Enhancement of Known IV Attack for WEP-like Systems

Ching-Nung Yang, Chen-Chin Kuo and Tsung-Yuan Cheng
Department of Computer Science & Information Engineering
National Dong Hwa University
1, Sec.2, Da Hsueh Rd., Shou-Feng, Hualien
Taiwan, Republic of China
TEL: 886-3-8634025 Fax: 886-3-8634010
cnyang@mail.ndhu.edu.tw

Abstract

Wired Equivalent Privacy (WEP) protocol is used to assure the privacy of IEEE 802.11a/b based on RC4 stream cipher. Fluhrer [1] proposed the known IV attack using the first output of RC4 in WEP to derive the secret key. However, if one avoids the usage of the first output of RC4, the attacker will get nothing using this type of attack. In this paper we extend the known IV attack to use other outputs of RC4. We also use the different outputs of RC4 simultaneously to reduce the required number of weak IVs in the known IV attack from 60 to 48 for 50% successful probability of deriving the secret key.

Keywords : IEEE 802.11a/b, Wired Equivalent Privacy, WEP2, RC4, Known IV attack.

1. Introduction

The key component of WEP is RC4 [3] which is the most widely used stream cipher and also used in other applications, i.e., Secure Socket Layer (SSL). The reason of using RC4 stream cipher in WEP is that the mobile device is battery-limited and cannot provide powerful computation. So, WEP adopts RC4 instead of the block cipher or public key system to speed up encryption. The input of RC4 in WEP is the public value known as an initialization vector (IV) (24 bits in WEP) followed by the secret key (40 or 104 bits in WEP). The output of RC4 (called a "keystream") is used to exclusive-or (XOR) the plaintext. In the meantime, there are two effective methods to crack WEP. One is "Keystream Reused Attack" [4] and the other is "known IV attack" [1]. The reason for the flaw of keystream reused attack is due to IV collisions. Unfortunately, the collisions will happen too often in real wireless environment. To overcome the flaw, WEP2 [5] is proposed to improve WEP by extending the length of IV from 24 bits to 128 bits and the secret key from 40 bits to 128 bits.

The increase of length for IV and key is not effective against the other flaw, the known IV attack, because the known IV attack uses the first output word of RC4 to derive the key. However, this type of attack may become useless if the user does not use the first output. Here, we extend the attack to use other outputs of RC4 for key extraction and also reduce the number of required IVs in the known IV attack.

This paper is organized as the following. In

Section 2 we describe the WEP protocol and the known IV attack. In Section 3 the extended known IV attack and the simulated results are given. In Section 4 we reduce the number of required IVs for recovering the correct key by combining all effective outputs simultaneously. Finally, a conclusion is given in section 5.

2. The WEP Protocol and Known IV Attack in IEEE 802.11a/b

2.1 WEP Protocol Review

WEP is used to assure the privacy in IEEE 802.11a/b on peer-to-peer or Ad-hoc environment. Figure 1 shows the WEP protocol.

WEP almost relies on RC4. It is important to know how RC4 operates. RC4 algorithm consists of Key Scheduling Algorithm (KSA) and Pseudo Random Generation Algorithm (PRGA). Figure 2 shows both algorithms.

2.2 The Known IV Attack

The attacker can use the first output of RC4 to get the secret key in WEP [1]. First we describe a specific situation of S. Now if S is in a permutation like Figure 3, after the operation of KSA, the value A″ will be the first output word of PRGA. The situation is defined in [1] as the *resolved condition*. The attacker can use the IVs satisfying the *resolved condition* to derive the secret key.

Represent the permutation S after round i of KSA as S_i. If S_i satisfies the following two constraints:

1) $A=S_i[1]$,
2) $A+A'=S_i[1]+S_i[S_i[1]]$,

then the permutation S after the operation of KSA will satisfy the *resolved condition* with 5% probability. The reason is that the probability that none of S[1], S[A], and S[A+A′] will participate in any further swaps is greater than e^{-3} about 0.05.

For example, in WEP-like system, suppose that we have an I-byte IV and a (l-I)-byte secret key (a 3-byte public IV and 5-byte secret key in WEP and a 16-byte public IV and a 16-bye secret key in WEP2), and we want to derive $k[B]$, $0 \leqq B \leqq l\text{-}I\text{-}1$. Figure 4 shows the attacking process. In [1], it is shown that the IV with some certain format, i.e., weak IV, may satisfy the *resolved condition* with high probability. As discussed in [1], the weak IV pattern is as follows (B+3, 255, X), where X is any value from 0 and 255.

3. Extensive Known IV Attack

We require the length of IV with at least (m+1)-byte when using the known IV attack for the m-th output. The reason will be explained in section 3.2.

In this section we extend the known IV attack to use the m-th ($2 \le m \le I$-1, if IV is I-byte) output in WEP-like system. That is using the second output in WEP and the m-th output ($2 \le m \le 15$) in WEP2. We also define our new *resolved conditions* for the extensive known IV attack with different outputs.

3.1 Known IV Attack using the Second Output of RC4 in WEP

When the user does not use the first output of RC4 to encrypt the plaintext for avoiding the known IV attack, we herein extend the attack by using the second output to derive the secret key. When the array S is in a specific permutation like Figure 5, after swapping operation of KSA,

the A_2'' will be the second output of RC4. Here, we define this situation as the *2nd resolved condition* and use the IVs satisfying the *2nd resolved condition* to derive the secret key.

Now if S_i satisfies the following constraints:

1) $A_1+A_2= S_i[1]+S_i[2]$,
2) $A_2+A_2'=S_i[2]+ S_i[S_i[1]+S_i[2]]$,
3) $A_1 \neq 2, A_1 \neq A_1+A_2$,

then the permutation S after the operation of KSA will satisfy the *2nd resolved condition* with e^{-4} probability. The reason is that the probability that none of S[1], S[2], $S[A_1+A_2]$ and $S[A_2+A_2']$ will participate in any further swaps is greater than e^{-4}.

Figure 6 shows the attacking process. Step 1 and 2 are in search of IVs which cause $S_I[1]+S_I[2]<I$, $S_I[2]+S_I[S_I[1]+S_I[2]]=I+B$, and $S_I[1] \neq 2$, $S_I[1] \neq S_I[1]+S_I[2]$ after the first I rounds. If S_I cannot satisfy the constraints, we discard it and find another suitable one. After step $I+B+1$, S_I will be in the *2nd resolved condition* with high probability and the most probable second output of RC4 will be $S_{I+B}[j_{I+B}+k[B]+ S_{I+B}[I+B]]$. If we have the second output byte Z_2 (as explained in the previous section, it is reasonable to know the second output of RC4 in IEEE 802.11a/b due to the known values of some headers), we can calculate $k[B]$ as $S_{I+B}^{-1}[Z_2]-j_{I+B}-S_{I+B}[I+B]$.

Next, we estimate how many IVs we need to collect for 240 weak IVs with the format (X, 252-B, 4+2B-2X) in WEP system. Because the public IV is three bytes for WEP, no matter how IVs are generated in a little or big endian counter , we will collect one weak IV every $2^{8\times2}$ IVs. Therefore, we have to collect $240\times2^{8\times2}$ ($\approx2^{24}$) packets to get 240 weak IVs for deriving the secret key.

The extensive known IV attack can be extended to the m-th output ($3 \le m \le 15$) of RC4 to get the secret key (This part can be found in the final version).

3.2 Simulated Results

In this section, we compare the known IV attacks using the m-th ($1 \le m \le 15$) output of RC4 in WEP2. The compared parameters used in this section are defined as follows:

P_{rc}: the probability of the *m-th resolved condition* which is not disturbed after the state array S has been computed at the end of KSA.

N_{IV}: the minimal required number of weak IVs to achieve 50% successful probability of recovering the secret key.

N_{LEC}: the number of collected packets to get N_{IV} weak IVs to recover the key if the value of public IV is increased in a little endian counter.

N_{BEC}: the number of collected packets to get N_{IV} weak IVs to recover the key if the value of public IV is increased in a big endian counter.

The format of IVs used in our simulation are created by the following IV Generation Algorithm (IGA) shown in Figure 7 when $m \ge 2$. First, find S′ that satisfies the *m-resolved condition* and then uses S′ to calculate the required IV formats. For example, consider the 3-th output (m=3), I=6, and B=1, we will obtain the weak IV formats as (X_0, X_1-X_0-1, X_0-2X_1-19, 15-2X_0+X_1, X_2, …, X_{13}) where $X_i \in \{0,\ldots,255\}$ and $X_0 \neq X_1$.

We get N_{IV} by simulation. We write a function which returns the correct $k[B]$ with probability $P_{rc}+(1-P_{rc})/256$ and regard the most frequent value as the correct one. Then we do this experiment many times to get N_{IV}. In our

simulation, we do 100 times and get the correct result more than 50 times. We will say that the successful probability of recovering the secret key is 50%.

Table 1 shows the compared results for the known IV attack with *m*-th output in WEP2. P_{rc} is $e^{-(m+2)}$ because the number of elements that are not swapped in S for the *m-th resolved condition* is *m*+2. Hence, if we use the later output of RC4 to attack, P_{rc} is decreased and we have to collect more weak IVs, i.e., larger N_{IV}.

If IVs are generated in a little endian counter, we will collect one weak IV format generated by IGA for the *m*(≥2)*-th resolved condition* every 2^{8m} IVs. Therefore, we have to collect $N_{IV}\times 2^{8m}$ packets to get enough weak IVs. If IVs are generated in a big endian counter, we will get $2^{8(I-(m+1))}$ weak IVs every $2^{8(I-(m-1))}$ IVs. So, we have to collect $\left\lceil \frac{N_{IV}}{2^{8(I-(m+1))}} \right\rceil \times 2^{8(I-(m-1))}$ packets to derive the secret key.

For example *m*=2, if IVs are generated in a little endian counter, we will collect one weak IV with the format (X_0, 252-*B*, 4+2*B*-2X_0, X_1, X_2, …, X_{13}) every $2^{8\times 2}$ IVs. Therefore, we have to collect $240\times 2^{8\times 2}$ ($\approx 2^{24}$) packets to get 240 weak IVs. If IVs are generated in a big endian counter, we will collect $2^{8\times 13}$ IVs with the format (X_0, 252-*B*, 4+2*B*-2X_0, X_1, X_2, …, X_{13}) every $2^{8\times 15}$ IVs. So, we have to collect $\left\lceil \frac{240}{2^{8\times 13}} \right\rceil \times 2^{8\times 15}$ ($\approx 2^{120}$) packets to derive the secret key.

For *m*=13, 14, and 15, the values of N_{LEC} are greater than 2^{128}. We cannot get enough weak IVs within all possible 16-byte IVs and the known IV attack is useless.

Figure 8 shows the values of N_{BEC} and N_{LEC} using the *m*-th output to attack in WEP2. It is observed that if the value of IV is increased in little endian counter, using the first output to attack is the best choice. However, if one uses IV in big endian counter, using the later output to attack will be the optimal choice.

4. Improvement of the Extensive Known IV Attack

In section 3 we use a single output, the *m*-th output, in the Known IV attack. Here, we use two or more outputs simultaneously to reduce the value of N_{IV}.

4.1 Extensive Known IV Attack Using the Combination of the First Two Outputs for WEP System

For WEP system (3-byte IV, 5-byte secret key), *m*(≧3)-th output cannot be used because the length of public IV is only three. Hence, we consider the combination of the first and second outputs to enhance the known IV attack in WEP system.

Suppose that the attacker get n_2 IVs with the format (*X*, 252-*B*, 4+2*B*-2*X*) when already collecting n_1 IVs with the format (*B*+3, 255, *X*). He now analyzes n_1+n_2 IVs simultaneously to get the most frequent value as the secret key.

Consider that IV is generated by a big endian counter. Because N_{BEC} of the first and second outputs are 2^{19} and 2^{24}, respectively, we will first finfish the collection of $N_{IV(1st)}$ (N_{IV} of the first output) IVs. Since we can collect one IV with the format (*X*, 252-*B*, 4+2*B*-2*X*) every 2^{16} IVs, so we get only 8 ($=\frac{2^{19}}{2^{16}}$) IVs with the format (*X*, 252-*B*, 4+2*B*-2*X*) when already getting 60 IVs with the format (*B*+3, 255, *X*). Hence, we cannot use the second output to

reduce $N_{IV(1st)}$ because the number of weak IVs with the format (X, 252-B, 4+2B-2X) is too few.

Nevertheless, if IVs are generated by a little endian counter, we can use the second output to reduce $N_{IV(1st)}$. The number of collected weak IVs with the formats (X, 252-B, 4+2B-2X) and (B+3, 255, X) are same, because we can collect one (B+3, 255, X) IV and one (X, 252-B, 4+2B-2X) IV every 2^{16} IVs. So we can reduce $N_{IV(1st)}$ from 60 [1] (actual number posted on sci.crypt is 54) to 48 by doing the experiment shown in section 3 using the first and second outputs simultaneously.

We can also get the same result of reduced $N_{IV(1st)}$=48 using the following simple approach. Now, assume that we have collected n weak IVs for the first and second outputs, respectively, and use them to derive the secret key simultaneously for 50% successful probability. Since $N_{IV(2nd)}$ is four times $N_{IV(1st)}$, we can get an approximation as $n + \frac{n}{4} = 60$. The first term is due to the first output and the second term is contributed by the second output. Then, we will get the reduced $N_{IV(1st)}$, $n = 48$.

4.2 Known IV Attack Using the Combination of All Effective Outputs for WEP2 system

In section 4.1, we use the combination of the first and second outputs to reduce the required number of weak IVs in WEP system. We can also improve the original known IV attack by using the first I-1 outputs simultaneously for WEP2 system with I-byte public IV. Here we consider that the length of IV is 16 bytes in WEP2.

The attacking process is described as the following: First the attacker is in search of IVs satisfying the *1st, 2nd, ..., (I-1)-th resolved condition* after the first I steps. If S_I cannot satisfy the constraints, we discard it and find another suitable one. After step I+B+1, they will still be in the *(I-1)-th resolved condition* with high probability. Then he calculates $k[B]$ using the first, second, …, (I-1)-th outputs separately.

If IV is generated by a big endian counter, from observation of the value N_{BEC} in Table 1, we know that the weak IVs satisfying the *15-th resolved condition* will first arrive. Thus, we use other outputs to reduce $N_{IV(15th)}$. For our example, consider using the 11-th, 12-th, 13-th, and 14-th outputs to improve the attack with the 15-th output, i.e., reduce $N_{IV(15th)}$. Using the same approach in section 4.1, suppose that we have collected n weak IVs for the 15-th output, then we can get $n_{14}\left(=\left\lfloor n \times \frac{2^{16}}{2^{24}} \right\rfloor \times 2^{8}\right)$, $n_{13}\left(=\left\lfloor n \times \frac{2^{16}}{2^{32}} \right\rfloor \times 2^{16}\right)$, $n_{12}\left(=\left\lfloor n \times \frac{2^{16}}{2^{40}} \right\rfloor \times 2^{24}\right)$, and n_{11} $\left(=\left\lfloor n \times \frac{2^{16}}{2^{48}} \right\rfloor \times 2^{32}\right)$ weak IVs for the 14-th, 13-th, 12-th, and 11-th outputs, respectively (Note that we will get $2^{8(I-(m+1))}$ weak IVs every $2^{8(I-(m-1))}$ IVs for the m-th output). Then, we have

$$n_{11} \times \frac{N_{IV(15\text{-th})}}{N_{IV(11\text{-th})}} + n_{12} \times \frac{N_{IV(15\text{-th})}}{N_{IV(12\text{-th})}} + n_{13} \times \frac{N_{IV(15\text{-th})}}{N_{IV(13\text{-th})}} + n_{14} \times \frac{N_{IV(15\text{-th})}}{N_{IV(14\text{-th})}} + n = N_{IV(15\text{-th})}$$

So, we can get the reduced n.

If IV is increased in a little endian order, the first weak IV for m-th (≥3) output will happen after finishing the collection of $N_{BEC(1st)}$ IVs for the first output, so we cannot use other outputs, except the second output, to reduce $N_{IV(1st)}$ to 48

like WEP system in section 4.1.

5. Conclusion

In this paper, we define new *resolved conditions* and extend the known IV attack using other outputs in WEP-like systems. Moreover, we reduce the required number of weak IVs to derive the secret key by combining the first and second outputs in WEP and by combining the first 15 outputs in WEP2. Although the later output we use the more N_{IV} we need, our extended known IV attack still works when the first output of RC4 is discarded by the user. In fact, there are several tools available online can be used to crack WEP based on the known IV attack. For avoiding the vulnerability of the known IV attack, we can discard the first two outputs of RC4 in WEP and the first fifteen outputs of RC4 in WEP2 to enhance the WEP-like system. In addition, choosing the big endian increasing mode of IV can also enhance the security of WEP-like system.

References

[1] S. Fluhrer, I. Mantin and A. Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4", *8th Annual Workshop on Selected Areas in Cryptography*, 2001.

[2] IEEE Standard Board, "802 part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", 1999 Edition.

[3] R. L. Rivest, "The RC4 Encryption Algorithm", *RSA Data Security, Inc*. Mar. 1992.

[4] N. Borisov. I. Goldberg. D. Wanger. "Intercepting Mobile Communications: The Insecurity of 802.11", *The Seventh Annual International Conference on Mobile Computing and Networking*, July 16-21, 2001.

[5] IEEE 802.11i Draft, URL: http://www.ieee802.org/11/private/Draft_Standards /11i/802.11i-D3.0.pdf.

[6] C. Peikari and S. Fogie, Maximum Wireless Security, *Sams Publishing*, 2002.

Input
IV: a public initialization vector
k: a secret key
one packet
Output
D: transmitted data
Algorithm
Step 1: Generate the plaintext $P=M\|\mathrm{CRC}(M)$
Step 2: Generate the keystream $KS=\mathrm{RC4}(\mathrm{IV}, k)$
Step 3: Generate the ciphertext $C=P\oplus KS$
data $D=\mathrm{IV}\|C$
$\|$: concatenation, CRC: Cyclic Redundant Check, $\oplus$: exclusive-or

Figure 1. WEP Protocol

KSA(*K*)
Initialization:
For i=0…255
S[i]=i
j=0
Scrambling:
For i=0…255
j=(j+S[i]+K[i mod l]) mod 256
Swap(S[i], S[j])

PRGA(S)
Initialization:
i=0
j=0
Generation loop:
i=(i+1) mod 256
j=(j+S[i]) mod 256
Swap(S[i], S[j])
Output z=S[(S[i]+S[j]) mod 256]

Figure 2. RC4 Algorithm

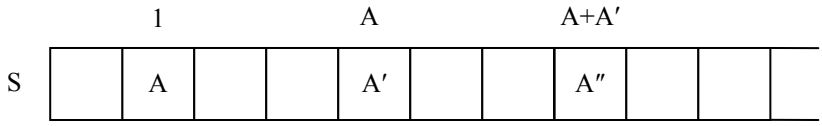

Figure 3. Specific permutation S for the known IV attack

Inputs
IV: a public I-byte initialization vector
Z_1: the first output of RC4
B: an index of secret key
Output
$k[B]$: a byte of secret key with index B
Algorithm
Step 1: Generate S_I
Step 2: If ($S_I[1]<I$ and $S_I[1]+S_I[S_I[1]]=I+B$) Go to step 3
Else Output "Discard this IV"
End
Step 3: Generate S_{I+B}
Step 4: Calculate $k[B]=S_{I+B}^{-1}[Z_1]-j_{I+B}-S_{I+B}[I+B]$

Figure 4. Algorithm of known IV attack using the first output

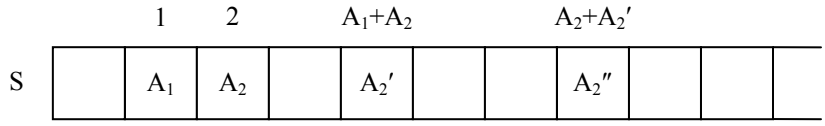

Figure 5. Specific permutation S for the known IV attack by the 2nd output

Inputs
IV: a public I-byte initialization vector
Z_2: the second output of RC4
B: an index of secret key
Output
$k[B]$: a byte of secret key with index B
Algorithm
Step 1: Generate S_I
Step 2: If (($S_I[1]+S_I[2]<I$ and $S_I[2]+S_I[S_I[1]+S_I[2]]=I+B$) and ($S_I[1]\neq 2$, $S_I[1]\neq S_I[1]+S_I[2]$))
Go to step 3
Else Output "Discard this IV"
End
Step 3: Generate S_{I+B}
Step 4: Calculate $k[B]=S_{I+B}^{-1}[Z_2]-j_{I+B}-S_{I+B}[I+B]$

Figure 6. Algorithm of known IV attack using the second output

IGA (m, I, B)

1. Initialization:
For i=0…255
  S[i]=i
j=0

2. Find S′:
For i=0…m-2
  $X_i \in \{0,\ldots,255\}$
  S′[i]=$X_i \neq \{X_0,\ldots,X_{i-1}\}$
S′[m]=I+B-S′[0]
$S'[m-1]=0-(\sum_{i=1}^{m-2} S'[i] + S'[m])$

3. Calculate IV:
For i=0…m
  IV[i]=S′[i]-j-S[i]
  j=S′[i]
For i=m-1…I-3
  $X_i \in \{0,\ldots,255\}$
  IV[i+2]= X_i
Output IV

Figure 7. IV Generation Algorithm ($m \geq 2$)

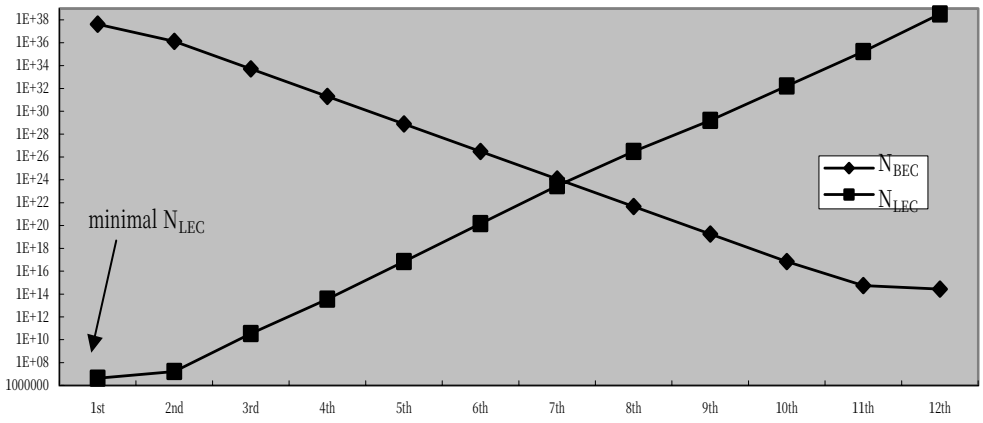

Figure 8. NBEC and NLEC of the first twelve outputs in WEP2

Table 1. The comparison for different output attack for WEP2 system (I=16, l=32)

Known output	P_{rc}	N_{IV}	N_{LEC}	N_{BEC}
1st output	e^{-3}	60	$60\times2^{8\times2}(\approx2^{22})$	$(16+16)\times2^{8\times15}(=2^{125})$
2nd output	e^{-4}	240	$240\times2^{8\times2}(\approx2^{24})$	$1\times2^{8\times15}(=2^{120})$
3rd output	e^{-5}	1200	$1200\times2^{8\times3}(\approx2^{35})$	$1\times2^{8\times14}(=2^{112})$
4th output	e^{-6}	7000	$7000\times2^{8\times4}(\approx2^{45})$	$1\times2^{8\times13}(=2^{104})$
5th output	e^{-7}	4.8×10^4	$(4.8\times10^4)\times2^{8\times5}(\approx2^{56})$	$1\times2^{8\times12}(=2^{96})$
6th output	e^{-8}	3.3×10^5	$(3.3\times10^5)\times2^{8\times6}(\approx2^{67})$	$1\times2^{8\times11}(=2^{88})$
7th output	e^{-9}	2.1×10^6	$(2.1\times10^6)\times2^{8\times7}(\approx2^{78})$	$1\times2^{8\times10}(=2^{80})$
8th output	e^{-10}	1.1×10^7	$(1.1\times10^7)\times2^{8\times8}(\approx2^{88})$	$1\times2^{8\times9}(=2^{72})$
9th output	e^{-11}	2.7×10^7	$(2.7\times10^7)\times2^{8\times9}(\approx2^{97})$	$1\times2^{8\times8}(=2^{64})$
10th output	e^{-12}	1×10^8	$(1\times10^8)\times2^{8\times10}(\approx2^{107})$	$1\times2^{8\times7}(\approx2^{56})$
11th output	e^{-13}	4.9×10^8	$(4.9\times10^8)\times2^{8\times11}(\approx2^{117})$	$1\times2^{8\times6}(\approx2^{48})$
12th output	e^{-14}	2.7×10^9	$(2.7\times10^9)\times2^{8\times12}(\approx2^{128})$	$161\times2^{8\times5}(\approx2^{47})$
m-th output	$e^{-(m+2)}$	*	**	***

*: simulation (use 1.6GHz Pentium IV processor and 256MB SDRAM; the simulated result is done up to m=12)

**: $N_{LEC} = N_{IV}\times2^{8m}$

***: $N_{BEC} = I\times2^{8(I-1)}+(l-I)\times2^{8(I-1)}$ for m=1, $N_{BEC} = \left\lceil \frac{N_{IV}}{2^{8(I-(m+1))}} \right\rceil \times2^{8(I-(m-1))}$ for m>1