

## Re: Help – I'm at wit's end...

**Source:** <http://www.derkeiler.com/Newsgroups/sci.crypt/2004-12/1952.html>

---

**From:** John Savard ([jsavard\\_at\\_excxn.aNOSPAMb.cdn.invalid](mailto:jsavard_at_excxn.aNOSPAMb.cdn.invalid))

**Date:** 12/31/04

Date: Fri, 31 Dec 2004 07:14:15 GMT

On Wed, 29 Dec 2004 21:14:31 GMT, Don Bruder <[dakidd@sonic.net](mailto:dakidd@sonic.net)> wrote, in part:

*>for any data  
>block ("message") whose expected hash contains the value 0x0D or 0x0A, I  
>get an "almost-correct" hash. "Almost-correct" in that anyplace the byte  
>0x0D appears in the "expected" hash value, I find the byte 0x0A in the  
>calculated hash, and anyplace that the byte 0x0A is expected, the  
>calculated hash has a 0x0D byte, with all other bytes of the calculated  
>hash matching the expected hash perfectly.*

Clearly, if, say, 0x0D and 0x0A were swapped in the input, you would get a hash that was completely scrambled.

So their *\*presence\** in the input is triggering their being swapped in the output.

This is not something that could be triggered by a programming bug in a SHA implementation. Test your code in a "non-real-world" situation where the inputs and outputs are supplied in hexadecimal.

I think what's happening is something to do with the input and output bit streams not being open as binary. 0x0D is binary for carriage return, 0x0A is binary for line feed.

John Savard

<http://home.ecn.ab.ca/~jsavard/index.html>