TC-linearisation of tweakable polynomials

Josef Bárta, Michal Hojsík

***Abstract*—Based on the Cube Attack by Itai Dinur and Adi Shamir and another, in the essence similar, method we devised a new polynomial linearisation technique, which proved to be more powerful, than the Cube Attack alone. Moreover, we present detailed description with formal proof not only of our findings, but also of the Cube Attack. Finally, we demonstrate the results of our efforts on a Trivium variant that is reduced in key and initialisation vector bit count. We managed to linearise polynomials representing a keystream bit output after up to 621 initialisation rounds using purely techniques described in this paper, compared to 581 initialisation rounds with original attack.**

***Index Terms*—Cube Attacks, cryptanalysis, stream ciphers, lightweight cryptography, Boolean functions, linearisation, tweakable polynomials**

I. INTRODUCTION

IN this paper we present a detailed description of Adi Shamir's Cube Attack and then to devise a generalisation, which could help push the boundaries of usability of the Cube Attack. Other important target of ours are of course the polynomials as such. Therefore we decided to actually compute the polynomial expression of the state and keystream bits of Trivium reduced in the number of the bits used as variables and then to analyse them without having to do any guessing. More specifically, we wanted to assess, whether the polynomials are linearisable using the techniques devised by us and whether they are any more effective than the original Cube Attack.

In the next section there is described the basic notation we use throughout the paper. Thereafter we "translate" the Cube Attack into our notation and we present its detailed description. In further sections we describe in the same manner another technique that can be used for attack in a similar way to the Cube Attack, which we simplify into two easier, but nonetheless effective techniques. The details about analysis of the polynomials and a description of the cryptosystem they represent can be found in the second to last section.

II. TC-LINEARISATION OF TWEAKABLE POLYNOMIALS

In this section we describe the theory behind the Cube Attack. Further in the text we define a technique, which is in itself very simple, but in its full variant could prove to be a very powerful way of linearising polynomials, if it was not for the computational complexity of the algorithm the equivalent condition yields. Nevertheless, we also present two simple variants, one of which proves in the next section to be quite powerful, when teamed up with the Cube Attack.

Supported by grant VF20102015006

Manuscript submitted on 11 September 2014, accepted 20 February 2015.
The authors are with Department of Algebra, Faculty of Mathematics and Physics, Charles University in Prague, 186 75 Praha 8, Sokolovska 83, Czech Republic, josef@barta.cz, hojsik@karlin.mff.cuni.cz

A. Tweakable polynomials

In this section, we introduce some notation and define the classes of polynomials we will be working with.

Throughout this paper, we denote by $[n]$ the set $\{0, 1, \ldots, n-1\}$ for any $n \in \mathbb{N}$. The set of all Boolean functions in n variables is denoted by $\mathcal{B}_n$, i.e. $\mathcal{B}_n = \{f : \mathbb{F}_2^n \to \mathbb{F}_2\}$.

Algebraic normal form (ANF) of a Boolean function f is its representation as a polynomial $f(x_0, \ldots, x_{n-1}) \in \mathbb{F}_2[x_0, \ldots, x_{n-1}]$ such that none of its monomials contain any variable in degree greater than one. For each Boolean function, there exist a unique algebraic normal form.

For $I \subseteq [n]$, we will use x_I to denote the monomial $\prod_{i \in I} x_i$. So for every Boolean function $f \in \mathcal{B}_n$ there exists a unique set $\mathcal{I} \subseteq \mathcal{P}([n])$ such that $f(x_0, \ldots, x_{n-1}) = \sum_{I \in \mathcal{I}} x_I$. We will write $x_I \in f$ if $I \in \mathcal{I}$.

Definition II-A.1. *Let $m, n \in \mathbb{N}$. We define set of secret variables $X = \{x_i; i \in [n]\}$ and set of public variables $Y = \{y_j; j \in [m]\}$.*

Later on, the secret variables will represent the secret key, while the public variables will represent the initialisation vector of a stream cipher, which is public and can be potentially set by an attacker.

In the rest of the paper we will use the ANF representation of Boolean functions and use the notation $\mathcal{B}[X]$, $\mathcal{B}[Y]$, $\mathcal{B}[X, Y]$ for Boolean functions (polynomials) in variables X, Y or $X \cup Y$ respectively.

Definition II-A.2. *We call a polynomial p tweakable, if $p \in \mathcal{B}[X, Y]$ and fully tweakable, if $p \in \mathcal{B}[Y]$.*

B. Basic Cube Attack

This section describes the basic principles of the Cube Attack in the same way it was done in [1]. For demonstration purposes we use fully tweakable polynomials.

Definition II-B.1. *[1] Let $p \in \mathcal{B}[Y]$ be a polynomial and $J \subseteq [m]$ a variable index subset. A superpoly of J in p is a polynomial $p_{S(J)} \in B[Y]$ such that*

$$p(Y) = y_J \cdot p_{S(J)}(Y) + q_J(Y) \qquad \text{(II-B.1)}$$

where $q = \sum_{J \not\subseteq J'} b_{J'} y_{J'}$, $b_{J'} \in \mathbb{F}_2$. We call y_J a maxterm, if the superpoly $p_{S(J)}$ is a linear, non-constant polynomial.

Note II-B.2. *The superpoly $p_{S(J)}$ does not contain any variables indexed by J.*

Example II-B.3. *Let $p \in \mathcal{B}[Y]$ be a polynomial*

$$p = y_0y_3y_4y_5 + y_1y_3y_4y_5 + y_0y_3y_5 + y_0y_2y_3y_4 + y_0y_1 + y_1y_2y_4 + y_2y_3y_4 + y_3y_4 + y_3y_5 + y_3 + y_4 \ .$$

We can factor out the monomial $y_J = y_3y_4y_5$ so we get

$$p = \underbrace{y_3y_4y_5}_{y_J} \cdot \overbrace{(y_0 + y_1)}^{p_{S(J)}(Y)} + \underbrace{+ y_0y_3y_5 + y_0y_2y_3y_4 + y_0y_1 + y_1y_2y_4 + y_2y_3y_4 + y_3y_4 + y_3y_5 + y_3 + y_4}_{q_J(Y) \text{ continued}}$$

In this case, y_J is a maxterm of J in p.

Definition II-B.4. *For an index subset $J \subseteq [m]$, $|J| = k$ we define a summation cube C_J as the set of k-tuples of variables $y_j : j \in J$ where all possible combinations of values of variables y_j are assigned. We can also understand C_J as a vector space $\mathbb{F}_2^k$ with information about indices of the variables. Hence we set $dim(C_J) = k$.*

Definition II-B.5. *[1] For every polynomial $p \in \mathcal{B}[Y]$ and for any k-dimensional summation cube C_J, $J \subseteq [m]$ we define $p_J := \sum_{v \in C_J} p_{|v}$ where $p_{|v}$ is a derived polynomial with $m - k$ variables $\{y_j : j \in [m] \setminus J\}$ and the variables indexed with J are assigned values from the k-tuple v.*

Now we can present a vital property of the superpoly of J in p, which is the main theorem in [1].

Proposition II-B.6. *[1] For any polynomial $p \in \mathcal{B}[Y]$ and variable subset J, $p_J = p_{S(J)}$.*

For our purposes, from now on, we shall call this technique of summing (partial) evaluations of a polynomial the **C-linearisation of fully tweakable polynomials**.

C. C-linearisation of tweakable polynomials

In this section we describe the C-linearisation (cube attack) on tweakable polynomials. We present a clear description of what makes a polynomial C-linearisable. In [1] this part was skipped, for they dealt with black box polynomials which demand a different approach than polynomials the explicit representation of which is known.

Definition II-C.1. *We call a polynomial $p \in \mathcal{B}[X, Y]$ C-linearisable, if there exists $J \subseteq [m]$ such that $p_J(X, Y = (1, ..., 1))$ is linear.*

For purposes of C-linearisation we present the following grouping of monomials: Let $p \in \mathcal{B}[X, Y]$ be a tweakable polynomial. Then we can write

$$p = \sum_{(I,J) \in \mathcal{I}} x_I y_J = \sum_{l \in L_p} l + \sum_{b \in B_p} b + \sum_{h \in H_p} h$$

where $\mathcal{I} \subseteq \mathcal{P}([n]) \times \mathcal{P}([m])$ and $B_p = \{x_I y_J \in p : |I| = 0\}$, $L_p = \{x_I y_J \in p : |I| = 1\}$ and $H_p = \{x_I y_J \in p : |I| > 1\}$.

The set B_p contains all monomials consisting purely of public variables and the free monomial. L_p contains all monomials consisting of exactly one secret and any number and combination of public variables. H_p consists of monomials with two or more secret variables. We can plainly see that $L_p \cup B_p \cup H_p$ contains all monomials of p.

Before we present the condition which describes precisely a C-linearisable polynomial, we present a simple lemma about C-linearisability:

Lemma II-C.2. *Let $p \in B[X, Y]$ be a tweakable polynomial. If $L_p = \emptyset$, then p is not C-linearisable.*

Proof. If there is no monomial that is linear in secret variables, there is definitely no monomial y_J, $J \subseteq [m]$, such that $p_{S(J)}$ is linear in secret variables. □

Now we can propose an equivalent definition of a C-linearisable tweakable polynomial:

Proposition II-C.3. *A tweakable polynomial $p \in \mathcal{B}[X, Y]$ is C-linearisable if and only if*

$$\exists x_i y_J \in L_p, \ : (\forall y_{J'} x_I \in H_p : y_J \not| y_{J'})$$

Proof. We shall prove the first implication by contradiction, the second directly:
"⇒": For contradiction, we assume that p is C-linearisable and $\forall y_J x_i \in L_p \exists y_{J'} x_I \in H_p : y_J | y_{J'}$. This implies that for every choice of J will in the superpoly $p_{S(J)}$ remain a monomial that is non-linear in secret variables, i.e. the superpoly will contain $\frac{y_{J'}}{y_J} x_I$ and $|I| \geq 2$ as $y_{J'} x_I \in H_p$. Thus the contradiction.
"⇐": We assume that $\exists y_J x_i \in L_p \ \forall y_{J'} x_I \in H_p : y_J \not| y_{J'}$. That implies y_J is a maxterm, which yields a superpoly $p_{S(J)}$ that is linear in secret variables. □

Example II-C.4. *In this example we rewrite the polynomial from previous example into the notation of the tweakable polynomials with distinguished secret and public variables. We shall have $m = n = 3$. So let $p \in \mathcal{B}[X, Y]$ be a tweakable polynomial:*

$$p = x_0y_0y_1y_2 + x_1y_0y_1y_2 + x_0y_0y_2 + x_0x_1y_0y_1 + x_1x_2y_1 + x_2y_0y_1 + y_0y_1 + y_0y_2 + y_0 + y_2$$

We can factor out $y_I = y_0y_1y_2$, so we obtain

$$p = y_0y_1y_2 \cdot (x_0 + x_1) + x_0y_0y_2 + x_0x_1y_0y_1 + x_1x_2y_1 + x_2y_0y_1 + y_0y_1 + y_0y_2 + y_0 + y_2$$

where $x_0 + x_1$ is the linear superpoly of $I = \{0, 1, 2\}$ in p and y_I is a maxterm.

D. T-linearisation of tweakable polynomials

Now we present T-linearisation, a technique we devised and describe to aid the C-linearisation to be as effective as possible when linearising a polynomial.

Definition II-D.1. *We call a polynomial $p \in \mathcal{B}[X, Y]$ T-linearisable, if*

$$\exists J \subseteq [m] : (\exists v \in C_J : p_{|v} \text{ is linear in secret variables})$$

In other words there exists a (partial) evaluation of the polynomial in public variables that results in $p_{|v}$ being linear in secret variables.

Proposition II-D.2. *Let $p \in \mathcal{B}[X, Y]$. If*

$$\exists l \in L_p,\ l = x_i y_J :\ (\forall h \in H_p\ \exists j \in [m] \setminus J : y_j | h)$$

then p is T-linearisable. Specially, we say that p is T1-linearisable.

Proof. If there is such l that the condition holds, then every $h \in H_p$ can be eliminated by setting respective $y_j = 0$ while the secret part of l will be kept in the polynomial by setting the public variables indexed by a smallest $J' \subseteq J$ such that $x_i y_{J'} \in L_p$ to one and the remaining ones to zero. □

We recall the polynomial from previous example to demonstrate the T1-linearisation:

Example II-D.3. *Let*

$$p = x_0y_0y_1y_2 + x_1y_0y_1y_2 + x_0y_0y_2 + x_0x_1y_0y_1 + x_1x_2y_1 + x_2y_0y_1 + y_0y_1 + y_0y_2 + y_0 + y_2$$

We can linearise this polynomial in secret variables by setting $y_1 = 0$, which gives us a non-constant polynomial, that is linear in secret variables

$$p_{|y_1=0} = x_0y_0y_2 + y_0y_2 + y_0 + y_2$$

and finally, by setting $y_0 = y_2 = 1$ we get

$$p_{|y_1=0, y_0=y_2=1} = x_0 + 1$$

which is a linear polynomial in secret variables only.

Corrolary II-D.4. *C-linearisability does not imply T1-linearisability.*

Proof. Consider polynomial $p = x_0y_0y_1 + x_0x_1y_1$. This polynomial is clearly not T1-linearisable, but it is obviously C-linearisable using $J = \{0, 1\}$. □

Clearly T1-linearisability is not a necessary condition for T-linearisability. Consider $p \in \mathcal{B}[X, Y]$,

$$p = x_0x_1y_0 + x_0x_1 + x_2 = (y_0 + 1)x_0x_1 + x_2.$$

This polynomial obviously is T-linearisable by setting $y_0 = 1$, but due to the monomial x_0x_1 T1-linearisation does not work here.

Definition II-D.5. *For any index subset $I \subseteq [n]$ and polynomial $p \in B[X, Y]$ we define the set of public monomials of p relative to I as $E_p(I) = \{y_J : x_I y_J \in p\}$ and the tweaking polynomial $p_{E_p(I)} \in \mathcal{B}[Y]$ as $p_{E_p(I)} = \sum_{y_J \in E_p(I)} y_J$.*

Using this we can express any tweakable polynomial $p \in \mathcal{B}[X, Y]$ as $p = \sum_{I \subseteq [n]} p_{E_p(I)} x_I$. In other words, $p_{E_p(I)}$ is the coefficient of x_I in p if seen as $p \in \mathcal{B}[Y][X]$.

Clearly, if we want to obtain a linear polynomial in X, we need to evaluate all $B[Y]$ coefficients of $x_I,\ |I| \geq 2$ to zero. In the following proposition we use this to present an equivalent definition of T-linearisation.

Proposition II-D.6. *Tweakable polynomial $p \in \mathcal{B}[X, Y]$ is T-linearisable if and only if*

$$\exists J \subseteq [m] :$$
$$(\exists v \in C_J : [\forall I \subseteq [n],\ |I| \geq 2 : p_{E_p(I)|v} = 0$$
$$\wedge\ (\exists I' \subseteq [n],\ |I'| = 1 : p_{E(I')|v} \neq 0)])$$

In other words, a tweakable polynomial is T-linearisable, if there is a solution to the system of polynomial equations yielded by $p_{E_p(I)}$'s, where $|I| \geq 2$ for that some of the $p_{E_p(I')},\ |I'| = 1$ does not evaluate to zero.

Proof. We prove the forward implication by contradiction, the backward directly.
"⇒": Let's assume that p is T-linearisable and

$$\forall a \in \mathbb{F}_2^m\ [(\exists I \subseteq [n], |I| \geq 2 : p_{E_p(I)}(a) = 1)$$
$$\vee\ (\forall I' \subseteq [n], |I'| = 1 : p_{E(I')}(a) = 0)]$$

That means that after any partial evaluation in public variables there either remains some monomial that is not linear in secret variables or the resulting polynomial is a constant. Hence the contradiction.
"⇐": If there exists such J and $v \in C_J$, then we can eliminate all the monomials that are non-linear in secret variables by partial evaluation in v and there is at least one monomial, that is linear in secret variables that remains in the polynomial after the partial evaluation. So $p_{|v}$ is a polynomial that is linear in secret variables, hence p is T-linearisable. □

This proposition yields a very compelling way of T-linearising a tweakable polynomial. First, we find the solutions for the system of polynomial equations defined by the $p_{E_p(I)}$'s for $|I| \geq 2$. Then we choose those solutions, for which there exist I' such that $|I'| = 1$ and $p_{E_p(I')}$ is non-zero after the partial evaluation. Naturally, this may be ineffective or even impossible, as shown in following example:

Example II-D.7. *Because our usual example polynomial is, as demonstrated, T-linearisable, for purposes of this example, we present a different polynomial, $q \in \mathcal{B}[X, Y]$, $q = x_0 + x_0x_1y_0 + x_1x_2y_0 + x_1x_2$. Obviously,*

$$q = x_0 + x_0x_1 \cdot y_0 + x_1x_2 \cdot (y_0 + 1)$$

which yields an equation system

$$y_0 + 1 = 0$$
$$y_0 = 0$$

which has no solution.

Because solving a system of polynomial equations over $\mathbb{F}_2$ in general is computationally ineffective, we present a simpler version, which we might be able to solve in a more efficient manner (given that the system of equations actually has a solution, otherwise we just conclude that there is none).

Corrolary II-D.8. *Let $p \in \mathcal{B}[X, Y]$. If*

$$\forall x_I y_J \in H_p : |J| \leq 1 \wedge$$
$$\wedge\ [\exists a \in \mathbb{F}_2^m : (\forall I' \subseteq [n], |I'| \geq 2 : p_{E(I')}(a) = 0) \wedge (\exists i \in [n] : p_{E(\{i\})}(a) = 1)]$$

then p is T-linearisable. Specially, we say that p is T2-linearisable.

This means, that a polynomial is T2-linearisable, if is T-linearisable and the tweaking polynomials for all I's, such that $|I| \geq 2$, are linear or constant.

Corrolary II-D.9. *T2-linearisability does not imply T1-linearisability.*

Proof. Polynomial $p = x_0y_0y_1 + x_0x_1y_0 + x_0x_1 = x_0y_0y_1 + x_0x_1(y_0 + 1)$ is clearly T2-linearisable (set $y_0 = y_1 = 1$), but not T1-linearisable. □

Corrolary II-D.10. *T1-linearisability does not imply T2-linearisability*

Proof. Polynomial $p = x_0y_0 + x_0x_1y_0y_1$ is clearly T1-linearisable (set $y_1 = 0$), but not T2-linearisable. □

E. TC-linearisation of tweakable polynomials

In this section we present TC-linearisation, our generalisation of Shamir's Cube Attack's C-linearisation.

In order to proceed to the definition of a TC-linearisable polynomial, we first define more general version of a maxterm.

Definition II-E.1. *Let $J \subseteq [m]$ be an index subset. We call the monomial y_J a T1-/T2-/T-maxterm, if the superpoly of J in p is a T1-/T2-/T-linearisable polynomial.*

Definition II-E.2. *Let $p \in \mathcal{B}[X, Y]$ be a tweakable polynomial. Then p is TC1-/TC2-/TC-linearisable if and only if there exists $J \subseteq [m]$ such that y_J is a T1-/T2-/T-maxterm respectively.*

Note, that TC1-/TC2-/TC-linearisation with $J = \emptyset$ equals T1-/T2-/T-linearisation since $p_{S(\emptyset)} = p$.

Example II-E.3. *In this example we use the same polynomial $p \in \mathcal{B}[X, Y]$ as previously:*

$$p = x_0y_0y_1y_2 + x_1y_0y_1y_2 + x_0y_0y_2 + x_0x_1y_0y_1 + x_1x_2y_1 + x_2y_0y_1 + y_0y_1 + y_0y_2 + y_0 + y_2$$

This polynomial is TC-linearisable, because we can either set $y_0 = y_2 = 1,\ y_1 = 0$ and obtain a non-constant linear polynomial $x_0 + 1$ by T-linearisation only or get for example $x_0 + x_1$ by summing over the cube defined by $J = \{0, 1, 2\}$. There is also the possibility of using a combination of both, as is made possible using TC-linearisation:

$$p = y_0 \cdot (x_0y_1y_2 + x_1y_1y_2 + x_0y_2 + x_0x_1y_1 + x_2y_1 + y_1 + y_2 + 1) + x_1x_2y_1 + y_2$$

and then we set $y_1 = 0 \wedge y_2 = 1$ to get x_0 as our linear polynomial. In this particular case it would of course be more efficient to use T-linearisation only, because x_0 is

Note II-E.4. *We call a tweakable polynomial $p \in \mathcal{B}[X, Y]$ TC1-/TC2-/TC-linearisable, if we can derive a polynomial that is linear in secret variables from it by using partial evaluation as described in the equivalent definitions of T1-/T2-/T-linearisation and cube summation presented as C-linearisation combined.*

Clearly, if a tweakable polynomial does not contain any monomials linear in secret variables, then we can not apply any of the presented techniques:

Corrolary II-E.5. *Let $p \in \mathcal{B}[X, Y]$ be a tweakable polynomial. If $L_p = \emptyset$, then p is not T-, C- or TC-linearisable.*

III. LINEARISING TRIVIUM KEYSTREAM POLYNOMIALS

With the previous one dealing with the underlying theory, this section describes the experimental part of the paper. At first we describe the cryptosystem we will be attacking and after that we present the attack itself. Since we wanted only to test our linearisation methods, we have not attempted the key-recovery, which is the aim of the full attack. In other words we show, how far we got with the techniques devised by us and compare them to the Cube Attack.

A. Trivium

Before we present the attack itself, we describe the cryptosystem we are about to attack. It is a reduced variation of a stream cipher Trivium [4]. At first we describe the original cryptosystem and after that we describe reduced variation we will attack.

Trivium is a very simple stream cipher with three non-linear feedback registers, 80 bit key and 80 bit initialisation vector (IV). The cipher produces a keystream $\{z_i\},\ z_i \in \mathbb{F}_2$ which is added to the plaintext to produce the ciphertext. The detailed description is to be found in [4].

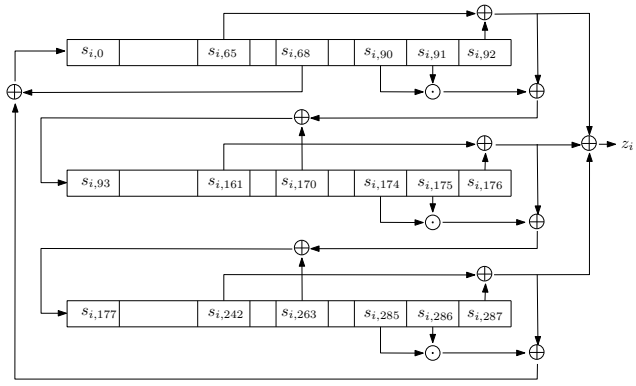

Fig. 1. Trivium cipher scheme

For our endeavour we had to reduce the original cryptosystem. Trivium-8 is a version of Trivium with shortened key and IV to 8 bits each. Aside from the shortened key and IV, it is the same Trivium as described above, so it has a 288 bit inner state. The length of the key and IV was chosen to be 8, i.e. $K = (k_0, \ldots, k_7)$ and $IV = (IV_0, \ldots, IV_7)$, since this should be enough to make the polynomials $p_i(X, Y)$ reasonably complex (interesting) while maintaining them small enough for the computation to be feasible with a standard PC.

In order to demonstrate the capabilities of the presented techniques we will assume that the keystream generation starts without any initialisation rounds (there are 1152 initialisation rounds in Trivium).

B. Attack description

As already mentioned, we will use the described linearisation methods to attack Trivium-8, a Trivium reduced in key and IV bits.

We assume that an attacker has access to Trivium-8 with fixed unknown key K. He can repeatedly choose the IV and

obtain the keystream $z_i = p_i(K, IV)$ (chosen IV attack). His goal is to find K by solving the respective equation systems.

For this purpose we need to compute the polynomials $g_{i,j}(X, Y)$ for all inner state bits and use them to compute the polynomials $p_i(X, Y)$ expressing the keystream bits. To obtain the actual values of z_i, we need an implementation of Trivium-8 that computes the keystream bits from the K and IV.

Attack on black-box polynomials, i.e. polynomials explicit representation of which is unknown to us, is out of scope of this paper and subject of future work.

1) Attack using T-linearisation: This part is pretty straightforward. Assume that $IV \in \mathbb{F}_2^m$ and $i \geq 0$ such that $p_i(X, IV)$ is linear in X. Then we get the value of $z_i = p_i(K, IV)$ and form a linear equation $p_i(X, IV) = z_i$.

If the J from the definition of T-linearisation is such that $J \neq [m]$, we add to the values of respective $v \in C_J$ values for the remaining public variables arbitrarily.

2) Attack using C-linearisation: Assume that for an $J \subseteq [m]$ the y_J is a maxterm of $p_i(X, Y)$ so by assigning ones to all y_j, $j \notin J$ we obtain a linear superpoly. We set $U = \{(u_0, \ldots, u_{m-1}) \in \mathbb{F}_2^m; u_j = 1 \ \forall j \notin J\}$ and obtain $z_i(a) = p_i(K, u)$ for all $u \in U$. The equation obtained by C-linearisation is then $p_{S(J)}(X, v) = \sum_{u \in U} z_i(u)$ where v is an element of U. Note that $p_{S(J)}(X, Y)$ can depend only on public variables with indices not in J, hence the equation does not depend on the particular choice of $v \in U$.

3) Attack using TC-linearisation: In C-linearisation, we set all public variables with indices not in J to one. In TC-linearisation, we assume that there exists $w \in C_{[m]\setminus J}$, such that $p_{S(J)|w}$ is linear. I.e. when summing over all k-tuples from the cube C_J, we have to extend each k-tuple with the T-linearising bits for partial evaluation, that remain fixed during the whole summing.

C. Experimental results

In this section we present results we got when we applied the presented linearisation techniques on Trivium-8 as presented in respective section.

Since solving a system of linear equations is simple, in our experiments we shall concentrate on whether we can actually obtain any linear, non-constant polynomials, from which we could build such.

For polynomial multiplication to build the representation we used a variant of algorithm from [2].

1) C-linearisation: The C-linearisation proved to be, as expected, very effective. We could use it to linearise the polynomials representing keystream bits with indices up to 609. It is important to note, that by not all keystream polynomials up to the 610^{th} are C-linearisable. Moreover, it is only effective up to the 582^{nd} keystream bit, because after that the linearising cube has dimension 8. making the linearisation uneffective.

2) T1-linearisation: T1-linearisation did not prove to be especially effective. We could T1-linearise polynomials representing up to the 361^{st} keystream bit.

This is actually a result we expected: If T1-linearisation would be effective on the polynomials, even if in relatively early stages of the initialisation, it would mean, that there would be no monomial $x_I y_\emptyset$, $I \subseteq [n]$. This is highly improbable though, because it would mean that none of a set of 2^n monomials would be present, which happens with probability of 2^{-2^n} if dealing with a random polynomial, which we assume the polynomials in the later stages of initialisation to be.

3) T2-linearisation: In spite of T2-linearisation using an approach that significantly differs from that used in T1-linearisation, we managed to linearise keystream polynomials only up to p_{361}. It is easy to see, why this technique was not any more successful: it demands, that for the polynomial $p \in \mathcal{B}[X, Y]$ that we are attempting to linearise every monomial in H_p has degree at most one in public variables. In fact then, this is a surprisingly good result.

4) TC1-linearisation: With TC1-linearisation, the situation is a bit more complicated. It is at least as effective as C-linearisation, but it could at some point prove to be able to reduce the cube and therefore the complexity of the linearisation. This happened only when the polynomial was T1-linearisable, so this technique turned out to be a bit disappointing.

5) TC2-linearisation: As we already hinted, TC2-linearisation is the technique, that really does improve the C-linearisation. We managed to linearise polynomials representing the keystream of Trivium-8 with indices up to 622, which is slightly more, than we managed using C-linearisation (609).

In the figure below, we present our results graphically. The dashed line denotes, where there are only such C-linearisable polynomials that are linearisable with a cube of dimension 8 only. Clearly, we consider the results achieved with TC2-linearisation to be a great success.

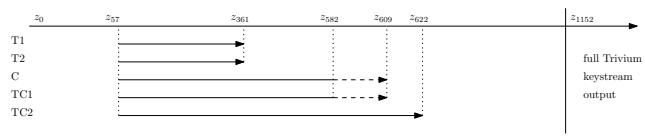

Fig. 2. Range of effectiveness of linearisation techniques

IV. CONCLUSION

In this paper we presented a detailed description of Cube Attack devised by Adi Shamir et. al. and its generalisation that proved to be slightly more effective when tested on key- and IV-reduced Trivium variant. However, none of these techniques is advanced enough to linearise keystream polynomials after full 1152 initialisation rounds.

REFERENCES

[1] Itai Dinur and Adi Shamir: Cube Attacks on Tweakable Black Box Polynomials, Cryptology ePrint Archive, Report 2008/385, 2008.
[2] Subhabrata Samajder and Palash Sarkar: Fast Multiplication of the Algebraic Normal Forms of Two Boolean Functions, International Workshop on Coding and Cryptography 2013, Bergen (Norway), 2013.
[3] Claude Carlet: Boolean Functions for Cryptography and Error Correcting Codes, chapter of the monography "Boolean Models and Methods in Mathematics, Computer Science, and Engineering", Cambridge University Press, 2006.
[4] Christophe de Cannière and Bart Preneel: Trivium, eSTREAM: the ECRYPT Stream Cipher Project, 2005.

Josef Bárta Josef Bárta has received bachelor degree in mathematics from Charles University in Prague and has been accepted to continue his studies at the Royal Holloway University of London.

His research interests include symmetric cryptanalysis, lightweight cryptography, authentication protocols and smart cards. He is currently working as a software engineer and preparing for his studies at the Royal Holloway.

Michal Hojsík Michal Hojsík has received master degree in mathematics from Charles University in Prague and PhD in computer science from University of Bergen, Norway.

His primary research interests are block ciphers and stream ciphers and lately also lightweight cryptography and authentication schemes. He is currently working as a cryptographic engineer.

CALL FOR PAPERS

Special Issue on Advanced wireless and mobile technologies and services

We have been witnessing a rapid development of wireless and mobile technologies and services during the past two decades. 4G mobile services are penetrating and mobile access is becoming an increasingly important way for accessing the Internet and it is expected to become the dominant one. The progress continues. 5G mobile systems are underway. Although many of the new technologies have already been incorporated in practical systems, there is still enough room for research and experimentation, in particular in the areas of cognitive radio, self-organizing networks, M2M communications, cross-layer optimization, just to name a few.

Topics of interest include but are not limited to:

- Cross-layer issues in wireless networks
- Cognitive radio for wireless communications
- QoS and resource allocation in wireless networks
- Mobile/wireless networks modeling and simulation
- Localization and positioning in wireless scenarios
- Topology control, self-organizing wireless networks
- Tools for modeling and analysis of wireless systems
- Personal wireless communications beyond 5G
- Software defined wireless networks and re-configurability
- M2M communications and the Internet of Things
- Storage, smart caching, and cloud for wireless
- Wireless social networks, participatory computing
- Molecular and nano-scale wireless communications
- New disruptive concepts for wireless systems

Selected papers from the European Wireless 2015 conference, *http://ew2015.european-wireless.org* will be invited to submit extended journal versions of their papers to this Special Issue, but high quality papers are welcome from open call too. Submissions will be peer reviewed according to the journal policy and international standards. Instructions for authors can be found on the journal website: *www.infocommunications.hu*.

Deadline for submission of manuscripts: June 30, 2015.

Tentative publication date: end of September, 2015.

Guest Editors:

SÁNDOR IMRE [M'93] is Professor and Head of Dept. of Networked Systems and Sevices at the Budapest University of Technology (BME). He obtained Dr. Univ. degree in in probability theory and statistics 1996, Ph.D. degree in 1999 and DSc degree from the Hungarian Academy of Sciences in 2007. He is Chairman of Telecommunication Scientific Committee of Hungarian Academy of Sciences. He participates on the Editorial Board of two journals: Infocommunications Journal and Hungarian Telecommunications. He was invited to join the Mobile Innovation Centre as R&D director in 2005. His research interests include mobile and wireless systems, quantum computing and communications. Especially he has contributions on different wireless access technologies, mobility protocols, security and privacy, reconfigurable systems, quantum computing based algorithms and protocols.

HASSAN CHARAF received his PhD in 1998. He is an Associate Professor and fellow at the Department of Automation and Applied Informatics at the Budapest University of Technology and Economics. He is the head of the IT group. As an outstanding figure in teaching, research and development, he is in key positions at several organizations at the university. His research fields are: distributed systems, cloud computing, multiplatform application development methods, software modeling and data technologies.