

Re: Using Crypt::DSA

Source: <http://coding.derkeiler.com/Archive/Perl/comp.lang.perl.misc/2005-08/msg01720.html>

- *From:* Mike Friedman <mikef@xx>
 - *Date:* Sat, 27 Aug 2005 05:18:41 +0000 (UTC)
-

Sisyphus <sisyphus1@xxxxxxxxxxxxxxxxxxxx> wrote:

```
>
> "Mike Friedman" <mikef@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message
>
>> I want to be able to create a public key file and a signature in
>> formats that can be distributed to a user community whose
>> applications may be using other DSA implementations (e.g., Java
>> crypto lib, or other scripting languages besides perl). So, it's
>> important that the public key file be in a reasonably standard format.
>
> Which, afaik, would be a pem file. Crypt::DSA can parse them (as
> long as you have Convert::PEM) and so, presumably, can the other
> DSA implementations. To write a *public* key pem file with
> Crypt::DSA you just do (as in 04-pem.t):
>
> $key->priv_key(undef);
> $key->write( Type => 'PEM', Filename => $keyfile);
```

Rob,

I do have Convert::PEM.

I tried your suggestion above, but I get only a private key. In fact, my script first writes the key object (using \$key->write) returned from sign(), as-is, to one file, then does the above in attempt to write the *public* key to another file. But both files are identical!

Here's the entirety of my little test script:

```
-----
#!/usr/local/bin/perl

use Crypt::DSA;
use strict;

my $dsa = new Crypt::DSA;
my $key = Crypt::DSA::Key->new;
my $filename;
```

Re: Using Crypt::DSA

```
$filename = "./dsakey";

$key = $dsa->keygen (
    Size => 1024,
    Verbosity => 1,
) or die $dsa->errstr();

$key->write(
    Type => 'PEM',
    Filename => "$filename.priv",
);

$key->priv_key(undef);

$key->write(
    Type => 'PEM',
    Filename => "$filename.pub",
);

exit;
```

The result is that 'dsakey.priv' and 'dsakey.pub' have exactly the same contents, including the 'BEGIN DSA PRIVATE KEY', 'END DSA PRIVATE KEY' delimiters, even though I've undefined priv_key. What am I doing wrong?

```
>> Also, I'd be passing the signature as a base64-encoded string
>> via a web form field. Once the application base64-decodes it,
>> the signature should be in a format easily fed to its DSA
>> verify routine.
>
> Are you saying the signature part is not a problem ? I find some
> ambiguity with "the signature should be in a format easily fed to
> its DSA verify routine" – not sure whether that means the signature
> is already in a suitable format, or whether it means that it needs
> to be in a suitable format (but isn't).
```

I was just expressing my concern that the signature as produced in your earlier example (by writing out the separate components of the signature object) wouldn't be 'standard' in some sense. But I haven't yet gotten far enough to try all that out.

Right now, I'm interested in your idea of how to write out a public key file in PEM format; it seems it should work, but I'm not getting the right results. So, probably I'm overlooking something obvious.

Thanks for your help so far.

Mike

- **Follow-Ups:**
 - ◆ **Re: Using Crypt::DSA**
 - ◇ From: Sisyphus

- **References:**
 - ◆ **Using Crypt::DSA**
 - ◇ From: Mike Friedman
 - ◆ **Re: Using Crypt::DSA**
 - ◇ From: A. Sinan Unur
 - ◆ **Re: Using Crypt::DSA**
 - ◇ From: Mike Friedman
 - ◆ **Re: Using Crypt::DSA**
 - ◇ From: Sisyphus
 - ◆ **Re: Using Crypt::DSA**
 - ◇ From: Mike Friedman
 - ◆ **Re: Using Crypt::DSA**
 - ◇ From: Sisyphus
 - ◆ **Re: Using Crypt::DSA**
 - ◇ From: Mike Friedman
 - ◆ **Re: Using Crypt::DSA**
 - ◇ From: Sisyphus

- Prev by Date: **Re: Is there any performance benefit to...**
- Next by Date: **Template-Toolkit-2.14 fails make on FC3**
- Previous by thread: **Re: Using Crypt::DSA**
- Next by thread: **Re: Using Crypt::DSA**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**