NAME

      Net::DNS::RR::NSEC3 − DNS NSEC3 resource record

SYNOPSIS

```
use Net::DNS;
$rr = new Net::DNS::RR('name NSEC3 algorithm flags iterations salt hnxtname');
```

DESCRIPTION

      Class for DNSSEC NSEC3 resource records.

      The NSEC3 Resource Record (RR) provides authenticated denial of existence for DNS Resource Record Sets.

      The NSEC3 RR lists RR types present at the original owner name of the NSEC3 RR. It includes the next hashed owner name in the hash order of the zone. The complete set of NSEC3 RRs in a zone indicates which RRSets exist for the original owner name of the RR and form a chain of hashed owner names in the zone.

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

      The Hash Algorithm field is represented as an unsigned decimal integer. The value has a maximum of 255.

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

   flags

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

      The Flags field is an unsigned decimal integer interpreted as eight concatenated Boolean values.

      optout

```
$rr->optout(1);

if ( $rr->optout ) {
        ...
}
```

      Boolean Opt Out flag.

   iterations

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

      The Iterations field is represented as an unsigned decimal integer. The value is between 0 and 65535, inclusive.

   salt

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

      The Salt field is represented as a contiguous sequence of hexadecimal digits. A "−" (unquoted) is used in string format to indicate that the salt field is absent.

saltbin

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

The Salt field as a sequence of octets.

hnxtname

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

The Next Hashed Owner Name field points to the next node that has authoritative data or contains a delegation point NS RRset.

typelist

```
@typelist = $rr->typelist;
$typelist = $rr->typelist;
$rr->typelist( @typelist );
```

typelist() identifies the RRset types that exist at the domain name matched by the NSEC3 RR. When called in scalar context, the list is interpolated into a string.

typemap

```
$exists = $rr->typemap($rrtype);
```

typemap() returns a Boolean true value if the specified RRtype occurs in the type bitmap of the NSEC3 record.

match

```
$matched = $rr->match( 'example.foo' );
```

match() returns a Boolean true value if the hash of the domain name argument matches the hashed owner name of the NSEC3 RR.

covers

```
$covered = $rr->covers( 'example.foo' );
```

covers() returns a Boolean true value if the hash of the domain name argument, or ancestor of that name, falls between the owner name and the next hashed owner name of the NSEC3 RR.

encloser, nextcloser, wildcard

```
$encloser = $rr->encloser( 'example.foo' );
print "encloser: $encloser\n" if $encloser;
```

encloser() returns the name of a provable encloser of the query name argument obtained from the NSEC3 RR.

nextcloser() returns the next closer name, which is one label longer than the closest encloser. This is only valid after **encloser()** has returned a valid domain name.

wildcard() returns the unexpanded wildcard name from which the next closer name was possibly synthesised. This is only valid after **encloser()** has returned a valid domain name.

COPYRIGHT

LICENSE

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, RFC5155, RFC4648

Hash Algorithms <http://www.iana.org/assignments/dnssec-nsec3-parameters>