

**NAME**

Net::DNS::RR::DS – DNS DS resource record

**SYNOPSIS**

```
use Net::DNS;
$rr = new Net::DNS::RR('name DS keytag algorithm digtype digest');

use Net::DNS::SEC;
$ds = create Net::DNS::RR::DS(
    $dnskeyrr,
    digtype => 'SHA256',
    ttl      => 3600
);
```

**DESCRIPTION**

Class for DNS Delegation Signer (DS) resource record.

**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.

**keytag**

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

The 16-bit numerical key tag of the key. (RFC2535 4.1.6)

**algorithm**

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

Decimal representation of the 8-bit algorithm field.

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

**digtype**

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

Decimal representation of the 8-bit digest type field.

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

**digest**

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

Hexadecimal representation of the digest over the label and key.

**digestbin**

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

Binary representation of the digest over the label and key.

**babble**

```
print $rr->babble;
```

The **babble()** method returns the 'BubbleBabble' representation of the digest if the `Digest::BubbleBabble` package is available, otherwise an empty string is returned.

BubbleBabble represents a message digest as a string of plausible words, to make the digest easier to verify. The “words” are not necessarily real words, but they look more like words than a string of hex characters.

The 'BubbleBabble' string is appended as a comment when the string method is called.

#### **create**

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

$dsrr = create Net::DNS::RR::DS($keyrr, digtype => 'SHA-256' );
$keyrr->print;
$dsrr->print;
```

This constructor takes a key object as argument and will return the corresponding DS RR object.

The digest type defaults to SHA-1.

#### **verify**

```
$verify = $dsrr->verify($keyrr);
```

The boolean verify method will return true if the hash over the key RR provided as the argument conforms to the data in the DS itself i.e. the DS points to the DNSKEY from the argument.

### **COPYRIGHT**

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

Portions Copyright (c)2013 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, RFC4034, RFC3658

Algorithm    Numbers    <<http://www.iana.org/assignments/dns-sec-alg-numbers>>,    Digest    Types  
<<http://www.iana.org/assignments/ds-rr-types>>