivial lower bounds have been previously known for $m_n(\alpha)$. In this article we show that $m_n(\alpha) \geq \lceil \log_2(\nu(\alpha)) \rceil$. Recall that $\nu(\alpha)$ is the Hamming weight of α in the binary notation. The basic idea is to first show that Hamming weight is invariant in a cyclotomic class. To obtain the bound, we then use this result along with the simple fact that when two positive integers are added, then the Hamming weight of sum is at most the sum of the Hamming weights. Similar techniques have been used in [20].

Lemma 1. *Let $n \in \mathbb{N}$, $\alpha \in \{0, 1, \ldots, 2^n - 2\}$, and C_α be the cyclotomic class of α (w.r.t. n). If $\beta \in C_\alpha$, then $\nu(\beta) = \nu(\alpha)$.*

Proof. This follows from a well-known observation that the multiplication of α by 2 modulo $2^n - 1$ is same as the cyclic left shift of the n-bit binary representation of α.

As an illustration, consider the cyclotomic class C_3 of $\alpha = 3$ w.r.t. $n = 5$. $C_3 = \{3, 6, 12, 24, 17\}$. Note that $17 \cdot 2 \equiv 3 \pmod{31}$. In the binary representation,

$$C_3 = \left\{ \langle 00011 \rangle_2, \langle 00110 \rangle_2, \langle 01100 \rangle_2, \langle 11000 \rangle_2, \langle 10001 \rangle_2 \right\}. \tag{6}$$

The following proposition gives a lower bound for $m_n(\alpha)$.

Proposition 1. $m_n(\alpha) \geq \lceil \log_2(\nu(\alpha)) \rceil$.

Proof. From Lemma 1 and, the fact that the Hamming weight of sum of two positive integers is at most the sum of the Hamming weights, we obtain that the CC-addition chain of length at most r (3) can only contain integers having Hamming weight at most 2^r. This is because elements of C_1 have Hamming weight 1 and at each step the Hamming weight can at most double. Therefore, in order for α to be present in a CC-addition chain, then the chain's length must be at least $\lceil \log_2(\nu(\alpha)) \rceil$. □

As a consequence of the above proposition, we now disprove the claim made in [4, pp. 373]. Their claim was that given a (fixed) value of n, $m_n(2^n - 2) \geq m_n(\alpha)\ \forall \alpha = 1, \ldots, 2^n - 3$, i.e., $2^n - 2$ has the longest min-length CC-addition chain among the integers modulo $2^n - 1$.

Proposition 2. *Let $n = 2^t + 1$ for some $t \in \mathbb{N}$ and $t > 2$. Then $m_n(2^n - 2) = t$. In particular, $m_9(510) = 3 < m_9(508) = 4$.*

Proof. In Appendix A.

2.3 Monotonicity of $m_n(\alpha)$

It is natural to ask how the value of $m_n(\alpha)$ varies with n. As mentioned previously, $m_n(\alpha)$ is defined only for $n \geq \lceil \log_2(\alpha+2) \rceil$. Is the value of $m_n(\alpha)$ independent of n for a given value of α? This is not true since we have already seen the counterexample $m_5(23) = 2$ but $m_6(23) = 3$. The example $m_7(83) = 3$ but $m_9(83) = 2$ shows that $m_n(\alpha)$ can also decrease as n increases. We can generalize the above examples to obtain infinitely many examples. For instance, consider $m_n\Big(\langle 1\underbrace{0\ldots0}_{n-4}111\rangle_2\Big) = m_n\Big(\langle \underbrace{0\ldots0}_{n-4}1111\rangle_2\Big) = 2$ but $m_{n+1}\Big(\langle 01\underbrace{0\ldots0}_{n-4}111\rangle_2\Big) = m_{n+1}\Big(\langle \underbrace{0\ldots0}_{n-4}11101\rangle_2\Big) = 3$, where $n \geq 5$.

But we can still show that $m_n(\alpha) \leq m_{n'}(\alpha)$ if $n \,|\, n'$, i.e. if n divides n'.

Theorem 1. *Let $\alpha, n, n' \in \mathbb{N}$, $n \,|\, n'$ and $\lceil \log_2(\alpha+2) \rceil \leq n \leq n'$. Then $m_n(\alpha) \leq m_{n'}(\alpha)$.*

Proof. In Appendix B.

Theorem 1 suggests that, to find a minimum length CC-addition chain w.r.t. n', first try to find one w.r.t. a divisor n of n'. Since $\mathbb{F}_{2^n}$ is a smaller field than $\mathbb{F}_{2^{n'}}$, it may be advantageous to work in $\mathbb{F}_{2^n}$. Once a minimum length CC-addition chain w.r.t. n' is found, then check if it is a CC-addition chain w.r.t. n'. If it is the case, then it will be a minimum length chain.

3 Polynomial Evaluation and Masking Complexity

3.1 $\mathbb{F}_{2^n}$-Polynomial Chain

The masking complexity of an S-box (Definition 5) corresponds to the min-length CC-addition chain of the exponent when it can be represented as a power function. However when the S-box has a general polynomial representation, a notion similar to CC-addition chain is required. For evaluating polynomials (over $\mathbb{R}$) the notion of *polynomial chain* is given in [10, Section 4.6.4]. In case of polynomials in $\mathbb{F}_{2^n}[x]$, we define the notion of $\mathbb{F}_{2^n}$*-polynomial chain*, where we do not count addition, scalar multiplication and squaring operations. Note that if $x, y \in \mathbb{F}_{2^n}$, then $x^{2^n} = x$ and $(x+y)^2 = x^2 + y^2$.

Definition 4. *A* $\mathbb{F}_{2^n}$-polynomial chain *S for a polynomial $P(x) \in \mathbb{F}_{2^n}[x]$ is defined as*

$$\lambda_{-1} = 1,\ \lambda_1 = x,\ \ldots\ ,\ \lambda_r = P(x) \tag{7}$$

where

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

Note that here $\cdot$ and $\odot$ both perform the same operation, multiplication in $\mathbb{F}_{2^n}$. However in order to differentiate the non-linear operation we use $\odot$ for scalar multiplication. Here $\lambda_j \cdot \lambda_k$ denotes a non-linear multiplication. Let the number of non-linear multiplications involved in chain S be $\mathcal{N}(S)$. Then the **non-linear complexity** *of $P(x)$ (over $\mathbb{F}_{2^n}$), denoted by $\mathcal{M}(P(x))$, is defined as $\mathcal{M}(P(x)) = \min_{S} \mathcal{N}(S)$, where S computes $P(x)$.*

Proposition 3. *Let $P(x) := \sum_{i=0}^{2^n-1} a_i\, x^i$ be a polynomial in $\mathbb{F}_{2^n}[x]$. Then*

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

Proof. To prove the proposition, we just need to prove the following claim. Let $\sigma_k^n := \{\alpha \,|\, m_n(\alpha) \ \leq k\}$. We claim that, with at most k non-linear multiplications, we can evaluate only those polynomials of the form $\sum_i a_i x^i$, where $i \in \sigma_k^n$ and $a_i \in \mathbb{F}_{2^n}$. It is easy to see that with zero non-linear multiplications, only those polynomials of the form $\sum_i a_i x^i$, where $i \in \sigma_0^n = \{2^j \mid 0 \leq j \leq n-1\}$. Let us assume that the above claim is true up to $k-1$ non-linear multiplications. Consider the set of polynomials $T := \{p(x) \mid p(x) = \sum_j b_j x^j,\, j \in \sigma_{k-1}^n,\, b_j \in \mathbb{F}_{2^n}\}$. Since squaring is a linear operation in $\mathbb{F}_{2^n}[x]$, the set T is closed under additions, scalar multiplications and squaring operations. Hence if we allow only one more non-linear multiplication, then exponents in the resulting polynomial can only be from σ_k^n. Note that $m_n(\alpha)$ is defined only for $0 < \alpha < 2^n - 1$ and $x^{2^n-1} = 1$ if $x \neq 0$. This proves the claim. □

3.2 Masking Complexity: Well-definedness and Lower Bounds

The *masking complexity* of an S-box is formally defined as follows.

Definition 5. [Masking Complexity] *Let $m, n \in \mathbb{N}$ with $m \leq n$. The* masking complexity *of an (n, m)-S-box is the non-linear complexity of $P(x)$, where $P(x)$ is the polynomial representation of the S-box over $\mathbb{F}_{2^n}$.*

Note that the above definition has been intuitively described in [4, Definition 1] as the minimum number of non-linear multiplications needed to evaluate the polynomial representation. Once the bit strings are identified naturally with the elements of $\mathbb{F}_{2^n}$ (given a field representation), then we can apply Lagrange interpolation technique to compute the (unique) polynomial of degree at most $2^n - 1$ representing the S-box in the corresponding field.

Well-definedness The well-definedness and relevance of the above definition of masking complexity is guranteed because of the following reasons.

1. A natural question is - *does masking complexity change with the irreducible polynomial used to represent $\mathbb{F}_{2^n}$?* Note that under the natural mapping of bit strings to the field elements, the same S-box may correspond to different polynomials over $\mathbb{F}_{2^n}$ for different representations of the field. However we show in Theorem 2 that masking complexity does not depend on the field representation.

2. It is relatively straightforward to mask affine functions. In $\mathbb{F}_{2^n}$, squaring is linear, and affine functions are free from any "non-linear" multiplications.

The n-bit strings can be naturally mapped to field elements of $\mathbb{F}_{2^n}$ represented as polynomials over $\mathbb{F}_2$ modulo a degree n irreducible polynomial $f_1(y)$. Formally, $\mathcal{B}_1 : \{0,1\}^n \to \mathbb{F}_2[y]/f_1(y)$ is defined as

$$\mathcal{B}_1\left(\langle b_{n-1}b_{n-2}\dots b_0\rangle\right) := \sum_{i=0}^{n-1} b_i\, y^i + \left(\mathbb{F}_2[y] \cdot f_1(y)\right), \tag{8}$$

where $b_i \in \{0,1\}$. The m-bit strings ($m \leq n$) are appended with leading zeros to identify them with n-bit strings. Later we shall see that it suffices if $\mathcal{B}_1$ is some $\mathbb{F}_2$-linear bijection. Note that $(\{0,1\}^n, \oplus)$ may be viewed as a vector space over $\mathbb{F}_2$.

Remark 2. It was claimed in [4, Remark 3] that the property of independence of masking complexity w.r.t. the irreducible polynomial used to represent $\mathbb{F}_{2^n}$ follows from the fact that field isomorphisms are $\mathbb{F}_2$-linear bijections. This reason is not enough and a formal proof requires more arguments, as we shall see in the proof of Theorem 2.

Let $f_1(y)$ and $f_2(z)$ be two irreducible polynomials of degree n over $\mathbb{F}_2$. Then $\mathbb{F}_2[y]/f_1(y)$ and $\mathbb{F}_2[z]/f_2(z)$ are two representations for $\mathbb{F}_{2^n}$. Let $\mathcal{B}_1 : \{0,1\}^n \to \mathbb{F}_2[y]/f_1(y)$ be as in (8), and $\mathcal{B}_2 : \{0,1\}^n \to \mathbb{F}_2[z]/f_2(z)$ be analogously defined for $f_2(z)$. Note that $\mathcal{B}_1$ and $\mathcal{B}_2$ are $\mathbb{F}_2$-linear isomorphisms between vector spaces. The corresponding inverse maps $\mathcal{B}_1^{-1}$ and $\mathcal{B}_2^{-1}$ are also $\mathbb{F}_2$-linear isomorphisms of vector spaces.

Let $\mathcal{U} : \{0,1\}^n \to \{0,1\}^n$ be any function on n-bit strings. For instance, $\mathcal{U}$ may represent an (n,m)-S-box (upon padding m-bit strings with leading zeros). The maps $\mathcal{U}$ and $\mathcal{B}_1$ will "induce" a map $\mathcal{U}_1 : \mathbb{F}_2[y]/f_1(y) \to \mathbb{F}_2[y]/f_1(y)$. More precisely,

$$\mathcal{U}_1 = \mathcal{B}_1 \circ \mathcal{U} \circ \mathcal{B}_1^{-1}. \tag{9}$$

Similarly we can define

$$\mathcal{U}_2 = \mathcal{B}_2 \circ \mathcal{U} \circ \mathcal{B}_2^{-1}. \tag{10}$$

Let $P_1(x)$ and $P_2(x)$ be the polynomial representations (of degree at most $2^n - 1$) of $\mathcal{U}_1$ and $\mathcal{U}_2$, respectively. We now prove the following theorem.

Theorem 2. *$\mathcal{M}(P_1(x)) = \mathcal{M}(P_2(x))$, where $P_1(x)$ and $P_2(x)$ are as defined above. In other words, the masking complexity of an S-box (in general, any function on bit strings) is invariant w.r.t. field representations.*

Proof. In Appendix C.

Lemma 2. [4, Proposition 1] *The masking complexity of an S-box (in general, any function) cannot increase when it is composed with affine functions. When composed with affine bijections, then masking complexity remains the same.*

Remark 3. Note that Lemma 2 holds only when the evaluation of affine functions over $\mathbb{F}_{2^n}$does not involve any non-linear multiplication. For the sake of completeness, this property is proved in Lemma 6.

Note that in the proof of Theorem 2 the only property of the maps $\mathcal{B}_1$ and $\mathcal{B}_2$ used is that they are $\mathbb{F}_2$-linear bijections. Hence if $\mathcal{B}_1$ and $\mathcal{B}_2$ are any linear bijections, even then the masking complexity of an S-box remains invariant.

Lower Bounds We represent the fields $\mathbb{F}_{2^4}$, $\mathbb{F}_{2^6}$ and $\mathbb{F}_{2^8}$ using irreducible polynomials $y^4 + y^3 + 1$, $y^6 + y^4 + y^3 + y + 1$, $y^8 + y^4 + y^3 + y + 1 \in \mathbb{F}_2[y]$, respectively. From Theorem 2, we know that the masking complexity is invariant w.r.t. the field representations.

The polynomials corresponding to eight DES S-boxes are polynomials of degree 62 in $\mathbb{F}_{2^6}[x]$, and the one for the PRESENT S-box is a polynomial of degree 14 in $\mathbb{F}_{2^4}[x]$. Since $m_6(62) = 3$ and $m_4(14) = 2$, from Proposition 3 we obtain the following corollary.

Corollary 1. *Masking complexity of a DES S-box is at least* 3*, and that of the PRESENT S-box is at least* 2*.*

The AES S-box can be written as an affine permutation composed with the polynomial $x^{254} \in \mathbb{F}_{2^8}[x]$. From Lemma 2, the masking complexity of AES S-box is $\mathcal{M}\left(x^{254}\right)$ over $\mathbb{F}_{2^8}$. Using arguments similar to the proof of Lemma 4 (in Appendix A), we obtain the following corollary.

Corollary 2. *Masking complexity of the AES S-box is at least 4.*

The above corollary was shown by exhaustive search in [4].

4 Improved Generic Higher-Order Masking of S-boxes

In [4], Carlet et al. used the cyclotomic class method to get a masking scheme for S-boxes. They also gave parity-split method to evaluate polynomials efficiently. In this section we apply a *divide-and-conquer* method to obtain an efficient solution to the same problem. The main idea of this approach is to express the polynomial(say, having degree N) as a function of several lower degree polynomials, each of degree at most k (for some fixed k).

Let $P(x)$ be the polynomial of degree N which we want to evaluate. Then we start by dividing the polynomial with x^{kt} where $N = k(2t-1)$. The remainder obtained by this will have degree at most $kt-1$ and degree of the quotient will be $kt - t = k(t-1)$. Next we can add the term $x^{k(t-1)}$ to the remainder and divide the sum by the quotient. This allows us to express the remainder by polynomials having degree at most $k-1$ and $k(t-1)-1$. Now the term $x^{k(t-1)}$ together with the other lower degree polynomials will allow us to apply the method recursively when $t = 2^l$.

In [14] this divide-and-conquer approach for monic polynomials is proposed. For the sake of completeness, a brief description of this general method is given in the Appendix D. However, we observe that in our case the restriction of polynomial being monic is not necessary. Also it turns out that we can adapt that algorithm even if the condition $N = k(2t-1)$ is not satisfied. We describe this with specific examples of DES, AES and some other well-known S-boxes.

4.1 DES S-boxes

Let $P_{DES}(x)$ be the polynomial in $\mathbb{F}_{2^6}[x]$ corresponding to an S-box of DES. Note that for all the S-boxes the corresponding polynomial has degree 62. We express P_{DES} as

$$P_{DES}(x) = q(x) \cdot x^{36} + R(x) \tag{11}$$

where $deg(R) \leq 35$ and $deg(q) = 26$. Now if we divide the polynomial $R(x) - x^{27}$ with $q(x)$, we get $c(x)$ and $s(x)$ satisfying

$$R(x) - x^{27} = c(x) \cdot q(x) + s(x) \tag{12}$$

where $deg(c) \leq 9$ and $deg(s) \leq 25$. Substituting (12) in (11), we get

$$P_{DES}(x) = (x^{36} + c(x)) \cdot q(x) + x^{27} + s(x) \tag{13}$$

Further continuing in the same way we first divide $q(x)$ with x^{18} to obtain

$$q(x) = q_1(x) \cdot x^{18} + R_1(x), \tag{14}$$

and then divide $R_1(x) - x^9$ by $q_1(x)$ to obtain

$$R_1(x) - x^9 = c_1(x) \cdot q_1(x) + s_1(x). \tag{15}$$

Combining (14) and (15) we get

$$q(x) = (x^{18} + c_1(x)) \cdot q_1(x) + x^9 + s_1(x). \tag{16}$$

Where $deg(R_1) \leq 17$, $deg(q_1) = 8$, $deg(c_1) \leq 9$ and $deg(s_1) \leq 7$. Similarly proceeding with $x^{27} + s(x)$, we get $q_2(x)$, $R_2(x)$, $c_2(x)$, and $s_2(x)$ satisfying

$$\begin{aligned} x^{27} + s(x) &= q_2(x) \cdot x^{18} + R_2(x) \\ R_2(x) - x^9 &= c_2(x) \cdot q_2(x) + s_2(x) \end{aligned} \tag{17}$$

where $deg(R_2) \leq 17$, $deg(q_2) = 9$, $deg(c_2) \leq 8$ and $deg(s_2) \leq 8$. Combining them we get

$$x^{27} + s(x) = (x^{18} + c_2(x)) \cdot q_2(x) + x^9 + s_2(x) \tag{18}$$

Finally combining equations (18), (16) and (13), we obtain

$$\begin{aligned} P_{DES}(x) =& (x^{36} + c(x)) \cdot \Big(((x^{18} + c_1(x)) \cdot q_1(x)) + (x^9 + s_1(x)) \Big) \\ &+ \Big((x^{18} + c_2(x)) \cdot q_2(x) + (x^9 + s_2(x)) \Big) \end{aligned} \tag{19}$$

In (19) the number of non-linear multiplications equals $3 + l'$, where l' is the number of non-linear multiplications involved in evaluating the monomials in (19) of degree at most 9, together with monomials x^{18} and x^{36}.
Consider the monomials $x, x^2, x^3, \ldots, x^9$. The number of non-linear multiplications required to evaluate them is 4. From x^9, we can compute $x^{18} = (x^9)^2$ and $x^{36} = (x^{18})^2$ using only squarings. Hence $l' = 4$. Therefore the number of non-linear multiplications for evaluating P_{DES} is $3 + 4 = 7$. Note that this also improves an upper bound on the masking complexity of DES S-boxes.

4.2 AES and other 8-bit S-boxes

Applying the above technique for 8-bit S-boxes leads to significant reduction in the number of non-linear multiplications required, in majority of the cases. To compute the polynomials corresponding to 8-bit S-boxes, we use the field representation $\mathbb{F}_{2^8} = \mathbb{F}_2[y]/(y^8 + y^4 + y^3 + y + 1)$.

CAMELLIA cipher [2] uses four 8-bit S-boxes and all the corresponding polynomials have degree 254. On the other hand, CLEFIA cipher [19] uses two 8-bit S-boxes and one of the corresponding polynomials (for S-box S_0) has degree 252 while the other has degree 254. We treat the above polynomials as if they are having degree $255 = 17 \times \left(2^4 - 1\right)$ and start by dividing with x^{136} (and then adding the term x^{119}). The process continues as done for DES above. For polynomials of degree 254, we need to precompute the powers x^i $(1 \leq i \leq 17)$, whereas for the polynomial of degree 252 we need to precompute until x^{19}. As Table 1 indicates, we require 15 non-linear multiplications for all the corresponding S-boxes except the S-box S_0 (corresponding to the polynomial of degree 252) of CLEFIA, which requires 16 non-linear multiplications. Previously, these S-boxes required 22 non-linear multiplications by the parity-split method of [4].

Table 1. Comparison of the number of non-linear multiplications required for masking various S-boxes

Method	S-box(es)					
	AES	CAMELLIA	CLEFIA	DES	PRESENT	SERPENT
Cyclotomic [4]	4	33	33	11	3	3
Parity-Split [4]	6	22	22	10	4	4
This Paper	**4**	**15**	**16** (S_0)/**15** (S_1)	**7**	**3**	**3**

The polynomial $P_{AES}(x)$ corresponding to the non-linear function of AES S-box is $x^{254} \in \mathbb{F}_{2^8}[x]$. Initially compute x, x^2, x^4, x^8, x^{16}, $x^{17} = x^{16} \cdot x$, $x^{34} = \left(x^{17}\right)^2$, $x^{68} = \left(x^{34}\right)^2$ and $x^{136} = \left(x^{68}\right)^2$. To compute this list only one non-linear multiplication is required. Write $P_{AES}(x) = x^{254} = q(x) \cdot x^{136}$, where $q(x) = x^{118}$. Further, $q(x) = x^{118} = q_1(x) \cdot x^{68}$, where $q_1(x) = x^{50}$. Finally, $q_1(x) = x^{50} = x^{16} \cdot x^{34}$. Hence

$$P_{AES}(x) = \left(\left(x^{16} \cdot x^{34}\right) \cdot x^{68}\right) \cdot x^{136}.$$

Given the initially computed list of powers, the above computation can be done with three non-linear multiplications. So four non-linear multiplications are required all together for the AES S-box, which is exactly equal to its masking complexity. The cyclotomic method of [4] also achieves the optimal number.

4.3 PRESENT and SERPENT S-boxes

We have also considered the application of above techniques to 4-bit S-boxes of PRESENT and SERPENT ciphers. PRESENT cipher has a single S-box, whose corresponding polynomial over $\mathbb{F}_{2^4}[x]$ is of degree 14. We use the representation $\mathbb{F}_{2^4} = \mathbb{F}_2[y]/(y^4 + y + 1)$. SERPENT uses eight 4-bit S-boxes and the corresponding polynomials have degree 14 (for two polynomials), 13 (for five) or 12 (for one). In all the cases we require 3 non-linear multiplications. The cyclotomic method also requires the same number.

An outline of the method is as follows. Initially compute the list x, x^2, $x^3 = x^2 \cdot x$, x^4, $x^5 = x^4 \cdot x$, $x^6 = \left(x^3\right)^2$, $x^{10} = \left(x^5\right)^2$, using two non-linear multiplications. Divide the polynomial by x^{10}, and proceed as done in the case of DES. This process stops at the first level itself, requiring only one non-linear multiplication. This method totally requires three non-linear multiplications.

4.4 Cost of linear operations

The technique presented in this section to evaluate the polynomials corresponding to specific S-boxes has lead to an improvement (or remain the same) in the number of non-linear multiplications required. We would like to note that this method does *not* incur significant overhead with respect to the linear operations. For instance, in the case of DES S-boxes, we need about 63 additions, 58 scalar multiplications, and 6 squarings. Both the cyclotomic method as well as the parity-split method of [4] require about 62 additions and 62 scalar multiplications. The number of squarings for the cyclotomic method is about 50, and it is about 7 for the parity-split method.

An estimate in general for the two methods of [4] is as follows. The number of additions required by both the methods is equal to the number of terms in the polynomial less one, while the number of scalar multiplications is the number of non-monic coefficients less one (for the constant term). Hence for dense polynomials (where most of the 2^n terms are present) both these quantities will be about the degree of the polynomial. The number of squarings for the cyclotomic method is about $2^n - 1$ less the number of cyclotomic classes, while for the parity-split method it is about $2^{\lceil \frac{n}{2} \rceil - 1} + \lfloor \frac{n}{2} \rfloor$ (for dense polynomials).

In our case, if the degree d of a polynomial is approximately $k \cdot (2^m - 1)$, then the number of additions is about $(k+1) \cdot (2^m - 1)$. The number of non-linear multiplications is about $k \cdot (2^m - 1)$. The number of squarings is about $\frac{k}{2} + \log_k d$. Hence if $k \approx \sqrt{d}$, then this is about $\frac{\sqrt{d}}{2} + 2$. Hence for dense polynomials (as is the case for many S-boxes), there is no significant overhead with respect to the linear operations.

5 Conclusion

In this work we have formalized the idea of polynomial chain in $\mathbb{F}_{2^n}$. Using this notion we give bounds on the masking complexity of polynomials corresponding to several S-boxes. The idea of polynomial chain is more generic (in the context of polynomial evaluation). This gives a better way of analyzing the masking complexity for S-boxes which do not correspond to some power function, as is the case for many S-boxes used in popular block ciphers. The polynomial evaluation method described in Section 4 results into more efficient generic higher-order masking scheme for many S-boxes, compared to the algorithms/heuristics provided in [4]. Also our analysis gives insight into the polynomial evaluation methods in $\mathbb{F}_{2^n}$, which could be of independent interest.

Acknowledgements. We would like to thank Jean-Sébastien Coron and Praveen K. Vadnala for suggesting us the FSE 2012 paper of Carlet et al.. We are thankful to the anonymous reviewers of CHES 2013 for their valuable comments and suggestions. Second author would also like to thank Srinivas Karthik for useful discussions.

References

1. Mehdi-Laurent Akkar and Christophe Giraud. An implementation of des and aes, secure against some attacks. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *CHES*, volume 2162 of *Lecture Notes in Computer Science*, pages 309–318. Springer, 2001.
2. Kazumaro Aoki, Tetsuya Ichikawa, Masayuki Kanda, Mitsuru Matsui, Shiho Moriai, Junko Nakajima, and Toshio Tokita. Camellia: A 128-bit block cipher suitable for multiple platforms - design and analysis. In Douglas R. Stinson and Stafford E. Tavares, editors, *Selected Areas in Cryptography*, volume 2012 of *Lecture Notes in Computer Science*, pages 39–56. Springer, 2000.
3. Alfred Brauer. On addition chains. *Bull. Amer. Math. Soc*, 45(10):736–739, 1939.
4. Claude Carlet, Louis Goubin, Emmanuel Prouff, Michaël Quisquater, and Matthieu Rivain. Higher-order masking schemes for s-boxes. In Anne Canteaut, editor, *FSE*, volume 7549 of *Lecture Notes in Computer Science*, pages 366–384. Springer, 2012.
5. Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Wiener [23], pages 398–412.
6. Jean-Sébastien Coron, Emmanuel Prouff, and Matthieu Rivain. Side channel cryptanalysis of a higher order masking scheme. In Pascal Paillier and Ingrid Verbauwhede, editors, *CHES*, volume 4727 of *Lecture Notes in Computer Science*, pages 28–44. Springer, 2007.
7. Louis Goubin and Jacques Patarin. Des and differential power analysis (the "duplication" method). In Çetin Kaya Koç and Christof Paar, editors, *CHES*, volume 1717 of *Lecture Notes in Computer Science*, pages 158–172. Springer, 1999.
8. Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer, 2003.

9. HeeSeok Kim, Seokhie Hong, and Jongin Lim. A fast and provably secure higher-order masking of aes s-box. In Bart Preneel and Tsuyoshi Takagi, editors, *CHES*, volume 6917 of *Lecture Notes in Computer Science*, pages 95–107. Springer, 2011.
10. Donald E. Knuth. *The Art of Computer Programming, Volume II: Seminumerical Algorithms, 3rd Edition.* Addison-Wesley, 1997.
11. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Wiener [23], pages 388–397.
12. Thomas S. Messerges. Securing the aes finalists against power analysis attacks. In Bruce Schneier, editor, *FSE*, volume 1978 of *Lecture Notes in Computer Science*, pages 150–164. Springer, 2000.
13. Yassir Nawaz, Kishan Chand Gupta, and Guang Gong. Algebraic immunity of s-boxes based on power mappings: analysis and construction. *IEEE Transactions on Information Theory*, 55(9):4263–4273, 2009.
14. Mike Paterson and Larry J. Stockmeyer. On the number of nonscalar multiplications necessary to evaluate polynomials. *SIAM J. Comput.*, 2(1):60–66, 1973.
15. Matthieu Rivain, Emmanuelle Dottax, and Emmanuel Prouff. Block ciphers implementations provably secure against second order side channel analysis. In Kaisa Nyberg, editor, *FSE*, volume 5086 of *Lecture Notes in Computer Science*, pages 127–143. Springer, 2008.
16. Matthieu Rivain and Emmanuel Prouff. Provably secure higher-order masking of aes. In Stefan Mangard and François-Xavier Standaert, editors, *CHES*, volume 6225 of *Lecture Notes in Computer Science*, pages 413–427. Springer, 2010.
17. Akashi Satoh, Sumio Morioka, Kohji Takano, and Seiji Munetoh. A compact rijndael hardware architecture with s-box optimization. In Colin Boyd, editor, *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 239–254. Springer, 2001.
18. Kai Schramm and Christof Paar. Higher order masking of the aes. In David Pointcheval, editor, *CT-RSA*, volume 3860 of *Lecture Notes in Computer Science*, pages 208–225. Springer, 2006.
19. Taizo Shirai, Kyoji Shibutani, Toru Akishita, Shiho Moriai, and Tetsu Iwata. The 128-bit blockcipher clefia (extended abstract). In Alex Biryukov, editor, *FSE*, volume 4593 of *Lecture Notes in Computer Science*, pages 181–195. Springer, 2007.
20. Joachim von zur Gathen. Efficient and optimal exponentiation in finite fields. *Computational Complexity*, 1:360–394, 1991.
21. Joachim von zur Gathen and Michael Nöcker. Exponentiation in finite fields: Theory and practice. In Teo Mora and Harold F. Mattson, editors, *AAECC*, volume 1255 of *Lecture Notes in Computer Science*, pages 88–113. Springer, 1997.
22. Joachim von zur Gathen and Michael Nöcker. Computing special powers in finite fields. *Math. Comput.*, 73(247):1499–1523, 2004.
23. Michael J. Wiener, editor. *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*. Springer, 1999.

A Proof of Proposition 2

Proposition *Let $n = 2^t + 1$ for some $t \in \mathbb{N}$ and $t > 2$. Then $m_n(2^n - 2) = t$. In particular, $m_9(510) = 3 < m_9(508) = 4$.*

Proof. The proof proceeds in two steps. In lemma 3 below, we first show that $m_n(2^n - 2) = t$. As a result, $m_9(510) = 3$. Then in Lemma 4, we prove that $m_9(508) = 4$. This will complete the proof of the proposition. □

Lemma 3. *$m_n(2^n - 2) = t$, where $n = 2^t + 1$, $t \in \mathbb{N}$ and $t > 2$.*

Proof. From Proposition 1, we have $m_n(2^n - 2) \geq \log_2(\nu(2^n - 2)) = t$. A CC-addition chain of length t for $2^n - 2$ (w.r.t. n) can be constructed as follows

$$C_1,\, C_{2^2-1},\, C_{2^4-1},\, C_{2^8-1},\, \dots\,,\, C_{2^{2^t}-1} = C_{2^n-2}. \tag{20}$$

Note that $C_{2^{2^t}-1} = C_{2^n-2}$ because $2^n - 2 = 2\left(2^{2^t} - 1\right)$. Why the above sequence is indeed a CC-addition chain can be readily seen if we look at the n-bit-representations of the representatives of the cyclotomic classes in the above sequence. In the proof of Proposition 1 and the example in (6), we have observed that all the elements of a given cyclotomic class can be obtained by (left) cyclic shifts of the n-bit-representation of any one element of the class. Consider an integer sequence

$$\begin{aligned}&\langle 1\rangle_2 \overset{\times}{\to} \langle 10\rangle_2 \overset{+}{\to} \langle 11\rangle_2 \overset{\times}{\to} \langle 1100\rangle_2 \overset{+}{\to} \langle 1111\rangle_2 \to \\ &\dots \to \langle\underbrace{11\dots11}_{2^t}\rangle_2 \overset{\times}{\to} \langle\underbrace{11\dots11}_{2^t}0\rangle_2.\end{aligned} \tag{21}$$

In the above sequence, those arrows marked with × correspond to multiplying by a power of 2 (i.e. left shift) and hence such a step is not a separate step in the corresponding CC-addition chain. But those marked with + correspond to addition of two distinct integers and hence count as one step in the CC-addition chain. This shows that the sequence in (20) is a CC-addition chain for $2^n - 2$ (w.r.t. n), and hence $m_n(2^n - 2) = t$. □

Lemma 4. *$m_9(508) = 4$.*

Proof. From Proposition 1, we have $m_9(508) \geq \lceil \log_2(7) \rceil = 3$. We now rule out the possibility that $m_9(508) = 3$. Let there be a CC-addition chain for 508 (w.r.t. 9) of length 3. The only possibility is that in such a chain, the Hamming weight doubles after each of the first two (addition) steps. But in the last step, we must have two integers $a = \left\langle a_8 \dots a_0 \right\rangle_2$ and $b = \left\langle b_8 \dots b_0 \right\rangle_2$ such that $508 = a + b$, $\nu(a) = \nu(b)$, and both must come from the same cyclotomic class. Hence the bit-patterns of a and b must be cyclic shifts of each other. We just need to make sure that the bit-pattern $508 = \langle 111111100\rangle_2$ cannot be obtained. There are four possible cases:

1. $a_0 = b_0 = 1$: then $a_1 = 1$ or $b_1 = 1$ (but not both). Hence with remaining 5 ones, it is not possible to obtain ones at the remaining 7 positions in the sum.
2. $a_0 = b_0 = 0$ and $a_1 = b_1 = 0$: now there are 8 ones for 7 positions. Hence a zero will appear in the sum when there is a one in the same position.

3. $a_0 = b_0 = 0$, $a_1 = b_1 = 1$ and $a_2 = b_2 = 1$: in this case it is not possible to get ones in 6 positions in the sum with only 4 ones.
4. $a_0 = b_0 = 0$, $a_1 = b_1 = 1$ and $a_2 = b_2 = 0$: by symmetry, we can set $a_3 = 1$ and $b_3 = 0$. Now there are 2 ones for a that can occur in any of the five remaining positions. Hence there are $\binom{5}{2} = 10$ choices. Once the two positions are fixed for a, then for b, the remaining three ones must be in the other three remaining positions of the sum. One can easily check in all the 10 cases that a and b are not cyclic shifts of each other.

Hence we obtain $m_9(508) > 3$. The CC-addition chain

$$\langle 1\rangle_2 \overset{\times}{\rightarrow} \langle 10\rangle_2 \overset{+}{\rightarrow} \langle 11\rangle_2 \overset{\times}{\rightarrow} \langle 1100\rangle_2 \overset{+}{\rightarrow} \langle 1111\rangle_2 \overset{\times}{\rightarrow} \langle 111100\rangle_2$$
$$\overset{+}{\rightarrow} \langle 111111\rangle_2 \overset{\times}{\rightarrow} \langle 1111110\rangle_2 \overset{+}{\rightarrow} \langle 1111111\rangle_2 \overset{\times}{\rightarrow} \langle 111111100\rangle_2.$$

shows that $m_9(508) \leq 4$. Hence $m_9(508) = 4$ □

B Proof of Theorem 1

Theorem *Let $\alpha, n, n' \in \mathbb{N}$, $n \,|\, n'$ and $\lceil \log_2(\alpha + 2)\rceil \leq n \leq n'$. Then $m_n(\alpha) \leq m_{n'}(\alpha)$.*

Proof. The basic idea is to transform *any* CC-addition chain for α w.r.t. n' into a CC-addition chain for α w.r.t. n such that the length of the resulting chain is at most the length of the original one. This implies that $m_n(\alpha) \leq m_{n'}(\alpha)$. Let

$$C'_{b_0} = C'_1,\ C'_{b_1},\ C'_{b_{2,}}\ \cdots,\ C'_{b_r} = C'_\alpha \tag{22}$$

be a CC-addition chain for α w.r.t. n'. Let $a_i := b_i \pmod{2^n - 1}$ and C_{a_i} be the cyclotomic class of a_i w.r.t. n, $\forall i = 0, 1, \ldots, r$. Consider the sequence

$$C_{a_0},\ C_{a_1},\ C_{a_{2,}}\ \cdots,\ C_{a_r}. \tag{23}$$

The claim is that the above sequence in (23) is a CC-addition chain for α w.r.t. n. In particular, we need to prove that the sequence in (23) satisfies two properties. First is the CC-addition chain property (w.r.t. n), i.e. Definition 3, and the second one is $C_{a_r} = C_\alpha$ (w.r.t. n).

Claim. The sequence in (23) satisfies Definition 3 w.r.t. n.

Proof. First we need to show that the mapping $C'_{b_j} \mapsto C_{a_j}$ is well-defined. This is because the cyclotomic class C'_{b_j} may be represented as $C'_{\beta_{j''}}$, where $\beta_{j''} \in C'_{b_j}$. From Definition 3, $\beta_{j''} \in C'_{b_j}$ iff $\beta_{j''} = b_j \cdot 2^{j''} \left(\bmod 2^{n'} - 1\right)$ for some $j'' \in \mathbb{N}$. Since $b_j \equiv a_j \pmod{2^n - 1}$, we have $\beta_{j''} \equiv b_j \cdot 2^{j''} \equiv a_j \cdot 2^k \pmod{2^n - 1}$, where $k := j'' \pmod{n}$. This proves the well-definedness property of the mapping of cyclotomic classes. Next, to prove the additivity property, observe that for every $i = 1, 2, \ldots, r$, there exist $0 \leq j, k < i$, $\beta_{i'} \in C'_{b_i}$, $\beta_{j'} \in C'_{b_j}$, and $\beta_{k'} \in C'_{b_k}$

such that $\beta_{i'} \equiv \beta_{j'} + \beta_{k'} \left(\bmod 2^{n'} - 1\right)$. This is because the sequence in (22) is a CC-addition chain (w.r.t. n'). From the reasoning above, we can write $\beta_{i'} \equiv a_i \cdot 2^{i'} \pmod{2^n - 1}$, $\beta_{j'} \equiv a_j \cdot 2^{j'} \pmod{2^n - 1}$ and $\beta_{k'} \equiv a_k \cdot 2^{k'} \pmod{2^n - 1}$. Since $n \mid n'$, we have $2^n - 1 \mid 2^{n'} - 1$. Hence $\beta_{i'} \equiv \beta_{j'} + \beta_{k'} \pmod{2^n - 1}$.Therefore, $a_i \cdot 2^{i'} \equiv a_j \cdot 2^{j'} + a_k \cdot 2^{k'} \pmod{2^n - 1}$. This proves the additivity property of the sequence in (23). □

Claim. $C_{a_r} = C_\alpha$.

Proof. Since $C'_{b_r} = C'_\alpha$ (w.r.t. n') from (22), we have $\alpha \equiv b_r 2^t \left(\bmod 2^{n'} - 1\right)$ for some $t \in \mathbb{N}$ and $t < n'$. Since $2^n - 1 \mid 2^{n'} - 1$, we have $\alpha \equiv b_r 2^t \equiv a_r 2^{t'} \pmod{2^n - 1}$. Therefore, $C_{a_r} = C_\alpha$. □

This completes the proof of Theorem 1. □

C Proof of Theorem 2

Theorem *$\mathcal{M}(P_1(x)) = \mathcal{M}(P_2(x))$, where $P_1(x)$ and $P_2(x)$ are as defined in Section 3.2. In other words, the masking complexity of an S-box (in general, any function on bit strings) is invariant w.r.t. field representations.*

Proof. Let the maps $\mathcal{B}_1, \mathcal{B}_2, \mathcal{U}, \mathcal{U}_1$ and $\mathcal{U}_2$ be as defined in (9) and (10). Since two finite fields of the same order are isomorphic, there exists a field isomorphism $\psi : \mathbb{F}_2[y]/f_1(y) \to \mathbb{F}_2[z]/f_2(z)$. Note that the map ψ is also an $\mathbb{F}_2$-linear isomorphism between vector spaces that is compatible with the multiplication operation of the fields. Let $\mathcal{H} : \mathbb{F}_2[y]/f_1(y) \to \mathbb{F}_2[z]/f_2(z)$ be defined as

$$\mathcal{H} = \mathcal{B}_2 \circ \mathcal{B}_1^{-1}. \tag{24}$$

Since $\mathcal{B}_1$ and $\mathcal{B}_2$ are $\mathbb{F}_2$-linear bijections, so will be $\mathcal{H}$. Note that $\mathcal{H}$ need not be a field isomorphism. Also define the maps $\mathcal{H}^*$,$\mathcal{U}_1^* : \mathbb{F}_2[z]/f_2(z) \to \mathbb{F}_2[z]/f_2(z)$ as

$$\mathcal{H}^* = \mathcal{H} \circ \psi^{-1}, \tag{25}$$

$$\mathcal{U}_1^* = \psi \circ \mathcal{U}_1 \circ \psi^{-1}. \tag{26}$$

Intuitively, the maps $\mathcal{H}^*$ and $\mathcal{U}_1^*$ are analogues of $\mathcal{H}^*$ and $\mathcal{U}_1^*$ that are maps from $\mathbb{F}_2[z]/f_2(z)$ to itself. From (9), (10), (24) and (26), we have

$$\mathcal{U}_1 = \psi^{-1} \circ \mathcal{U}_1^* \circ \psi = \mathcal{H}^{-1} \circ \mathcal{U}_2 \circ \mathcal{H}.$$

Hence from (26), we get

$$\mathcal{U}_2 = \mathcal{H}^* \circ \mathcal{U}_1^* \circ {\mathcal{H}^*}^{-1}. \tag{27}$$

Let $P_{\mathcal{H}^*}(x)$, $P_{{\mathcal{H}^*}^{-1}}(x)$ and $P_{\mathcal{U}_1^*}(x)$ be polynomials over $\mathbb{F}_2[z]/f_2(z)$ of degree at most $2^n - 1$ representing $\mathcal{H}^*$, ${\mathcal{H}^*}^{-1}$ and $\mathcal{U}_1^*$, respectively. From the above relation, we obtain

$$P_2(x) = P_{\mathcal{H}^*}\left(P_{\mathcal{U}_1^*}\left(P_{{\mathcal{H}^*}^{-1}}(x)\right)\right). \tag{28}$$

It is precisely to get the above relation that we had to introduce the maps $\mathcal{H}^*$ and $\mathcal{U}_1^*$. The following two lemmas show that $\mathcal{M}\left(P_{\mathcal{U}_1^*}(x)\right) = \mathcal{M}\left(P_1(x)\right)$ and $\mathcal{M}\left(P_{\mathcal{H}^{*-1}}\right) = \mathcal{M}\left(P_{\mathcal{H}^*}\right) = 0$.

Lemma 5. $\mathcal{M}\left(P_{\mathcal{U}_1^*}(x)\right) = \mathcal{M}\left(P_1(x)\right)$.

Proof. Let $P_1(x) = \sum_{i=0}^{2^n-1} a_i\, x^i$, where $a_i \in \mathbb{F}_2[z]/f_2(z)$. From the definition of $\mathcal{U}_1^*$ in (26), it follows that $P_{\mathcal{U}_1^*}(x) = \sum_{i=0}^{2^n-1} \psi(a_i)\, x^i$. Using the field isomorphisms ψ and ψ^{-1}, any polynomial chain to evaluate $P_1(x)$ can be converted to one that evaluates $P_{\mathcal{U}_1^*}(x)$, and vice-versa. Hence the lemma follows. □

Lemma 6. *Let $\mathcal{A} : \mathbb{F}_2[z]/f_2(z) \rightarrow \mathbb{F}_2[z]/f_2(z)$ be an affine function and $P_{\mathcal{A}}(x)$ be the corresponding polynomial representation of degree at most $2^n - 1$. Then $P_{\mathcal{A}}(x) = \sum_{i=0}^{n-1} a_i\, x^{2^i}$, for some $a_i \in \mathbb{F}_2[z]/f_2(z)$ $(0 \leq i \leq n-1)$, and $\mathcal{M}\left(P_{\mathcal{A}}(x)\right) = 0$.*

Proof. Since $\mathcal{A}$ is an affine function, it can be written as $\mathcal{A} = \mathcal{A}' + a_0$, where $\mathcal{A} : \mathbb{F}_2[z]/f_2(z) \rightarrow \mathbb{F}_2[z]/f_2(z)$ is a $\mathbb{F}_2$-linear map, and $a_0 \in \mathbb{F}_2[z]/f_2(z)$. It is enough to show that the polynomial $P_{\mathcal{A}'}(x)$ corresponding to $\mathcal{A}'$ is of the form $\sum_{i=1}^{n-1} a_i\, x^{2^i}$. Suppose that there exists a term x^m in $P_{\mathcal{A}'}(x)$ whose coefficient is non-zero, and $m \neq 2^j$ for $0 \leq j \leq n-1$. Let x^m be the largest among such terms and write $m = 2^t \cdot k$, where k is odd. Define the polynomial

$$P'(x) = P_{\mathcal{A}'}(x+1) - P_{\mathcal{A}'}(x) - P_{\mathcal{A}'}(1).$$

We have $P'(x) \not\equiv 0$ since the coefficient of the term $x^{2^t(k-1)}$ will not be zero. This is because in the binomial expansion of $(x+1)^m$, the coefficient of $z^{2^t(k-1)}$ will be 1 (note that the characteristic of $\mathbb{F}_{2^n}$ is two), and because of the fact that x^m is the largest among the terms that are not of the form x^{2^i}, we have that $x^{2^t(k-1)}$ will not be cancelled by any other term. By the linearity of $\mathcal{A}'$, we require $P'(\alpha) = 0$ for all $\alpha \in \mathbb{F}_2[z]/f_2(z)$. But this is not possible since $deg(P') < 2^n$ and a polynomial of degree d can have at most d roots over $\mathbb{F}_{2^n}$. Hence $P'(x) \equiv 0$ and $P_{\mathcal{A}'}(x) = \sum_{i=1}^{n-1} a_i\, x^{2^i}$ and $\mathcal{M}\left(P_{\mathcal{A}'}(x)\right) = \mathcal{M}\left(P_{\mathcal{A}}(x)\right) = 0$. By the linearity of $P_{\mathcal{A}'}(x)$, we have $P_{\mathcal{A}'}(0) = 0$. Hence the lemma follows. □

From (28), Lemma 5 and Lemma 6, we have $\mathcal{M}\left(P_2(x)\right) \leq \mathcal{M}\left(P_1(x)\right)$. From (27), we get $\mathcal{U}_1^* = {\mathcal{H}^*}^{-1} \circ \mathcal{U}_2 \circ \mathcal{H}^*$. Hence $\mathcal{M}\left(P_1(x)\right) \leq \mathcal{M}\left(P_2(x)\right)$. Therefore, $\mathcal{M}\left(P_1(x)\right) = \mathcal{M}\left(P_2(x)\right)$. This completes the proof of Theorem 2. □

D Divide-and-Conquer Strategy for Polynomial Evaluation

Let $P(x)$ be a polynomial having degree $N = k(2t-1)$. We divide $P(x)$ by x^{kt} and express $P(x)$ as following

$$P(x) = Q(x) \cdot x^{kt} + R(x) \tag{29}$$

where Q is monic and $\deg(Q) = k(t-1)$, $\deg(R) \leq kt - 1$. Now we divide $R(x) - x^{k(t-1)}$ by $Q(x)$ and obtain $C(x)$, $R_1(x)$ as following

$$R(x) - x^{k(t-1)} = C(x) \cdot Q(x) + R_1(x) \tag{30}$$

where $\deg(C) \leq k - 1$, $\deg(R_1) \leq k(t-1) - 1$. So $P(x)$ can be written as

$$P(x) = (x^{kt} + c(x)) \cdot Q(x) + x^{k(t-1)} + R_1(x) \tag{31}$$

Note that $(x^k)^t + c(x))$ is already a function of polynomials having degree at most k. Assume that $t = 2^{i-1}$, then having computed $x^2, x^3, ..., x^k$ we can compute x^{kt} for "free"(without non-linear multiplications).

Next we apply the same technique to $Q(x)$ and $x^{k(t-1)} + R_1(x)$ (both having degree $k(t-1)$) recursively. In general, if $i \leq m$ then the number of non-linear multiplications can be calculated from the relation

$$\mathcal{T}(k(2^i - 1)) = 2\mathcal{T}(k(2^{i-1} - 1)) + 1 \tag{32}$$

where $\mathcal{T}(\gamma)$ is the number of non-linear multiplications required to evaluate a polynomial having degree γ, using the above technique. This gives $\mathcal{T}(k(2^m - 1)) = 2^{m-1} - 1 \approx N/2k$. Hence the total number of non-linear multiplications is about $\frac{1}{2}(k + N/k)$.