

**NAME**

Net::DNS::RR::SIG – DNS SIG resource record

**SYNOPSIS**

```
use Net::DNS;
$r = new Net::DNS::RR('name SIG typecovered algorithm labels
                        orgttl sigexpiration siginception
                        keytag signature');

use Net::DNS::SEC;
$sigrr = create Net::DNS::RR::SIG( $string, $keypath,
                                   sigval => 10    # minutes
                                   );

$sigrr->verify( $string, $keyrr ) || die $sigrr->vrfyerrstr;
$sigrr->verify( $packet, $keyrr ) || die $sigrr->vrfyerrstr;
```

**DESCRIPTION**

Class for DNS digital signature (SIG) resource records.

In addition to the regular methods inherited from Net::DNS::RR the class contains a method to sign packets and scalar data strings using private keys (create) and a method for verifying signatures.

The SIG RR is an implementation of RFC2931. See Net::DNS::RR::RRSIG for an implementation of RFC4034.

**METHODS**

The available methods are those inherited from the base class augmented by the type-specific methods defined in this package.

Use of undocumented package features or direct access to internal data structures is discouraged and could result in program termination or other unpredictable behaviour.

**algorithm**

```
$algorithm = $rr->algorithm;
```

The algorithm number field identifies the cryptographic algorithm used to create the signature.

**algorithm()** may also be invoked as a class method or simple function to perform mnemonic and numeric code translation.

**sigexpiration and siginception times****sigex sign signal**

```
$expiration = $rr->sigexpiration;
$expiration = $rr->sigexpiration( $value );
```

```
$inception = $rr->siginception;
$inception = $rr->siginception( $value );
```

The signature expiration and inception fields specify a validity time interval for the signature.

The value may be specified by a string with format 'yyyymmddhhmmss' or a Perl **time()** value.

Return values are dual-valued, providing either a string value or numerical Perl **time()** value.

**keytag**

```
$keytag = $rr->keytag;
$rr->keytag( $keytag );
```

The keytag field contains the key tag value of the KEY RR that validates this signature.

**signature**

```
$signature = $rr->signature;
$rr->signature( $signature );
```

The signer name field value identifies the owner name of the KEY RR that a validator is supposed to use to validate this signature.

### signature

#### sig

```
$sig = $rr->sig;
$rr->sig( $sig );
```

The Signature field contains the cryptographic signature that covers the SIG RDATA (excluding the Signature field) and the subject data.

#### sigbin

```
$sigbin = $rr->sigbin;
$rr->sigbin( $sigbin );
```

Binary representation of the cryptographic signature.

### create

Create a signature over scalar data.

```
use Net::DNS::SEC;

$keypath = '/home/olaf/keys/Kbla.foo.+001+60114.private';

$sigrr = create Net::DNS::RR::SIG( $data, $keypath );

$sigrr = create Net::DNS::RR::SIG( $data, $keypath,
                                   signal => 10
                                   );

$sigrr->print;

# Alternatively use Net::DNS::SEC::Private

$private = Net::DNS::SEC::Private->new($keypath);

$sigrr= create Net::DNS::RR::SIG( $data, $private );
```

**create()** is an alternative constructor for a SIG RR object.

This method returns a SIG with the signature over the data made with the private key stored in the key file.

The first argument is a scalar that contains the data to be signed.

The second argument is a string which specifies the path to a file containing the private key as generated using `dnssec-keygen`, a program that comes with the ISC BIND distribution.

The optional remaining arguments consist of ( name => value ) pairs as follows:

```
signin  => 20191201010101,      # signature inception
sigex   => 20191201011101,      # signature expiration
sigval  => 10,                  # validity window (minutes)
```

The `signin` and `sigex` values may be specified as Perl time values or as a string with the format 'yyyymmddhhmmss'. The default for `signin` is the time of signing.

The `sigval` argument specifies the signature validity window in minutes ( `sigex` = `signin` + `sigval` ).

By default the signature is valid for 10 minutes.

- Do not change the name of the private key file. The `create` method uses the filename as generated by `dnssec-keygen` to determine the keyowner, algorithm, and the keyid (keytag).

**verify**

```
$verify = $sigrr->verify( $data, $keyrr );
$verify = $sigrr->verify( $data, [$keyrr, $keyrr2, $keyrr3] );
```

The **verify()** method performs SIG0 verification of the specified data against the signature contained in the `$sigrr` object itself using the public key in `$keyrr`.

If a reference to a `Net::DNS::Packet` is supplied, the method performs a SIG0 verification on the packet data.

The second argument can either be a `Net::DNS::RR::KEYRR` object or a reference to an array of such objects. Verification will return successful as soon as one of the keys in the array leads to positive validation.

Returns false on error and sets `$sig->vrfyerrstr`

**vrfyerrstr**

```
$sig0 = $packet->sigrr || die 'not signed';
print $sig0->vrfyerrstr unless $sig0->verify( $packet, $keyrr );

$sigrr->verify( $packet, $keyrr ) || die $sigrr->vrfyerrstr;
```

**REMARKS**

The code is not optimised for speed.

If this code is still around in 2100 (not a leap year) you will need to check for proper handling of times after 28th February.

**ACKNOWLEDGMENTS**

Although their original code may have disappeared following redesign of `Net::DNS`, `Net::DNS::SEC` and the OpenSSL API, the following individual contributors deserve to be recognised for their significant influence on the development of the SIG package.

Andy Vaskys (Network Associates Laboratories) supplied code for RSA.

T.J. Mather provided support for the DSA algorithm.

**COPYRIGHT**

Copyright (c)2001–2005 RIPE NCC, Olaf M. Kolkman

Copyright (c)2007–2008 NLnet Labs, Olaf M. Kolkman

Portions Copyright (c)2014 Dick Franks

All rights reserved.

Package template (c)2009,2012 O.M.Kolkman and R.W.Franks.

**LICENSE**

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of the author not be used in advertising or publicity pertaining to distribution of the software without specific prior written permission.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**SEE ALSO**

`perl`, `Net::DNS`, `Net::DNS::RR`, `Net::DNS::SEC`, `RFC4034`, `RFC3755`, `RFC2535`, `RFC2931`, `RFC3110`, `RFC3008`, `Net::DNS::SEC::DSA`, `Net::DNS::SEC::RSA`

Algorithm Numbers <<http://www.iana.org/assignments/dns-sec-alg-numbers>>

BIND 9 Administrator Reference Manual <<http://www.bind9.net/manuals>>