Linear Approximations of Addition Modulo 2^n

Johan Wallén

Laboratory for Theoretical Computer Science
Helsinki University of Technology
P.O.Box 5400, FIN-02015 HUT, Espoo, Finland
johan@tcs.hut.fi

Abstract. We present an in-depth algorithmic study of the linear approximations of addition modulo 2^n. Our results are based on a fairly simple classification of the linear approximations of the carry function. Using this classification, we derive an $\Theta(\log n)$-time algorithm for computing the correlation of linear approximation of addition modulo 2^n, an optimal algorithm for generating all linear approximations with a given non-zero correlation coefficient, and determine the distribution of the correlation coefficients. In the generation algorithms, one or two of the selection vectors can optionally be fixed. The algorithms are practical and easy to implement.
Keywords. Linear approximations, correlation, modular addition, linear cryptanalysis.

1 Introduction

Linear cryptanalysis [8] is one of the most powerful general cryptanalytic methods for block ciphers proposed by date. Since its introduction, resistance against this attack has been a standard design goal for block ciphers. Although some design methodologies to achieve this goal have been proposed—for example [12, 10, 4, 13]—many block ciphers are still designed in a rather ad hoc manner, or dictated by other primary design goals. For these ciphers, it it important to have efficient methods for evaluating their resistance against linear cryptanalysis.

At the heart of linear cryptanalysis lies the study of the correlation of linear approximate relations between the input and output of functions. Good linear approximations of ciphers are usually found heuristically by forming trails consisting of linear approximations of the components of the cipher. In order to search the space of linear trails, e.g. using a Branch-and-bound algorithm (see e.g. [5, 9, 1]), we need efficient methods for computing the correlation of linear approximations of the simplest components of the cipher, as well as methods for generating the relevant approximations of the components. Towards this goal, we study a few basic functions often used in block ciphers.

Currently, block ciphers are usually build from local nonlinear mappings, global linear mappings, and arithmetic operations. The mixture of linear mappings and arithmetic operations seems fruitful, since they are suitable for software implementation, and their mixture is difficult to analyse mathematically.

While the latter property intuitively should make standard cryptanalysis intractable , it also makes it difficult to say something concrete about the security of the cipher.

Perhaps the simplest arithmetic operations in wide use are addition and subtraction modulo 2^n. Interestingly, good tools for studying linear approximations of even these simple mappings have not appeared in the literature to date. In this paper, we consider algorithms for two important problems for linear approximations of these operations: for computing the correlation of any given linear approximation and for generating all approximations with a correlation coefficient of a given absolute value. Our results are based on a fairly simple classification of the linear approximations of the carry function. Using this classification, we derive $\Theta(\log n)$-time algorithms for computing the correlation of of linear approximations of addition and subtraction modulo 2^n in a standard RAM model of computation. The classification also gives optimal (that is, linear in the size of the output) algorithms for generating all linear approximations of addition or subtraction with a given non-zero correlation. In the generation algorithms, one or two of the selection vectors may optionally be fixed. As a simple corollary, we determine closed-form expressions for the distribution of the correlation coefficients. We hope that our result will facilitate advanced linear cryptanalysis of ciphers using modular arithmetic.

Similar results with respect to differential cryptanalysis [2] are discussed in [7, 6]. The simpler case with one addend fixed is considered in [11] with respect to both linear and differential cryptanalysis.

In the next section, we discuss linear approximations and some preliminary results. In Sect. 3, we derive our classification of linear approximations of the carry function, and the corresponding results for addition and subtraction. Using this classification, we then present the $\Theta(\log n)$-time algorithm for computing the correlation of linear approximations in Sect. 4, and the generation algorithms in Sect. 5.

2 Preliminaries

2.1 Linear Approximations

Linear cryptanalysis [8] views (a part of) the cipher as a relation between the plaintext, the ciphertext and the key, and tries to approximate this relation using linear relations. The following standard terminology is convenient for discussing these linear approximations.

Let $f, g\colon \mathbb{F}_2^n \to \mathbb{F}_2$ be Boolean functions. The *correlation* between f and g is defined by

$$c(f,g) = 2^{1-n}\big|\{x \in \mathbb{F}_2^n \mid f(x) = g(x)\}\big| - 1 \ .$$

This is simply the probability taken over x that $f(x) = g(x)$ scaled to a value in $[-1,1]$. Let $u = (u_{m-1}, \dots, u_0)^t \in \mathbb{F}_2^m$ and $w = (w_{n-1}, \dots, w_0)^t \in \mathbb{F}_2^n$ be binary column vectors, and let $h\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$. Let $w \cdot x = w_{n-1}x_{n-1} + \cdots + w_1x_1 + w_0x_0$ denote the standard dot product. Define the linear function $l_w\colon \mathbb{F}_2^n \to \mathbb{F}_2$ by

$l_w(x) = w \cdot x$ for all $w \in \mathbb{F}_2^n$. A *linear approximation* of h is an approximate relation of the form $u \cdot h(x) = w \cdot x$. Such a linear approximation will be denoted by the formal expression $u \xleftarrow{h} w$, or simply $u \leftarrow w$ when h is clear from context. Its efficiency is measured by its *correlation* $\mathcal{C}(u \xleftarrow{h} w)$ defined by

$$\mathcal{C}(u \xleftarrow{h} w) = c(l_u \circ h, l_w) \ .$$

Here, u and w are the *output* and *input selection vectors*, respectively.

2.2 Fourier Analysis

There is a well-known Fourier-based framework for studying linear approximations [3]. Let $f \colon \mathbb{F}_2^n \to \mathbb{F}_2$ be a Boolean function. The corresponding real-valued function $\hat{f} \colon \mathbb{F}_2^n \to \mathbb{R}$ is defined by $\hat{f}(x) = (-1)^{f(x)}$. With this notation, $c(f,g) = 2^{-n} \sum_{x \in \mathbb{F}_2^n} \hat{f}(x)\hat{g}(x)$. Note also that $f + g \leftrightarrow \hat{f}\hat{g}$. Recall that an algebra $\mathcal{A}$ over a field $\mathbb{F}$ is a ring, such that $\mathcal{A}$ is a vector space over $\mathbb{F}$, and $a(xy) = (ax)y = x(ay)$ for all $a \in \mathbb{F}$ and $x, y \in \mathcal{A}$.

Definition 1. *Let $\mathcal{B}_n = \langle \hat{f} \mid f \colon \mathbb{F}_2^n \to \mathbb{F}_2 \rangle$ be the real algebra generated by the n-variable Boolean functions. As usual, the addition, multiplication, and multiplication by scalars are given by $(\xi + \eta)(x) = \xi(x) + \eta(x)$, $(\xi\eta)(x) = \xi(x)\eta(x)$ and $(a\xi)(x) = a(\xi(x))$ for all $\xi, \eta \in \mathcal{B}_n$ and $a \in \mathbb{R}$.*

The algebra $\mathcal{B}_n$ is of course unital and commutative.

The vector space $\mathcal{B}_n$ is turned into an inner product space by adopting the standard inner product for real-valued discrete functions. This inner product is defined by

$$\langle \xi, \eta \rangle = 2^{-n} \sum_{x \in \mathbb{F}_2^n} (\xi\eta)(x) \ , \ \forall \xi, \eta \in \mathcal{B}_n \ .$$

For Boolean functions, $f, g \colon \mathbb{F}_2^n \to \mathbb{F}_2$, $\langle \hat{f}, \hat{g} \rangle = c(f,g)$. Since the set of linear functions $\{\hat{l}_w \mid w \in \mathbb{F}_2^n\}$ forms an orthonormal basis for $\mathcal{B}_n$, every $\xi \in \mathcal{B}_n$ has a unique representation as

$$\xi = \sum_{w \in \mathbb{F}_2^n} \alpha_w \hat{l}_w \ , \text{ where } \alpha_w = \langle \xi, \hat{l}_w \rangle \in \mathbb{R} \ .$$

The corresponding Fourier transform $\mathcal{F} \colon \mathcal{B}_n \to \mathcal{B}_n$ is given by

$$\mathcal{F}(\xi) = \Xi \ , \text{ where } \Xi \text{ is the mapping } w \mapsto \langle \xi, \hat{l}_w \rangle \ .$$

This is usually called the *Walsh-Hadamard transform* of ξ. For a Boolean function $f \colon \mathbb{F}_2^n \to \mathbb{F}_2$, the Fourier transform $\hat{F} = \mathcal{F}(\hat{f})$ simply gives the correlation between f and the linear functions: $\hat{F}(w) = c(f, l_w)$.

For $\xi, \eta \in \mathcal{B}_n$, their *convolution* $\xi \otimes \eta \in \mathcal{B}_n$ is given by

$$(\xi \otimes \eta)(x) = \sum_{t \in \mathbb{F}_2^n} \xi(x+t)\eta(t) \ .$$

Clearly, $\mathcal{B}_n$ is a commutative, unital real algebra also under convolution as multiplication. The unity is the function δ such that $\delta(0) = 1$ and $\delta(x) = 0$ for $x \neq 0$. As usual, the Fourier transform is an algebra isomorphism between the commutative, unital real algebras $\langle \mathcal{B}_n, +, \cdot \rangle$ and $\langle \mathcal{B}_n, +, \otimes \rangle$.

Let $f \colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a Boolean function. Since the correlation of a linear approximation of f is given by $\mathcal{C}(u \xleftarrow{f} w) = \mathcal{F}(\widehat{l_u f})(w)$, the correlation of linear approximations can conveniently be studied using the Fourier transform. Since $l_u f$ can be expressed as $\sum_{i:u_i=1} f_i$, where f_i denotes the ith component of f, we have the convolutional representation

$$\mathcal{C}(u \xleftarrow{f} w) = \bigotimes_{i:u_i=1} \hat{F}_i \enspace ,$$

where $\hat{F}_i = \mathcal{F}(\hat{f}_i)$. Especially when using the convolutional representation, it will be convenient to consider $\mathcal{C}(u \xleftarrow{f} w)$ as a function of w with u fixed.

3 Linear Approximations of Addition Modulo 2^n

3.1 k-Independent Recurrences

We will take a slightly abstract approach to deriving algorithms for studying linear approximations of addition modulo 2^n, since this approach might turn out to be useful also for some related mappings. The key to the algorithms are a certain class of *k-independent recurrences.* The name comes from the fact that they will be used to express the correlation of linear approximations of functions whose ith output bit is independent of the $(i+k)$th input bit an higher.

We let $e_i \in \mathbb{F}_2^n$ denote a vector whose ith component is 1 and the other 0. If $x \in \mathbb{F}_2^n$, $\overline{x}$ denotes the component-wise complement of x: $\overline{x}_i = x_i + 1$. Let $\mathrm{eq} \colon \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2^n$ be defined by $\mathrm{eq}(x,y)_i = 1$ if and only if $x_i = y_i$. That is, $\mathrm{eq}(x,y) = \overline{x+y}$. For $x, y \in \mathbb{F}_2^n$, we let $xy = (x_{n-1}y_{n-1}, \dots, x_1y_1, x_0y_0)^t$ denote their component-wise product.

Definition 2. *A function $f \colon \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{R}$ is* k-independent*, if $f(x,y) = 0$ whenever $x_j \neq 0$ or $y_j \neq 0$ for some $j \geq k$. Let $r_0, r \colon \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{R}$ be k-independent functions. A recurrence $R_i = R_i^{r_0,r}$ is* k-independent*, if it has the form*

$$R_0(x,y) = r_0(x,y) \enspace , \textit{ and}$$
$$R_{i+1}(x,y) = \frac{1}{2}\Big(r(x^{i+k}, y) + r(x, y^{i+k}) + R_i(x,y) - R_i(x^{i+k}, y^{i+k}\Big)$$

for $i > 0$, where we for compactness have denoted $z^{i+k} = z + e_{i+k}$. Note that R_j is a $k+j$-independent function for all j.

Note that k-independent recurrences can be efficiently computed, provided that we efficiently can compute the base cases r and r_0. The crucial observation is that at most one of the terms in the expression for R_{i+1} is non-zero, and that we can determine which of the four terms might be non-zero by looking only at x_{i+k} and y_{i+k}. The four terms consider the cases $(x_{i+k}, y_{i+k}) = (1,0)$, $(0,1)$, $(0,0)$, and $(1,1)$, respectively. This observation yields the following lemma.

Lemma 1. *Let $R_i = R_i^{r_0,r}$ be a k-independent recurrence. Then*

$$R_0(x,y) = r_0(x,y) \ , \ \textit{and}$$
$$R_{i+1}(x,y) = \begin{cases} \frac{1}{2} r(x\overline{e_{i+k}}, y\overline{e_{i+k}}) \ , & \textit{if } x_{i+k} \neq y_{i+k} \textit{ and} \\ \frac{1}{2}(-1)^{x_{i+k}} R_i(x\overline{e_{i+k}}, y\overline{e_{i+k}}) \ , & \textit{if } x_{i+k} = y_{i+k} \ . \end{cases}$$

It turns out that the k-independent recurrences of interest can be solved by finding a certain type of common prefix of the arguments. Towards this end, we define the common prefix mask of a vector.

Definition 3. *The* common prefix mask $\mathrm{cpm}_i^k \colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ *is for all j defined by* $\mathrm{cpm}_i^k(x)_j = 1$ *if and only if $k \le j < k+i$ and $x_\ell = 1$ for all $j < \ell < k+i$.*

Let $w_{\mathrm{H}}(x) = \big|\{i \mid x_i \neq 0\}\big|$ denote the Hamming weight of $x \in \mathbb{F}_2^n$.

Lemma 2. *Let $R_i = R_i^{r_0,r}$ be a k-independent recurrence. Denote $r_1 = r$, and let $z = \mathrm{cpm}_i^k(\mathrm{eq}(x,y))$, $\ell = w_{\mathrm{H}}(z)$ and $s = (-1)^{w_{\mathrm{H}}(zxy)}$. Let $b = 0$, if $xz = yz$ and let $b = 1$ otherwise. Then*

$$R_i(x,y) = s \cdot 2^\ell r_b(x\overline{z}, y\overline{z}) \ .$$

Proof. For $i = 0$, $\mathrm{cpm}_0^k(\mathrm{eq}(x,y)) = 0$, $\ell = 0$, $s = 1$, and $b = 0$. Thus, the lemma holds for $i = 0$, so consider $i+1$. Let $x' = x\overline{e_{i+k}}$, $y' = y\overline{e_{i+k}}$, $z' = \mathrm{cpm}_i^k(\mathrm{eq}(x',y'))$, $\ell' = w_{\mathrm{H}}(z')$, $s' = (-1)^{w_{\mathrm{H}}(z'x'y')}$, and $b' = 0$, if $x'z' = y'z'$ and $b' = 1$ otherwise. By Lemma 1, there are two cases to consider. If $x_{i+k} \neq y_{i+k}$, $z = e_{i+k}$, $\ell = 1$, $s = 1$, and $b = 1$. In this case $s \cdot 2^\ell r_b(x\overline{z}, y\overline{z}) = \frac{1}{2} r(x\overline{e_{i+k}}, y\overline{e_{i+k}}) = R_{i+1}(x,y)$. If $x_{i+k} = y_{i+k}$, $z = e_{i+k} + z'$, $\ell = \ell' + 1$, $s = s'(-1)^{x_{i+k}}$, and $b = b'$. In this case, $s \cdot 2^\ell r_b(x\overline{z}, y\overline{z}) = \frac{1}{2}(-1)^{x_{i+k}} \cdot s' 2^{\ell'} r_{b'}(x'\overline{z'}, y'\overline{z'}) = \frac{1}{2}(-1)^{x_{i+k}} R_i(x', y') = R_{i+1}(x,y)$. □

We will next consider the convolution of k-independent recurrences.

Lemma 3. *Let $R_i = R_i^{\delta,\delta}$ be a 0-independent recurrence, and let $f \colon \mathbb{F}_2^n \to \mathbb{R}$ be k-independent. Define $S_i = R_{i+k} \otimes f$, $s = f$, and $s_0 = R_k \otimes f$. Then $S_i = S_i^{s_0,s}$ is a k-independent recurrence.*

Proof. Clearly, s_0 and s are k-independent. Furthermore, $S_0 = R_k \otimes f = s_0$ by definition. Finally, $2S_{i+1}(x,y) = 2R_{(i+k)+1}(x,y) \otimes f(x,y) = (\delta(x^{i+k}, y) + \delta(x, y^{i+k}) + R_{i+k}(x,y) - R_{i+k}(x^{i+k}, y^{i+k})) \otimes f(x,y) = f(x^{i+k}, y) + f(x, y^{i+k}) + (R_{i+k} \otimes f)(x,y) - (R_{i+k} \otimes f)(x^{i+k}, y^{i+k})$, where we have used the notation $z^{i+k} = z + e_{i+k}$. □

3.2 Linear Approximations of the Carry Function

In this subsection, we derive a classification of the linear approximations of the carry function modulo 2^n. It will turn out that the correlation of arbitrary linear approximations of the carry function can be expressed as a recurrence of the type studied in the previous subsection. We will identify the vectors in $\mathbb{F}_2^n$ and the elements in $\mathbb{Z}_{2^n}$ using the natural correspondence

$$(x_{n-1}, \ldots, x_1, x_0)^t \in \mathbb{F}_2^n \leftrightarrow x_{n-1}2^{n-1} + \cdots + x_1 2^1 + x_0 2^0 \in \mathbb{Z}_{2^n} \ .$$

To avoid confusion, we sometimes use $\oplus$ and $\boxplus$ to denote addition in $\mathbb{F}_2^n$ and $\mathbb{Z}_{2^n}$, respectively.

Definition 4. *Let* carry: $\mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2^n$ *be the* carry function *for addition modulo* 2^n *defined by* $\text{carry}(x, y) = x \oplus y \oplus (x \boxplus y)$*, and let* $c_i = \text{carry}_i$ *denote the* i*th component of the carry function for* $i = 0, \ldots, n-1$.

Note that the ith component of the carry function can be recursively computed as $c_0(x, y) = 0$, and $c_{i+1}(x, y) = 1$ if and only if at least two of x_i, y_i and $c_i(x, y)$ are 1. By considering the 8 possible values of x_i, y_i and $c_i(x, y)$, we see that $\hat{c}_0(x, y) = 1$ and $\hat{c}_{i+1}(x, y) = \frac{1}{2}\big((-1)^{x_i} + (-1)^{y_i} + \hat{c}_i(x, y) - (-1)^{x_i + y_i}\hat{c}_i(x, y)\big)$. Thus we have

Lemma 4. *The Fourier transform of the carry function* $\hat{c}_i$ *is given by the recurrence*

$$\begin{aligned} \hat{C}_0(v, w) &= \delta(v, w) \ , \ \textit{and} \\ \hat{C}_{i+1}(v, w) &= \frac{1}{2}\Big(\delta(v + e_i, w) + \delta(v, w + e_i) + \hat{C}_i(v, w) - \hat{C}_i(v + e_i, w + e_i)\Big) \ , \end{aligned}$$

for $i = 0, \ldots, n-1$.

Note that this indeed is a 0-independent recurrence.

In the sequel, we will need a convenient notation for stripping off ones from the high end of vectors.

Definition 5. *Let* $x \in \mathbb{F}_2^n$ *and* $\ell \in \{0, \ldots, n\}$*. Define* $\text{strip}(x)$ *to be the vector in* $\mathbb{F}_2^n$ *that results when the highest component that is* 1 *in* x *(if any) is set to* 0*. By convention,* $\text{strip}(0) = 0$*. Similarly, let* $\text{strip}(\ell, x)$ *denote the vector that results when all but the* ℓ *lowest ones in* x *have been set to zero. For example,* $\text{strip}(2, 1011101) = 0000101$.

Let $u \in \mathbb{F}_2^n$ and let $\{i \mid u_i = 1\} = \{k_1, \ldots, k_m\}$ with $k_\ell < k_{\ell+1}$. Define $j_0 = 0$ and $j_{\ell+1} = k_{\ell+1} - k_\ell$ for $\ell = 0, \ldots, m-1$. Then

$$\mathcal{C}(u \xleftarrow{\text{carry}} v, w) = \bigotimes_{i : u_i = 1} \hat{C}_i(v, w) = \bigotimes_{i=1}^{m} \hat{C}_{k_i}(v, w) \ .$$

Define a sequence of recurrences $S_{0,i}, \dots, S_{m,i}$ by

$$S_{0,i} = \delta \ , \text{ and}$$
$$S_{\ell+1,i} = \hat{C}_{i+k_\ell} \otimes S_{\ell,j_\ell} \ ,$$

for $\ell = 0, \dots, m-1$. The crucial observation is that

$$S_{\ell,j_\ell}(v,w) = \mathcal{C}(\text{strip}(\ell,u) \xleftarrow{\text{carry}} v, w)$$

for all ℓ. Thus, $\mathcal{C}(u \xleftarrow{\text{carry}} v, w) = S_{m,j_m}(v,w)$.

Lemma 5. *Let $S_{\ell,i}$, j_ℓ, and k_ℓ be as above. Define s_ℓ, s'_ℓ by $s_1 = s'_1 = \delta$, and for $\ell > 0$ by $s_{\ell+1} = S_{\ell,j_\ell}$ and $s'_{\ell+1} = s_\ell$. Then $S_{\ell,i} = S^{s'_\ell,s_\ell}_{\ell,i}$ is a $k_{\ell-1}$-independent recurrence for all $\ell > 0$, where $k_0 = 0$.*

Proof. For $\ell = 1$, the result is clear. If $S_{\ell,i} = S^{s'_\ell,s_\ell}_{\ell,i}$ is a $k_{\ell-1}$-independent recurrence for some $\ell \geq 1$, then S_{ℓ,j_ℓ} is a $j_\ell + k_{\ell-1} = k_\ell$-independent function. By Lemma 3, $S_{\ell+1,i} = S^{f_0,f}_{\ell+1,i}$ is a k_ℓ-independent recurrence with $f = S_{\ell,j_\ell} = s_{\ell+1}$ and $f_0 = \hat{C}_{k_\ell} \otimes S_{\ell,j_\ell} = \hat{C}_{k_\ell} \otimes (\hat{C}_{j_\ell + k_{\ell-1}} \otimes S_{\ell-1,j_{\ell-1}}) = \hat{C}_{k_\ell} \otimes \hat{C}_{k_\ell} \otimes S_{\ell-1,j_{\ell-1}} = S_{\ell-1,j_{\ell-1}} = s'_{\ell+1}$. □

For any function f, we let f^0 denote the identity function and $f^{i+1} = f \circ f^i$. Lemmas 2 and 5 now give

Lemma 6. *The correlation of any linear approximation of the carry function is given recursively as follows. First, $\mathcal{C}(0 \xleftarrow{\text{carry}} v, w) = \delta(v,w)$. Second, if $u \neq 0$, let $j \in \{0, \dots, n-1\}$ be maximal such that $u_j = 1$. If $\text{strip}(u) \neq 0$, let k be maximal such that $\text{strip}(u)_k = 1$. Otherwise, let $k = 0$. Denote $i = j - k$. Let $z = \text{cpm}^k_i(\text{eq}(v,w))$, $\ell = w_{\mathrm{H}}(z)$, and $s = (-1)^{w_{\mathrm{H}}(zvw)}$. If $vz = wz$, set $b = 2$. Set $b = 1$ otherwise. Then*

$$\mathcal{C}(u \xleftarrow{\text{carry}} v, w) = s \cdot 2^{-\ell} \mathcal{C}(\text{strip}^b(u) \xleftarrow{\text{carry}} v\overline{z}, w\overline{z}) \ .$$

Our next goal is to extract all the common prefix masks computed in the previous lemma, and combine them into a single common prefix mask depending on u. This gives a more convenient formulation of the previous lemma.

Definition 6. *The* common prefix mask $\text{cpm} \colon \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2^n$ *is defined recursively as follows. First, $\text{cpm}(0,y) = 0$. Second, if $x \neq 0$, let j be maximal such that $x_j = 1$. If $\text{strip}(x) \neq 0$, let k be maximal such that $\text{strip}(x)_k = 1$. Otherwise, let $k = 0$. Denote $i = j - k$ and $z = \text{cpm}^k_i(y)$ If $zy = z$, set $b = 2$. Set $b = 1$ otherwise. Then*

$$\text{cpm}(x,y) = \text{cpm}^k_i(y) + \text{cpm}(\text{strip}^b(x), y) \ .$$

Theorem 1. *Let $u, v, w \in \mathbb{F}_2^n$, and let $z = \text{cpm}(u, \text{eq}(v,w))$. Then*

$$\mathcal{C}(u \xleftarrow{\text{carry}} v, w) = \begin{cases} 0 \ , & \text{if } v\overline{z} \neq 0 \text{ or } w\overline{z} \neq 0, \text{ and} \\ (-1)^{w_{\mathrm{H}}(vw)} \cdot 2^{-w_{\mathrm{H}}(z)} \ , & \text{otherwise.} \end{cases}$$

Since the only nonlinear part of addition modulo 2^n is the carry function, it should be no surprise that the linear properties of addition completely reduce to those of the carry function. Subtraction is also straightforward. When we are approximating the relation $x \boxminus y = z$ by $u \xleftarrow{\boxminus} v, w$, we are actually approximating the relation $z \boxplus y = x$ by $v \xleftarrow{\boxplus} u, w$. With this observation, it is trivial to prove

Lemma 7. *Let $u, v, w \in \mathbb{F}_2^n$. The correlations of linear approximations of addition and subtraction modulo 2^n are given by*

$$\mathcal{C}(u \xleftarrow{\boxplus} v, w) = \mathcal{C}(u \xleftarrow{\text{carry}} v + u, w + u) \ , \ \textit{and}$$
$$\mathcal{C}(u \xleftarrow{\boxminus} v, w) = \mathcal{C}(v \xleftarrow{\text{carry}} u + v, w + v) \ .$$

Moreover, the mappings $(u, v, w) \mapsto (u, v + u, w + u)$ and $(u, v, w) \mapsto (v, u + v, w + v)$ are permutations in $(\mathbb{F}_2^n)^3$.

4 The Common Prefix Mask

4.1 RAM Model

We will use a standard RAM model of computation consisting of n-bit memory cells, logical and arithmetic operations, and conditional branches. Specifically, we will use bitwise and ($\wedge$), or ($\vee$), exclusive or ($\oplus$) and negation ($\bar{\cdot}$), logical shifts ($\ll$ and $\gg$), and addition and subtraction modulo 2^n ($\boxplus$ and $\boxminus$). As a notational convenience, we will allow our algorithms to return values of the form $s2^{-k}$, where $s \in \{0, 1, -1\}$. In our RAM model, this can be handled by returning s and k in two registers.

4.2 Computing cpm

To make the domain of cpm clear, we write $\text{cpm}_n = \text{cpm} \colon \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2^n$. We will extend the definition of cpm to a 3-parameter version.

Definition 7. *Let $\text{cpm}_n \colon \{0,1\} \times \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2^n$ be defined by $\text{cpm}_n(b, x, y) = (z_{n-1}, \ldots, z_0)^t$, where $z = \text{cpm}_{n+1}((b, x)^t, (0, y)^t)$.*

Lemma 8 (Splitting lemma). *Let $n = k + \ell$ with $k, \ell > 0$. For any vector $x \in \mathbb{F}_2^n$, let $x^L \in \mathbb{F}_2^k$ and $x^R \in \mathbb{F}_2^\ell$ be such that $x = (x^L, x^R)^t$. Then*

$$\text{cpm}_n(x, y) = (\text{cpm}_k(x^L, y^L), \text{cpm}_\ell(b, x^R, y^R))^t \ ,$$

where $b = x_0^L$ if and only if $(y_0^L, \text{cpm}_k(x^L, y^L)_0) \neq (1, 1)$.

Proof. Let $w = w_{\text{H}}(x^L)$ and $z^L = \text{cpm}_k(x^L, y^L)$. If $w = 0$, the result is trivial. If $w = 1$ and $x_0^L = 1$, $b = 1$ and the result holds. If $w = 1$ and $x_0^L = 0$, $b = 1$ if and only if $z_0^L = 1$ and $y_0^L = 1$. If $w = 2$ and $x_0^L = 1$, $b = 0$ if and only if $z_0^L = 1$ and $y_0^L = 1$. Finally, if $w = 2$ and $x_0^L = 0$, or $w > 2$, the result follows by induction. □

Using this lemma, we can easily come up with an $\Theta(\log n)$-time algorithm for computing $\mathrm{cpm}_n(x, y)$. For simplicity, we assume that n is a power of two (if not, the arguments can be padded with zeros). The basic idea is to compute both $\mathrm{cpm}_n(0, x, y)$ and $\mathrm{cpm}_n(1, x, y)$ by splitting the arguments in halves, recursively compute the masks for the halves in parallel in a bit-sliced manner, and then combine the correct upper halves with the correct lower halves using the splitting lemma. Applying this idea bottom-up gives the following algorithm.

Theorem 2. *Let n be a power of 2, let $\alpha^{(i)} \in \mathbb{F}_2^n$ consist of blocks of 2^i ones and zeros starting from the lest significant end (e.g. $\alpha^{(1)} = 0011 \cdots 0011$), and let $x, y \in \mathbb{F}_2^n$. The following algorithm computes $\mathrm{cpm}(x, y)$ using $\Theta(\log n)$ time and constant space in addition to the $\Theta(\log n)$ space used for the constants $\alpha^{(i)}$.*

```
1. Initialise $\beta = 1010 \cdots 1010$, $z_0 = 0$, and $z_1 = \boxminus 1$.
2. For $i = 0, \ldots, \log_2 n - 1$, do
   (a) Let $\gamma_b = ((y \wedge z_b \wedge \overline{x}) \vee (\overline{y \wedge z_b} \wedge x)) \wedge \beta$ for $b \in \{0,1\}$.
   (b) Set $\gamma_b \leftarrow \gamma_b \boxminus (\gamma_b \gg 2^i)$ for $b \in \{0,1\}$.
   (c) Let $t_b = (z_b \wedge \overline{\alpha^{(i)}}) \vee (z_0 \wedge \overline{\gamma_b} \wedge \alpha^{(i)}) \vee (z_1 \wedge \gamma_b)$ for $b \in \{0,1\}$.
   (d) Set $z_b \leftarrow t_b$ for $b \in \{0,1\}$.
   (e) Set $\beta \leftarrow (\beta \gg 2^i) \wedge \overline{\alpha^{(i+1)}}$.
3. Return $z_0$.
```

Note that $\alpha^{(i)}$ and the values of β used in the algorithm only depend on n. For convenience, we introduce the following notation. Let $\beta^{(i)} \in \mathbb{F}_2^n$ be such that $\beta^{(i)}_\ell = 1$ iff $\ell - 2^i$ is a non-negative multiple of 2^{i+1} (e.g. $\beta^{(1)} = 0100 \cdots 01000100$). For $b \in \{0,1\}$, let

$$z^{(i)}(b, x, y) = (\mathrm{cpm}_{2^i}(b, x^{(n/2^i-1)}, y^{(n/2^i-1)}), \ldots, \mathrm{cpm}_{2^i}(b, x^{(0)}, y^{(0)}))^t \ ,$$

where $x = (x^{(n/2^i-1)}, \ldots, x^{(0)})^t$ and $y = (y^{(n/2^i-1)}, \ldots, y^{(0)})^t$. We also let $x \to y, z$ denote the function "if x then y else z". That is, $x \to y, z = (x \wedge y) \vee (\overline{x} \wedge z)$.

Proof (of Theorem 2). The algorithm clearly terminates in time $\Theta(\log n)$ and uses constant space in addition to the masks $\alpha^{(i)}$. The initial value of β can also be constructed in logarithmic time. We show by induction on i that $\beta = \beta^{(i)}$ and $z_b = z^{(i)}(b, x, y)$ at the start of the ith iteration of the for-loop. For $i = 0$, this clearly holds, so let $i \geq 0$. Consider the vectors x, y and z_b split into 2^{i+1}-bit blocks, and let x', y', and z'_b denote one of these blocks. After step 2a, $\gamma_{b,\ell} = (y_\ell \wedge z_{b,\ell}) \to \overline{x}_\ell, x_\ell$ when $\ell - 2^i$ is a multiple of 2^{i+1}, and $\gamma_{b,\ell} = 0$ otherwise. Let ξ denote the bit of γ_b corresponding to the middle bit of the block under consideration. By induction and the splitting lemma, $\mathrm{cpm}(b, x', y') = (\mathrm{cpm}(b, x'^L, y'^L), \mathrm{cpm}(\xi, x'^R, y'^R))^t$. After step 2b, a block of the form $\chi 00 \cdots 0$ in γ_b has been transformed to a block of the form $0\chi\chi \cdots \chi$. In step 2c, the upper half of each block z'_b is combined with the corresponding lower half of the block z'_ξ to give $t'_b = \mathrm{cpm}(b, x', y')$. That is, $t_b = z^{(i+1)}(b, x, y)$. Finally, $\beta = \beta^{(i+1)}$ after step 2e. $\square$

Since the Hamming weight can be computed in time $O(\log n)$, we have the following corollary.

Corollary 1. *Let $u, v, w \in \mathbb{F}_2^n$. The correlation coefficients $\mathcal{C}(u \overset{\boxplus}{\longleftarrow} v, w)$ and $\mathcal{C}(u \overset{\boxminus}{\longleftarrow} v, w)$ can be computed in time $\Theta(\log n)$ (using the algorithm in Theorem 2 and the expressions in Theorem 1 and Lemma 7).*

5 Generating Approximations

In this section, we derive a recursive description of the linear approximations $u \xleftarrow{\text{carry}} v, w$ with a given non-zero correlation coefficient. For simplicity, we only consider the absolute values of the correlation coefficients. The recursive description immediately gives optimal generation algorithms for the linear approximations. By Theorem 1, the magnitude of $\mathcal{C}(u \xleftarrow{\text{carry}} v, w)$ is either zero or a power of $\frac{1}{2}$. Thus, we start by considering the set of vectors $(u, v, w) \in (\mathbb{F}_2^n)^3$ such that $\mathcal{C}(u \xleftarrow{\text{carry}} v, w) = \pm 2^{-k}$.

We will use the splitting lemma to determine the approximations $u \xleftarrow{\text{carry}} v, w$ with non-zero correlation and $w_{\mathrm{H}}(\mathrm{cpm}_n(u, \mathrm{eq}(v, w))) = k$. Note that

$$\mathrm{cpm}_n(x, y) = (\mathrm{cpm}_{n-1}(x^L, y^L), \mathrm{cpm}_1(b, x_0, y_0))^t \ ,$$

where $b = x_0^L$ iff $(y_0^L, \mathrm{cpm}_{n-1}(x^L, y^L)_0) \neq (1, 1)$. Now, $\mathrm{cpm}_1(b, x_0, y_0) = 1$ iff $b = 1$ iff either $x_0^L = 1$ and $(y_0^L, \mathrm{cpm}_{n-1}(x^L, y^L)_0) \neq (1, 1)$ or $x_0^L = 0$ and $(y_0^L, \mathrm{cpm}_{n-1}(x^L, y^L)_0) = (1, 1)$. Let the $\{0, 1\}$-valued $b_n(x, y) = 1$ iff $x_0 = 1$ and $(y_0, \mathrm{cpm}_n(x, y)_0) \neq (1, 1)$ or $x_0 = 0$ and $(y_0, \mathrm{cpm}_n(x, y)_0) = (1, 1)$, let $F(n, k) = \{(u, v, w) \in (\mathbb{F}_2^n)^3 \mid \mathcal{C}(u \leftarrow v, w) = \pm 2^{-k}, b_n(u, \mathrm{eq}(v, w)) = 1\}$, and let $G(n, k) = \{(u, v, w) \in (\mathbb{F}_2^n)^3 \mid \mathcal{C}(u \leftarrow v, w) = \pm 2^{-k}, b_n(u, \mathrm{eq}(v, w)) = 0\}$. Let $A(n, k) = \{(u, v, w) \in (\mathbb{F}_2^n)^3 \mid \mathcal{C}(u \leftarrow v, w) = \pm 2^{-k}\}$. Then $A(n, k)$ is formed from $F(n-1, k-1)$ and $G(n-1, k)$ by appending any three bits to the approximations in $F(n-1, k-1)$ (since u_0 and $\mathrm{eq}(v, w)_0$ are arbitrary, and $\mathrm{cpm}_n(u, \mathrm{eq}(v, w))_0 = 1$) and by appending $\{(0, 0, 0), (1, 0, 0)\}$ to the approximations in $G(n-1, k)$ (since u_0 is arbitrary and $\mathrm{cpm}_n(u, \mathrm{eq}(v, w))_0 = 0$). Let $S = \{(0, 0, 0), (0, 1, 1), (1, 0, 1), (1, 1, 0)\}$, $T = \{(0, 0, 1), (0, 1, 0), (1, 0, 0), (1, 1, 1)\}$, and denote $y = \mathrm{eq}(v, w)$. We denote concatenation simply by juxtaposition.

The set $F(n, k)$ can be divided into two cases.

1. The vectors with $w_{\mathrm{H}}(\mathrm{cpm}_{n-1}(u^L, y^L)) = k$, $b_{n-1}(u^L, y^L) = 0$, and $b_n(u, y) = 1$. Since $(u_0, y_0) \in \{(1, 0), (1, 1)\}$ and $\mathrm{cpm}_n(u, y)_0 = 0$, this set equals $G(n-1, k)(1, 0, 0)$.
2. The vectors with $w_{\mathrm{H}}(\mathrm{cpm}_{n-1}(u^L, y^L)) = k - 1$, $b_{n-1}(u^L, y^L) = 1$ and $b_n(x, y) = 1$. Since $(u_0, y_0) \in \{(0, 1), (1, 0)\}$ and $\mathrm{cpm}_n(u, y)_0 = 1$, this set equals $F(n-1, k-1)S$.

That is,

$$F(n, k) = G(n-1, k)(1, 0, 0) \cup F(n-1, k-1)S \ .$$

Clearly, $F(1, 0) = \{(1, 0, 0)\}$ and $F(n, k) = \emptyset$ when $k < 0$ or $k \geq n$.

Similarly, $G(n, k)$ can be divided into two cases:

1. The vectors with $w_{\mathrm{H}}(\mathrm{cpm}_{n-1}(u^L, y^L)) = k$, $b_{n-1}(u^L, y^L) = 0$, and $b_n(u, y) = 0$. Since $(u_0, y_0) \in \{(0,0),(0,1)\}$ and $\mathrm{cpm}_n(u,y)_0 = 0$, this set equals $G(n-1,k)(0,0,0)$.
2. The vectors with $w_{\mathrm{H}}(\mathrm{cpm}_{n-1}(u^L, y^L)) = k-1$, $b_{n-1}(u^L, y^L) = 1$ and $b_n(u, y) = 0$. Since $(u_0, y_0) \in \{(0,0),(1,1)\}$ and $\mathrm{cpm}_n(u,y)_0 = 1$, this set equals $F(n-1,k-1)T$.

That is,

$$G(n,k) = G(n-1,k)(0,0,0) \cup F(n-1,k-1)T \ .$$

Clearly, $G(1,0) = \{(0,0,0)\}$ and $G(n,k) = \emptyset$ when $k < 0$ or $k \geq n$.

Theorem 3. *Let* $A(n,k) = \{(u,v,w) \in (\mathbb{F}_2^n)^3 \mid \mathcal{C}(u \xleftarrow{\text{carry}} v,w) = \pm 2^{-k}\}$. *Then*

$$A(n,k) = F(n-1,k-1)(\mathbb{F}_2 \times \mathbb{F}_2 \times \mathbb{F}_2) \cup G(n-1,k)\{(0,0,0),(1,0,0)\} \ ,$$

where F *and* G *are as follows. Let* $S = \{(0,0,0),(0,1,1),(1,0,1),(1,1,0)\}$ *and* $T = \{(0,0,1),(0,1,0),(1,0,0),(1,1,1)\}$. *First,* $F(1,0) = \{(1,0,0)\}$, $G(1,0) = \{(0,0,0)\}$, *and* $F(n,k) = G(n,k) = \emptyset$ *when* $k < 0$ *or* $k \geq n$. *Second, when* $0 \leq k < n$,

$$\begin{aligned} F(n,k) &= G(n-1,k)(1,0,0) \cup F(n-1,k-1)S \ , \ \text{and} \\ G(n,k) &= G(n-1,k)(0,0,0) \cup F(n-1,k-1)T \ . \end{aligned}$$

Here, juxtaposition denotes concatenation.

From this theorem, it can be seen that there are $8(n-1)$ linear approximations $u \xleftarrow{\text{carry}} v,w$ with correlation $\pm\frac{1}{2}$. In the notation of formal languages, these are the 8 approximations of the form

$$0^{n-2}1a \xleftarrow{\text{carry}} 0^{n-2}0b, 0^{n-2}0c$$

for arbitrary $a,b,c \in \{0,1\}$, and the $8(n-2)$ approximations of the form

$$0^{n-i-3}1d0^ig \xleftarrow{\text{carry}} 0^{n-i-3}0e0^i0, 0^{n-i-3}0f0^i0$$

for $(d,e,f) \in \{(0,0,1),(0,1,0),(1,0,0),(1,1,1)\}$, $g \in \{0,1\}$ and $i \in \{0,\dots,n-3\}$.

The recursive description in Theorem 3 can easily be used to generate all linear approximations with a given correlation. The straightforward algorithm uses $O(n)$ space and is linear-time in the number of generated approximations. Clearly, this immediately generalise to the case where one or two of the selection vectors are fixed. By Lemma 7, this also generalise to addition and subtraction modulo 2^n.

Corollary 2. *The set of linear approximations with correlation* $\pm 2^{-k}$ *of the carry function, addition, or subtraction modulo* 2^n *can be generated in optimal time (that is, linear in the size of the output) and* $O(n)$ *space in the RAM model (by straightforward application of the recurrence in Theorem 3 and the expressions in Lemma 7). Moreover, one or two of the selection vectors can be optionally fixed.*

Theorem 3 can also be used to determine the distribution of the correlation coefficients.

Corollary 3. *Let* $N(n,k) = \left|\{(u,v,w) \in (\mathbb{F}_2^n)^3 \mid \mathcal{C}(u \leftarrow v,w) = \pm 2^{-k}\}\right|$. *Then*

$$N(n,k) = 2^{2k+1}\binom{n-1}{k}$$

for all $0 \leq k < n$ *and* $N(n,k) = 0$ *otherwise. Thus, the number of linear approximations with non-zero correlation is* $2 \cdot 5^{n-1}$.

Proof. Based on Theorem 3, it is easy to see that

$$N(n,k) = \begin{cases} 0 \ , & \text{if } k < 0 \text{ or } k \geq n, \\ 2 \ , & \text{if } n = 1 \text{ and } k = 0, \text{ and} \\ 4N(n-1,k-1) + N(n-1,k) \ , & \text{otherwise.} \end{cases}$$

The claim clearly holds for $n = 1$. By induction, $N(n,k) = 4N(n-1,k-1) + N(n-1,k) = 4 \cdot 2^{2(k-1)+1}\binom{n-2}{k-1} + 2^{2k+1}\binom{n-2}{k} = 2^{2k+1}\binom{n-1}{k}$. Finally, $\sum_{k=0}^{n-1} N(n,k) = 2\sum_{k=0}^{n-1}\binom{n-1}{k}4^k = 2 \cdot 5^{n-1}$. □

If we let X be a random variable with the distribution

$$\Pr[X = k] = \Pr_{u,v,w}\left[-\log_2|\mathcal{C}(u \leftarrow v,w)| = k \mid \mathcal{C}(u \leftarrow v,w) \neq 0\right] \ ,$$

we see that

$$\Pr[X = k] = \binom{n-1}{k}\left(\frac{4}{5}\right)^k\left(\frac{1}{5}\right)^{n-1-k}$$

for all $0 \leq k < n$, since $2 \cdot 5^{n-1}\binom{n-1}{k}\left(\frac{4}{5}\right)^k\left(\frac{1}{5}\right)^{n-1-k} = 2^{2k+1}\binom{n-1}{k}$. Thus, X is binomially distributed with mean $\frac{4}{5}(n-1)$ and variance $\frac{4}{25}(n-1)$.

6 Conclusions

In this paper, we have considered improved algorithms for several combinatorial problems related to linear approximations of addition modulo 2^n. Our approach might seem unnecessarily complicated considering the surprising simplicity of the results (especially Theorem 3), but should lead to natural generalisations to other recursively defined function. This generalisation and applications to block ciphers are, however, left to later papers. A reference implementation of the algorithms is available from the author.

Acknowledgements. This work was supported by the Finnish Defence Forces Research Institute of Technology.

References

1. Kazumaro Aoki, Kunio Kobayashi, and Shiho Moriai. Best differential characteristic search for FEAL. In *Fast Software Encryption 1997*, volume 1267 of *LNCS*, pages 41–53. Springer-Verlag, 1997.
2. Eli Biham and Adi Shamir. *Differential Cryptanalysis of the Data Encryption Standard*. Springer-Verlag, 1993.
3. Florent Chabaud and Serge Vaudenay. Links between differential and linear cryptanalysis. In *Advances in Cryptology—Eurocrypt 1994*, volume 950 of *LNCS*, pages 356–365. Springer-Verlag, 1995.
4. Joan Daemen. *Cipher and Hash Function Design: Methods Based on Linear and Differential Cryptanalysis*. PhD thesis, Katholieke Universiteit Leuven, March 1995.
5. E.L. Lawler and D.E. Wood. Branch-and-bound methods: a survey. *Operations Research*, 14(4):699–719, 1966.
6. Helger Lipmaa. On differential properties of Pseudo-Hadamard transform and related mappings. In *Progress in Cryptology—Indocrypt 2002*, volume 2551 of *LNCS*, pages 48–61. Springer-Verlag, 2002.
7. Helger Lipmaa and Shiho Moriai. Efficient algorithms for computing differential properties of addition. In *Fast Software Encryption 2001*, volume 2355 of *LNCS*, pages 336–350. Springer-Verlag, 2002.
8. Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In *Advances in Cryptology—Eurocrypt 1993*, volume 765 of *LNCS*, pages 386–397. Springer-Verlag, 1993.
9. Mitsuru Matsui. On correlation between the order of S-boxes and the strength of DES. In *Advances in Cryptology—Eurocrypt 1994*, volume 950 of *LNCS*, pages 366–375. Springer-Verlag, 1995.
10. Mitsuru Matsui. New structure of block ciphers with provable security against differential and linear cryptanalysis. In *Fast Software Encryption 1996*, volume 1039 of *LNCS*, pages 205–218. Springer-Verlag, 1996.
11. Hiroshi Miyano. Addend dependency of differential/linear probability of addition. *IEICE Trans. Fundamentals*, E81-A(1):106–109, 1998.
12. Kaisa Nyberg. Linear approximations of block ciphers. In *Advances in Cryptology—Eurocrypt 1994*, volume 950 of *LNCS*, pages 439–444. Springer-Verlag, 1995.
13. Serge Vaudenay. Provable security for block ciphers by decorrelation. In *STACS 1998*, volume 1373 of *LNCS*, pages 249–275. Springer-Verlag, 1998.