Ruby trunk - Feature #12802

Add BLAKE2 support to Digest

10/01/2016 05:49 PM - bascule (Tony Arcieri)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	

Description

BLAKE2 is a fast, modern hash function, based on improvements to the BLAKE function, which was a SHA3 finalist. BLAKE2 performs about twice as fast in software as SHA3 winner Keccak:

https://blake2.net/

BLAKE2 has received an informational RFC 7693 from the IETF:

https://tools.ietf.org/html/rfc7693

It was added to the Python standard library in Python 3.6:

https://docs.python.org/3.6/library/hashlib-blake2.html

If there's interest in supporting BLAKE2 in the Ruby standard library, I can put together a patch.

History

#1 - 10/26/2016 11:26 AM - ferdinandrosario@gmail.com (ferdinand rosario)

- Assignee set to ruby-core

#2 - 10/26/2016 11:59 AM - hsbt (Hiroshi SHIBATA)

- Assignee deleted (ruby-core)

#3 - 10/26/2016 03:52 PM - nobu (Nobuyoshi Nakada)

Is BLAKE2b unavailable if 64-bit integer is unavailable?

#4 - 10/26/2016 04:47 PM - bascule (Tony Arcieri)

BLAKE2b will work on 32-bit CPUs but is optimized for 64-bit CPUs

#5 - 10/27/2016 01:32 AM - nobu (Nobuyoshi Nakada)

I meant that BLAKE2b seems to need uint64_t.
If a compiler does not support such large integer type, isn't BLAKE2b usable on that platform?

#6 - 10/27/2016 07:10 AM - rhenium (Kazuki Yamaguchi)

Is there still a supported environment without 64-bit integer support? ext/digest/sha2/sha2.c is already using 64-bit integers. It is compiled only when OpenSSL (or CommonCrypto) is not available, but the current versions of OpenSSL also require uint64_t.

But, I'm wondering, why not add the SHA-3 winner but only BLAKE2?

#7 - 10/27/2016 07:44 AM - nobu (Nobuyoshi Nakada)

https://github.com/nobu/ruby/tree/feature/digest/sha3

#8 - 10/29/2016 06:58 PM - bascule (Tony Arcieri)

Nice @ SHA-3 branch. I think it makes sense to add both.

The reason to add BLAKE2 in addition to SHA-3 is that BLAKE2 is, in some cases, over 3X faster than SHA-3 at an equivalent security level:

https://blake2.net/sandy.png

BLAKE2b on a modern Intel CPU is even faster than MD5, but provides substantially better security.

Something of a grassroots movement is pushing for the use of BLAKE2 anywhere speed is important, such as checksumming large files.

#9 - 12/12/2016 03:43 PM - naruse (Yui NARUSE)

> Nobuyoshi Nakada wrote:
>
> > I meant that BLAKE2b seems to need uint64_t.
> > If a compiler does not support such large integer type, isn't BLAKE2b usable on that platform?
> > Kazuki Yamaguchi wrote:
> > Is there still a supported environment without 64-bit integer support? ext/digest/sha2/sha2.c is already using 64-bit integers. It is compiled only when OpenSSL (or CommonCrypto) is not available, but the current versions of OpenSSL also require uint64_t.
>
> But, I'm wondering, why not add the SHA-3 winner but only BLAKE2?

Many part of CRuby requires int64_t/uint64_t.
The point seems acceptable.

#10 - 12/21/2016 02:11 PM - shyouhei (Shyouhei Urabe)

We looked at this issue at today's developer meeting. We could not be sure if we need our own implementation of BLAKE2 (or SHA3). Maybe a matter of time? We might need to use SHA3 someday, but OpenSSL should also have one at that time (BLAKE2 is there already).

Isn't it better for us to encourage people switching from Digest to OpenSSL?