

**NAME**

ldns\_dnssec\_zone\_sign, ldns\_dnssec\_zone\_sign\_nsec3, ldns\_dnssec\_zone\_mark\_glue,  
 ldns\_dnssec\_name\_node\_next\_nonglue, ldns\_dnssec\_zone\_create\_nsecs,  
 ldns\_dnssec\_remove\_signatures, ldns\_dnssec\_zone\_create\_rrsigs - sign ldns\_dnssec\_zone

**SYNOPSIS**

```
#include <stdint.h>
```

```
#include <stdbool.h>
```

```
#include <ldns/ldns.h>
```

```
ldns_status ldns_dnssec_zone_sign(ldns_dnssec_zone *zone, ldns_rr_list *new_rrs, ldns_key_list
*key_list, int (*func)(ldns_rr *, void *), void *arg);
```

```
ldns_status ldns_dnssec_zone_sign_nsec3(ldns_dnssec_zone *zone, ldns_rr_list *new_rrs,
ldns_key_list *key_list, int (*func)(ldns_rr *, void *), void *arg, uint8_t algorithm, uint8_t flags,
uint16_t iterations, uint8_t salt_length, uint8_t *salt);
```

```
ldns_dnssec_zone_mark_glue();
```

```
ldns_rbnode_t* ldns_dnssec_name_node_next_nonglue(ldns_rbnode_t *node);
```

```
ldns_status ldns_dnssec_zone_create_nsecs(ldns_dnssec_zone *zone, ldns_rr_list *new_rrs);
```

```
ldns_dnssec_rrs* ldns_dnssec_remove_signatures(ldns_dnssec_rrs *signatures, ldns_key_list
*key_list, int (*func)(ldns_rr *, void *), void *arg);
```

```
ldns_status ldns_dnssec_zone_create_rrsigs(ldns_dnssec_zone *zone, ldns_rr_list *new_rrs,
ldns_key_list *key_list, int (*func)(ldns_rr *, void*), void *arg);
```

**DESCRIPTION**

*ldns\_dnssec\_zone\_sign()* signs the given zone with the given keys

**zone:** the zone to sign

**key\_list:** the list of keys to sign the zone with

**new\_rrs:** newly created resource records are added to this list, to free them later

**func:** callback function that decides what to do with old signatures This function takes an *ldns\_rr\** and an optional *void \*arg* argument, and returns one of four values:

LDNS\_SIGNATURE\_LEAVE\_ADD\_NEW: leave the signature and add a new one for the

corresponding key LDNS\_SIGNATURE\_REMOVE\_ADD\_NEW: remove the signature and replace is with a new one from the same key LDNS\_SIGNATURE\_LEAVE\_NO\_ADD: leave the signature and do not add a new one with the corresponding key

LDNS\_SIGNATURE\_REMOVE\_NO\_ADD: remove the signature and do not replace

**arg:** optional argument for the callback function

Returns LDNS\_STATUS\_OK on success, an error code otherwise

*ldns\_dnssec\_zone\_sign\_nsec3()* signs the given zone with the given new zone, with NSEC3

**zone:** the zone to sign

**key\_list:** the list of keys to sign the zone with

**new\_rrs:** newly created resource records are added to this list, to free them later

**func:** callback function that decides what to do with old signatures

**arg:** optional argument for the callback function

**algorithm:** the NSEC3 hashing algorithm to use

**flags:** NSEC3 flags

**iterations:** the number of NSEC3 hash iterations to use

**salt\_length:** the length (in octets) of the NSEC3 salt

**salt:** the NSEC3 salt data

Returns LDNS\_STATUS\_OK on success, an error code otherwise

*ldns\_dnssec\_zone\_mark\_glue()*

*ldns\_dnssec\_name\_node\_next\_nonglue()* Finds the first dnssec\_name node in the rbtree that is not occluded. It \*does\* return names that are partially occluded.

**node:** the first node to check

Returns the first node that has not been marked as glue, or NULL if not found (TODO: make that LDNS\_RBTREE\_NULL?)

*ldns\_dnssec\_zone\_create\_nsecs()* Adds NSEC records to the given dnssec\_zone

**zone:** the zone to add the records to

**new\_rrs:** ldns\_rr's created by this function are added to this rr list, so the caller can free them later

Returns LDNS\_STATUS\_OK on success, an error code otherwise

*ldns\_dnssec\_remove\_signatures()* remove signatures if callback function tells to

**signatures:** list of signatures to check, and possibly remove, depending on the value of the callback

**key\_list**: these are marked to be used or not, on the return value of the callback  
**func**: this function is called to specify what to do with each signature (and corresponding key)  
**arg**: Optional argument for the callback function  
Returns s pointer to the new signatures rrs (the original passed to this function may have been removed)

*ldns\_dnssec\_zone\_create\_rrsigs()* Adds signatures to the zone

**zone**: the zone to add RRSIG Resource Records to  
**new\_rrs**: the RRSIG RRs that are created are also added to this list, so the caller can free them later  
**key\_list**: list of keys to sign with.  
**func**: Callback function to decide what keys to use and what to do with old signatures  
**arg**: Optional argument for the callback function  
Returns LDNS\_STATUS\_OK on success, error otherwise

## AUTHOR

The ldns team at NLnet Labs.

## REPORTING BUGS

Please report bugs to [dns-team@nlnetlabs.nl](mailto:dns-team@nlnetlabs.nl) or on GitHub at <https://github.com/NLnetLabs/ldns/issues>

## COPYRIGHT

Copyright (c) 2004 - 2006 NLnet Labs.

Licensed under the BSD License. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

## SEE ALSO

*ldns\_dnssec\_zone*. And **perldoc Net::DNS**, **RFC1034**, **RFC1035**, **RFC4033**, **RFC4034** and **RFC4035**.

## REMARKS

This manpage was automatically generated from the ldns source code.