On the Security of LBlock against the Cube Attack and Side Channel Cube Attack

Saad Islam, Mehreen Afzal, and Adnan Rashdi

National University of Sciences and Technology, Islamabad, Pakistan
{saadislam,mehreenafzal,adnanrashdi}@mcs.edu.pk

Abstract. In this research, a recently proposed lightweight block cipher LBlock, not tested against the cube attack has been analyzed. 7, 8 and 9 round LBlock have been successfully attacked with complexities of $O(2^{10.76})$, $O(2^{11.11})$ and $O(2^{47.00})$ respectively. For the case of side channel cube attack, full version of LBlock has been attacked using a single bit leakage model with the complexity of $O(2^{55.00})$ cipher evaluations. For this purpose, a generic practical platform has been developed to test various stream and block ciphers against the latest cube attack.

Keywords: Cube attack, Side channel cube attack, Lightweight block ciphers, LBlock.

1 Introduction

Cube attack has been recently introduced by Dinur and Shamir in 2009 [1,2]. Preliminarily cube attack has been applied successfully on stream ciphers. Several results can be found on the stream cipher Trivium [3,4], one of the finalists of the estream project [5]. Reduced versions of Trivium having 672, 735 and 767 initialization rounds have been attacked. In a similar research, Vielhaber worked on the concept named AIDA (Algebraic IV Differential Attack) and attacked One.Fivium(a variant of Trivium) [6]. His other contributions include [7,8,9,10]. Zhe et al. further improved results of Vielhaber on One.Fivium [11]. Other predecessors of cube attack include the work of Englund et al. who showed statistical weaknesses of Trivium up to 736 initialization rounds [12] and the attack on 672 round Trivium by Fischer et al. [13]. In 2011, Mroczkowski and Szmidt evaluated Trivium by applying the cube attack and used the concept of quadraticity tests [14]. Another LFSR-based lightweight stream cipher Hitag2 has been analyzed by Sun et al. against the cube attack in 2011 [15]. MICKEY [16], also a finalist of the estream project has been found secure by Stefan in [17].

After successful results of cube attack on Trivium, Shamir et al. proposed the concept of Cube testers in 2009 [18]. Cube testers are based on efficient property-testing algorithm. They detect nonrandom behavior rather than performing key extraction. They can also attack cryptographic schemes described by nonrandom polynomials of relatively high degree. The targets of the authors in the mentioned paper are Trivium and MD6 [19]. In 2010, Li et al. worked on cube testers on Bivium [20]. Shamir et al. worked on Grain-128 [21] and gave results for

A. Cuzzocrea et al. (Eds.): CD-ARES 2013 Workshops, LNCS 8128, pp. 105–121, 2013.

cube testers and dynamic cube attack in [22] and [23]. Conditional differential cryptanalysis by Knellwolf et al. is a predecessor to dynamic cube attack [24]. Standard cube attack finds the key by solving a system of linear equations in terms of key bits whereas the dynamic cube attack recovers the secret key by exploiting distinguishers obtained from cube testers. Recently in 2012, Shamir et al. has proposed the concept of robust cube attacks for stream ciphers in realistic scenarios and also suggested the use of generalized linearity tests instead of BLR tests [25].

For block ciphers, Dinur and Shamir proposed the idea of side channel cube attack model in which only one bit of information is available to the attacker after each round [26]. The time complexities for AES [27] and SERPENT [28] are found to be $O(2^{38})$ and $O(2^{18})$ for full key recovery. Due to the exponential increase in degree after every round, the standard cube attack becomes limited to reduced versions only while the side channel attack model is applicable to the full versions and thus more practical.

Lightweight block ciphers, which provide a good trade off between security and efficiency, have attained significant attention of researchers. These ciphers are mostly used in resource-constraint environments like RFID and sensor networks. RFID technology has been used in many aspects of life, such as access control, parking management, identification, goods tracking etc. The lightweight block ciphers evaluated against the cube attack include the KATAN family [29], NOEKEON [30], PRESENT [31] and Hummingbird-2 [32] in [33,34,35,36,37,38]. Mroczkowski and Szmidt have attacked the Courtois Toy Cipher CTC, designed by Courtois [39] against the cube attack [40,41]. Lightweight block ciphers which are not evaluated against the cube attack so far include LED [42], EPCBC [43], PRINCE [44], Piccolo [45], mCrypton [46], TWIS [47], MIBS [48], CGEN [49], PRINTcipher [50], KLEIN [51], FOX [52], HIGHT [53], ICEBERG [54], LCASE [55], MISTY [56], PUFFIN [57], SEA [58], TEA [59] and CLEFIA [60].

LBlock, a lightweight block cipher recently proposed in 2011 by Wu et al. has not yet been tested against the cube attack [61]. In the security evaluation of LBlock by the authors, five cryptanalysis techniques have been used. For differential cryptanalysis, there is no useful 15-round differential characteristic for LBlock. For linear cryptanalysis, it is difficult to find useful 15-round linear-hulls which can be used to distinguish LBlock from a random permutation. For impossible differential cryptanalysis, attacks on 20-round LBlock has been mounted using 14-round impossible differential distinguishers. Integral attack goes up to 20 rounds and the related key attack goes up to 14 rounds of LBlock. Impossible differential attack has been improved up to 21 and 22-round LBlock in [62] and [63]. Minier et al. improved the related key attack to 22 rounds of LBlock [64]. Liu et al. also worked on a similar concept on 22-round LBlock [65]. Biclique cryptanalysis has been performed by the authors of LBlock and new key scheduling algorithm has been proposed in [66].

The cube attack implementations include Paul Crowley's implementation [67,68] and a practical platform developed by Bo Zhu [69]. Zhu et al. has also created an online application which only works for Trivium and checks the cubes for linearity and generate the linear expressions [70]. However, the tool is not suitable to be extended for the complete cube attack on any generic structure. Cryptool 2.0 [71] includes the cube attack block having the option for Trivium and DES only.

Our Contribution. LBlock has been evaluated against the developed tool. We are able to successfully attack 7, 8 and 9 round LBlock with complexities of just $O(2^{10.76})$, $O(2^{11.11})$ and $O(2^{47.00})$ cipher evaluations. Full version of LBlock has been attacked using the single bit leakage side channel cube attack model with the complexity of $O(2^{55.00})$. Cube attack may also be extended to further rounds by using more efficient hardware resources like super computer, the use of GPUs and the concept of distributed computing.

We have developed a graphical user interface toolkit which can load any stream or block cipher into it(as a function) and can check its resistance against the cube attack. The tool shows how may rounds of the cipher can be attacked, and it outputs the cube expressions found in a text file. The options such as cube size, number of linearity tests, output bit index, public bit size and secret bit size can be set from the GUI. The tool is user friendly and can be used easily without the help of the developers. The developed tool is capable of detecting the total number of processors in the machine and can utilize all of them for efficient execution. The tool works on both x86 and x64 systems having any windows version as it is just an executable file. The algorithm of cube attack used in our implementation can be found in [17].

Organization of the Paper. The cube attack has been explained in Section 2. An introduction of the cipher LBlock is given in Section 3 and the results of cube attack against LBlock are given in Section 4. Section 5 contains the results of side channel cube attack against LBlock. Detail of our software toolkit is given in Section 6. Section 7 concludes the article and proposes some future work.

2 The Cube Attack

The Cube Attack is a chosen public key attack which means chosen IV attack for stream ciphers and chosen plaintext attack for block ciphers. Ciphers can be represented as black box polynomials in terms of secret and private variables. These black boxes can be attacked by hitting them with chosen input values and obtaining the output.

Definition 1. Assume some polynomial $p\left(x_1,...,x_n\right)$ and a set $I \subseteq \{1,...n\}$ of indices to the variables of p Let t_I be a subterm of p which is the product of the variables indexed by I. Then factorizing p by t_I yields Equation 1.

$$p\left(x_1,...,x_n\right) = t_I.p_{S(I)} + q\left(x_1,...,x_n\right) \tag{1}$$

where $p_{S(I)}$ is the superpoly of I in p and q is the linear combination of all terms which do not contain t_I.

For detailed description of the attack, refer to [2]. The attack consists of two phases, the preprocessing phase and the online phase.

2.1 Preprocessing Phase

In the Preprocessing stage of the attack, the target is to find the maximum number of linearly independent expressions in terms of key bits. These expressions are called maxterm equations. This phase is time consuming and may take several weeks. The precomputation phase consists of two parts, finding maxterms and the superpoly reconstruction.

Finding Maxterms. A maxterm or a cube is a set of positions of plaintext block bits for which $2^{cubesize}$ crafted plaintexts are generated. These plaintexts P_i are generated by inserting all the possible values at cube positions keeping all other positions zero or constant. Summing a fixed output bit C_j for all P_i's while setting a same random key K in $GF(2)$ is called a cube sum for key K with output bit index j. Cube sum or summing over a cube is an important terminology. Linear cubes are searched whose cube sums satisfy the linearity tests(Blum Luby Rubinfeld tests) [72]. BLR test checks for the condition $f(0) \oplus f(K_1) \oplus f(K_2) = f(K_1 \oplus K_2)$ where K_1 and K_2 are random keys and f is the cube sum with a certain key over a cube to be tested. The probability that f is linear for $3N$ tests is $1 - 2^{-N}$. If a cube satisfies all the linearity tests, it is placed in the results table with the corresponding output bit index and the reconstructed maxterm equation which is explained in the next part. For the selection of cubes, the authors have proposed a random walk process in [2].

Reconstructing Maxterm Equations. Reconstructing maxterm equations or the superpoly reconstruction in terms of key bits (e.g $1 \oplus k_3 \oplus k_4$) is the second part of the preprocessing phase. According to Theorem 2 in [2], the constant term can be easily computed by setting $K = 0$ and calculating the cube sum. If the sum is 1, the maxterm contains the free term 1, otherwise not. The coefficients of the key bit variables k_i can be found by setting each k_i to one and remaining zeros and calculating the cube sums. If the sum is different from that for $K = 0$, that k_i will be the part of the maxterm equation. This is because if the value of a variable in a linear expression is flipped, the value of the expression is also flipped.

2.2 Online Phase

In the online phase there is an unknown set key which has to be recovered and the adversary can only tweak the plaintext bits. The target of this phase is to determine the right hand sides of the found expressions and their solution. This stage consists of two phases, forming and solving a system of linear equations.

Forming System of Linear Equations. In this part, cube sums are calculated for the same cubes found in the preprocessing stage and their relevant output bit index. These sums make the right hand side of the expressions making a system of linear equations (e.g $1 \oplus k_3 \oplus k_4 = 0$).

Solving System of Linear Equations. The system of linear equations may be solved using Gaussian elimination. The number of key bits recovered is equal to the number of linearly independent relations found in the preprocessing phase. For finding further relations the time consumed by the first phase increases exponentially.

Attack Complexity. The attack complexity includes two things. One is the number of iterations of the cipher carried out in the formation of system of linear equations and the other is the complexity to solve the linear relations in the online phase. Hence, the total complexity becomes $O(2^{d-1}n + n^2)$ where d is the degree of the cryptosystem and n is the number of secret bits. Brute force complexity of the remaining unknown key bits is also added to the total.

3 LBlock: A Lightweight Block Cipher

LBlock, LuBan LOCK or Lightweight BLOCK cipher has been proposed by Wu and Zhang in 2011 [61]. The cipher is a good trade off between efficiency and security. The hardware implementation of LBlock requires about $1320GE$ on $0.18\mu m$ technology with a throughput of $200Kbps$ at $100KHz$ and its software implementation on 8-bit microcontroller requires about 3955 clock cycles to encrypt a plaintext block.

3.1 Specification of LBlock

LBlock has a Fiestel structure having block length of 64-bit, key length of 80-bit and 32 rounds see Figure 1, where concatenation of X_1 and X_0 represents the plaintext block, $K_1 - K_{32}$ are the 32 subkeys generated through a key scheduling procedure, $<<<$ 8 sign indicates 8-bit left cyclic shift operation, $\oplus$ is the XOR operation, X_{32} and X_{33} represents the concatenated ciphertext block. The round function F contains the confusion layer having eight 4×4 S-Boxes and a diffusion layer having permutation of eight 4-bit words.

4 The Cube Attack on LBlock

We have applied the cube attack on LBlock having 7, 8 and 9 rounds. The machine used throughout our analysis is Dell XPS 17 Laptop, 2nd generation Intel Core i7 2.20 GHz, 8GB DDR3, NVIDIA GeForce GT 550M 1GB graphics. Extension of the attack to further rounds has been constrained by the available computational capability. However, the concept of supercomputing and GPUs can greatly reduce the simulation time.

4.1 Results of the Preprocessing Phase

70 linearly independent relations in terms of key bits can be found in the preprocessing part for 8-round LBlock as shown in Table 1. 100 linearity tests have

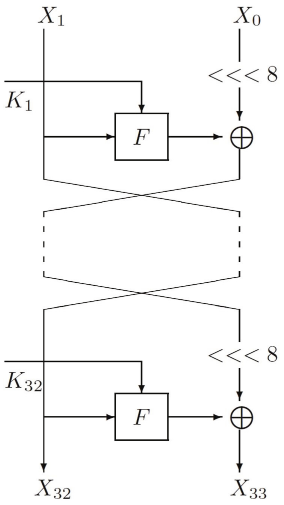


Fig. 1. Encryption Procedure for LBlock

been passed by each cube. The results have been confirmed by testing the attack for various random keys. Results for the preprocessing phase for 7 and 9 round LBlock are shown in Table 4 and Table 5 in Appendix-A.

4.2 Results of the Online Phase

In the online phase of the attack, the set of expressions obtained in the preprocessing phase are converted into the system of linear equations by determining the right hand sides. The values have been found by setting a random test key and summing over the same cubes found in the first phase. The example equations are shown below:

$$\begin{aligned}
&x1 = 0, x2 = 0, 1 + x3 + x4 = 1, x4 = 0, x5 = 0, x6 = 0, x7 = 0, x8 = 1,\\
&1 + x9 = 1, x10 = 0, 1 + x11 = 0, x12 = 0, 1 + x13 = 1, x14 = 0,\\
&1 + x15 + x16 = 1, x16 = 1, x17 = 0, x18 = 1, 1 + x19 = 1, 1 + x20 = 1,\\
&x21 = 0, x22 = 1, x23 + x24 = 1, 1 + x24 = 0, 1 + x25 = 1, x26 = 1,\\
&1 + x27 + x28 = 0, x28 = 0, x29 = 0, x30 = 1, 1 + x31 + x32 = 1, 1 + x32 = 0,\\
&x38 = 0, x2 + x39 = 0, x40 + x41 = 0, x41 = 1, x42 = 0, x43 = 1,\\
&1 + x10 + x44 + x45 = 0, x45 = 1, 1 + x46 = 1, x47 = 1, x47 + x48 + x49 = 1,\\
&1 + x22 + x49 = 1, x50 = 1, x30 + x51 = 1, 1 + x52 + x53 = 0, x53 = 1,\\
&x18 + x54 = 0, x55 = 0, 1 + x17 + x56 + x57 = 1, x57 = 1, x58 = 1,\\
&x26 + x59 = 0, 1 + x60 + x61 = 0, 1 + x61 = 0, x67 = 1, 1 + x68 = 0,\\
&x69 + x70 = 0, x70 = 1, x71 = 1, x72 = 0, x71 + x73 + x74 = 1, x74 = 1,\\
&1 + x75 = 1, x76 = 1, x76 + x77 + x78 = 1, x76 + x78 = 0, x79 = 0,\\
&x79 + x80 = 0
\end{aligned}\tag{2}$$

Table 1. Maxterms for 8-Round LBlock

Maxterm Equations	**Cube Indexes**	**Output Index**	**Maxterm Equations**	**Cube Indexes**	**Output Index**
x1	2,4,23,51	25	x41	2,3,4,23	25
x2	1,3,51,52	11	x42	9,10,11,15	15
1+x3+x4	1,2,51,52	11	x43	9,11,14,56	13
x4	1,2,51,52	12	1+x10+x44+x45	11,12,14,54	15
x5	6,7,19,43	1	x45	11,12,15,54	15
x6	5,7,19,43	1	1+x46	2,22,23,24	18
x7	5,6,19,43	1	x47	3,21,22,23	18
x8	5,6,18,42	4	x47+x48+x49	3,21,22,24	18
1+x9	10,11,15,54	13	1+x22+x49	2,23,24,57	18
x10	11,41,55,56	1	x50	26,29,31,32	6
1+x11	9,10,15,54	13	x30+x51	27,31,32,63	5
x12	9,10,14,55	13	1+x52+x53	27,30,31,32	5
1+x13	14,15,45,48	5	x53	27,29,31,62	6
x14	6,13,33,47	10	x18+x54	5,19,20,36	12
1+x15+x16	10,14,46,47	21	x55	6,17,18,19	9
x16	13,14,47,48	23	1+x17+x56+x57	7,18,19,33	9
x17	7,18,19,35	10	x57	7,18,19,33	10
x18	6,17,19,34	9	x58	25,27,28,31	29
1+x19	6,17,18,34	9	x26+x59	27,28,31,40	29
1+x20	17,18,33,36	9	1+x60+x61	26,27,28,31	29
x21	2,22,23,59	18	1+x61	26,27,28,31	30
x22	21,23,58,59	1	x67	41,43,44,62	60
x23+x24	3,21,22,58	18	1+x68	41,43,55,56	13
1+x24	21,22,59,60	1	x69+x70	42,43,44,62	60
1+x25	26,27,39,40	14	x70	42,43,44,63	57
x26	27,37,38,57	3	x71	49,50,51,55	47
1+x27+x28	25,26,39,40	14	x72	50,51,52,54	47
x28	25,26,30,38	29	x71+x73+x74	22,49,50,52,54	47
x29	31,32,50,51	27	x74	22,51,52,55,56	31
x30	26,28,29,62	5	1+x75	42,62,63,64	50
1+x31+x32	29,30,63,64	6	x76	43,61,62,63	50
1+x32	27,29,30,63	6	x76+x77+x78	43,61,62,64	50
x38	1,3,4,22	28	x76+x78	25,42,61,62,64	50
x2+x39	3,4,23,52	25	x79	34,37,39,40	38
x40+x41	2,3,4,22	28	x79+x80	31,35,37,38,40	37

Solving Equations 2, 70 key bits are recovered as shown below:
$x1 = 0, x2 = 0, x3 = 0, x4 = 0, x5 = 0, x6 = 0, x7 = 0, x8 = 1, x9 = 0, x10 = 0, x11 = 1, x12 = 0, x13 = 0, x14 = 0, x15 = 1, x16 = 1, x17 = 0, x18 = 1, x19 = 0, x20 = 0, x21 = 0, x22 = 1, x23 = 0, x24 = 1, x25 = 0, x26 = 1, x27 = 1, x28 = 0, x29 = 0, x30 = 1, x31 = 1, x32 = 1, x38 = 0, x39 = 0, x40 = 1, x41 = 1, x42 = 0, x43 = 1, x44 = 0, x45 = 1, x46 = 0, x47 = 1, x48 = 1, x49 = 1, x50 = 1, x51 = 0, x52 = 0, x53 = 1, x54 = 1, x55 = 0, x56 = 1, x57 = 1, x58 = 1, x59 = 1, x60 =$

$0, x61 = 1, x67 = 1, x68 = 1, x69 = 1, x70 = 1, x71 = 1, x72 = 0, x73 = 1, x74 = 1, x75 = 0, x76 = 1, x77 = 1, x78 = 1, x79 = 0, x80 = 0$

Remaining bits $x33, ..., x37, x62, ...x66$ may be recovered using quadraticity tests [14] or brute force search. The recovered bits can be further compared with the test key. The test key may be set to any random value. The online phase is not computationally expensive and just takes fraction of a second.

4.3 Attack Complexity

The total complexity includes the complexity of the online phase and of the brute force search. Precomputation is the one time effort and thus not included in the calculations. 66 out of 70 cubes are of size 4 having complexity equal to $66 \times 2^4 = 1056$. 4 out of 70 cubes are of size 5 having complexity equal to $4 \times 2^5 = 128$. Total becomes 1056 + 128 = 1184 iterations of LBlock. Brute force complexity for remaining 10 bits is 2^{10}. So final complexity becomes 1184 + 2^{10} = 2208 approximately equal to $O(2^{11.11})$ which is quite less. Similarly for 7-round LBlock the complexity becomes $17\times2^2+32\times2^3+18\times2^4+3\times2^5+2^{10} \approx O\left(2^{10.76}\right)$. For 9-round LBlock the complexity is $12\times2^4+11\times2^5+10\times2^6+2^{47} \approx O\left(2^{47.00}\right)$.

5 The Side Channel Cube Attack on LBlock

The side channel cube attack is a variant of the standard cube attack which is more practical in realistic scenarios. The standard cube attack is restricted on the reduced versions of the ciphers whereas the side channel cube attack is a threat in practical situations for the full versions. In this type of attack the adversary is able to get one bit of leakage information of the state after each round of an iterated block cipher. The process may be made possible via physical probing, power measurement, or any other type of side channel. However, this information is quite noisy and this problem is addressed by using error correction techniques like erasure codes [73].

5.1 Results of the Preprocessing Phase

LBlock achieves complete diffusion after 8 rounds, as mentioned by the authors of LBlock [61]. So, a single bit leakage after 8^{th} round may give maximum number of linear relations as compared to inner rounds. We have taken the MSB of the right half X_R of the state after 8^{th} round as the leakage bit and used for our analysis. Thousands of linear relations have been found but 25 linearly independent have been extracted using Gaussian elimination technique. The results of the preprocessing phase for full version of LBlock are shown in Table 3. Time consumed for searching all possible combinations for various cube sizes is shown in Table 2 where the number of linearity tests has been set to 100. The results are for the single core execution with the multi-processing feature disabled.

Table 2. Elapsed Times against the Cube Sizes for Preprocessing

Cube Size	Time in Seconds
3	2
4	53
5	1667
6	28739
7	704582

5.2 Attack Complexity

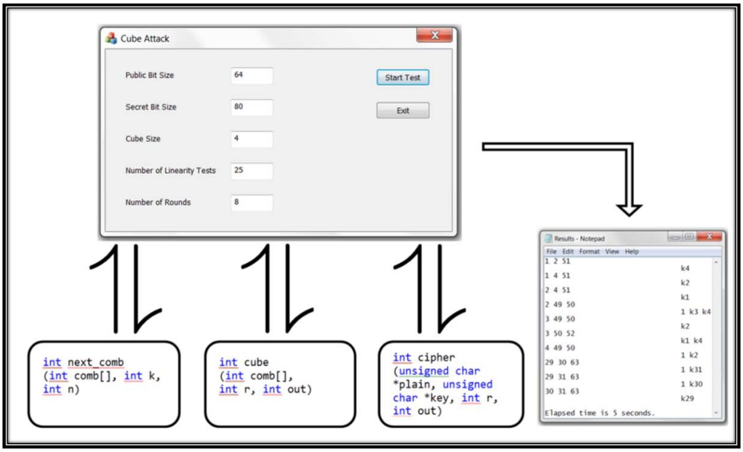
10 out of 25 cubes are of size 4 having complexity $= 10 \times 2^4 = 160$. Remaining 15 cubes are of sizes 5, 6, 7 and 8 having complexities, $5 \times 2^5 = 160, 4 \times 2^6 = 256, 3 \times 2^7 = 384$ and $3 \times 2^8 = 768$ respectively. Total becomes $160 + 160 + 256 + 384 + 768 = 1728$ iterations of LBlock. Brute force complexity for remaining 55 bits $= 2^{55}$. So final complexity becomes $1728 + 2^{55}$ approximately equal to $O(2^{55.00})$.

Table 3. Maxterms for Full LBlock using Leakage Bit after 8^{th} Round

Maxterm Equations	Cube Indexes	Maxterm Equations	Cube Indexes
x1	3,4,21,41,54,55	x22	21,23,58,59
x2	3,19,23,49,50,55	1+x23+x24	21,22,58,59
1+x3+x4	1,2,19,51,52,55	1+x24	21,22,59,60
x1+x4	3,18,23,50,51,55	x25+x26+x28	27,31,62,63,64
x5	6,7,19,43	x38	1,3,4,18,22,41,55
x6	5,7,19,43	1+x38+x39	1,2,4,23,41,52,55,56
x7	5,6,19,43	1+x40+x41	1,2,3,21,41,54,55
1+x5+x8	7,18,42,43	x41	1,2,3,9,23,24,41,49
1+x9	10,11,41,54,56	1+x71+x72	19,49,50,51,55
x10	11,41,55,56	1+x72	19,50,51,52,54
1+x11+x12	10,41,55,56	x71+x73+x74	1,2,3,17,18,22,50,54
x12	9,10,18,41,55	x74	1,2,3,19,22,50,55
1+x21	22,24,59,60		

6 Cube Attack Software Toolkit

We have developed a GUI based software tool using the MFC application in Microsoft Visual Studio 2010 Professional. There is a function named *cipher* which is to be replaced by any stream or block cipher to be tested. The function is capable of taking the plaintext, key and number of rounds as input and should return the ciphertext as the output. All inputs/outputs have to be in hexadecimal notation. After replacing the function, one has to start debugging and an executable file is made as the output. This executable will be able to run on any Windows version on both x86 and x64 platforms. In the GUI, we have five inputs public bit size, secret bit size, cube size, number of linearity tests and the number of rounds which can be set to any desired position. The results are compiled in a text file at the end of the simulation which include the cubes found, the output bit indexes, total simulation time in seconds and the reconstructed maxterms, see Figure 2.



Public Bit Size: 64
Secret Bit Size: 80
Cube Size: 4
Number of Linearity Tests: 25
Number of Rounds: 8

```
int next_comb
(int comb[], int k,
int n)
```

```
int cube
(int comb[],
int r, int out)
```

```
int cipher
(unsigned char
*plain, unsigned
char *key, int r,
int out)
```

```
1 2 51        k4
1 4 51        k2
2 4 51        k1
2 49 50       1 k3 k4
3 49 50       k2
3 50 52       k1 k4
4 49 50       1 k2
29 30 63      1 k31
29 31 63      1 k30
30 31 63      k29

Elapsed time is 5 seconds.
```

Fig. 2. Cube Attack Implementation Architecture

- After debugging the project with the embedded cipher function, the GUI is created and launched.
- The GUI takes the parameters from the user and interacts with the three functions next_comb, cube and cipher.
- next_comb function is responsible for randomly generating different cubes of the required size.
- cube function is responsible for testing the cubes for the linearity tests.
- cipher function is invoked millions of times to get the required output bits for the crafted plaintexts.
- Results are written in the results.txt file at the end of the simulation.

The number of rounds in the GUI represents the initialization or setup rounds in case of stream ciphers and the main rounds in case of block ciphers. Hence, the tool is generalized for both of them. The option to set number of rounds is for the variants or reduced versions of ciphers. This helps in better understanding about the resistivity of the ciphers.

The tool is intelligent to use all the available CPU cores in a system, thus decreasing the simulation time to a great extent. OpenMP (Open Multiprocessing) has been used to implement this task [74]. Another option has been added in the tool to work on multiple output bits on each iteration. The standard cube attack works on a single output bit model and the remaining block is not utilized. The concept has been explained in Figure 3. The same concept holds true for the stream ciphers and thus complete reinitialization of the cipher is not required to get each output bit. This feature increased the speed of the simulation 27 times in our experiments. The option is turned off when working on single bit leakage models.

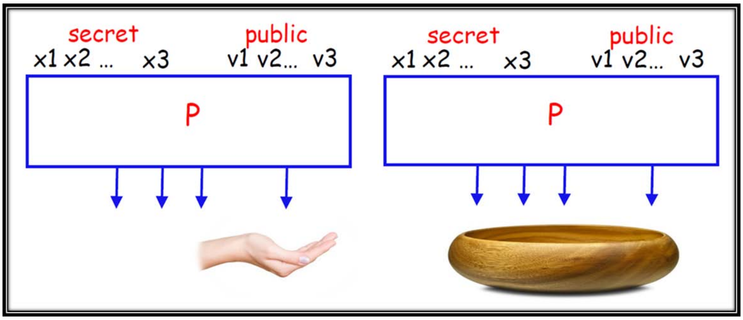


Fig. 3. Attacking Multiple Output Bits

7 Conclusion and Future Work

Cube Attack is a relatively new technique of cryptanalysis and its application on different new ciphers is important. 7, 8 and 9 rounds of LBlock have been attacked. The complexities of the attack for the three versions are $O(2^{10.76})$, $O(2^{11.11})$ and $O(2^{47.00})$ respectively. Full version of LBlock has been attacked using single bit side information after 8 rounds with a complexity of $O(2^{55})$. A software tool has been developed for the application of cube attack to any black box cipher. The tool can be easily used for testing and evaluation purposes.

Higher order tests like quadraticity tests may be implemented to recover more number of key bits where linearity tests have failed to produce the linear relations. BLR tests may be replaced by generalized linearity tests. The efficiency of the tool may be increased by the use of GPUs as their highly parallel structure makes them more effective than CPUs. The task can also be divided to a number of computers connected in a network, a concept known as distributed computing. Another solution is to use a super computer having a number of processors having multiple cores along with the powerful GPUs. Cube testers and dynamic cube attacks are the next steps after the cube attack.

Acknowledgments. We are thankful to the authors of LBlock especially Lei Zhang. They confirmed us that the security analysis of LBlock against the cube attack has not been done yet and we may try. We are also thankful to Michael Vielhaber, Deian Stefan and Bo Zhu who coordinated with us on email that helped us in understanding the cube attack.

References

1. Dinur, I., Shamir, A.: Cube attacks on tweakable black box polynomials. Cryptology ePrint Archive, Report 2008/385 (2008), `http://eprint.iacr.org/`
2. Dinur, I., Shamir, A.: Cube attacks on tweakable black box polynomials. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 278–299. Springer, Heidelberg (2009)
3. De Cannière, C., Preneel, B.: Trivium specifications. ECRYPT Stream Cipher Project Report 2005/030 (2005)
4. De Cannière, C.: Trivium: A stream cipher construction inspired by block cipher design principles. In: Katsikas, S.K., López, J., Backes, M., Gritzalis, S., Preneel, B. (eds.) ISC 2006. LNCS, vol. 4176, pp. 171–186. Springer, Heidelberg (2006)
5. eSTREAM: The ecrypt stream cipher project
6. Vielhaber, M.: Breaking one.fivium by aida an algebraic iv differential attack. Cryptology ePrint Archive, Report 2007/413 (2007), `http://eprint.iacr.org/`
7. Vielhaber, M.: Speeding up aida the algebraic iv differential attack by the fast reed-muller transform. In: Intelligent Decision Making Systems. World Scientific Proceedings Series on Computer Engineering and Information Science, vol. 2. World Scientific Publishing Co. (2010)
8. Vielhaber, M.: Aida vs. trivium 793: 1152 final score 980: 1152. Eurocrypt 2009 rump session (April 2009), `http://eurocrypt2009rump.cr.yp.to/`
9. Vielhaber, M.: Aida breaks bivium (a&b) in 1 minute dual core cpu time. Cryptology ePrint Archive, Report 2009/402 (2009), `http://eprint.iacr.org/`
10. Vielhaber, M.: Shamir's "cube attack": A remake of aida, the algebraic iv differential attack (2009)
11. Zhe, S., Shi-Wu, Z., Lei, W.: Chosen iv algebraic attack on one.fivium. In: 3rd International Conference on Intelligent System and Knowledge Engineering, ISKE 2008, pp. 1427–1431 (November 2008)
12. Englund, H., Johansson, T., Sönmez Turan, M.: A framework for chosen IV statistical analysis of stream ciphers. In: Srinathan, K., Pandu Rangan, C., Yung, M. (eds.) INDOCRYPT 2007. LNCS, vol. 4859, pp. 268–281. Springer, Heidelberg (2007)
13. Fischer, S., Khazaei, S., Meier, W.: Chosen IV statistical analysis for key recovery attacks on stream ciphers. In: Vaudenay, S. (ed.) AFRICACRYPT 2008. LNCS, vol. 5023, pp. 236–245. Springer, Heidelberg (2008)
14. Mroczkowski, P., Szmidt, J.: Corrigendum to: The cube attack on stream cipher trivium and quadraticity tests. Cryptology ePrint Archive, Report 2011/032 (2011), `http://eprint.iacr.org/`
15. Sun, S., Hu, L., Xie, Y., Zeng, X.: Cube cryptanalysis of hitag2 stream cipher. In: Lin, D., Tsudik, G., Wang, X. (eds.) CANS 2011. LNCS, vol. 7092, pp. 15–25. Springer, Heidelberg (2011)
16. Babbage, S., Dodd, M.: The MICKEY stream ciphers. In: Robshaw, M., Billet, O. (eds.) New Stream Cipher Designs. LNCS, vol. 4986, pp. 191–209. Springer, Heidelberg (2008)
17. Stefan, D.: Analysis and Implementation of ESTREAM and SHA-3 Cryptographic Algorithms. Cooper Union for the Advancement of Science and Art, Albert Nerken School of Engineering, Graduate Division (2011)
18. Aumasson, J.P., Dinur, I., Meier, W., Shamir, A.: Cube testers and key recovery attacks on reduced-round md6 and trivium. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 1–22. Springer, Heidelberg (2009)

19. Rivest, R.L., Agre, B., Bailey, D.V., Crutchfield, C., Dodis, Y., Elliott, K., Khan, F.A., Krishnamurthy, J., Lin, Y., Reyzin, L., Shen, E., Sukha, J., Sutherland, D., Tromer, E., Yin, Y.L.: The md6 hash function a proposal to nist for sha-3 (2008)
20. Li, S., Wang, Y., Peng, J.: Cube testers on bivium. In: 2010 International Conference on Communications and Intelligence Information Security (ICCIIS), pp. 121–124 (October 2010)
21. Hell, M., Johansson, T., Maximov, E., Meier, W.: A stream cipher proposal: Grain-128. In: ISIT
22. Aumasson, J.P., Dinur, I., Henzen, L., Meier, W., Shamir, A.: Efficient fpga implementations of high-dimensional cube testers on the stream cipher grain-128. Cryptology ePrint Archive, Report 2009/218 (2009), http://eprint.iacr.org/
23. Dinur, I., Shamir, A.: Breaking grain-128 with dynamic cube attacks. In: Joux, A. (ed.) FSE 2011. LNCS, vol. 6733, pp. 167–187. Springer, Heidelberg (2011)
24. Knellwolf, S., Meier, W., Naya-Plasencia, M.: Conditional differential cryptanalysis of NLFSR-based cryptosystems. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 130–145. Springer, Heidelberg (2010)
25. Dinur, I., Shamir, A.: Applying cube attacks to stream ciphers in realistic scenarios. Cryptography and Communications 4, 217–232 (2012)
26. Dinur, I., Shamir, A.: Side channel cube attacks on block ciphers. Cryptology ePrint Archive, Report 2009/127 (2009), http://eprint.iacr.org/
27. Daemen, J., Rijmen, V.: Aes proposal: Rijndael (1998)
28. Biham, E., Anderson, R.J., Knudsen, L.R.: Serpent: A new block cipher proposal. In: Vaudenay, S. (ed.) FSE 1998. LNCS, vol. 1372, pp. 222–238. Springer, Heidelberg (1998)
29. De Cannière, C., Dunkelman, O., Knežević, M.: KATAN and KTANTAN — A family of small and efficient hardware-oriented block ciphers. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 272–288. Springer, Heidelberg (2009)
30. Daemen, J., Peeters, M., Vanassche, G.: Nessie proposal: Noekeon. Submitted as an NESSIE Candidate Algorithm, http://www.cryptonessie.org
31. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An ultra-lightweight block cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
32. Engels, D., Saarinen, M.-J.O., Schweitzer, P., Smith, E.M.: The hummingbird-2 lightweight authenticated encryption algorithm. In: Juels, A., Paar, C. (eds.) RFIDSec 2011. LNCS, vol. 7055, pp. 19–31. Springer, Heidelberg (2012)
33. Bard, G.V., Courtois, N.T., Nakahara Jr., J., Sepehrdad, P., Zhang, B.: Algebraic, AIDA/Cube and side channel analysis of KATAN family of block ciphers. In: Gong, G., Gupta, K.C. (eds.) INDOCRYPT 2010. LNCS, vol. 6498, pp. 176–196. Springer, Heidelberg (2010)
34. Mroczkowski, P., Szmidt, J.: The algebraic cryptanalysis of the block cipher katan32 using modofied cube attack. In: Concepts and Implementations for Innovative Military Communications (2011)
35. Abdul-Latip, S., Reyhanitabar, M., Susilo, W., Seberry, J.: On the security of noekeon against side channel cube attacks. In: Kwak, J., Deng, R.H., Won, Y., Wang, G. (eds.) ISPEC 2010. LNCS, vol. 6047, pp. 45–55. Springer, Heidelberg (2010)
36. Yang, L., Wang, M., Qiao, S.: Side channel cube attack on present. In: Garay, J.A., Miyaji, A., Otsuka, A. (eds.) CANS 2009. LNCS, vol. 5888, pp. 379–391. Springer, Heidelberg (2009)

37. Zhao, X., Wang, T., Guo, S.: Improved side channel cube attacks on present. Cryptology ePrint Archive, Report 2011/165 (2011), `http://eprint.iacr.org/`
38. Fan, X., Gong, G.: On the security of hummingbird-2 against side channel cube attacks. In: Armknecht, F., Lucks, S. (eds.) WEWoRC 2011. LNCS, vol. 7242, pp. 18–29. Springer, Heidelberg (2012)
39. Courtois, N.T.: How fast can be algebraic attacks on block ciphers? In: Online Proceedings of Dagstuhl Seminar 07021, Symmetric Cryptography, pp. 7–12 (2006)
40. Mroczkowski, P., Szmidt, J.: Cube attack on courtois toy cipher. Cryptology ePrint Archive, Report 2009/497 (2009), `http://eprint.iacr.org/`
41. Mroczkowski, P., Szmidt, J.: The cube attack in the algebraic cryptanalysis of ctc2. Concepts and Implementations for Innovative Military Communications (2011)
42. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.: The LED block cipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 326–341. Springer, Heidelberg (2011)
43. Yap, H., Khoo, K., Poschmann, A., Henricksen, M.: EPCBC - A block cipher suitable for electronic product code encryption. In: Lin, D., Tsudik, G., Wang, X. (eds.) CANS 2011. LNCS, vol. 7092, pp. 76–97. Springer, Heidelberg (2011)
44. Borghoff, J., et al.: PRINCE – A low-latency block cipher for pervasive computing applications. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 208–225. Springer, Heidelberg (2012)
45. Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., Shirai, T.: *Piccolo*: An ultra-lightweight blockcipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 342–357. Springer, Heidelberg (2011)
46. Lim, C.H., Korkishko, T.: mCrypton – A lightweight block cipher for security of low-cost RFID tags and sensors. In: Song, J.-S., Kwon, T., Yung, M. (eds.) WISA 2005. LNCS, vol. 3786, pp. 243–258. Springer, Heidelberg (2006)
47. Ojha, S.K., Kumar, N., Jain, K., Sangeeta: TWIS – A lightweight block cipher. In: Prakash, A., Sen Gupta, I. (eds.) ICISS 2009. LNCS, vol. 5905, pp. 280–291. Springer, Heidelberg (2009)
48. Izadi, M., Sadeghiyan, B., Sadeghian, S., Khanooki, H.: MIBS: A new lightweight block cipher. In: Garay, J.A., Miyaji, A., Otsuka, A. (eds.) CANS 2009. LNCS, vol. 5888, pp. 334–348. Springer, Heidelberg (2009)
49. Robshaw, M.J.B.: Searching for compact algorithms: CGEN. In: Nguyên, P.Q. (ed.) VIETCRYPT 2006. LNCS, vol. 4341, pp. 37–49. Springer, Heidelberg (2006)
50. Knudsen, L., Leander, G., Poschmann, A., Robshaw, M.J.B.: PRINTCIPHER: A block cipher for IC-printing. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 16–32. Springer, Heidelberg (2010)
51. Gong, Z., Nikova, S., Law, Y.: Klein: A new family of lightweight block ciphers. In: Juels, A., Paar, C. (eds.) RFIDSec 2011. LNCS, vol. 7055, pp. 1–18. Springer, Heidelberg (2012)
52. Junod, P., Vaudenay, S.: FOX: A new family of block ciphers. In: Handschuh, H., Hasan, M.A. (eds.) SAC 2004. LNCS, vol. 3357, pp. 114–129. Springer, Heidelberg (2004)
53. Hong, D., et al.: Hight: A new block cipher suitable for low-resource device. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 46–59. Springer, Heidelberg (2006)
54. Standaert, F.-X., Piret, G., Rouvroy, G., Quisquater, J.-J., Legat, J.-D.: `ICEBERG`: An involutional cipher efficient for block encryption in reconfigurable hardware. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 279–299. Springer, Heidelberg (2004)

55. Tripathy, S., Nandi, S.: Lcase: Lightweight cellular automata-based symmetric-key encryption (2008)
56. Matsui, M.: New block encryption algorithm MISTY. In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 54–68. Springer, Heidelberg (1997)
57. Cheng, H., Heys, H.M., Wang, C.: Puffin: A novel compact block cipher targeted to embedded digital systems. In: Proceedings of the 2008 11th EUROMICRO Conference on Digital System Design Architectures, Methods and Tools, DSD 2008, pp. 383–390. IEEE Computer Society, Washington, DC (2008)
58. Standaert, F.X., Piret, G., Gershenfeld, N., Quisquater, J.J.: Sea: A scalable encryption algorithm for small embedded applications. In: Domingo-Ferrer, J., Posegga, J., Schreckling, D. (eds.) CARDIS 2006. LNCS, vol. 3928, pp. 222–236. Springer, Heidelberg (2006)
59. Wheeler, D., Needham, R.: Tea, a tiny encryption algorithm. In: Preneel, B. (ed.) FSE 1994. LNCS, vol. 1008, pp. 363–366. Springer, Heidelberg (1995)
60. Shirai, T., Shibutani, K., Akishita, T., Moriai, S., Iwata, T.: The 128-bit blockcipher CLEFIA (extended abstract). In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 181–195. Springer, Heidelberg (2007)
61. Wu, W., Zhang, L.: Lblock: A lightweight block cipher. In: Lopez, J., Tsudik, G. (eds.) ACNS 2011. LNCS, vol. 6715, pp. 327–344. Springer, Heidelberg (2011)
62. Liu, Y., Gu, D., Liu, Z., Li, W.: Impossible differential attacks on reduced-round Lblock. In: Ryan, M.D., Smyth, B., Wang, G. (eds.) ISPEC 2012. LNCS, vol. 7232, pp. 97–108. Springer, Heidelberg (2012)
63. Karakoç, F., Demirci, H., Harmancı, A.E.: Impossible differential cryptanalysis of reduced-round LBlock. In: Askoxylakis, I., Pöhls, H.C., Posegga, J. (eds.) WISTP 2012. LNCS, vol. 7322, pp. 179–188. Springer, Heidelberg (2012)
64. Minier, M., Naya-Plasencia, M.: A related key impossible differential attack against 22 rounds of the lightweight block cipher Lblock. Inf. Process. Lett. 112(16), 624–629 (2012)
65. Liu, S., Gong, Z., Wang, L.: Improved related-key differential attacks on reduced-round Lblock. In: Chim, T.W., Yuen, T.H. (eds.) ICICS 2012. LNCS, vol. 7618, pp. 58–69. Springer, Heidelberg (2012)
66. Wang, Y., Wu, W., Yu, X., Zhang, L.: Security on LBlock against biclique cryptanalysis. In: Lee, D.H., Yung, M. (eds.) WISA 2012. LNCS, vol. 7690, pp. 1–14. Springer, Heidelberg (2012)
67. Crowley, P.: Trivium, sse2, corepy, and the "cube attack" (December 2008), http://www.lshift.net/blog/2008/12/09/trivium-sse2-corepy-and-the-cube-attack
68. Corepy: Assembly programming in python, http://www.corepy.org/
69. Zhu, B., Yu, W., Wang, T.: A practical platform for cube-attack-like cryptanalyses. Cryptology ePrint Archive, Report 2010/644 (2010), http://eprint.iacr.org/
70. Zhu, B., Yu, W., Wang, T.: A practical platform for cube-attack-like cryptanalyses, http://cube-attack.appspot.com
71. Cryptool 2 cryptography for everybody, http://www.cryptool.org/en/cryptool2
72. Blum, M., Luby, M., Rubinfeld, R.: Self-testing/correcting with applications to numerical problems. In: Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing, STOC 1990, pp. 73–83. ACM, New York (1990)
73. Luby, M., Mitzenmacher, M., Shokrollahi, M., Spielman, D.: Efficient erasure correcting codes. IEEE Transactions on Information Theory 47(2), 569–584 (2001)
74. The openmp api specification for parallel programming, http://openmp.org/wp/

Appendix-A

Table 4. Maxterms for 7-Round LBlock

Maxterm Equations	**Cube Indexes**	**Output Index**	**Maxterm Equations**	**Cube Indexes**	**Output Index**
x1	3,4	4	x41	3,4,23,50	57
x2	1,50	4	x42	9,10,11,54	23
1+x3+x4	2,49,50	1	x43	10,11,12	23
1+x4	3,50	3	x10+x44+x45	11,12,54	23
x5	7,8	24	1+x45	50,51,63,64	39
x6	7,42	24	1+x46	23,24,60	11
1+x7+x8	6,41,42	21	x21+x47	22,23,60	11
x8	5,6,42	28	x22+x48+x49	1,23,24,46	15
1+x9	10,12,55	25	1+x22+x49	23,24,57	11
x10	11,53	27	x50	29,30,31,63	4
1+x11+x12	10,53	27	x30+x51	31,32,64	4
x12	9,10,55	25	1+x52+x53	31,32,62	4
1+x13	14,15,45	13	x53	27,29,31,62	38
x14	13,47	13	1+x18+x54	19,20,36	41
1+x15+x16	14,46	16	1+x55	19,20,33	14
x16	13,14,47	15	1+x17+x56+x57	18,19,33	14
1+x17	18,20,34	20	x57	7,18,19,33	41
x18	20,35	16	x58	25,26,27,40	5
1+x17+x19	6,20,34	41	1+x26+x59	27,28,40	5
x17+x20	19,34	18	1+x60+x61	26,27,28,31	61
x21	22,23,59	5	x61	26,27,28	5
x22	23,58	8	x67	41,43,44,62	4
1+x23+x24	22,58	8	1+x68	41,43,55,56	45
1+x24	21,22,59	6	x69+x70	18,43,44,62	3
x25	27,28	9	x70	18,43,44,63	1
x26	25,38	9	x71	49,50,51,55	23
x27	25,26,39	5	1+x72	49,52,55	21
1+x28	25,26,39	10	x71+x73+x74	50,52,55	21
x29	31,32	29	x74	22,51,52,55	23
x30	29,62	29	1+x75	27,43,63,64	21
1+x31	29,30,63	1	x76	41,42,62,63	21
x32	29,30,62	32	x76+x77+x78	14,41,61,65,64	47
x38	1,3,4,22	60	x76+x78	25,42,61,62,64	24
x2+x39	3,4,51	31	x79	34,37,39,40	14
x40+x41	3,4,50	31	x79+x80	31,35,37,38,40	13

Table 5. Maxterms for 9-Round LBlock

Maxterm Equations	**Cube Indexes**	**Output Index**	**Maxterm Equations**	**Cube Indexes**	**Output Index**
x1	2,3,41,43,44	28	x30	29,32,41,43,44,64	25
x2	29,35,37,38,39	6	x32	29,30,41,43,44,64	25
1+x3+x4	1,2,41,43,44	28	x67	41,43,44,62	28
x4	46,57,58,59	9	x67+x68	41,42,44,61,62	25
x5	6,7,41,61,63,64	17	x69+x70	42,43,44,62	28
x6	2,47,57,58,59	10	x70	42,43,44,63	25
x7	33,35,36,39	29	x71	49,50,51,55	15
x7+x8	7,33,34,36,39	29	x72	50,51,52,54	15
1+x9+x10	34,35,36,39	29	x71+x73+x74	22,49,50,52,54	15
x10	34,35,36,38	32	x74	22,49,50,52,55	15
1+x17	18,19,31,37,39,40	5	x75	26,42,61,62,63	18
x18	17,19,31,37,39,40	5	x76	43,61,62,63	18
x25	8,26,27,34,35,36	32	x76+x77+x78	43,61,62,64	18
x26	5,28,33,35,36,37	30	x76+x78	25,42,61,62,64	18
x9+x10+x27+x28	25,26,34,35,36,37	30	x79	34,37,39,40	6
1+x25+x28	27,39,40,57,59,60	11	x79+x80	31,35,37,38,40	5
x29	30,32,41,43,44,64	25			