Ruby - Bug #4418

OpenSSL::PKey::DH#public_key

02/21/2011 11:45 PM - ohai (Ippei Obayashi)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	ruby 1.9.2p180 (2011-02-18 revision 30909) [x86_64-linux]	**Backport:**

Description

=begin
OpenSSL::PKey::DH#public_key がその名前に反して DH パラメータのみを複製して返しています。
require 'openssl'
dh = OpenSSL::PKey::DH.generate(1024) # => パラメータと鍵の生成、時間がかかる
p dh.pub_key # => 公開鍵の整数を表示
p dh.public_key.pub_key # => nil

DH の場合、RSAなどのように「公開鍵とパラメータ」を取り出す需要はあまりない
ように思われるので、名前を変える、もしくはメソッドを廃止するのが良いのではと思われます。
=end

History

#1 - 02/22/2011 08:46 AM - MartinBosslet (Martin Bosslet)

- File fix_dh_dup.tar.gz added

=begin
Hi,

I had been doing some work in this area, so I looked into this. The problem
is that DH parameters are duplicated, but this only duplicates the generator
g and the prime p, but not the public key, which can be derived from g and p.

The easiest way to fix this is to simply dup the existing value and assign it
to the dup'ed DH instance.

Patch and test are attached.

Regards,
Martin
=end

#2 - 02/22/2011 09:45 AM - naruse (Yui NARUSE)

- Status changed from Open to Assigned

- Assignee set to nahi (Hiroshi Nakamura)

=begin

=end

#3 - 02/23/2011 01:12 AM - ohai (Ippei Obayashi)

=begin
Hi, Martin

Your fix is appropriate.
However, I feel no one needs this (copying only
parameters and a public key) method, so
it is also a reasonable idea that we simply remove
or rename the method.

=end

#4 - 03/08/2011 11:23 AM - MartinBosslet (Martin Bosslet)

=begin
Hi Ippei,

I see your point now. There is some confusion if we look
at the EC key agreement interface. There, EC#dh_compute_key
takes what is returned by EC#public_key as a parameter,
which could be assumed the natural equivalent to what
DH#public_key returns.
Maybe DH#compute_key could additionally support a
version where it takes the "public_key" instead of the
"pub_key". This way we would achieve consistency among
DH and ECDH interfaces and DH#public_key wouldn't be
as useless anymore :)

What do you think?

Regards,
Martin
=end

#5 - 03/13/2011 03:43 PM - ohai (Ippei Obayashi)

=begin
Hi, Martin

RSA#public_key returns a RSA object, DSA#public_key returns a
DSA object, but EC#public_key does not return a EC object (it returns a EC::Point object).
I feel this fact is also confusing.
=end

#6 - 03/18/2011 11:42 AM - MartinBosslet (Martin Bosslet)

=begin
Hello Ippei,

I thought this to be confusing, too - that EC#public_key is an EC::Point instead of an instance of EC itself. But when I had a closer look again, I noticed that EC::Point is in fact a subclass of EC, so the analogy to RSA and DSA is kept. So we could still have the version where DH#compute_key and EC#dh_compute_key take the return value of the corresponding #public_key methods, relying on API common to EC and DH.

Best regards,
Martin
=end

#7 - 06/20/2011 06:23 PM - nahi (Hiroshi Nakamura)

- Priority changed from Normal to 3

I agree with DH interface is confusing.

- DH#p ... DH parameter p
- DH#g ... DH parameter g
- DH#public_key ... DH parameter (DHParameterSpec in Java)
- DH#priv_key ... private value: S
- DH#pub_key ... exchange value: g^S mod p

We should have PKey::DH::Params class as same as PKey::EC::Point in the future though I don't know it's good to define it as a subclass of DH.

Back to the topic, DH#public_key is needed for exchanging DH parameters (p and g) so we cannot drop it. And we would need new method DH#params as a copy of DH#public_key when we implement DH::Params class.

#8 - 06/23/2011 08:36 PM - MartinBosslet (Martin Bosslet)

Hiroshi NAKAMURA wrote:

> I agree with DH interface is confusing.

Adding to the confusion is that DH implements the PKey interface in OpenSSL (OpenSSL itself, not Ruby's ext/openssl), but it conceptually is not really like the other PKey implementations.

1. PKey offers PKey#sign and PKey#verify as a common characteristic. DH responds to both in OpenSSL, but they ultimately lead to an error saying that signature/verification is not supported.
2. PKeys offer a public and a private "key", which at first glance is conceptually fine for DH, as there is also a public and a private part. But the

> analogy ends when it comes to en-/decoding their PEM/DER representation. The rest allows a "private" encoding as well as a X.509 "PUB_KEY" encoding, both of which DH does not support. As a consequence it also does not work with the new OpenSSL::PKey.read functionality.

This and the matters already discussed lead me to the conclusion that it might be a good idea to separate DH from the PKey implementations in ext/openssl and set up a separate KeyExchange module featuring two implementations, DH and ECDH (and possibly more in the future). By this separation, we could also clean up the confusion with PKey::EC, as in its current form it's some sort of hybrid, featuring both PKey and DH functionality.

The separation could also concentrate on Key Exchange/Agreement features better: We could add support for Key Derivation algorithms to simplify arbitrary-length symmetric key generation for Ciphers (a non-trivial task that needs to be taken care of manually right now), and it would be easier to design a nice API for supporting key agreement using static and ephemeral keys as outlined in NIST SP 800-56A.

What do you think about this (post 1.9.3, of course :)?

> We should have PKey::DH::Params class as same as PKey::EC::Point in the future though I don't know it's good to define it as a subclass of DH.

Great idea, and we could even call it params instead of public_key if we went the "separate module approach", making it possible to rename priv_key and pub_key to private_key and public_key!

Regards,
Martin

#9 - 06/26/2011 06:41 PM - nahi (Hiroshi Nakamura)

- Target version set to 2.0.0

#10 - 11/29/2012 10:05 PM - nahi (Hiroshi Nakamura)

- Assignee changed from nahi (Hiroshi Nakamura) to MartinBosslet (Martin Bosslet)

- Target version changed from 2.0.0 to 2.6

I like to keep ext/openssl just reflects OpenSSL API but we already have some exceptions in API for ease of use.

I postponed this to "next minor" but as we talked at RubyConf, we can try it at openssl gem (vaporgem ATM.)

#11 - 09/13/2015 03:20 AM - zzak (zzak _)

- Assignee changed from MartinBosslet (Martin Bosslet) to 7150

#12 - 11/10/2017 04:05 AM - rhenium (Kazuki Yamaguchi)

- Status changed from Assigned to Rejected

I agree the name 'public_key' was not a good choice, but at the same time I don't think the name being confusing is not strong enough justification to remove or rename now. I'll leave it as is.

Files

fix_dh_dup.tar.gz	633 Bytes	02/22/2011	MartinBosslet (Martin Bosslet)