

Re: doubts about rc4

Source: <http://www.derkeiler.com/Newsgroups/sci.crypt/2003-11/1854.html>

From: Gregory G Rose (ggr_at_qualcomm.com)

Date: 11/25/03

Date: 25 Nov 2003 11:59:53 -0800

In article <Pine.GSO.4.58.0311251310330.22703@turmalina.dcc.ufmg.br>, Leonardo Barbosa e Oliveira <leob@dcc.ufmg.br> wrote:
>Even though I am newbie about crypt., I think my results
>are a little bit strangers. I am using avr studio simulator
>for rc4. However, the number of cycles are varying as function
>of both message size and key size.

I don't know that this simulator is. But I do know that no-one has ever detected a cycle in RC4 if it was properly initialised.

>Concerning the first, I could not figure out what is happening, is it
>possible? cycles varying with message size?

>

>About the last, at first, I found curious the fact that increasing key
>size the number of cycles used to decrease. Afterward, however, I
>realized the array has to be filled with the key size and, in turn,
>the more is the key size, the lesser is number of instructions to fill
>the array. Is that line of reasoning right?

I don't think so. The array is first filled with sequential numbers, then a single pass over the entire array, using key bytes modulo key length, is made. Thus, keying requires 256 swap operations, no matter what the key length is.

>Any help will be very appreciated.

I suspect a bug in your implementation.

If you initialise RC4 with the 8-byte key "test key", and drop the first 256 output bytes, the next 44 bytes should be:

```
unsigned char expected[] = {  
    0xbd, 0xe9, 0x5c, 0xb5, 0x2b, 0x8d, 0xf8, 0xfb,  
    0xf2, 0xb7, 0x51, 0xf6, 0x5b, 0xe1, 0xdf, 0x3e,  
    0xd7, 0x4b, 0x45, 0x7a, 0xe9, 0x76, 0x4d, 0x26,
```

sci.crypt: Re: doubts about rc4

```
0x2f, 0x43, 0xa4, 0x70, 0x9a, 0x2a, 0xc9, 0x4e,  
0x11, 0x23, 0x89, 0x7b, 0x02, 0x2a, 0x4f, 0x07,  
0x80, 0x98, 0xa1, 0xa0,  
};
```

Greg.

```
--  
Greg Rose  
232B EC8F 44C6 C853 D68F E107 E6BF CD2F 1081 A37C  
Qualcomm Australia: http://www.qualcomm.com.au
```