The Improved Cube Attack on Grain-v1

Yongjuan Wang[1] Liren Ding[1] Wenbao Han[2] Xiangyu Wang[1]
（ 1. University for Foreign Language of Luo Yang, The Department of Language Engineering,
Luo Yang He Nan 471003；
2.Information Engineering University of Zheng Zhou, The Department of Information Study,
Zheng Zhou He Nan 450002）

Abstract：The crucial problem of cube attack is the selection of cube set, which also being the most time-consuming process. This paper designs a new search algorithm which generates several linear equations through one cube set and applies cube attack to simplified version of Grain-v1algorithem. Our attack directly recovers 14 bits of the secret key when the initialization rounds in Grain-v1is 75 and finds 5 linear expressions about another 28 bits of the key.
Keywords：cube attack，Grain-v1 algorithm，linearity test，cube sum

1 Introduction

Cube attack is a recently developed known plaintext attack based on the idea of algebra, which been presented by Dinur[1] and Shamir in 2008. The idea is to represent the investigated algorithm as a low degree polynomial $F(K,IV)$ about the secret key K and public variables IV(plaintext bits in block ciphers) over infinite field F_2. Then obtaining the linear relation of the key through the reasonably selected cube set in order to generate a system of linear equations about the key. At last recover a certain number of key bits by solving the system of linear equations. Cube attack is a previous attack which performed very well when applied to block ciphers[4,9]、stream ciphers[5,6,7] and hash functions[3], thus receiving extensive attention. In 2008, Dinur[1] applied cube attack to Trivium when the initialization rounds is 767 and recovered 35 bits of the key within time complexity of $O(2^{45})$, where the scale of cube set was 28-31. Haixin Song et al[7] applied cube attack to Grain-v1[8] when the initialization rounds is 70 and recovered 15 bits of the key. In 2009, Dinur et al[9] applied cube attack to block cipher DES、Serpent[10] and AES[11] after denoise process and recovered all of the key bits of the algorithms mentioned above. Aumasson [3] applied cube attack to MD6 when the initialization rounds is 14 and recovered 128 bits within time complexity of $O(2^{23})$, which being the best known attack of MD6. Lathrop et al[12] applied cube analysis to hash functions and successfully attacked the simplified SHA-3 and ESSENCE. In order to enhance the efficiency of cube attack, scholars keep improving its method, Yu Sun et al[13]proved the theoretic relationship between cube attack and high order differential attack. In 2010, Dinur[17] present dynamic cube attack method toward Grain-128 stream cipher algorithm. In this paper, we assign reasonable values to public variables and can obtain several linear equations through only one cube, thus dramatically improving the efficiency of cube attack. When applied to Grain-v1 when the initialization rounds is 75, our attack directly recovers 14 bits of the secret key and finds 5 linear expressions about another 28 bits of the key, which being the best known attack of Grain-v1.

2 Preliminaries

2.1 Cube Attack

Cube attack is a known plaintext attack based on the idea of algebra. It usually regards the investigated cryptosystem as a polynomial $P(v,k)$ over the vector space F_2^{m+n} with the secret key $k=(x_1,\cdots,x_n)$ and the public variable $v=(v_1,\cdots,v_m)$, the attacker can obtain and arbitrarily choose the public variables and get access to the output bits by inquiring the "black box". $P_t(v,k)=y_t$ represents the output of time t, assuming $I\subset\{1,2,\cdots,m\}$, $U=\{v_i,i\in I\}\subset\{v_1,\cdots,v_m\}$, the output of time t can be described as follows:

$$P_t(x_1,\cdots,x_n,v_1,\cdots,v_m)=v^I P_{S(I)}(x_1,\cdots x_n,V)+Q(x_1,\cdots,x_n,v_1,\cdots,v_m)$$

Note that $v^I=v_{i_1}v_{i_2}\cdots v_{i_k}$, $V=\{v_1,\cdots,v_m\}\setminus U$, polynomial $P_{S(I)}(\cdot)$ doesn't contain any variables from set U, v^I in polynomial $Q(\cdot)$ is a monomial of the factor and the set $C_I=\{(v_1,\cdots,v_m)\in F_2^m \mid v_i\in F_2, i\in I; v_i=0, i\notin I\}$ is called a Cube. Shamir[1] presents that the variables from V can be any value and for the sake of computational convenience they are usually assigned to 0. The variables from U walk over the values of $F_2^{|I|}$, obviously $|C_I|=2^{|I|}$, then we have:

$$\sum_{(v_1,\cdots,v_m)\in C_I} P(v_1,\cdots,v_m,x_1,\cdots,x_n)=\sum_{(v_1,\cdots,v_m)\in C_I} v^I P_{S(I)}(\cdot)+\sum_{(v_1,\cdots,v_m)\in C_I} Q(\cdot), \quad (1)$$

Cube attack mainly depends on the following observations:

Theorom 1[1] For any polynomial $P(v,k)$ and public variable, we have $\sum_{v\in C_I} P(v,k)\equiv P_{S(I)}(v,k)\bmod 2$.

Proof: Since every term from Q doesn't contain $v_{i_1}v_{i_2}\cdots v_{i_k}$, the value after 2^k times sum of every term from Q is 0, thus $\sum_C Q$ is equal to 0. Next, only if $v_{i_1}v_{i_2}\cdots v_{i_k}$ are all set to 1, the coefficient of $P_{S(I)}(\cdot)$ is 1.#

We can obtain a polynomial $P_{S(I)}(\cdot)$ of the key K and the remain variables V through this kind of sum, while the variables from V are all set to 0, there being a equation of the key K and the output bits. If $P_{S(I)}(\cdot)$ is linear, then v^I is a **maxterm** and the polynomial $P_{S(I)}(\cdot)$ is a **superpoly**.

Example 1 Let $F(x_1,x_2,x_3,v_1,v_2,v_3)=x_1v_1v_2+x_3v_1v_2+x_1v_1v_3+v_1v_2+x_2+1$, where (x_1,x_2,x_3) being the key and (v_1,v_2,v_3) being the public variables, and let I={1,2}, then we have $F(x_1,x_2,x_3,v_1,v_2,v_3)=v_1v_2(x_1+x_3+1)+x_1v_1v_3+x_2+1$, set $C_I=\{(0,0,0),(0,1,0),(1,0,0),(1,1,0)\}$, then

$$\sum_{v\in C_I} F_t(x_1,x_2,x_3,v_1,v_2,v_3)=P_{S(I)}(x_1,x_2,x_3,v_3)=x_1+x_3+1$$

The polynomial of investigated cryptosystem is unknown during the attack, and the whole attack is regarded as an attack to a "black box". Cube attack consists of two phases, the preprocessing phase[14] and the on-line phase[16]. In the preprocessing phase, the attacker can obtain relative output by assigning values to public variables and key bits, and defining appropriate cube to get access to the equation about key bits and public variables. Then he ensure whether $P_{S(I)}(\cdot)$ is linear using the linearity test. In the on-line phase, he can only control the

public variables and the main task is to establish linear equations as many as possible. Due to the selection of different cube sets, he can establish linear equations and then recover the secret key bits or narrow the search space by solving the equation system.

2.2 Grain-v1 Algorithm

Due to the co-creative endeavor of Swedish scholar Hell, Johansson and Swiss scholar Meier, Grain-v1 is a stream cipher algorithm based on hardware implementation. It consists of two shift register with the degree of 80, one of which is linear feedback shift register and the other is non-linear feedback shift register. They generate the key stream after the filter of a non-linear output function. Figure 1 depictures Grain-vl algorithm:

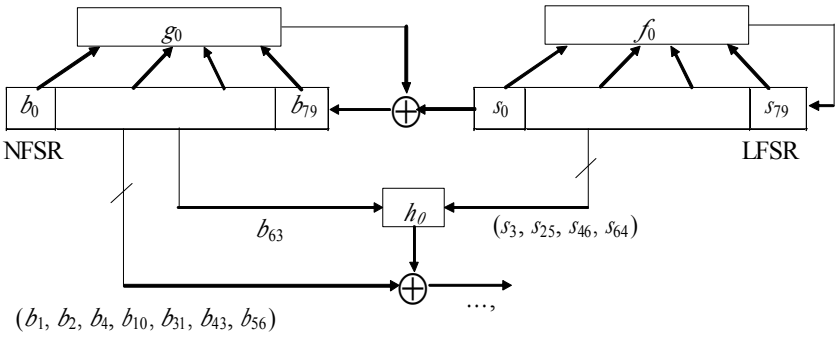

Figure 1 Grain-v1 algorithm

Grain v1 has a 80-bit key and 64-bit initiative value IV. Since the feedback polynomial $f_0(x)$ of LFSR is a premitive polynomial with the degree of 80, the sequence generated by it is a m-sequence with a period of 2^{80}-1. $f_0(x)$ is defined as follows:

$$f_0(x) = 1+x^{18}+x^{29}+x^{42}+x^{18}+x^{57}+x^{67}+x^{80}$$

Relatively, the update of LFSR's inner state is:

$$s_{i+80} = s_{i+62}+ s_{i+51}+ s_{i+38}+ s_{i+23}+ s_{i+13}+ s_i$$

The feedback polynomial $g_0(x)$ of NFSR is a balanced Boonlean function, which defined as follows:

$g_0(x)=1+x^{18}+x^{20}+x^{28}+x^{35}+x^{43}+x^{47}+x^{52}+x^{59}+x^{66}+x^{71}+x^{80}+x^{17}x^{20}+x^{43}x^{47}+x^{65}x^{71}+x^{20}x^{28}x^{35}+x^{47}x^{52}x^{59}+x^{17}x^{35}$
$x^{52}x^{71}+x^{20}x^{28}x^{43}x^{47}+x^{17}x^{20}x^{59}x^{65}+x^{17}x^{20}x^{28}x^{35}x^{43}+x^{47}x^{52}x^{59}x^{65}x^{71}+x^{28}x^{35}x^{43}x^{47}x^{52}\,x^{59}$.

The inner state of NFSR is depend on itself and the output of LFSR, its update is : $b_{i+80}=s_i+b_{i+62}+b_{i+60}+b_{i+52}+b_{i+45}+b_{i+37}+b_{i+33}+b_{i+28}+b_{i+21}+b_{i+14}+b_{i+9}+b_i+b_{i+63}+b_{i+60}+b_{i+37}b_{i+33}+b_{i+15}b_{i+9}+b_{i+60}b_{i+52}b_{i+45}+b_{i+60}b_{i+52}b_{i+45}+b_{i+33}b_{i+28}b_{i+21}+b_{i+63}b_{i+45}b_{i+28}b_{i+9}+b_{i+60}b_{i+52}b_{i+37}b_{i+33}+b_{i+63}b_{i+60}b_{i+21}b_{i+15}+b_{i+63}b_{i+60}b_{i+52}b_{i+45}b_{i+37}+b_{i+33}b_{i+28}b_{i+21}b_{i+15}b_{i+9}+b_{i+52}\,b_{i+45}b_{i+37}b_{i+33}b_{i+28}b_{i+21}$

Inner states of LFSR and NFSR generate the key stream bits through filter function $h_0(x)$, which is a balanced Boonlean function with the correlation immunity degree of 1, algebra degree of 3 and linear complexity of 12. $h_0(x)$ is defined as follows:

$$h_0(x_0 ,\ldots, x_4)= x_1+x_4+x_0x_3+x_2x_3+x_3x_4+x_0x_1x_2+x_0x_2x_3+x_0x_2x_4+x_1x_2x_4+x_2x_3x_4$$

The output of key stream z_t^0 is:

$$z_t^0 = \sum_{k\in A} b_{i+k} + h_0(s_{i+3}, s_{i+25}, s_{i+46}, s_{i+64}, b_{i+63})$$

where A= {1,2,4,10,31,43,56}.

The generation of Grain-v1's key stream consists of two phases: firstly, Grain-v1 initializes the inner state using secret key and IV, and respectively feed the output bits back to LFSR and NFSR. Then the system empty runs 160 rounds and outputs the key stream.

The security of Grain-vl algorithm is previously acknowledged by academics after three rounds of selection and recent attack experiments. Recently, the main attacks toward this algorithm aiming at reducing the initialization rounds, paper[12] applied cube attack to Grain v1 when the initialization rounds is 70 and recover 15 bits of the key. Based on the standard cube attack, this paper can obtain several linear equations using only one cube set.

3. The Improved Cube Attack

According to theorem 1, in standard cube attack we can get one residual polynomial $P_{S(I)}$ by one given cube set, then decide whether it is linear using linearity test, if not, then we renew a C_I to find new residual polynomial. Therefore, only by reasonable selection of index set I, confirming the dimension and maxterm of cube set can we ensure the polynomial is linear. Given an index set I, we can obtain at most one linear equation using standard cube attack, in this paper, we introduce an improved method, through which we can obtain several linear equations with the help of reasonable selected I.

Given a cryptosystem $P(v,k)$, where $v=(v_1,v_2,\cdots,v_m)\in F_2^m$ is public variable, $k=(x_1,\cdots,x_n)\in F_2^n$ is secret key, selecting index set $I\subset\{1,\cdots,m\}$, in general, letting $I=\{1,2,\cdots,k\}$, then we have $v^I=v_1v_2\cdots v_k$, letting set $J=\{1,2,\cdots,k-1\}\subset I$, discussing the relationship between $P_{S(I)}(v,k)$ and $P_{S(J)}(v,k)$:

$$P(v,k)=v^I P_{S(I)}(v_{k+1},\cdots v_m,k)+Q(v,k) \tag{2}$$

Note that $Q(v,k)$ does not contain the multiplicator v^I, combining the monomials that contain the multiplicator v^J in $Q(v,k)$ we have:

$$Q(v,k)=v^J\dot{P}_{S(J)}(v_k,\cdots,v_m,k)+Q'(v,k)$$

Substituting to (2), we have:

$$\begin{aligned}P(v,k)&=v^I P_{S(I)}(v_{k+1},\cdots v_m,k)+v^J\dot{P}_{S(J)}(v_k,\cdots,v_m,k)+Q'(v,k)\\&=v^J\cdot v_k P_{S(I)}(v_{k+1},\cdots v_m,k)+v^J\dot{P}_{S(J)}(v_k,\cdots,v_m,k)+Q'(v,k)\\&=v^J(v_k P_{S(I)}(v_{k+1},\cdots v_m,k)+\dot{P}_{S(J)}(v_k,\cdots,v_m,k))+Q'(v,k)\end{aligned} \tag{3}$$

As for the index set J, factorizing $P(v,k)$ directly, then we have:

$$P(v,k)=v^J P_{S(J)}(v_k,\cdots v_m,k)+Q'(v,k) \tag{4}$$

Combining (3) and (4), the following relationship can be confirmed:

$$P_{S(J)}(v_k,\cdots v_m,k)=v_k P_{S(I)}(v_{k+1},\cdots v_m,k)+\dot{P}_{S(J)}(v_k,\cdots,v_m,k) \tag{5}$$

According to the conclusion of theorem 1, we can conclude as follows:

Lemma 1 The definitions of set I and J are the same as mentioned above, so do the polynomials, when $v_k=0$ we have:

$$P_{S(J)}(x_1,\cdots,x_n)=\dot{P}_{S(J)}(x_1,\cdots,x_n)。$$

Proof: It is confirmed as (5). #

Extending the conclusion of lemma 1 to all of the possible subsets of I, we can conclude as follows, which also being the most important conclusion of this paper:

Theorem 2 Assuming $P(v,k)$ is a Boonlean function over F_2^{m+n}, where $v=(v_1,v_2,\cdots,v_m)\in F_2^m$ being public variable and $k=(x_1,\cdots,x_n)\in F_2^n$ is the key, selecting index set $I\subset\{1,\cdots,m\}$, $|I|=d$, then the polynomial can be uniquely factorized as follows:

$$P(v,k)=\sum_{J\subseteq I} v^J P'_{S(J)}(v,k)$$

For any $J\subseteq I$, in cube set C_J when $v_i=0, i\in I\setminus J$,we have $P'_{S(J)}(v,k)=P_{S(J)}(v,k)$.

Proof: Firstly, ordering the n dimensional vector, letting $a=(a_1,\cdots,a_n),b=(b_1,\cdots,b_n)\in F_2^n$, if $wt(a)>wt(b)$, then $a>b$; when $wt(a)=wt(b)$, if $(a_1,\cdots,a_n)_2>(b_1,\cdots,b_n)_2$, then $a>b$, where $(a_1,\cdots,a_n)_2=\sum_{i=1}^{n} a_i 2^{i-1}$. Mapping the subset of I to vector space $F_2^d:\{J\mid J\subseteq I\}\to F_2^d$, subset J is mapped to a d dimensional vector, the values of corresponding positions of number in J are 1, the values of other positions are 0. Foe example I={1,2,3} $\mapsto$ (1,1,1); subset J={1,3} $\mapsto$ (1,0,1).

Given a index set I, ordering all of the possible subset of I according to the image order of the map mentioned above:

$$I=I_0\supset I_1\supset\cdots\supset I_{2^d-1}=\phi$$

Then we have polynomials:

$$P(v,k)=v^J P_{S(J)}(v,k)+Q_0(v,k)$$

$$Q_0(v,k)=v^{J_1} P'_{S(J_1)}(v,k)+Q_1(v,k)$$

$$Q_1(v,k)=v^{J_2} P'_{S(J_2)}(v,k)+Q_2(v,k)$$

……

Substituting them to the premitive polynomial we have:

$$P(v,k)=\sum_{J\subseteq I} v^J P'_{S(J)}(v,k)$$

According to lemma 1, for any $J\subseteq I$, in cube set C_J when $v_i=0, i\in I\setminus J$,we have $P'_{S(J)}(v,k)=P_{S(J)}(v,k)$. #

Letting $k=x$ in(1), $p(v,k)$ is simplified as polynomial $p_{I,x}(\upsilon)$ containing cube variables. In theorem 1, when public variables walk over C_I, we can obtain the truth table of $p_{I,x}(\upsilon)$ if the values of output function are stored. Then we have $p'_{S(J)}(x),\forall J\subseteq I$ through converting truth table to ANF. By doing so, not only do we improve the efficiency of attack, but also avoid a crucial problem: the dimension of index set I. After the linearity test, we keep linear functions and leave out non-linear functions or constants. According to theorem 2, we can dramatically improve the efficiency of cube attack since the procedure of repeatedly selecting of index set I is left out.

4. Application of Improved Cube Attack to Simplified Grain v1

According to the analysis above, when applying improved cube attack, the attacker only need to select index set with the largest dimension within his computational ability for the searching of

linear superpoly. This paper applies improved cube attack to Grain-v1 when the initialization rounds is 75 using the PC with its dominant frequency of 2.1GHz, RAM of 1G and dual-core, the procedures are as follows:

Preprocessing phase. The attacker can change the values of public variables and secret key, in order to find appropriate index set. The main tasks including selection of index set, linearity test and solve ANF.

1）、Selection of index set. According to the computational ability, we choose index set with $d_{\max} = 16$, then using random method of searching.

2）、Linearity test. We apply BLR linearity test to this paper, arbitrarily, independently and evenly select vector x, y over F_2^m to verify the equation:

$$p_{SI}(x) \oplus p_{SI}(y) \oplus p_{SI}(0^{(n)}) = p_{SI}(x \oplus y)$$

The attacker ensures that $p_{SI}(k)$ is linear, if the test always succeeds.

According to lemma 1 and theorem 2, when testing $p_{SI}(k)$, we can also test all $p_{SJ}(k), J \subseteq I$. We find 19 linear superpolies in nearly a month, details are in table 1.

3）、Solve ANF. According to the index set obtained from last step, we compute their ANF respectively, the procedures are as follows:

a 、 Coefficient of constant terms. Letting key $k^{(n)} = 0^{(n)}$, we have $a = \sum_{v \in C_I} p(v, \omega, 0^{(n)})$;

b、Coefficient of terms with first degree. Letting $k_i = 1, k_j = 0, j \in \{1, 2, \cdots, m\} \setminus \{i\}$, we have $a_i = \sum_{v \in C_I} p(v, \omega, k)$.

According to the procedures above, the ANFs of linear superpolies are in the following table:

Table 1 Attack Result of Grain-v1 （19 index sets）

Index set	Superpoly
12, 15, 20, 34, 40, 46, 62,	63,
0, 12, 15, 17, 20, 23, 34, 37, 40, 44, 49,	66, 1,
3, 9, 12, 15, 18, 30, 34, 37, 40, 45, 50,	67, 1,
0, 11, 12, 15, 27, 32, 34, 40, 44, 51,	68, 1,
7, 8, 12, 15, 19, 26, 27, 34, 37, 40, 52, 56,	69, 1,
12, 15, 33, 34, 40, 53, 62,	70, 1,
6, 12, 15, 16, 30, 34, 39, 40, 43, 45, 54,	71, 1,
1, 14, 15, 34, 36, 37, 39, 40, 48, 56, 59, 62,	73, 1,
11, 12, 15, 30, 32, 34, 36, 40, 42, 43, 57,	74, 1,
3, 5, 9, 12, 13, 15, 18, 30, 34, 37, 40, 45,	75,
4, 11, 12, 15, 19, 21, 31, 32, 33, 34, 40, 48, 59,	76, 1,
12, 14, 15, 25, 32, 34, 39, 40, 60,	77,

Index set	Superpoly
8, 9, 12, 15, 16, 19, 21, 31, 34, 40, 44, 45,	78,
12, 14, 15, 23, 27, 31, 34, 36, 40, 45, 49, 57,	79,
2, 5, 8, 13, 22, 23, 27, 33, 34, 37, 40, 45, 54,	2, 3, 5, 11, 32, 44, 57, 1,
7, 8, 12, 15, 19, 26, 34, 37, 40, 48, 56,	5, 6, 9, 11, 14, 17, 35, 38, 47, 50, 60, 63, 1,
8, 12, 15, 19, 26, 27, 34, 37, 40, 48, 56,	8, 9, 11, 17, 38, 50, 63,
7, 12, 13, 14, 18, 19, 29, 30, 36, 37, 39, 48, 62,	9, 10, 12, 18, 39, 51, 64,
7, 13, 14, 18, 19, 29, 30, 36, 37, 39, 46, 48, 62,	10, 11, 13, 19, 40, 52, 65,

On-line phase. We obtain p_I through "black box" then establish equation $\sum a_i k_i \oplus a = p_I$. At last, we recover the secret key bits or narrow the search space by solving the equation system.

For Grain-v1 algorithm when initialization rounds is 75, according to the result obtained from preprocessing phase and the equations established at on-line phase, we can directly recover 14 bits of secret key and find 5 linear expressions of another 28 bits of the key, which means we can recover 19 bits of the key in all. In addition, we applied improved cube attack to Grain-v1when the initialization rounds ranking from 70 to 74, costing nearly a month and the results are as follows:

Table 2 Attack Results of Simplified Grain-v1

Initialization rounds	70	71	72	73	74
Number of linear polynomials	22	22	21	19	17

We can conclude that both the efficiency and result of our attack have remarkable improvement when compared to those of standard cube attack.

5. Conclusion

In this paper, we simply introduced Grain-v1algorithm and the improved cube attack. We applied the improved cube attack to Grain-v1and directly recovered the key bits as well as obtained linear equations of key bits. Besides, how to effectively find superpolies and analyze the security index of the resistance of cube attack are the following research focus.

6. References

[1]. Dinur I, shamir A. Cube attack on Tweakable Black Box Polynomials[C]. In EUROCRYPT 2009, LNCS5479, 278-299.

[2]. Aumasson J P, Meier W, Dinur I. Cube Testers and Key Recovery Attacks on Reduced Round MD6 and Trivium[C]. International Workshop on Fast Software Cryptology Encryption 2009, Springer 2009 LNCS, 1-22.

[3]. Mroczkowski P, Szmidt J. The Cube Attack on Courtois Toy Cipher[EB/OL]. Cryptology ePrint Archive, 2009. http://eprint.iacr.org/2009/497.

[4]. Karmakar S, Mukhopadhyay D, Chowdhury D R. Cube Attack on A Simplified Version of Trivium[C]. National Workshop of Cryptology 2010, Coimbatore, India.

[5]. Canniere D C, Preneel B. Trivium – a stream cipher construction inspired by block cipher design principles[C]. New Stream Cipher Designs--The eSTREAM Fanalists, Springer, 2008.
[6]. Haixin Song, Xiubin Fan, Chuankun Wu, Dengguo Feng. Cube Attack on Stream Cipher Algorithm Grain[J]. Journal of Software, 2012, 23(1), 171-176.
[7]. Heel M, Johansson T, Meier W. Grain—A Stream Cipher for Constrained Environments[C]. New Stream Cipher Designs--The eSTREAM Fanalists, Springer, 2008.
[8]. Dinur I, Shamir A. Side Channel Cube Attacks on Block Ciphers[EB/OL]. Cryptology ePrint Archive, 2009. http://eprint.iacr.org/2009/127.
[9]. Daemen Rijmen Vincent. AES Proposal: Rijndael[C]. Technical Evaluation, CD-1: Documentation, 1998.
[10]. Anderson R, Biham E, Knudsen L. Serpent: A Proposal for The Advanced Encryption Standard[C]. First Advaced Enryption Standard(AES) Conference, 1998.
[1]. Lathrop J, Cube Attacks on Cryptographic Hash Functions[EB/OL]. Master's Thesis in Computer Science, 2009. http://www.cs.rit.edu/~jal6806/thesis/.
[11]. Yu Sun, Yongjuan Wang. The Theory and Improvement of Cube Attack[J], Computer Science, 2012, No 6A, 77-80.
[12]. Englund H, Johansson T, Turan M. A Framwork for Chosen IV Statistical Analysis of Stream Cipher[C]. INDOCRYPT, LNCS 4859, Springer, 2007, 268-281.
[13]. Vielhaber M. Breaking ONE. FIVIUM by AIDA An Algebraic IV Differential Attack[EB/OL]. Cryptology ePrint Archive, 2007. http://eprint.iacr.org/2007/413.
[14]. Fischer S, Khazaei S, Meier W. Chosen IV Statistical Analysis for Key Recovery Attacks on Stream Cipher[C]. ADEICACRYPT, LNCS5023, Springer, 2008, 236-245.
[15]. Dinur I, Shamir A. Breaking Grain-128 with Dynamic Cube Attacks[C]. International Workshop on Fast Software Cryptology Encryption, 2011.