RC6 CIPHERING

Mohammud A.AL-Jabbar A.AL-Wahab
Iraqia university/Islamic science Faculty
AL-Hadeeth dept.

Abstract:

The focus of this research concentrated on the issue of encryption using clever methods to increase the proportion of the complexity of the algorithm. This is done by achieving more than the implementation phase and more than one key so as to make the issue of breach the code is very difficult to know the content of the letter of information, and there are three main factors that control in this topic a number of encryption sessions, the key length, the length of the encrypted word, it was synthesized according to the algorithm .

1.Introduction

These comments are an expanded version of the comments we submitted to the Third AES Candidate Conference , security is not only the most important, but also the most difficult characteristic to compare . In the absence of any theoretical ways of measuring security [1], we can only fall back on estimates & guesses. " I can't break this algorithm , & all those other smart people can't either" is the best we can say. Hence , all discussions about security rely on this type of non-rigorous argument.

When looking at the published cryptanalysis on the AES finalists, it is important to keep in mind what the data mean.

Historically , cryptanalytic results against any logarithm have improved over time initial results might cryptanalyze a simplified variant of the algorithm, or a version of the algorithm with fewer round [5] . Later results improve on those initial results : more rounds or less simplification . Finally, there may be a successful attack against the full algorithm .

This is why the published cryptanalysis against the AES finalists , even though none of the results approach practicability & none of the attacks are of any use against the full version of the algorithms , are so important. By comparing how close the published attacks come to break the full algorithms, we can get some inkling about how the algorithms' security compares. This is not a perfect comparison by any means.

Safety factors

The best measure of security that we have come across is the **safety factor** first compared the AES candidates in this manner when he calculated the "minimal secure rounds" [1]. Lars Knudsen also used this factor when he discussed the AES candidates in his first-round comments [10].

2. The RC6 Algorithm

Like RC5, RC6 is a fully parameterized family of encryption algorithm. A version of RC6 is more accurately speci_ed as RC6-w/r/b where the word size is w bits, encryption consist of a non-negative number of rounds r, & b denotes the length of the encryption key in bytes. Since the AES submission is targeted at w = 32 & r = 20, we shall use RC6 as shorthand to refer to such versions .

When any other value of w or r is intended in the text , the parameter values will be speci_ed as RC6-w/r. of particular relevance to the AES e_ort will be the versions of RC6 with 16-, 24-, & 32 byte keys.

For all variants , RC6-w/r/b operates on units of four w-bit words using the following six basic operations. The base-two logarithm of w will be denoted by lgw.

a + b integer addition modulo 2w

a b integer subtraction modulo 2w

a _ b bitwise exclusive-or of w-bit words

a _ b integer multiplication modulo 2w

a<<<b rotate the w-bit word a to the left by amount given by the least signi_cant lgw bits of b**.**

a>>>b rotate the w-bit word a to the right by the amount given by the least signi_cant lgw bits of b .

note that in the description of RC6 the term \round" is somewhat analogous to the usual DES-like idea of a round : half of the data is updated by the other half ; & the two are then swapped . In RC5, the term \half-round" was used to describe this style of action , & an RC5 round was deemed to consist of two half-rounds. This seems to have become a potential cause of confusion , & so RC6 reverts to use the term \round" in the more established way.

Key schedule

The key schedule of RC6-w/r/b is practically identical to the key schedule of RC5-w/r/b. indeed, the only di_erence is that for RC6-w/r/b, more words are derived from the user-supplied key for use during encryption & decryption .

The user supplies a key of b bytes, where 0 _ b _ 255. From this key , 2r + 4 words (w bit each) are derived & stored in the array S[0; : : : **;** 2r + 3]. This array is used in both encryption & decryption.

Encryption & decryption

RC6 works with four w_bit registers A;B;C;D which contain the initial input plaintext as well as the output chiphertext at the end of encryption . The _rst byte of plaintext or ciphertext is placed in the least-signi_cant byte of A; the last byte of plaintext or ciphertext is placed into the most-

signi_cant byte of D . We use (A;B;C;D) = (B;C;D;A) to mean the parallel assignment of values on the right to registers on the left .

Encryption with RC6-w/r/b

Input : plaintext stored in four w-bit input registers A;B;C;D number of rounds
w-bit round keys S[0;:::;2r + 3]
Output ciphertext stored in A;B;C;D
Procedure : B =B + S[0]
D = D + S[1]
for I = 1 to r do
f
t = (B _ (2B + 1))<<<lg w
u = (D_(2D + 1))<<<lg w
A = ((A _ t)<<<u) + S[2i]
C = ((C _ u)<<<t) + S[2i+ 1]
(A;B;C;D) = (B;C;D;A)
g
A = A + S[2r + 2]
C = C + S[2r + 3]

Our decorder will decript four bytes at a time. The decryption is governed by a multi-bit key. This key is used to pre-calculate a set of constants that are used through the decryption (once the constants are calculated , the key is no longer needed).

For this research , you don't need to worry about how the constants are calculated ; I have already done that & will just give you the constants , & you will hard-code them in your chip. For a decoder with four rounds (such as ours) there arec12 constants ; 8 inside the datapath & 4 outside the datapath. If you want to read more about the constant calculation , see the abovereferenced document on the web.

The block cipher RC6 was proposed by Rivest et al . in [8] to meet the requirements of the Advanced Encryption Standard (AES) & is one of the finalists of the AES candidates . It has been admired for its high-level security & high speed software implementation especially on intel CPU .

RC6 is designed based on the block cepher RC5 [7] which makes essential use of arithmetic key additions & data-dependent rotations.

As additional primitive operations to RC6, the inclusion of arithmetic multiplications & fixed rotations is believed to contribute the strength of the security of RC6. there are some crypt analysis of RC6 : resistance against differential attack , Related Key Attack [2, 3],Linear Attack[1, 2, 3] , Mod n Attack [5], & statistical Attack [4]. Shimoyama et al. [9] evaluated the resistance of RC6 with 256-bit key against multiple linear attack & showed that the target key of 14-round RC6 can be recovered &

also that the target key of 18-round RC6 with weak keys , which exits with probability 1=290 at least, can be recovered . One of the most effective attacks is an attack based on Â2 test. This attack was originally proposed by Vaudenay [10], & was applied to RC6 by Gilbert et al. [4] & Knudsen & Meier [6] , independently. In [6] , Knudsen & Meier can cryptanalyze up to 15-round RC6 with general keys & 17-round RC6 with weak keys. Wecall this attack "Â2 attack" shortly in this paper.

We enumerate attacks on RC6 in table 1. In thispaper we study Â2 attack against RC6 more precisely.

Knudsen & Meier [6] experimented with 2- , 4- & 6-round RC6 by Â2 test & estimated the sample complexity necessary to distinguish (2r + 3)-round RC6 from random permutation at 216:2r+13:8. However , the way of security evaluation against at Â2 attack has not been known except the computer experiment . We analyze the sample complexity (in the sense of chosen plaintext) of Knudsen & Meier's Â2 attack on RC6 more precisely . We introduce a novel technique "transition matrix computing " to evaluate the expected Â2 value , & to estimate the sample complexity with respect to any fixed key . We show that the sample complexity with respect to the average key to distinguish (2r+3)-round RC6 from random permutation is at most 216:0198r+13:1094. We note that the sample complexity 216:2r+13:8 estimated by Knudsen & Meier is quite close to our results though their value is drown from the 20 trials . In addition , Knudsen & Meier indicated that the key , by which the least significant five bits of round key is zero , is weaker than keys in average . and , for these weak keys , they estimated 17-round RC6 can be distinguished from a random permutation with less than the sample complexity 2118. Using our method , we can show that such weak keys , mentioned by Knudsen & Meier , are actual "weakest keys" . Moreover , we show that there exist weak keys in 17-round RC6 whose fraction is 1=269:8747, which can be distinguished by using less sample complexity than 2118 . Therefore , it is said that weak key ratio is about 1024 times larger than the weak key ratio mentioned by Knudsen & Meier (1=280).

We set m = 1024 & compute μ(Á(r)) from randomly chosen 1000 user keys . We also compute the averages & the derivations from the experimental data . By using these averages , we calculate the value of n satisfying that E[Â2(Xn)] = μ(Á(r)) ¢ n + 1023 = 1098 , that means the average sample value exceeds 1098 . The logarithm of the sample complexity to distinguish 2r- round RC6-32 from random permutation is almost linear in r. using the least square method , we obtain that the sample complexity to distinguish (2r + 3)-round RC6-32 from random permutation is 216:0198r+13:1094.

We note that the sample complexity 216:2r+13:8 estimated by Knudsen & Meier is quite close to the theoretical value though their value is drown from the 20 trials for 2- & 4-round RC6-32.

In [6] , Knudsen & Meier estimated the sample complexity in case of 2-round RC6-32 at 213 . They also experimented with 4-round RC6-

Then , we can show that there exist weak keys in 17-round RC6-32 whose fraction is one over 269:8474 which can be distinguished by using less sample complexity than 2118.

Therefore , it is said that the weak key ratio is about 1024 times larger than the weak key ratio (one in 280 keys) mentioned by Knudsen & Meier [6] . on the other hand , for 19-round RC6-32 , the ratio of such weak keys is one in 2569:3812, that is much fewer than whole size of user keys. So such weak keys don't exist .

Now we study about the weak keys mentioned by Knudsen & Meier such that the least significant five bits of extended keys are zero for every 2 rounds. In our method, the value of E[Â2(Xn)] depends only on each of the input value Isb5(A;C) & lsb5(S[4i + 2]; S[4i + 3]) for i from 0 to br=4c. Therefore, it is possible to evaluate the security of RC6-32 against the Â2 attack for any key. Our results of the "alert" security evaluation for almost all keys imply that the weak keys mentioned by Knudsen & Meier are actual "weakest keys". The column of "weakest" in table 7 shows its security level for each round . Table 7 shows that the distinguishing attack can not be applicable to RC6-32 more than 18 rounds even in the case of the weakest key . Therefore , we are able to prove that the 20-round RC6-32 is secure against Â2 attack if it is shown the 3- round elimination attack (including the key-recovery algorithm) can not be applicable .

For the end of this section , we comment the randomness of extended keys of RC6-32. *the equation lsb5(S[4i + 2]; S[4i + 3]) = 0 holds. We avoid the difficult* In this paper, we use the number 2¡10 for the ratio of the user keys such that the related extended keys S[4i + 2]; S[4i + 3] satisfy *equation* lsb5(S[4i + 2]; S[4i + 3]) = 0 holds. We avoid the difficulty of theoretical analysis about the distribution of extended keys, since the key schedule part of RC6-32 is very complex. Instead, we adopt computer-experimental ratio. Our results show that the user keys, for which isb5(S[2i]; S[2i + 1]) 0 hold, exist one in 29:9994 on average , & its variance is 2¡34:296. These results are obtained from the experiment with sampling on 230 user keys of RC6-32 using 128-bit key such that (key[0];key[1];key[2];key[3]) = (0; 0; 0; 0); ¢ ¢ ¢; (0; 0; 0; 0; 230 ¡ 1).

From this , we can say that each of the least significant five bits are uniformly distributed. This implies validity of our assumption .

*4.***Software implementation by c++**

We have two programs one for encryption named final RC6 encryption & another for decryption named final RC6 decryption & text file named rc6enc.text saved in partition c:\ directly for encryption & decryption saving for example write iraqiraqiraq in the text file.

At start must do the following :-

1-open the file final rc6 encryption

2-press f7 to compile the source code to get zero error message.

3-press control+f5 to on the program, then the following background appear as shown in figure (1).

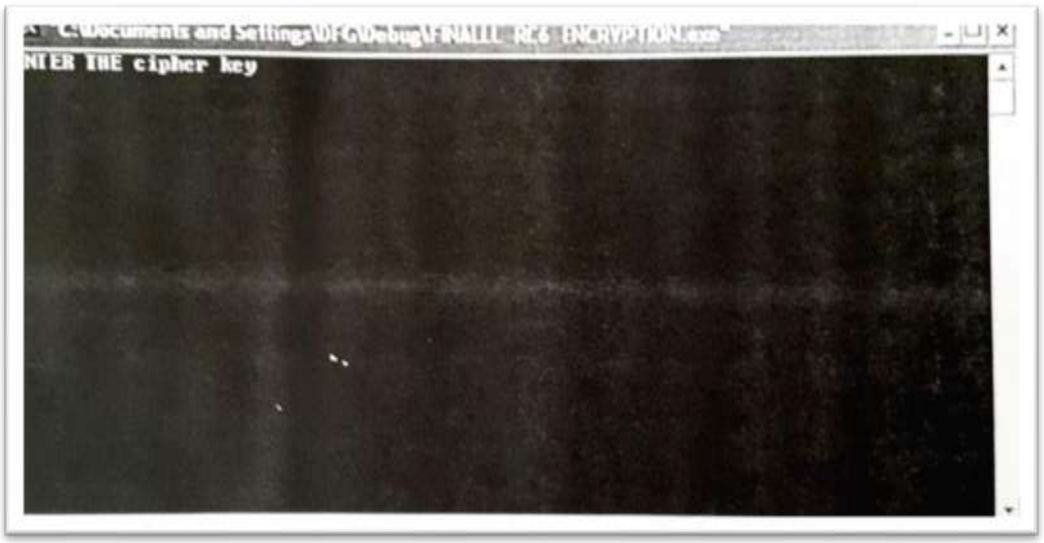

Figure (1) Window for entering cipher key in encryption step

4-enter any key you of length >=4 like asdfghjk as a key.

5-press enter.

The following appear as shown in figure (2).

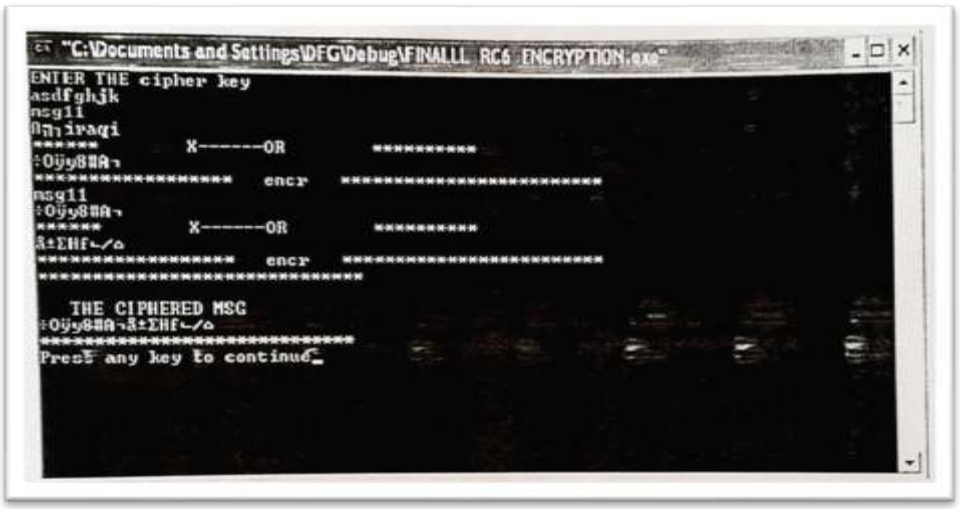

Figure (2) Window shows the encryption message

This ciphered text is saved in the file rc6enc.text

The decryption steps

1-open the file final rc6 decryption .
2-press f7 to compile the source code to get zero error message.
3-press control+f5 to on the program , then the following background appear as shown in figure (3).

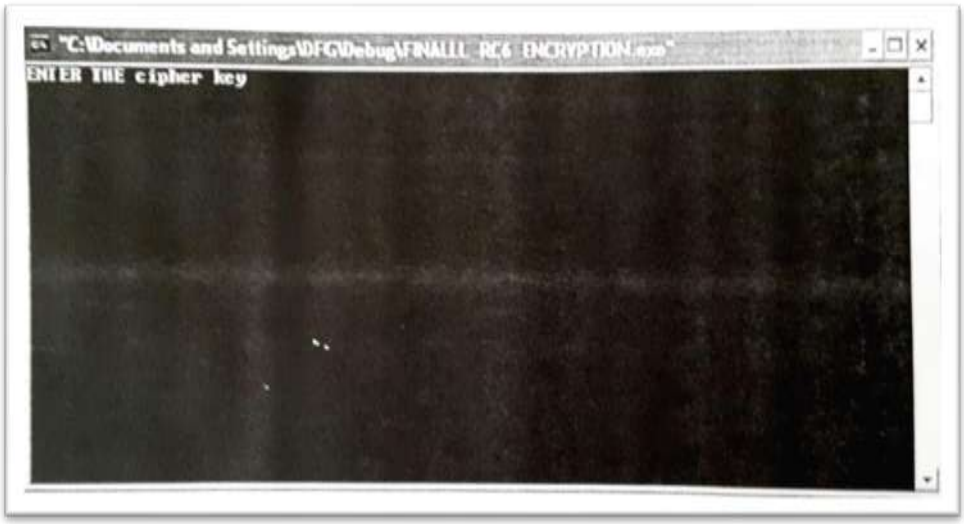

Figure (3) Window for entering cipher key in decryption step

4-enter the same key you entered above.
5-press enter.

The following appear the same as printed firstly iraqiraqiraq as shown in figure (4).

Figure (4) decrypted message in decryption step

5- References

1. E. Biham & A. Shamir. Di_erential cryptanalysis of the data encryption stan-dard. Springer-Verlag, New York , 1993.
2. A. Biryukov & E. kushilevitz. Improved cryptanalysis of RC5. In K. Nyberg, editor , Advances in cryptology | Eurocript '98, volume 1403 Lecture Notes in computer science, pages 85{99, 1998. Springer verlag.
3. S. contini , R.L. Rivest , M.J.B. Robshaw & Y.L. Yin. The security of the RC6 TM Block Cipher. Version 1.0. August 20, 1998.
4. Hewlett Packard . strategy description , May, 22 , 1997.
5. M.H. Heys. Linearly weak keys of RC5. IEE Electronic Letters, Vol. 33, pages 836{ 838, 1997.
6. Intel corporation. MCS 51 Microcontroller Family User's Manual . February 1994.
7. Intel corporation. The Next Generation of Microprocessor Architecture. October, 1997.
8. B.S. Kaliski & Y.L. Yin. On di_erential & linear cryptanalysis of the RC5 encryption algorithm. In D. coppersmith , editor , Advances in cryptology| crypto '95, volume 963 of Lecture Notes in computer science , pages 171{184,1995. Springer Verlag.
9. B.S. Kaliski & Y.L. Yin. On the security of the RC5 Encryption Algorithm.
10. L.R. Knudsen . Truncated & higher order di_erentials. In B. preneel , editor , Fast software encryption, volume 1006 of Lecture Notes in computer science , pages 196{211 , 1994. Springer Verlag.

التشفير باستعمال خوارزمية : (RC6)

م. محمد عبد الجبار عبد الوهاب
الجامعة العراقية/ كلية العلوم الإسلامية

الملخص:

يتمركز محور هذا البحث حول مسالة التشفير باستخدام اساليب ذكية لزيادة نسبة تعقيد الخوارزمية، ويتم هذا من طريق تحقيق اكثر من مرحلة تنفيذ، وبأكثر من مفتاح، وذلك لجعل مسالة اختراق الشفرة صعب جداً؛ لمعرفة ما تتضمنه الرسالة من معلومات، وهناك ثلاثة عوامل رئيسة تتحكم في هذا الموضوع، وهي عدد دورات التشفير، وطول المفتاح، وطول الكلمة المشفرة التي جرى توليفها حسب الخوارزمية.