

Re: Cohen's paper on byte order

Source: <http://www.derkeiler.com/Newsgroups/sci.crypt/2003-05/0558.html>

From: Mok-Kong Shen (mok-kong.shen_at_t-online.de)

Date: 05/08/03

Date: Thu, 08 May 2003 11:49:08 +0200

"Douglas A. Gwyn" wrote:

>
> *Mok-Kong Shen wrote:*
> > *"Douglas A. Gwyn" wrote:*
> >> *Mok-Kong Shen wrote:*
> >>> *Now allow me again a detailed argument. Let's*
> >>> *consider the (internal) bit sequence 01001111. At*
> >>> *first, as you also mentioned, it means just 8 bits.*
> >> *You should write it as 0,1,0,0,1,1,1,1,... and also*
> >> *recognize that there are not just 8 bit values but*
> >> *rather a large number of them, of which you have*
> >> *chosen to show just the first 8, presumably because*
> >> *we can agree that whatever I/O processing is done,*
> >> *the FIPS clearly specifies that it be done in groups*
> >> *of 8 successive input bits each.*
> >>> *Excepting its sequential ordering, the whole hasn't*
> >>> *yet a 'meaning' as a binary integer, let alone MSB*
> >>> *or LSB. (It has instead leftmost bit and rightmost*
> >>> *bit).*
> >> *No! A sequence has a "first" or an "earliest" bit,*
> >> *which is also, when indexed by the natural numbers*
> >> *(as is done by the FIPS), a "lowest numbered" bit.*
> >> *If you want to call it a "leftmost" bit you are at*
> >> *that point assuming a big-endian convention (i.e.*
> >> *associating the first bit with the most significant*
> >> *bit in a binary numeration system), which begs the*
> >> *question.*
>
> > *I don't understand you at all. You wrote 'should*
> > *write it as 0,1,0,0,1,1,1,1'. What does that*
> > *mean exactly?*
>
> *That's not what I said! The "... " is important to*
> *remind us that we are not at that point talking about*
> *exactly 8 bits, but introduce an 8-bit grouping for a*
> *specific purpose, as I explained. The reason for the*
> *commas is that that is standard mathematical usage*

- > *when indicating a sequence by exhibiting its first*
- > *few terms; as you should have understood from the*
- > *further discussion, when you elided the ,s and ,...*
- > *you changed from a bit sequence to what appears to be*
- > *a positive integer expressed in binary notation, and*
- > *by so doing you have tacitly adopted a "big endian"*
- > *convention. But that is the very step that is under*
- > *dispute, so it is not acceptable to assume that.*

But what just 'appears' to you 'need' not always be taken into consideration by you! (You certainly know the diverse optical illusions.) If you recognize that 'fundamentally' what one has to start with is 'only' a sequence of 128 bits that are 'independent' and there is 'only' a 'sequential' ordering relation among them (if the key, say, is not 'random', then there are 'correlations' among the bits, but that's not relevant here), then everything you get on interpreting the sequence with some 'additional' 'context'/viewpoint of your own choice is something 'artificial', something newly created, something extra, isn't it? When you see a long OTP, do you regard it as a huge binary integer? Certainly you could do that and consequently also perform some big-integer operations on it. But that's something 'extra' of yours, i.e. stemming from your own 'particular' 'will' and actions that are not germane ('propre', 'eigen') to what is actually being given by the user at the start. Do you agree? See also a couple of my response to Gladman and in particular my analogy of the kitchen knives there.

Look also in the following way. In my code I use an array of unsigned char to represent a bit array, since C doesn't have bit as basic data type. This is waste of storage space. If one looks into that array one would see

```
00000000 00000001 00000000 00000000 00000001 .....
```

Now do a compression on it and one obtains 01001111, which is exactly what I have in the variable bt. Do you think that one should have 11110010, namely in the reverse order instead? But why (without invoking any 'interpretation' as integer)? If reversal at all, why a reversal on 8 bit chunks and not a reversal of order of the whole bunch of 128 bits?

- >
- > > *... For convenient reference here*
- > > *again is the program (that you snipped) ...*

- >
- > *But it isn't relevant, since it has for its input a*
- > *bit array, not a multibit array. All along it has*
- > *been agreed that there is no problem when the I/O data*
- > *is externally organized as a bit sequence.*

But the user input IS at the beginning a bit array of size 128. Do you agree or not? So what's wrong with my program?

BTW, do you find any problems with the art of presentation of the AES algorithm as is done in W. Stallings' paper in Cryptologia?

M. K. Shen