

Re: Blowfish Sign Extension implementation risk

Source: <http://www.derkeiler.com/Newsgroups/sci.crypt/2004-04/2759.html>

From: William Wallace (msm30_at_yahoo.com)

Date: 04/30/04

Date: 29 Apr 2004 20:10:09 -0700

Tom St Denis <tom@seurescience.net> wrote in message
news:<Rs4kc.316708\$2oI1.177921@twister01.bloor.is.net.cable.rogers.com>...

> *Tom St Denis wrote:*

> > *I didn't say you said that. I put that in quotes to capture the essence
> > of the thread. That you think XOR is the solution to the problem at
> > hand. Then you get all defensive when I suggest that just implementing
> > the algorithm correctly in the first place is the better course of action.*

>

> *I want to add something here I think you overlooked.*

>

> *In the Blowfish paper he doesn't specify that a key must be loaded by
> shifting and OR'ing. He specifies a key is loaded in network byte order
> into 18 32-bit variables.*

>

Yes, I think you made that point twice in this thread. I haven't read his paper, but I have read two Dr. Dobbs articles, and two applied cryptography books, by Bruce. In the 2nd edition, he says:

(2) XOR P1 with the first 32 bits of the key, XOR P2 with the second 32 bits of the key, an so on for all bits of the key (up to P18). Repeatedly cycle through the key bits until the entire P-array has been XORed with the key bits.

So, you have a point. However, does your implementation follow that description? I highly doubt it, but it might. Specifically, this step says key *bits*, not key bytes. Do you allow, for example, a 121-bit key in your crypto library? Do you rotate key bits or key bytes before XORing into the P array? Or do you follow Bruce's code from the 2nd edition. Specifically:

```
for(k=0;k<4;k++){
  data = (data <<8) | k[j]; /* note, 2nd edition has sign extension
bug too */
//...
}
```

Again, let me be very clear in my question: Does your implementation follow his written description, or the source code?

> *In the relevance of this thread that's a big difference.*

Maybe, maybe not. Depends on how you answer the last question.

> *You're claiming [from what I gather] that Blowfish wasn't designed > robustly because an implementation is wrong.*

No, I claimed that the algorithm could have been designed to be more robust to implementation errors than it was, as was one of the stated goals (see Dr. Dobbs, April, 1994). You have a point that I might be including Bruce's de facto reference implementation as part of the algorithm description, since the algorithm, as it is available to the mainstream, was published with C source code (again, see Dr. Dobbs, Applied Cryptography, etc.).

> *I'm simply stating that the implementation is wrong and the design [while > terse and lacking test vectors] is correct.*

Good point. But I am interested, is your implementation correct? Do you shift in key bits? Or do you interpret (2) above to mean that the key must be mod 32 bits? Or did you look at the C source code provided by Bruce for a clue?

>
> *Tom*