

## Re: Don't use S-boxes!

**Source:** <http://www.derkeiler.com/Newsgroups/sci.crypt/2004-11/0925.html>

---

*karl\_m\_at\_acm.org*

**Date:** 11/19/04

Date: 19 Nov 2004 12:16:48 -0800

David Wagner wrote:

> > *The specific values of cycles-per-byte for the encryption function with a pre-expanded key increased from 375 to 750 cycles per byte on my reference implementations.*

>

> *That counts as extremely slow -- so slow as to render this implementation strategy pretty useless, most likely.*

They're slightly smaller if you time the function after it is loaded into the processor's cpu-cache: 226 and 628 cycles per byte.

> > *I'm unable to duplicate the published attack, and frankly don't understand how the "simple cache-timing attack on AES" keeps the S-BOX values out of the cache for reload-timings during its run-after-run collections of cycle counts. Perhaps Professor Bernstein has not implemented something correctly, or has a hardware misconfiguration where the Level 1 or Level 2 cache is turned-off. Has anyone been able to duplicate his result? I'll post sample code on my web site shortly.*

>

> *Are you running on the same architecture as him? The attack might be architecture-dependent. His paper includes some comments about the fact that Athlon caches are 2-way associative. Caches on a Pentium might well be organized differently, leading to different timing effects (I don't know).*

I've tried four different Athlon platforms and two Pentium: AMD Athlon 800MHz, AMD Athlon ? MHz, AMD Athlon XP 2000, AMD Athlon 1GHz, Pentium-S 100MHz, Pentium-M. I get nothing but random numbers for the S-Box table lookup version, with or without table lookups for the Xtime function calls.

sci.crypt: Re: Don't use S-boxes!

There seems to be some difference of opinion as to the difficulty of obtaining a constant time AES implementation. You say it's trivial, while Professor Bernstein says it's nearly impossible. I would say that expecting a single implementation strategy to be sufficient for all platforms is idealism. The only real platform where timing attacks should be a problem is the stand-alone device, like the DES PC cards used for electronic banking, that have 8-bit processors without caches. Of course today, one would expect engineers to design using embedded processors with cache. On this platform, absolute constant time would be necessary, or randomized timing.

>>*From the experience of adding timing attack resistance to my reference implementation, constant time on a pentium/athlon platform is going to be impossible. My guess is that the market for 8-bit processors isn't going away any time soon.*

karl m