NAME

Net::DNS::SEC::Keyset − DNSSEC Keyset object class

SYNOPSIS

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

DESCRIPTION

A keyset is an ‘‘administrative’’ unit used for DNSSEC maintenance.

This class provides interfaces for creating, reading and writing keysets.

Object methods are provided to extract DNSKEY, RRSIG and DS records.

Note that this class is still being developed. Attributes and methods are subject to change.

new (from file)

```
$keyset = Net::DNS::SEC::Keyset->new( $filename );
$keyset = Net::DNS::SEC::Keyset->new( $filename, $directory );
die Net::DNS::SEC::Keyset->keyset_err unless $keyset;
```

Constructor method which reads the specified keyset file and returns a keyset object.

The optional second argument specifies the filename base directory.

Sets keyset_err and returns undef on failure.

new (by signing keys)

```
$keyset = Net::DNS::SEC::Keyset->new( [@keyrr], $privatekeypath );
```

Creates a keyset object from the keys provided through the reference to an array of Net::DNS::RR::DNSKEY objects.

The method will create and self-sign the whole keyset. The private keys as generated by the BIND dnssec-keygen tool are assumed to be in the current directory or, if specified, the directory indicated by `$privatekeypath`.

Sets keyset_err and returns undef on failure.

new (from key and sig RRsets)

```
$keyset = Net::DNS::Keyset->new( [@keyrr], [@sigrr] );
```

Creates a keyset object from the keys provided through the references to arrays of Net::DNS::RR::DNSKEY and Net::DNS::RR::RRSIG objects.

Sets keyset_err and returns undef on failure.

new (from Packet)

```
$res = Net::DNS::Resolver->new;
$res->dnssec(1);

$packet = $res->query ( "example.com", "DNSKEY", "IN" );

$keyset = Net::DNS::SEC::Keyset->new( $packet )
```

Creates a keyset object from a Net::DNS::Packet that contains the answer to a query for the apex key records.

This is the method you should use for automatically fetching keys.

Sets keyset_err and returns undef on failure.

keys

```
@keyrr = $keyset->keys;
```

Returns an array of Net::DNS::RR::DNSKEY objects.

sigs

```
@sigrr = $keyset->sigs;
```

Returns an array of Net::DNS::RR::RRSIG objects.

extract_ds

```
@ds = $keyset->extract_ds;
die $keyset->keyset_err unless @ds;
```

Extracts DS records from the keyset. Note that the keyset will be verified during extraction. All keys will need to have a valid self-signature.

The method sets keyset_err if verification fails.

verify

```
@keytags = $keyset->verify();
die $keyset->keyset_err unless @keytags;

$keyset->verify( $keytag ) || die $keyset->keyset_err;
```

If no arguments are given:

- Verifies if all signatures present verify the keyset.
- Verifies if there are DNSKEYs with the SEP flag set, there is at least one RRSIG made using that key.
- Verifies that if there are no DNSKEYs with the SEP flag set there is at least one RRSIG made with one of the keys from the keyset.

If an argument is given, it is should be the numeric keytag of the key in the keyset which will be verified using the corresponding RRSIG.

The method returns a list of keytags of verified keys in the keyset.

The method sets keyset_err and returns empty list if verification fails.

keyset_err

```
$keyset_err = Net::DNS::SEC::Keyset->keyset_err;

$keyset_err = $keyset->keyset_err;
```

Returns the keyset error string.

string

```
$string = $keyset->string;
```

Returns a string representation of the keyset.

print

```
$keyset->print;             # similar to print( $keyset->string )
```

Prints the keyset.

writekeyset

```
$keyset->writekeyset;
$keyset->writekeyset( $path );
$keyset->writekeyset( $prefix );
$keyset->writekeyset( $prefix, $path );
```

Writes the keyset to a file named "keyset–<domain>." in the current working directory or directory defined by the optional `$path` argument.

The optional `$prefix` argument specifies the prefix that will be prepended to the domain name to form the keyset filename.

COPYRIGHT

LICENSE