The ANF of the Composition of Addition and Multiplication mod 2^n with a Boolean Function

An Braeken[1] and Igor Semaev[2]

[1] Department Electrical Engineering, ESAT/COSIC,
Katholieke Universiteit Leuven, Kasteelpark Arenberg 10,
B-3001 Heverlee-Leuven, Belgium
an.braeken@esat.kuleuven.ac.be

[2] Selmer Center, Inst. for Informatikk,
University of Bergen, Bergen 5020 Norway
Igor.Semaev@ii.uib.no

Abstract. Compact formulas are derived to represent the Algebraic Normal Form (ANF) of $f(\overline{x} + \overline{a} \mod 2^n)$ and $f(\overline{x} \times \overline{a} \mod 2^n)$ from the ANF of f, where f is a Boolean function on $\mathbb{F}_2^n$ and $\overline{a}$ is a constant of $\mathbb{F}_2^n$. We compare the algebraic degree of the composed functions with the algebraic degree of the original function f. As an application, the formula for addition modulo 2^n is applied in an algebraic attack on the summation generator and the E_0 encryption scheme in the Bluetooth keystream generator.

1 Introduction

Addition and multiplication modulo 2^n are operations which are very often used in cryptosystems like e.g. in the block ciphers Safer [20] and Idea [22], in the key stream generators natural sequence generator [12], summation generator [25] and E_0 encryption scheme of the Bluetooth keystream generator, and in the stream ciphers Turing [24] and Helix [14].

Recently, algebraic attacks [5, 6] have been applied successfully to stream ciphers and to some block ciphers. The central idea in the algebraic attacks is to find low degree equations or approximations of the cipher and then to solve an over-determined system of nonlinear multivariate equations of low degree by efficient methods such as XL [5], simple linearization [7] or by Gröbner Bases techniques [11].

By having compact formulas for representing the algebraic normal form of the composition of a Boolean function f with addition and multiplication modulo 2^n, we can better understand the structure of the polynomial equations of the cipher and also the consequences of mixing operations from different rings. Moreover, we give a precise criteria to avoid that the degree of the composed functions will not decrease with respect to the degree of f. As an example, we apply our formulas in order to derive the algebraic relations used in an algebraic attack

H. Gilbert and H. Handschuh (Eds.): FSE 2005, LNCS 3557, pp. 112–125, 2005.

on the summation generator and the E_0 encryption scheme of the Bluetooth keystream generator.

The paper is organised as follows. Some definitions and preliminaries that will be used later in the paper are described in Sect. 2. In Sect. 3, we derive the compact formulas for the algebraic normal forms of $f(\overline{x}+\overline{a} \mod 2^n)$ and $f(\overline{x}\times\overline{a} \mod 2^n)$ from the algebraic normal form of f, where f is a Boolean function on $\mathbb{F}_2^n$ and $\overline{a}, \overline{b}$ are constants of $\mathbb{F}_2^n$. In Sect. 4, we compare the algebraic degree of the composed functions with the algebraic degree of the original function f. In Sect. 5, the formula for addition modulo 2^n is applied in order to find the algebraic equations for the summation generator and the E_0 encryption scheme of the Bluetooth keystream generator. Finally, we present some conclusions and open problems in Sect. 6.

2 Definitions and Preliminaries

For the sake of clarity, we use " $\oplus$ " for the addition in characteristic 2 and " $+$ " for the addition modulo 2^n or in $\mathbb{R}$. The multiplication modulo 2^n is represented by " $\times$ ".

Let $\mathbb{F}_2^n$ be the set of all n-tuples of elements in the field $\mathbb{F}_2$ (Galois field with two elements), endowed with the natural vector space structure over $\mathbb{F}_2$. The correspondence between $\mathbb{F}_2^n$ and $\mathbb{Z}_{2^n}$ is defined by

$$\psi : \mathbb{F}_2^n \rightarrow \mathbb{Z}_{2^n} : \overline{u} = (u_0, \ldots, u_{n-1}) \mapsto u = \sum_{i=0}^{n-1} u_i 2^{i-1}.$$

The partial ordering $\overline{x} \preceq \overline{a}$ means that x precedes a or also $x_i \leq a_i$ for all $i \in \{0, \ldots, n-1\}$.

Let $f(\overline{x})$ be a Boolean function on $\mathbb{F}_2^n$. Any Boolean function f can be uniquely expressed in the algebraic normal form (ANF). Namely,

$$f(\overline{x}) = \bigoplus_{u \in \mathbb{Z}_{2^n}} h_u \overline{x}^u, \qquad h_u \in \mathbb{F}_2,$$

where $\overline{x}^u$ denotes $x_0^{u_0} \cdots x_{n-1}^{u_{n-1}}$. The coefficients h_u are defined by the Möbius inversion principle, $h_u = h(\overline{u}) = \sum_{\overline{x} \preceq \overline{u}} f(\overline{x})$ for any $u \in \mathbb{Z}_{2^n}$. The *algebraic degree* of f, denoted by $\deg(f)$, is equal to to the number of variables in the longest term $x_0^{u_0} \cdots x_{n-1}^{u_{n-1}}$ in the ANF of f, or simply as the maximum Hamming weight of $\overline{u}$ (denoted as $\mathrm{wt}(\overline{u})$) for which $h_u \neq 0$. The Hamming weight of a binary vector is equal to the number of nonzero components.

A vectorial Boolean function $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$ (also called (n, m) S-box or shortly S-box) can be represented by the vector $(f_1, f_2, \ldots, f_m)$, where f_i are Boolean functions from $\mathbb{F}_2^n$ into $\mathbb{F}_2$ for $1 \leq i \leq m$. The functions $(f_i)_{1 \leq i \leq m}$ are called the component functions of the S-box.

The composition $f \circ F$ of a Boolean function f on $\mathbb{F}_2^m$ with an (n, m) S-box F leads to a Boolean function on $\mathbb{F}_2^n$. Here, we will study the composition of an arbitrary Boolean function on $\mathbb{F}_2^n$ and addition respectively multiplication modulo 2^n with a fixed constant $\overline{a} \in \mathbb{F}_2^n$. The addition modulo 2^n, i.e., $\overline{r} = \overline{x} + \overline{a} \mod 2^n$ is defined by

$$\begin{aligned} &r_0 + r_1 \cdot 2 + \cdots + r_{n-1} \cdot 2^{n-1} = \\ &(x_0 + x_1 \cdot 2 + \cdots + x_{n-1} \cdot 2^{n-1}) + (a_0 + a_1 \cdot 2 + \cdots + a_{n-1} \cdot 2^{n-1}) \mod 2^n, \end{aligned} \tag{1}$$

with components $(r_0, \dots, r_{n-1})$ recursively defined by

$$\begin{aligned} r_0 &= x_0 \oplus a_0 \oplus c_0, \qquad c_0 = 0, \\ r_i &= x_i \oplus a_i \oplus c_i, \qquad c_i = x_{i-1}a_{i-1} \oplus x_{i-1}c_{i-1} \oplus a_{i-1}c_{i-1}, \\ &\forall i \in \{1, \dots, n-1\}. \end{aligned}$$

The multiplication $\overline{s} = \overline{x} \times \overline{a} \mod 2^n$ is defined by

$$\begin{aligned} &s_0 + s_1 \cdot 2 + \cdots + s_{n-1} \cdot 2^{n-1} = \\ &(x_0 + x_1 \cdot 2 + \cdots + x_{n-1}2^{n-1}) \times (a_0 + a_1 \cdot 2 + \cdots + a_{n-1}2^{n-1}) \mod 2^n, \end{aligned} \tag{2}$$

with components $(s_0, \dots, s_{n-1})$ equal to

$$\begin{aligned} s_0 &= x_0a_0, \\ s_1 &= x_1a_0 \oplus x_0a_1 \oplus c_1(x_0, a_0), \\ &\vdots \\ s_{n-1} &= x_{n-1}a_0 \oplus x_{n-2}a_1 \oplus \cdots \oplus x_0a_{n-1} \oplus c_{n-1}(x_0, \dots, x_{n-1}, a_0, \dots, a_{n-1}), \end{aligned}$$

where $c_i()$ is a function of its arguments which defines the carry bit. The number of terms of c_i grows exponentially for increasing i. We write for instance c_i for $i = 1, 2, 3$ explicitely:

$$\begin{aligned} c_1 &= c_1(x_0, a_0) = 0, \\ c_2 &= c_2(x_0, x_1, a_0, a_1) = a_0a_1x_0x_1, \\ c_3 &= c_3(x_0, \dots, x_2, a_0, \dots, a_2) = a_0a_1x_0x_1 \oplus a_0a_1x_1x_2 \oplus a_0a_1x_0x_1x_2 \\ &\qquad \oplus a_0a_2x_0x_2 \oplus a_1a_2x_0x_1 \oplus a_0a_1a_2x_0x_1. \end{aligned}$$

In this paper we will study the equations formed by the composition of addition or multiplication with a Boolean function. This corresponds with studying the ANF of $f(\overline{x} + \overline{a})$ and $f(\overline{x} \times \overline{a})$. For instance, if the Boolean function is defined by the ANF $\overline{x}^u = x_0^{u_0} \cdots x_{n-1}^{u_{n-1}}$, then the corresponding equation of $f(\overline{x}+\overline{a})$ is equal to to $r_0^{u_0} \cdots r_{n-1}^{u_{n-1}}$, where $r_0, \dots, r_{n-1}$ are functions in variables $(x_0, \dots, x_{n-1})$ defined by (1).

3 Algebraic Normal Form of $f(\overline{x} + \overline{a})$ and $f(\overline{x} \times \overline{a})$

In this section, we deduce compact formulas for representing the ANF of the functions $f(\overline{x} + \overline{a})$ and $f(\overline{x} \times \overline{a})$ using the ANF of f.

3.1 The ANF of $f(\overline{x}+\overline{a})$

Theorem 1. *If the ANF of $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2 : \overline{x} \mapsto f(\overline{x})$ is given by the monomial $\overline{x}^u$ ($u \in \mathbb{Z}_{2^n}$), then the ANF of $f(\overline{x}+\overline{a})$ with $\overline{a} \in \mathbb{F}_2^n$ a fixed constant is given by*

$$f(\overline{x}+\overline{a}) = \bigoplus_{c=0}^{u} \overline{x}^{u-c}\overline{a}^{c}, \tag{3}$$

where $u - c$ represents subtraction in $\mathbb{R}$.

Proof. To prove the theorem, we need two lemmas which can be proven by induction.

Lemma 1. *For $\overline{x}, \overline{a}_0 \in \mathbb{F}_2^n$, with $\overline{a}_0 = (a_0, 0, \ldots, 0)$, we have that*

$$(\overline{x}+\overline{a}_0)^u = \overline{x}^u \oplus \overline{x}^{u-1}\overline{a}_0.$$

Proof. (*Lemma 1*)If $n = 1$, the lemma is trivial. Suppose the lemma is true for dimension less or equal than $n-1$. We will show that the lemma holds for dimension n. If $u < 2^{n-1}$, the lemma is true by induction, otherwise write u as $2^{n-1} + u_1$, where $0 < u_1 < 2^{n-1}$, and thus

$$(\overline{x}+\overline{a}_0)^{2^{n-1}+u_1} = (\overline{x}+\overline{a}_0)^{2^{n-1}}(\overline{x}+\overline{a}_0)^{u_1}.$$

On the second term of the product, we apply induction. For the first term, we use the definition of addition (1) to compute $(\overline{x}+\overline{a}_0) = (x_0 \oplus a_0, x_1 \oplus a_0x_0, x_2 \oplus a_0x_0x_1, \ldots, x_{n-1} \oplus a_0x_0 \cdots x_{n-2})$. Taking the 2^{n-1}-th power is equal to selecting the $(n-1)$-th component in the binary representation. As a result we have

$$\begin{aligned}(\overline{x}+\overline{a}_0)^u &= (\overline{x}^{2^{n-1}} \oplus \overline{a}_0\overline{x}^{2^{n-1}-1})(\overline{x}^{u_1} \oplus \overline{x}^{u_1-1}\overline{a}_0)\\ &= \overline{x}^{2^{n-1}+u_1} \oplus \overline{x}^{2^{n-1}+u_1-1}\overline{a_0},\end{aligned}$$

where we used the fact that $\overline{a}_0\overline{x}^{2^{n-1}-1}\overline{x}^{u_1} = \overline{a}_0\overline{x}^{2^{n-1}-1}\overline{x}^{u_1-1} = \overline{a}_0\overline{x}^{2^{n-1}-1}$ in the last reduction step. This equality is due to the fact that $u_1 \preceq 2^{n-1}-1$ and $u_1 - 1 \preceq 2^{n-1}-1$. □

Lemma 2. *Denote $\overline{x} = \overline{x}_0 + 2 \times \overline{x}'$ with $\overline{x}_0 = (x_0, 0, \ldots, 0)$ and $\overline{x}' = (x_1, \ldots, x_{n-1}, 0)$. Similarly, denote $\overline{a} = \overline{a}_0 + 2 \times \overline{a}'$ with $\overline{a}_0 = (a_0, 0, \ldots, 0)$ and $\overline{a}' = (a_1, \ldots, a_{n-1}, 0)$, then*

$$(2 \times (\overline{x}' + \overline{a}'))^u = \begin{cases} 0 & \text{if } u \text{ is odd,} \\ \bigoplus_{v=0}^{\frac{u}{2}} (2 \times \overline{x}')^{u-2v}(2 \times \overline{a}')^{2v} & \text{if } u \text{ is even.} \end{cases}$$

Proof. (*Lemma 2*) We prove the lemma by induction on the number n of variables. Because multiplication by 2 only shifts the vector over one position, it follows that

$$(2 \times \overline{x}')^u = \begin{cases} 0 & \text{if } u \text{ is odd,} \\ (\overline{x}')^{\frac{u}{2}} & \text{if } u \text{ is even.} \end{cases} \tag{4}$$

By induction on n, we have for even u that

$$(\overline{x}' + \overline{a}')^{\frac{u}{2}} = \bigoplus_{v=0}^{\frac{u}{2}} \overline{x}'^{\frac{u}{2}-v}\overline{a}'^{v}.$$

If we rescale the previous formula using (4), we get the formula of the lemma. □

By using the previous lemmas, we are now able to prove the theorem. We start with applying Lemma 1 repeatedly.

$$\begin{aligned}(\overline{x} + \overline{a})^u &= (\overline{x}_0 + 2 \times \overline{x}' + \overline{a}_0 + 2 \times \overline{a}')^u \\ &= (2 \times \overline{x}' + 2 \times \overline{a}' + \overline{x}_0)^u \oplus (2 \times \overline{x}' + 2 \times \overline{a}' + \overline{x}_0)^{u-1}\overline{a}_0 \\ &= (2 \times \overline{x}' + 2 \times \overline{a}')^u \oplus (2 \times \overline{x}' + 2 \times \overline{a}')^{u-1}\overline{x}_0 \\ &\quad \oplus \overline{a}_0((2 \times \overline{x}' + 2 \times \overline{a}')^{u-1} \oplus (2 \times \overline{x}' + 2 \times \overline{a}')^{u-2}\overline{x}_0). \end{aligned} \tag{5}$$

Note that multiplication modulo 2^n is distributive with respect to addition modulo 2^n, i.e. $(2 \times \overline{x}' + 2 \times \overline{a}') = 2 \times (\overline{x}' + \overline{a}')$. As a consequence, we can apply Lemma 2 on Equation (5). This implies that we need to distinguish the case u is odd and the case u is even. We give here the proof for u odd. The proof for u even is similar.

$$\begin{aligned}(\overline{x} + \overline{a})^u &= \overline{x}_0(2 \times \overline{x}' + 2 \times \overline{a}')^{u-1} \oplus \overline{a}_0(2 \times \overline{x}' + 2 \times \overline{a}')^{u-1} \\ &= \overline{x}_0 \bigoplus_{v=0}^{\frac{u-1}{2}} (2 \times \overline{x}')^{u-2v-1}(2 \times \overline{a}')^{2v} \oplus \overline{a}_0 \bigoplus_{v=0}^{\frac{u-1}{2}} (2 \times \overline{x}')^{u-2v-1}(2 \times \overline{a}')^{2v}. \end{aligned} \tag{6}$$

The following equalities hold

$$\begin{aligned}(2 \times \overline{x}')^{u-2v-1} &= \overline{x}^{u-2v-1}, \\ \overline{x}_0(2 \times \overline{x}')^{u-2v-1} &= \overline{x}^{u-2v}, \end{aligned} \tag{7}$$

because $2 \times \overline{x}' = (0, x_1, \ldots, x_{n-1})$ and $u - 2v - 1$ is even. The same argument holds for $2 \times \overline{a}'$ and thus

$$\begin{aligned}(2 \times \overline{a}')^{2v} &= \overline{a}^{2v}, \\ \overline{a}_0(2 \times \overline{a}')^{2v} &= \overline{a}^{2v+1}. \end{aligned} \tag{8}$$

After substituting the equalities (7) and (8) in Equation (6) and collecting the terms, we find Formula (3). □

Remark 1. If $u = 2^i$, Formula (3) expresses the i-th component of the sum $\overline{x}+\overline{a}$. Similarly, if $u = 2^i + 2^j$, Formula (3) expresses the product of the i-th and j-th component of the sum $\overline{x} + \overline{a}$. Note that Formula (3) consists only of all terms for which the integer sum of the exponents of $\overline{x}$ and $\overline{a}$ is exactly equal to u. The formula can be easily generalized for the addition of n elements $\overline{y}_1, \ldots, \overline{y}_n$ of $\mathbb{F}_2^n$

by applying Formula (3) recursively. Again, the result is equal to the sum of all terms with sum of exponents equal to u.

$$f(\overline{y}_1 + \cdots + \overline{y}_n) = \bigoplus_{\substack{k_0,\ldots,k_{n-1}\geq 0 \\ k_0+\cdots+k_{n-1}=u}} \overline{y}_1^{k_0}\overline{y}_2^{k_1}\cdots\overline{y}_n^{k_{n-1}} \tag{9}$$

We now generalize Theorem 1 for Boolean functions where the ANF consists of an arbitrary number of terms. By collecting the terms in a right way, we obtain the following formula.

Corollary 1. *If the ANF of $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is given by $\bigoplus_{u\in\mathbb{Z}_{2^n}} h_u\overline{x}^u, h_u \in \mathbb{F}_2$, the ANF of $f(\overline{x}+\overline{a})$ is given by*

$$f(\overline{x}+\overline{a}) = \bigoplus_{v}(\bigoplus_{u\geq v} h_u\overline{a}^{u-v})\overline{x}^v, \tag{10}$$

where $u - v$ represents the subtraction modulo 2^n.

Example 1. Consider the ANF of the function $f(x_0, x_1, x_2) = \overline{x}^5 \oplus \overline{x}^1$. The ANF of $f(\overline{x}+\overline{a})$ is then determined by the previous corollary:

$$f(\overline{x}+\overline{a}) = (\overline{a}^1 \oplus \overline{a}^5) \oplus \overline{x}^1(\overline{a}^0 \oplus \overline{a}^4) \oplus \overline{x}^2\overline{a}^3 \oplus \overline{x}^3\overline{a}^2 \oplus \overline{x}^4\overline{a}^1 \oplus \overline{x}^5,$$

which can also be written as

$$f(\overline{x}+\overline{a}) = (a_0 \oplus a_0a_2) \oplus x_0(1 \oplus a_2) \oplus x_1a_0a_1 \oplus x_0x_1a_1 \oplus x_2a_0 \oplus x_0x_2.$$

3.2 The ANF of $f(\overline{x} \times \overline{a})$

Theorem 2. *If the ANF of $f : \mathbb{F}_2^n \to \mathbb{F}_2 : \overline{x} \mapsto f(\overline{x})$ is given by the monomial $\overline{x}^u$ ($u \in \mathbb{Z}_{2^n}$), then the ANF of $f(\overline{x}\times\overline{a})$ with $\overline{a} \in \mathbb{F}_2^n$ a fixed constant is given by*

$$f(\overline{x}\times\overline{a}) = \bigoplus_{k^u=[k_0,\ldots,k_{n-1}]} \overline{a}^{r^k}\overline{x}^{s^k}, \tag{11}$$

where $k^u = [k_0, \ldots, k_{n-1}]$ satisfies

$$\begin{aligned} &k_0 \geq 0, \ldots, k_{n-1} \geq 0; \\ &k_0 + 2k_1 + \cdots + 2^{n-1}k_{n-1} = u. \end{aligned} \tag{12}$$

The integers $r^k = r_0^k + 2r_1^k + \cdots + r_{n-1}^k$ and $s^k = s_0^k + 2s_1^k + \cdots + s_{n-1}^k$ are defined by the following $(n+1)\times(n+1)$-table:

s_{n-1}^k	$k_{0,n-1}$	0	$\cdots$	0
s_{n-2}^k	$k_{0,n-2}$	$k_{1,n-2}$	$\cdots$	0
		$\ddots$		
s_0^k	$k_{0,0}$	$k_{1,0}$	$\cdots$	$k_{n-1,0}$
	r_0^k	r_1^k	$\cdots$	r_{n-1}^k

For each $k^u = [k_0, \ldots, k_{n-1}]$ that satisfies the properties as described by (12), we fill the table with the binary representation of $k_0 = (k_{0,0}, \ldots, k_{0,n-1}), \ldots, k_{n-1} = (k_{n-1,0}, \ldots, k_{n-1,n-1})$. The digit r_i^k (resp. s_i^k) for all $i \in \{0, \ldots, n-1\}$ is equal to to 1 if the corresponding column (resp. row) is different from the all-zero vector and is equal to to 0 otherwise. The integer r^k is then defined by the binary representation $(r_0^k, \ldots, r_{n-1}^k)$ and the integer s_k by $(s_0^k, \ldots, s_{n-1}^k)$.

Proof. Note that the multiplication $\overline{a} \times \overline{x}$ can be written as a sum:

$$\overline{a} \times \overline{x} = a_0 \cdot \overline{x} + a_1 \cdot (2 \times \overline{x}) + \cdots + a_{n-1} \cdot (2^{n-1} \times \overline{x}).$$

By formula (9) for the addition of n points, we obtain

$$f(\overline{x} \times \overline{a}) = \bigoplus_{\substack{k_0,\ldots,k_{n-1} \geq 0 \\ k_0+\cdots+k_{n-1}=u}} (a_0 x)^{k_0} (a_1 \cdot (2 \times \overline{x}))^{k_1} \cdots (a_{n-1} \cdot (2^{n-1} \times \overline{x}))^{k_{n-1}}.$$

As explained in the proof of Lemma 2, we have for the general case i, $i \in \{0, \ldots, n-1\}$ that $(2^i \times \overline{x})^{k_i}$ shifts the components of x over i positions, which means that

$$(2^i \times \overline{x})^{k_i} = \begin{cases} \overline{x}^{\frac{k_i}{2^i}} & \text{if } k_i \equiv o(2^i) \\ 0 & \text{otherwise.} \end{cases}$$

Consequently, we can write the above equation for $f(\overline{x} \times \overline{a})$ as:

$$f(\overline{x} \times \overline{a}) = \bigoplus_{\substack{k_0,\ldots,k_{n-1} \geq 0 \\ k_1 \equiv o(2),\ldots,k_{n-1} \equiv o(2^{n-1}) \\ k_0+\cdots+k_{n-1}=u}} (a_0 \overline{x})^{k_0} (a_1 \overline{x})^{\frac{k_1}{2}} \cdots (a_{n-1} \overline{x})^{\frac{k_{n-1}}{2^{n-1}}},$$

where $k_i \equiv o(2^i)$ means that 2^i is a divisor of k_i. This representation contains mixed terms, i.e. terms which consists of powers of vector x and powers of components of a. Moreover because $x_i^2 = x_i$, we can very often reduce the powers of x. However by translating this form in the representation given by (11), we avoid these disadvantages. This can be seen by the definition of the vectors r^k and s^k. The value r_i^k (resp s_i^k) for all $i \in \{0, \ldots, n-1\}$ is equal to to 1 if the corresponding column (resp. row) is different from the all-zero vector and is equal to to 0 otherwise. □

Remark 2. We note that the formula of multiplication, unlike the formula of addition, does not immediately give the full reduced form of the ANF because some terms can cancel out. For instance (see also Example 2), if the pattern $\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}$ appears in the represenation table of the exponents of a term, then also the pattern $\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}$ satisfies the same conditions of (12) and will give the same exponents. Consequently both terms will cancel out. Another clear example is the pattern $\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}$ which is equivalent with the pattern $\begin{smallmatrix} 0 & 1 \\ 1 & 1 \end{smallmatrix}$. However, the formula is still much more practical than using explicitly the definition of multiplication.

We now generalize Theorem 2 for Boolean functions where the ANF consist of an arbitrary number of terms.

Corollary 2. *If the ANF of $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$ is given by $\bigoplus_{u \in \mathbb{Z}_{2^n}} h_u \overline{x}^u, h_u \in \mathbb{F}_2$, the ANF of $f(\overline{x} \times \overline{a})$ is given by*

$$f(\overline{x} \times \overline{a}) = \bigoplus_{u \in \mathbb{Z}_{2^n}} h_u \left(\bigoplus_{k^u = [k_0, \ldots, k_{n-1}]} \overline{a}^{r_{k,u}} \overline{x}^{s_{k,u}} \right),$$

where k^u, $r_{k,u}$ and $s_{k,u}$ for each u (corresponding with a non-zero h_u in the ANF of f) are defined as in Theorem 2.

Example 2. Consider the ANF of the function $f(x_0, x_1, x_2) = \overline{x}^5$. To compute the ANF of $f(\overline{x} \times \overline{a})$, we first determine all k^5 that satisfy the properties of (12). There are 4 different possibilities for $k = (k_0, k_1, k_2)$, i.e. $k = (5, 0, 0), k = (3, 1, 0), k = (1, 2, 0), k = (1, 0, 1)$. For each k, we compute the corresponding exponent of $\overline{a}$ and $\overline{x}$ by computing its corresponding table:

1	1 0 0
0	0 0 0
1	1 0 0
	1 0 0

0	0 0 0
1	1 0 0
1	1 1 0
	1 1 0

0	0 0 0
1	0 1 0
1	1 0 0
	1 1 0

0	0 0 0
0	0 0 0
1	1 0 1
	1 0 1

$k = (5, 0, 0), k = (3, 1, 0), k = (1, 2, 0), k = (1, 0, 1)$

As a consequence, we get the following ANF of f:

$$\begin{aligned} f(\overline{x} \times \overline{a}) &= \overline{a}_1 \overline{x}^5 \oplus \overline{a}_3 \overline{x}^3 \oplus \overline{a}_3 \overline{x}^3 \oplus \overline{a}_5 \overline{x}^1. \\ &= \overline{a}_1 \overline{x}^5 \oplus \overline{a}_5 \overline{x}^1. \end{aligned}$$

4 Comparison of the Degrees

In this section, we compare the degrees of $f(\overline{x})$ at the one side with the degrees of $f(\overline{x} + \overline{a})$ and $f(\overline{x} \times \overline{a})$ at the other side.

4.1 Degrees of $f(\overline{x})$ and $f(\overline{x} + \overline{a})$

Theorem 3. *If $f(\overline{x}) = \bigoplus_{u \in \mathbb{Z}_{2^n}} h_u \overline{x}^u$, define $u_m = \max_{\substack{h_u \neq 0 \\ u \in \mathbb{Z}_{2^n}}} u$. The degree of the function $f_{\overline{a}} : \mathbb{F}_2^n \rightarrow \mathbb{F}_2 : \overline{x} \rightarrow f(\overline{x} + \overline{a})$ will be for all values of $\overline{a} \in \mathbb{F}_2^n$ in the interval*

$$\begin{array}{ll} [\mathrm{wt}(u_m), \lfloor \log_2 u_m \rfloor] & \textit{if } \mathrm{wt}(u_m) \leq \lfloor \log_2 u_m \rfloor \textit{ or } u_m \neq 2^{\lceil \log_2 u_m \rceil} - 1, \\ [\mathrm{wt}(u_m), \mathrm{wt}(u_m)] & \textit{otherwise.} \end{array}$$

Proof. From Corollary 1, we have that

$$\begin{aligned} f(\overline{x}+\overline{a}) &= \bigoplus_{v} \left(\bigoplus_{u \geq v} h_u \overline{a}^{u-v} \right) \overline{x}^{v} \\ &= \overline{x}^{u_m} \oplus \bigoplus_{v<u_m} \left(\bigoplus_{u \geq v} h_u \overline{a}^{u-v} \right) \overline{x}^{v}. \end{aligned} \tag{13}$$

From (13), the lowerbound for the degree of $f_{\overline{a}}$ is equal to to $\mathrm{wt}(u_m)$, because the term $\overline{x}^{u_m}$ always appears and does not cancel out in the ANF of $f_{\overline{a}}$. The degree of the function $f_{\overline{a}}$ is exactly equal to $\mathrm{wt}(u_m)$ if the term $\bigoplus_{u \geq v} h_u \overline{a}^{u-v}$ from (13) is equal to to zero for all v with weight greater than $\mathrm{wt}(u_m)$.

The upperbound is equal to to the maximum weigth of v for which $v < u_m$. This is equal to $\lfloor \log_2 u_m \rfloor$ for $u \neq 2^{\lceil \log_2 u_m \rceil} - 1$. The degree of $f_{\overline{a}}$ is exactly equal to $\lfloor \log_2 u_m \rfloor$ if the term $\bigoplus_{u \geq v} h_u \overline{a}^{u-v}$ from (13) is equal to to one for at least one v with weight equal to $\lfloor \log_2 u_m \rfloor$. □

Example 3. Consider the function $f : \mathbb{F}_2^7 \to \mathbb{F}_2 : \overline{x} \mapsto \overline{x}^{64} \oplus \overline{x}^{62}$. The degree of this function is 5, while u_m is equal to to 64 with weight one. We now show that the degree of the function $f_{\overline{a}}$ is between one and six according to Theorem 3 and depending on the value $\overline{a}$.

For odd a, i.e. $a_0 = 1$, the term x^{63} appears in the ANF of $f_{\overline{a}}$, and thus the corresponding functions have degree 6. If $a_0 = 0, a_1 = 0$, the function $f_{\overline{a}}$ has degree 5 because of the term x^{62} in the ANF of the function. For functions $f_{\overline{a}}$ with $a_0 = 0, a_1 = 1, a_2 = 0$, the resulting degree is equal to to 4. If $a_0 = 0, a_1 = 1, a_2 = 1, a_3 = 0$ the degree of $f_{\overline{a}}$ is 3, and if $a_0 = 0, a_1 = 1, a_2 = 1, a_3 = 1, a_4 = 0$ the degree of $f_{\overline{a}}$ becomes 2. Finally for $a = (0, 1, 1, 1, 1, 0, 1)$ and $a = (0, 1, 1, 1, 1, 0, 0)$ the function $f_{\overline{a}}$ has degree 1.

In order to diminish the degeneration of the degree of the function $f(\overline{x}+\overline{a})$ for $\overline{a} \in \mathbb{F}_2^n$ with respect to the degree of the function $f(\overline{x})$, we need to take care that $|\,\mathrm{wt}(u_m) - \deg(f)|$ is small. The condition that a function satisfies $\mathrm{wt}(u_m) = \deg(f)$ will appear for instance if f is of degree d and contains the monomial $x_{n-d-1} \cdots x_{n-1}$.

4.2 Degrees of $f(\overline{x})$ and $f(\overline{x} \times \overline{a})$

Theorem 4. *If $a_0 \neq 0$ then the degree of $f(\overline{x} \times \overline{a})$ will be greater or equal than the weight of u_m^0, where $u_m^0 = \max_{\substack{h_u \neq 0 \\ u \in \mathbb{Z}_{2^n}}} u$. If $a_0 = 0$ and $a_1 \neq 0$, then the degree of $f(\overline{x} \times \overline{a})$ will be greater or equal than the weight of u_m^1, where $u_m^1 = \max_{\substack{h_u \neq 0 \\ u \in \mathbb{Z}_{2^n} \\ u \equiv o(2)}} u$. In general, if $a_0 = \cdots a_{i-1} = 0$ and $a_i \neq 0$, then the degree of $f(\overline{x} \times \overline{a})$ will be greater or equal than the weight of u_m^i, where $u_m^i = \max_{\substack{h_u \neq 0 \\ u \in \mathbb{Z}_{2^n} \\ u \equiv o(2^i)}} u$.*

Proof. We give the proof for the degree of $f(\overline{x} \times \overline{a})$ and for the case $a_0 \neq 0$. All other cases can be proven in the same way. In the following, we denote by $g(\overline{x})$ the function $g(\overline{x}) = (a_0\overline{x})^{k_0}(a_1\overline{x})^{\frac{k_1}{2}} \cdots (a_{n-1}\overline{x})^{\frac{k_{n-1}}{2^{n-1}}}$. From Theorem 2, we can write

$$\begin{aligned}
f(\overline{x} \times \overline{a}) &= \bigoplus_{u \in \mathbb{Z}_{2^n}} h_u \bigoplus_{\substack{k_0,\ldots,k_{n-1} \geq 0 \\ k_1 \equiv o(2),\ldots k_{n-1} \equiv o(2^{n-1}) \\ k_0 + \cdots k_{n-1} = u}} g(\overline{x}) \\
&= \bigoplus_{u \in \mathbb{Z}_{2^n}} h_u(a_0\overline{x}^u) \oplus \bigoplus_{u \in \mathbb{Z}_{2^n}} h_u \bigoplus_{\substack{k_0,\ldots,k_{n-1} \geq 0 \\ k_0 \neq u, k_1 \equiv o(2),\ldots k_{n-1} \equiv o(2^{n-1}) \\ k_0 + \cdots k_{n-1} = u}} g(\overline{x}) \\
&= a_0\overline{x}^{u_{m_0}} \oplus \bigoplus_{u \in \mathbb{Z}_{2^n}} h_u \bigoplus_{\substack{k_0,\ldots,k_{n-1} = 0 \\ k_0 \neq u_{m_0}, k_1 \equiv o(2),\ldots k_{n-1} \equiv o(2^{n-1}) \\ k_0 + \cdots k_{n-1} = u}} g(\overline{x})
\end{aligned}$$

□

5 Algebraic Attacks

Algebraic attacks exploit the existence of low degree equations. Once a system of nonlinear multivariate equations of low degree is obtained, it is solved by efficient methods such as XL [5], simple linearization [7] or by Gröbner Bases techniques [11]. We will derive in this section low degree equations for the summation generator and the E_0 encryption scheme in the Bluetooth key stream generator.

5.1 Algebraic Attack on the Summation Generator

Consider a summation generator, proposed by Rueppel [25], that consists of n binary Linear Feedback Shift Registers (LFSR). The output bit of the j-th LFSR at time t will be denoted by x_j^t. The binary output bit z^t is defined by

$$z^t = x_1^t \oplus \cdots \oplus x_n^t \oplus c_0^t, \tag{14}$$

where c_0^t is the 0-th bit of the carry $\overline{c}^t = (c_0^t, \ldots, c_{k-1}^t)$ with $k = \lceil \log_2 n \rceil$. The carry for the next stage $t+1$ is computed by

$$\overline{c}^{t+1} = \left\lfloor (x_1^t + \cdots + x_n^t + \overline{c}^t)/2 \right\rfloor. \tag{15}$$

The summation generator is an (n, k)-combiner, which is a stream cipher that combines n LFSRs and has k bits of memory. The summation generator produces a key stream with linear complexity close to its period, which is equal to the product of the periods of the n LFSRs. Moreover, the generator has maximum algebraic degree and maximum order of correlation-immunity (cf Siegenthaler's inequality $t \leq n-d-1$ for combiners without memory). For this reason, summation generators are very interesting building blocks in stream ciphers. We here describe the algebraic attack as presented in [17], but by using the formulas for addition modulo 2^n as given is Subsection 3.1, which makes the analysis and the proofs from [17] much shorter.

To simplify notations, we denote by σ_i^t for $1 \leq i \leq n$, the symmetric polynomial that contains all terms of degree i in the variables $x_1^t, \ldots, x_n^t$, i.e.

$$\begin{aligned}
\sigma_1^t &= \oplus_{i=1}^n x_i^t, \\
\sigma_2^t &= \oplus_{1 \leq i_1 < i_2 \leq n}^n x_{i_1}^t x_{i_2}^t, \\
&\vdots \\
\sigma_n^t &= x_1^t x_2^t \cdots x_n^t.
\end{aligned}$$

We now show how we can use Formula (3) in order to simplify the proof of the main theorem in [17].

Theorem 5. *For a summation generator of $n = 2^k$ LFSRs we can write an algebraic equation connecting LFSR output bits and $k+1$ consecutive key stream bits of degree upperbounded by 2^k in the LFSR output bits.*

Proof. By using formula (9), we immediately determine $c_0^{t+1}, \ldots, c_{k-1}^{t+1}$, which are by defintion (15) the first until the $(k-1)$-th components of the sum $x_1^t + \cdots + x_n^t + \overline{c}^t$.

$$\begin{aligned}
c_0^{t+1} &= \sigma_2^t \oplus c_0^t \sigma_1^t \oplus c_1^t, && (16)\\
c_1^{t+1} &= \sigma_4^t \oplus c_0^t \sigma_3^t \oplus c_1^t \sigma_2^t \oplus c_0^t c_1^t \sigma_1^t \oplus c_2^t, && (17)\\
c_2^{t+1} &= \sigma_8^t \oplus c_0^t \sigma_7^t \oplus c_1^t \sigma_6^t \oplus c_0^t c_1^t \sigma_5^t \oplus c_2^t \sigma_4^t \\
&\quad \oplus c_0^t c_2^t \sigma_3^t \oplus c_1^t c_2^t \sigma_2^t \oplus c_0^t c_1^t c_2^t \sigma_1^t \oplus c_3^t, && (18)\\
&\vdots \\
c_{k-1}^{t+1} &= \sigma_{2^k}^t \oplus c_0^t \sigma_{2^k-1}^t \oplus \cdots \oplus c_0^t \cdots c_{k-1}^t \sigma_1^t. && (19)
\end{aligned}$$

As a consequence, c_i^{t+1} for $0 \leq i \leq k-1$ can be expressed by an equation of degree 2^{i+1} in the LFSR output bits $x_1^t, \ldots, x_n^t$ because it contains the term $\sigma_{2^{i+1}}^t$.

From (14), we derive an equation for c_0^t of degree one in the variables $x_1^t, \ldots, x_n^t$,

$$c_0^t = \sigma_1^t \oplus z^t. \tag{20}$$

Substitution of the equations for c_0^t (20) and c_0^{t+1} ((20) shifted over one position) in Equation (16), results in an equation for c_1^t of degree 2 in the variables $x_1^t, \ldots, x_n^t, x_1^{t+1}, \ldots, x_n^{t+1}$, i.e.,

$$c_1^t = \sigma_2^t \oplus (z^t \oplus 1)\sigma_1^t \oplus \sigma_1^{t+1} \oplus z^{t+1}. \tag{21}$$

Substitution of the equations for c_0^t (20), c_1^t and c_1^{t+1} (21) in Equation (17), results in an equation for c_2^t of degree 4 in the variables $x_1^t, \ldots, x_n^t, x_1^{t+1}, \ldots, x_n^{t+1}$,

$x_1^{t+2}, \ldots, x_n^{t+2}$. This process is repeated and in the last step we substitute the equations for $c_0^t, \ldots, c_{k-1}^t, c_{k-1}^{t+1}$ in Equation (19) which results in an equation in the LFSR output bits $x_1^t, \ldots, x_n^t, \ldots, x_1^{t+k}, \ldots, x_n^{t+k}$. □

We want to note that a similar approach for deriving the equations can also be used on two versions of stream ciphers which are derived from the summation generator: the improved summation generator with 2-bit memory [18] and the parallel stream cipher for secure high-speed communications [19].

5.2 Algebraic Attack on Bluetooth Key Stream Generator

The E_0 encryption system used in the Bluetooth specification [3] for wireless communication is derived from the summation generator and consists of 4 LFSRs. The variables z^t, x_i^t, σ_i^t have the same meaning as explained by the summation generator. Now the initial state consists of 4 memory bits, denoted by $(c_0^{t+1}, c_1^{t+1}, S_0^{t+1}, S_1^{t+1})$. In order to obtain the output and the initial state, the following equations are derived:

$$\begin{aligned}
z^t &= \sigma_1^t \oplus c_0^t \\
c_0^{t+1} &= S_0^{t+1} \oplus c_0^t \oplus c_0^{t-1} \oplus c_1^{t-1} \\
c_1^{t+1} &= S_1^{t+1} \oplus c_1^t \oplus c_0^{t-1} \\
(S_0^{t+1}, S_1^{t+1}) &= \left\lfloor \frac{x_1^t + x_2^t + x_3^t + x_4^t + c_0^t + 2c_1^t}{2} \right\rfloor
\end{aligned}$$

Using our formula for addition, we immediately find the algebraic equations for S_0^{t+1}, S_1^{t+1}:

$$\begin{aligned}
S_0^{t+1} &= \sigma_4^t \oplus \sigma_3^t c_0^t \oplus \sigma_2^t c_1^t \oplus \sigma_1^t c_0^t c_1^t \\
S_1^{t+1} &= \sigma_2^t \oplus \sigma_1^t c_0^t \oplus c_1^t
\end{aligned}$$

In [1], these equations are justified by comparing the truth tables of both sides, but no formal proof was given. The next step is to manipulate the equations in such a way that an equation is obtained where all memory bits are eliminated. These equations have degree 4 and are used in the algebraic attack.

6 Conclusions

We have computed compact formulas for representing the ANF of the composition of a Boolean function with addition modululo 2^n, multiplication modulo 2^n and a combination of both, from the ANF of the original function. We have shown that comparing the degrees of the compositions and the original function is not possible in general. If the function satisfies the property that its degree is equal to the weight of the highest coefficient modulo 2^n in its ANF representation, then the degree of the composition with addition modulo 2^n and multiplication with odd constants modulo 2^n will always be higher or equal than the degree of the original function. Finally, we have used our formula of addition modulo 2^n for finding low degree equations of the summation generator and the E_0 encryption scheme in the Bluetooth key stream generator.

An open problem is to further simplify the formula for multiplication modulo 2^n. Further research is required to investigate if those formulas could be used for finding efficient low degree equations in other cryptosystems. For instance, an application of our formulas on the T-functions of Shamir and Klimov [15] seems to be possible for getting better insight in the algebraic equations.

Acknowledgments

The authors would like to thank Bart Preneel and the anonymous referees for the helpful remarks and comments. An Braeken is research assistant of the Fund for Scientific research - Flanders (Belgium) and Igor Semaev was staying at the University of Leuven under the project Flanders FWO G.0186.02.

References

1. F. Armknecht, A Linearization Attack on the Bluetooth Key Stream Generator, Cryptology ePrint Archive, Report 2002/191, http://eprint.iacr. org/2002/191, 2002.
2. F. Armknecht, M. Krause, Algebraic Attacks on Combiners with Memory, Crypto 2003, LNCS 2729, Springer-Verlag, pp. 162–175, 2003.
3. Bluetooth SIG, Specification of the Bluetooth System, Version 1.1, 1 Febrary 22, 2001, available at http://www.bluetooth.com.
4. D.H. Bailey, K. Lee, H.D. Simon, Using Strassen's Algorithm to Accelerate the Solution of Linear Systems, J. of Supercomputing, Vol. 4, pp. 357–371, 1990.
5. N. Courtois, Higher Order Correlation Attacks, *XL* Algorithm and Cryptanalysis of *Toyocrypt*, Asiacrypt 2002, LNCS 2587, Springer-Verlag, pp. 182–199, 2002.
6. N. Courtois, W. Meier, Algebraic Attacks on Stream Ciphers with Linear Feedback, *Eurocrypt'03*, LNCS 2656, Springer-Verlag, pp. 345-359, 2003.
7. N. Courtois, A. Klimov, J. Patarin, A. Shamir, Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations, Eurocrypt'00, LNCS 1807, Springer-Verlag, pp. 392-407, 2000.
8. N. Courtois, Fast Algebraic Attacks on Stream Cipher with Linear Feedback, Crypto 2003, LNCS 2729, Springer-Verlag, pp. 176-194, 2003.
9. N. Courtois, Algebraic Attacks on Combiners with memory and Several Outputs, eprint archive, 2003/125
10. N. Courtois, J. Pieprzyk, Cryptanalysis of Block Ciphers with Overdefined Systems of Equations, Asiacrypt 2002, LNCS 2501, Springer-Verlag, pp. 267–287, 2002.
11. A. Joux, J.-C. Faugére, Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Grobner Bases, Crypt 2003, LNCS 2729, Springer-Verlag 2003, pp. 44–60, 2003.
12. T.W. Cusick, C. Ding, A. Renvall, Stream Ciphers and Number Theory, Elsevier, Amsterdam, 1998.

13. C. Ding, The Differential Cryptanalysis and Design of the Natural Stream Ciphers, Fast Software Encryption 1994, LNCS 809, Springer-Verlag, pp. 101–115, 1994.
14. N. Ferguson, D. Whiting, B. Schneier, J. Kelsey, S. Lucks, T. Kohno, Helix: Fast Encryption and Authentication in a Single Cryptographic Primitive, Fast Software Encryption 2003, LNCS 2887, Springer-Verlag, pp. 345–362, 2003.
15. A. Klimov, A. Shamir, New Cryptographic Primitives Based on Multiword T-Functions, Fast Software Encryption 2004, LNCS 3017, Springer-Verlag, pp. 1–15, 2004.
16. A. Klapper, M. Goresky, Cryptanalysis Based on 2-adic Rational Approximation, Crypto 1995, LNCS 963, Springer-Verlag, pp. 262–273, 1995.
17. D.H. Lee, J. Kim, J. Hong, J.W. Han, D. Moon, Algebraic Attacks on Summation Generators, Fast Software Encryption 2004, LNCS 3017, Srpinger-Verlag, pp. 3448, 2004.
18. H. Lee, S. Moon, On an Improved Summation Generator with 2-bit Memory, Signal Processing 80, pp. 211-217, 2000.
19. H. Lee, S. Moon, Parallel Stream Cipher for Secure High-Speed Communications, Signal Processing 82, pp. 259–265, 2002.
20. J.L. Massey, SAFER K-64: A Byte-Oriented Block-Ciphering Algorithm, Fast Software Encryption, Cambridge Security Workshop Proceedings, Springer-Verlag, 1994, pp. 1–17.
21. W. Meier, O. Staffelbach, Correlation Properties of Combiners with Memory in Stream Cipher, Journal of Cryptology, Vol. 5, pp. 67–86, 1992.
22. X. Lai, J.L. Massey, A Proposal for a New Block Encryption Standard, Eurocrypt 1990, LNCS 473, Springer-Verlag 1990, pp. 389–404, 1991.
23. D.J. Newman, Analytic Number Theory, Springer-Verlag New York, 1998.
24. G. Rose, P. Hawkes, Turing: a Fast Stream Cipher, Fast Software Encryption 2003, Fast Software Encryption 2003, LNCS 2887, Springer-Verlag, pp. 307-324, 2003.
25. R.A. Rueppel, Correlation Immunity and the Summation Generator, Crypto 1985, LNCS 218, Springer-Verlag 1986, pp. 260–272, 1985.
26. B. Schneier, Applied Cryptography, Wiley, New York, 1996.