Searching Cubes for Testing Boolean Functions and Its Application to Trivium

Meicheng Liu, Dongdai Lin and Wenhao Wang
State Key Laboratory of Information Security
Institute of Information Engineering
Chinese Academy of Sciences
Beijing 100093, P. R. China
Email: meicheng.liu@gmail.com

***Abstract*—In this paper, we describe a sub-maximal degree monomial test and propose a heuristic algorithm for searching favourable cubes, for testing Boolean functions formed by stream ciphers. We apply them to Trivium, and mount a distinguisher on Trivium reduced to 839 rounds with 2^{37} complexity, which is so far the best distinguisher on reduced Trivium.**

***Index Terms*—Stream cipher, Trivium, distinguisher.**

I. Introduction

In a stream cipher, a pseudo-random digit stream called keystream is generated independently of the plaintext and ciphertext, and then combined with the plaintext to encrypt or the ciphertext to decrypt. In the most common form, binary digits are used, and the keystream is combined with the plaintext using the exclusive-or (XOR) operation. This is called a binary additive stream cipher. There is abundant theoretical knowledge on stream ciphers, and various design principles for stream ciphers have been proposed and extensively analyzed. For instance, in the modern stream ciphers, a secret key accompanied with a public initialization vector (IV) are used to avoid time-memory trade-off attacks and time-memory-data trade-off attacks. Generally, the security of a stream cipher is closely connected to how well this sequence of bits resembles a truly random sequence.

In a distinguishing attack, the cryptanalyst tries to detect a behaviour of the generated sequence giving evidence that this is not a truly random sequence. In [7], Filiol introduced a test to evaluate the statistical properties of symmetric ciphers using the number of the monomials in the Boolean functions that simulate the action of a given cipher. In [11], Saarinen extended these ideas to a chosen IV statistical attack, called the d-monomial test, and used it to find weaknesses in several stream ciphers. In [6] Englund *et al.* generalized Saarinen's idea and proposed a framework for chosen IV statistical attacks using a polynomial description. Their basic idea is to select a subset of IV bits as variables. Assuming all other IV values as well as the key being fixed, one can write a keystream symbol as a Boolean function of the selected IV variables. By running through all possible values of these bits and creating a keystream output for each of them, the truth table of this Boolean function is created. It is hoped that this Boolean function has some statistical weaknesses that can be detected. Based on these observations, Englund *et al.* proposed two new tests, called the monomial distribution test and the maximal degree monomial test, and then applied them on some stream cipher proposals of the eSTREAM project. In particular, they experimentally detected statistical weaknesses in the keystream of Grain-128 with IV initialization reduced to 192 rounds as well as in the keystream of Trivium using an initialization reduced to 736 rounds.

In [1] Aumasson *et al.* introduced cube tester to mount distinguishers or to detect nonrandomness in cryptographic primitives, which combines the cube attack [4] with efficient property tester, and showed a distinguisher on 790-round Trivium using a cube of size 30. Stankovski [12] proposed a greedy distinguisher for 806-round Trivium in 2^{44} operations. Recently Vardasbi *et al.* [14] used a so-called multi-χ^2 test for testing the monomials of suplypolys over a set of cubes and mounted distinguishing attack on 830-round Trivium with a complexity 2^{39}. This is so far the best result of distinguishing attack on reduced-round Trivium. Besides, Knellwolf *et al.* [10] used conditional differential cryptanalysis to distinguish 868-round and 961-round Trivium respectively for 2^{31} and 2^{26} weak keys both with a complexity 2^{25}.

The purpose of this paper is to further study chosen IV statistical analysis of stream ciphers by considering a keystream bit as a Boolean function. A sub-maximal degree monomial test combined the Moebius transform is described. The test runs in 2^n cipher operations and requires 2^n bits memory. A heuristic algorithm for searching favourable cubes for testing Boolean functions is also proposed. Applying this algorithm to Trivium, a cube of size 28 is found, whose superpoly at round 790 is constant and which can also be used to detect the unbalancedness of the output of round 806. The results are already better than [4]. Using the tool of Moebius transform and sub-maximal degree monomial test, the results are further improved. Table I summarizes our results on Trivium, comparing them with the previous distinguishing attacks. As shown in Table I, all our announced complexities are lower than previous attacks. All our announced attacks are fully tested and verified.

This work was supported by the National 973 Program of China under Grant 2011CB302400, the National Natural Science Foundation of China under Grant 61303258, and the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDA06010701.

TABLE I
SUMMARY OF THE BEST KNOWN DISTINGUISHING ATTACKS ON TRIVIUM

	#Rounds	Time
[6]	736	2^{33}
[14]	760	2^{26}
[1]	785	2^{27}
This paper	789	2^{26}
[1]	790	2^{30}
[14]	795	2^{31}
[12]	806	2^{44}
This paper	807	2^{31}
[14]	830	2^{39}
This paper	839	2^{37}

II. PRELIMINARIES

A. Hypothesis Testing

A hypothesis test is a method of statistical inference used for testing a statistical hypothesis. A test result is called statistically significant if it has been predicted as unlikely to have occurred by chance alone, according to a threshold probability—the significance level. Hypothesis tests are used in determining what outcomes of a study would lead to a rejection of the null hypothesis for a pre-specified level of significance. In the Neyman-Pearson framework, the process of distinguishing between the null hypothesis and the alternative hypothesis is aided by identifying two conceptual types of errors (type 1 and type 2), and by specifying parametric limits on e.g. how much type 1 error will be permitted.

Pearson's χ^2 test is used to assess two types of comparison: tests of goodness of fit and tests of independence. What we are interested in is the former. A test of goodness of fit establishes whether or not an observed frequency distribution differs from a theoretical distribution.

Let the probabilities of various classes in a distribution be p_i $(1 \leq i \leq k)$ with observed frequencies n_i $(1 \leq i \leq k)$. The quantity

$$\chi_s^2 = \sum_{i=1}^{k} \frac{(n_i - Np_i)^2}{Np_i}$$

is therefore a measure of the deviation of a sample from expectation, where N is the sample size. Pearson proved that the limiting distribution of χ_s^2 is a chi-squared distribution [13]. The chi-squared statistic can then be used to calculate a p-value by comparing the value of the statistic to a chi-squared distribution. The number of degrees of freedom s is equal to the number of classes k minus the reduction in degrees of freedom.

B. Boolean functions

Let $\mathbb{F}_2$ denote the binary field and $\mathbb{F}_2^n$ the n-dimensional vector space over $\mathbb{F}_2$. An n-variable Boolean function is a mapping from $\mathbb{F}_2^n$ into $\mathbb{F}_2$. Denote by $\mathbf{B}_n$ the set of all n-variable Boolean functions. An n-variable Boolean function f can be uniquely represented as its truth table, i.e., a binary string of length 2^n,

$$f = [f(0,0,\cdots,0), f(1,0,\cdots,0), \cdots, f(1,1,\cdots,1)].$$

An n-variable Boolean function f can also be uniquely represented as a multivariate polynomial over $\mathbb{F}_2$,

$$f(x_1,\cdots,x_n) = \bigoplus_{c=(c_1,\cdots,c_n)\in\mathbb{F}_2^n} a_c \prod_{i=1}^{n} x_i^{c_i},\ a_c \in \mathbb{F}_2,$$

called the algebraic normal form (ANF). The algebraic degree of f, denoted by $\deg(f)$, is defined as $\max\{\mathrm{wt}(c) \mid a_c \neq 0\}$, where $\mathrm{wt}(x)$ denotes the Hamming weight of x.

1) Moebius transforms: There is a good relationship between a Boolean function's truth values and its ANF. The coefficients a_c's can be represented as

$$a_c = \bigoplus_{b\in\mathbb{F}_2^n, \mathrm{supp}(b)\subset\mathrm{supp}(c)} f(b), \tag{1}$$

where $\mathrm{supp}(x)$ denotes the set of indexes $\{i|x_i \neq 0\}$. This is called the Moebius transform or the Reed-Muller transform of the Boolean function f. A fast Moebius transform algorithm can be found in [9]. The algorithm has a time complexity of $n2^{n-1}$ bit operations and uses a memory of 2^n bits. The 32-bit implementation presented in [9] performs roughly 32 times less operations.

Data: Truth table S of Boolean function f
Result: Coefficients of the ANF of f

```
for i from 0 to n − 1 do
    Sz ← 2^i, Pos ← 0
    while Pos < 2^n do
        for j from 0 to Sz − 1 do
            S[Pos+Sz+j] ← S[Pos+j]⊕S[Pos+Sz+j]
        end
        Pos ← Pos + 2Sz
    end
end
return S
```

Algorithm 1: Moebius transform algorithm

C. Cube attacks and cube testers

A keystream bit of a stream cipher with an m-bit secret key $K = (k_1, k_2, \cdots, k_m)$ and an l-bit IV $V = (v_1, v_2, \cdots, v_l)$ can be seen as a Boolean function $f(K, V)$ with K and V as variables. The Boolean function f are usually analyzed under a fixed key K for mounting a successful attack on the cipher. The function $f(K, V)$ with valued K is a Boolean function on V, denoted by $f_K(V)$, which can be written by

$$f_K(V) = \bigoplus_{c=(c_1,\cdots,c_l)\in\mathbb{F}_2^l} g_c(K) \prod_{i=1}^{l} v_i^{c_i},$$

where g_c is a Boolean function on K. By (1) we can see that

$$g_c(K) = \bigoplus_{b\in\mathbb{F}_2^n, \mathrm{supp}(b)\subset\mathrm{supp}(c)} f_K(b). \tag{2}$$

The indexes set $\mathrm{supp}(c)$ or the set of variables with indexes inside of $\mathrm{supp}(c)$ is called a cube in [4]. In a more general

case, one can fix the IV variables V_0 outside of a cube V_c, and rewrites $f(K,V)$ as

$$f_{K,V_0}(V_c) = g_c(K,V_0) \prod_{i\in \mathrm{supp}(c)} v_i \oplus \bigoplus_{\mathrm{supp}(c')\subsetneq \mathrm{supp}(c)} g_{c'}(K,V_0) \prod_{i\in \mathrm{supp}(c')} v_i, \quad (3)$$

where g_c is a Boolean function on K and V_0. The function g_c is called a superpoly in [4]. By (1) it follows that

$$g_c(K,V_0) = \bigoplus_{b\in\mathbb{F}_2^n} f_{K,V_0}(b), \quad (4)$$

where n equals the cardinality of $\mathrm{supp}(c)$. As a matter of fact, using the Moebius transform described previously, the values of $g_c(K,V_0)$ and all $g_{c'}(K,V_0)$'s for a fixed K and V_0 can be computed at once.

In a chosen-IV key recovery attack, the adversary can control the values of V to exploit the information of g_c and then recover the secret key K, e.g. cube attacks [4]. In a chosen-IV distinguishing attack, the attacker can control the values of V to exploit the information of g_c and then distinguishing the cipher from a random one, e.g. cube testers [1]. The target of cube attacks is finding a set of linear functions g_c's on the secret key K and recovering the key by solving this linear system. The goal of cube testers is to mount distinguishers or detect the nonrandomness of g_c.

III. Sub-maximal Degree Monomial Test

In [11], Saarinen proposed d-monomial test for testing the balancedness of coefficients of monomials with the same degrees, and used it to find weaknesses in several stream ciphers. In [6] Englund *et al.* generalized Saarinen's idea and proposed two tests, called the monomial distribution test and the maximal degree monomial test, and applied them on some stream cipher proposals of the eSTREAM project. Cube testers further generalizes these tests. We refer to [1] for more details of cube testers.

In this paper, we look inside the sub-maximal degree monomial test combined with the Moebius transform. The Moebius transform was suggested in [5] by Dinur and Shamir to be used to compute every single subcube of a large cube at once, and was used in [8] by Fouque and Vannet to optimize cube attacks and mount key recovery attack to 784 and 799 Rounds of Trivium with a complexity 2^{39} and 2^{62} respectively, which is so far the best key recovery attack on reduced-round Trivium.

The reasons we are interested in sub-maximal degree monomial test are twofold. On one hand, we can compute the coefficients of these monomials at once. It costs $n2^{n-1}$ bit operations and a memory of 2^n bits for the Moebius transform. To obtain the truth table of the Boolean function f_{K,V_0} generated by the cipher, it needs 2^n cipher operations, which costs much more time than the Moebius transform. While for separately testing n monomials with algebraic degree $n-1$, it needs $n2^{n-1}$ cipher operations. This is the reason why we do not use maximal degree monomial test (noting that a monomial with algebraic degree $n-1$ achieves maximum degree over a cube of size $n-1$). On the other hand, the nonrandomness appears more likely in the monomials with sub-maximal degree than the monomials with lower degrees, especially for the ciphers based on nonlinear feedback shift registers (NLFSRs).

Given N samples $z=(z_0,z_1,\cdots,z_{N-1})$ of coefficients of monomials with degree d, our two hypotheses are

H_0: z is sampled from a random function;
H_1: z is not sampled from a random function.

By Pearson's χ^2 test as described in Section II-A, we calculate the quantity

$$\chi^2 = \frac{(n_0-\frac{N}{2})^2}{\frac{N}{2}} + \frac{(n_1-\frac{N}{2})^2}{\frac{N}{2}} = 2\cdot\frac{(n_1-\frac{N}{2})^2}{\frac{N}{2}},$$

where n_0 and n_1 respectively denote the number of zeros and ones appearing in z. There is only 1 degree of freedom. The hypothesis is rejected if the test statistics χ^2 is greater than the tabulated $\chi^2(1-\alpha;1)$ value, for a significance level α.

We describe as blew an algorithm for sub-maximal degree monomial test, where the key of the cipher is unknown but fixed. In our case, $N=n$ and $d=n-1$.

Data: An n-element subset V_c of IV variables, i.e., a cube of size n
Result: Cipher or Random

Set up truth table S of length 2^n on V_c by 2^n enquiries on the cipher with fixed V_0
Call Algorithm 1 with S as inputs
$crt \leftarrow 0$
for i *from* 0 *to* $n-1$ **do**
  $Pos \leftarrow 2^n - 2^i - 1$
  if $S[Pos]=1$ **then**
    $crt \leftarrow crt+1$
  end
end
$\chi^2 \leftarrow 2\cdot\frac{(crt-\frac{n}{2})^2}{\frac{n}{2}}$
if $\chi^2 > \chi^2(1-\alpha;1)$ **then**
  return Cipher
else
  return Random
end

Algorithm 2: Sub-maximal degree monomial test

Noting that the approximation to the chi-squared distribution breaks down if expected frequencies are too low. When the degree of freedom equals 1, the approximation is not reliable if expected frequencies are below 10. This yet is not an issue in our test, since to mount distinguishers we usually need a cube of size $n \geq 20$. Even for $n < 20$, a better approximation can be obtained by reducing the absolute value of each difference between observed and expected frequencies by 0.5 before squaring; this is called Yates's correction for continuity [15].

As shown previously, Algorithm 2 requires 2^n cipher operations and 2^n-bit memory. To mount a successful test for most keys, it is sufficient to require the success rate of mounting a distinguisher for a random key to be greater than 0.5.

IV. Searching Cubes for Testers

Though there are various methods for testing the ANFs of Boolean functions formed by ciphers, there are few general methods in the literatures available for searching favourable cubes to optimize the testers. In [4] Dinur and Shamir used random walk to search cubes for setting up linear relationships between the key bits. In [2] Aumasson *et al.* programmed a evolutionary algorithm to search good cubes that maximize the number of rounds after which the superpoly is still not balanced. These methods both directly manipulate large cubes, and thus the outcomes and the searching time depend on the initial cubes, more or less. Stankovski [12] proposed greedy bit set algorithm with 2^{n+c} complexity. Fouque and Vannet [8] described an empirical reduction of density of the ANFs for Trivium, where the union of two disjoint subcubes was used and the principles of favourable subcubes were also discussed.

In this section, we propose a heuristic method for searching favourable cubes optimizing statistical testers, especially for NLFSR-based ciphers. Our method is different from the methods proposed in [4], [2], [12], but similar to Fouque and Vannet's. Our ideas are based on exhaustive search on small cubes and iteratively using the union of favourable subcubes.

For the phase of exhaustion, we estimate the maximum size n for exhaustive search by maximizing $2^n\binom{l}{n}$ within a reasonable time. The complexity of this phase is roughly $\mathcal{O}(2^{n_{\min}}\binom{l}{n_{\min}})$. The exact running time highly depends on the determiner $\mathcal{D}_0$, which is also a crucial part of the algorithm. An NLFSR-based cipher usually updates a few bits of its internal state at each round. Considering these bits as Boolean functions on the secret key and the IV, the corresponding superpolys over a cube c are always zeros at the first rounds, but become non-zeros after some rounds. We record the round at which for the first time at least one of its superpolys is non-zero, and use $\mathcal{D}_0$ to determine whether accepting it as a favourable subcube. There are some useful principles for $\mathcal{D}_0$. For instance, at the recorded round, the non-zero superpoly should be non-zero constant, and there should be a sharp gap r_n between this round and the maximum round where the cubes of the small size reach. These restrictions assure that the monomial x^c appears latter than most of cubes of the same size and does not divide a bigger monomial when it appears for the first time.

For the phase of union, it not necessary to require either the two subcubes c_1 and c_2 or the two sets C_1 and C_2 are disjoint. Our experiments on Trivium show that the union of joint subcubes usually goes further compared with the union of two disjoint subcubes, under a same size. The classification of C and the determiners $\mathcal{D}_i$ are crucial for efficient selection of cubes. Taking $C_1 = C_2 = C$ may be a good choice. In this case, however, $\mathcal{D}_i$ should fast rule out a large number of cubes. For a small size n, e.g. $n \leq 16$, $\mathcal{D}_i$ could be the same as $\mathcal{D}_0$. For a large size n, the goal of $\mathcal{D}_i$ is to select cubes that maximize the number of rounds after which the superpolys are constants, where it does not require the superpolys either being non-zero constants or appearing for the first time. The main functions may be generated from the state bits or the keystream bits, when we mention superpolys. We will illustrate our points by how we treat Trivium in the next section.

Data: Round function R of a cipher
Result: A set of cubes for testing the cipher

```
/* exhaustively searching for small cubes */
n_min ← maximum size of cubes for exhaustion
C ← empty set
for n from 1 to n_min do
    for each cube c of size n do
        if D_0(R, c) = 1 then
            add c to C
        end
    end
end
/* union of small cubes */
n_max ← maximum size of cubes for searching
i ← 1
while n < n_max do
    classify C into two classes C_1 and C_2
    C ← {c_1 ∪ c_2 | c_1 ∈ C_1, c_2 ∈ C_2}
    for c ∈ C do
        if D_i(R, c) = 0 then
            remove c from C
        end
    end
    n ← maximum size of c ∈ C
    i ← i + 1
end
return C
```

Algorithm 3: Searching cubes for testers

Why not use the unions of three or more cubes? On one hand, checking whether a large cube is desired or not is rather time consuming. On the other hand, there are much more choices for combining three or more subcubes than combining two subcubes. Therefore, we get rid of the unions of three or more cubes, unless there is an extremely efficient determiner for selecting such unions.

Furthermore, it is possible that using the Moebius transform gives a further improvement.

V. Experiments on Trivium

Trivium, a stream cipher designed by Cannière and Preneel [3], has been selected as one of the portfolio for hardware ciphers (Profile 2) by the eSTREAM project. Though Trivium is designed to provide a flexible trade-off between speed and gate count in hardware, it also provides extremely efficient software implementation. It generates up to 2^{64} bits of keystream by using an 80-bit key and an 80-bit initialization vector (IV). Trivium contains a 288-bit internal state and uses 1152 rounds

of initialization, and 66 rounds of Trivium can be implemented in parallel. In each round, three bits of its internal state are updated. It is the simplest eSTREAM submission, yet till now there is no successful attack faster than exhaustive search on its full version. We refer to [3] for more details of Trivium.

In this section, we first show the application of Algorithm 3 to Trivium, and then apply sub-maximal degree monomial test described as Algorithm 2 to testing Trivium.

A. Searching cubes for Trivium

For the phase of exhaustion on Trivium, we set $n_{\min} = 6$. Trivium has an 80-bit IV, so we need to test $\binom{80}{6} \approx 2^{28.2}$ cubes of size 6, $\binom{80}{5} \approx 2^{24.5}$ cubes of size 5, and all the cubes of smaller sizes. Use $\mathcal{D}_0$ as suggested previously, in which 100 random K and V_0 are tested for each cube c, and set $r_n = 14$ for $n \leq 6$. In this case, most of cubes have been eliminated after a few tests rather than 100 tests are done. Using a computer with a single CPU core running at 3.4 GHz, it costs about one hour for testing all the cubes of size 6 and less than two minutes for size $n \leq 5$. The maximum rounds for $n = 4, 5, 6$ turn out to be respectively $422, 472, 497$, where the number of rounds starts from 0. We discard the cubes of size less than 4.

For the phase of union, we set $n_{\max} = 28$. Use $C_1 = C_2 = C$, and discard the union c of c_1 and c_2 if c contains two adjacent indexes, e.g. $\{0, 1\}$. This is based on the structure of Trivium, whose nonlinear terms are generated by two adjacent bits, and it can efficiently eliminate a large number of c. Then use $\mathcal{D}_0$ to filter the rest cubes for $n \leq 12$, use a modified $\mathcal{D}_0$ for $n \leq 24$, where the condition of superpoly being non-zero constant is removed, and use another modified $\mathcal{D}_0$ for $n > 24$, where the keystream bits rather than updated bits of internal state are the targets.

Finally, we obtain a unique cube of size 28 as below,

$$\{0, 2, 4, 6, 8, 10, 12, 15, 17, 19, 21, 23, 25, 28, 30, 34, 36, 38, 42, 44, 47, 49, 51, 53, 57, 64, 71, 79\}.$$

Fixing the IV bits outside of this cube to be zeros, after testing 100 random keys, the module 2 summations of the first output bits of 790-round Trivium over this cube are all zeros, and at the same time we can detect unbalancedness of 806-round Trivium. Combining the observations on the frequencies of indexes in picked cubes of sizes 26 and 27 from the output C of Algorithm 3, we obtain the cubes of sizes 31, 34 and 37 as below,

$$\{0, 2, 4, 6, 8, 10, 12, 15, 17, 19, 21, 23, 25, 28, 30, 32, 34, 36, 38, 41, 45, 47, 51, 53, 56, 60, 64, 67, 71, 73, 79\},$$
$$\{0, 2, 4, 6, 8, 10, 12, 15, 17, 19, 21, 23, 25, 28, 30, 32, 34, 36, 38, 41, 43, 45, 47, 49, 51, 53, 56, 58, 60, 64, 66, 68, 71, 79\},$$
$$\{0, 2, 4, 6, 8, 10, 12, 15, 17, 19, 21, 23, 25, 28, 30, 32, 34, 36, 38, 41, 43, 45, 47, 49, 51, 53, 56, 58, 60, 62, 64, 66, 68, 71, 73, 75, 79\}.$$

We tested 100 random keys for sizes 31 and 34 on a computer with a single CPU core at 3.4 GHz. It costs less than 12 hours for size 31 and about 3.5 days for size 34. We also tested 96 random keys for size 37 on a computer with 8 CPU cores at 2.6 GHz. The computations take about 5.5 days. The results are listed in Table II, where UB is short for unbalanced. The cube of size 26 is the one of size 28 by the exclusion of indexes $\{8, 71\}$, using the tool of Moebius transform.

TABLE II
CUBES AND THEIR SUPERPOLIES OF REDUCED TRIVIUM

#Rounds	789	790	806	807	812	817	824	839
Cube size	26	28	28	31	31	34	34	37
Superpoly	0	0	UB	0	UB	0	UB	0^a

[a] It is almost zero constant.

B. Sub-maximal degree monomial tests on Trivium

In the tests, we set the significance level $\alpha = 10^{-3}$. Noting that at least 10 tries should be done to achieve a significance level of 2^{-10} even if the superpoly is constant. It means that the complexity is at least $10 \cdot 2^n$ when using a single cube of size n and testing a single bit each time. We verified for $n = 26, 31, 34, 37$ that the complexity is cut down to 2^n by using Algorithm 2, at the cost of decreasing success rate. However, as shown in Table III, the success rates are always greater than 0.5 when the superpolys are zero constants or almost zero constants.

TABLE III
COMPLEXITIES OF TESTS ON REDUCED TRIVIUM

#Rounds	789	807	817	824	839
Time	2^{26}	2^{31}	2^{34}	2^{34}	2^{37}
Success rate	0.58	0.67	0.78	0.24	0.56

REFERENCES

[1] J. Aumasson, I. Dinur, W. Meier, A. Shamir: Cube Testers and Key Recovery Attacks on Reduced-Round MD6 and Trivium. FSE 2009: 1–22

[2] J. Aumasson, I. Dinur, L. Henzen, *et al.*: Efficient FPGA Implementations of High-Dimensional Cube Testers on the Stream Cipher Grain-128. IACR Cryptology ePrint Archive 2009: 218 (2009)

[3] C. D. Cannière, B. Preneel: *Trivium*. The eSTREAM Finalists 2008: 244–266

[4] I. Dinur, A. Shamir: Cube Attacks on Tweakable Black Box Polynomials. EUROCRYPT 2009: 278–299

[5] I. Dinur, A. Shamir: Breaking Grain-128 with Dynamic Cube Attacks. FSE 2011: 167–187

[6] H. Englund, T. Johansson, M. S. Turan: A Framework for Chosen IV Statistical Analysis of Stream Ciphers. INDOCRYPT 2007: 268–281

[7] E. Filiol: A new statistical testing for symmetric ciphers and hash functions. In: R. Deng et al. (Eds.) ICICS 2002, LNCS 2513, pp. 342–353. Springer, Heidelberg (2002)

[8] P. Fouque, T. Vannet: Improving Key Recovery to 784 and 799 Rounds of Trivium Using Optimized Cube Attacks. FSE 2013: 502–517

[9] A. Joux: Algorithmic cryptanalysis. 1st ed., Chapman & Hall/CRC, 2009.

[10] S. Knellwolf, W. Meier, M. Naya-Plasencia: Conditional Differential Cryptanalysis of Trivium and KATAN. Selected Areas in Cryptography 2011: 200–212

[11] M.J.O. Saarinen: Chosen-IV statistical attacks on estream stream ciphers. eSTREAM, ECRYPT Stream Cipher Project, Report 2006/013 (2006)

[12] P. Stankovski: Greedy Distinguishers and Nonrandomness Detectors. INDOCRYPT 2010: 210–226

[13] K. Pearson. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. Philosophical Magazine Series 5 50(302): 157–175 (1900)

[14] A. Vardasbi, M. Salmasizadeh, J. Mohajeri: Superpoly algebraic normal form monomial test on Trivium. IET Information Security 7(3): 230–238 (2013)

[15] F. Yates: Contingency table involving small numbers and the χ^2 test. Supplement to the Journal of the Royal Statistical Society 1(2): 217–235 (1934)