FreeBSD kernel IPsec code Reference Manual

Generated by Doxygen 1.4.7

Sat Feb 24 19:57:33 2007

Contents

Chapter 1

FreeBSD kernel IPsec code Main Page

IMPORTANT: This API documentation may contain both functions which are public and functions that are for internal use only. Since we have not reviewed every part of the documentation yet, *some internal functions are not marked as such*. Until we finish reviewing the API documentation and add appropriate comments to functions which are only for internal use, you should take this into account. In case you want to use a function of this kernel subsystem in another kernel subsystem you should search for precedence of use outside this subsystem. If the function is not used outside this subsystem you should ask on the mailinglists about it, else you risk breaking something.

Chapter 2

FreeBSD kernel IPsec code Directory Hierarchy

2.1 FreeBSD kernel IPsec code Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

Chapter 3

FreeBSD kernel IPsec code Data Structure Index

3.1 FreeBSD kernel IPsec code Data Structures

Here are the data structures with brief descriptions:

Chapter 4

FreeBSD kernel IPsec code File Index

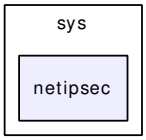
4.1 FreeBSD kernel IPsec code File List

Here is a list of all files with brief descriptions:

Chapter 5

FreeBSD kernel IPsec code Directory Documentation

5.1 /usr/src/sys/netipsec/ Directory Reference

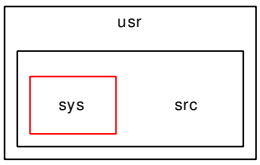


Files

- file ah.h
- file ah_var.h
- file esp.h
- file esp_var.h
- file ipcomp.h
- file ipcomp_var.h
- file ipip_var.h
- file ipsec.c
- file ipsec.h
- file ipsec6.h
- file ipsec_input.c
- file ipsec_mbuf.c
- file ipsec_osdep.h
- file ipsec_output.c
- file key.c
- file key.h
- file key_debug.c
- file key_debug.h
- file key_var.h
- file keydb.h

- file keysock.c
- file keysock.h
- file xform.h
- file xform_ah.c
- file xform_esp.c
- file xform_ipcomp.c
- file xform_ipip.c
- file xform_tcp.c

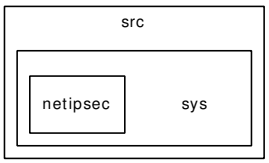
5.2 /usr/src/ Directory Reference



Directories

- directory sys

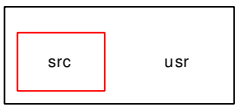
5.3 /usr/src/sys/ Directory Reference



Directories

- directory netipsec

5.4 /usr/ Directory Reference



Directories

- directory src

Chapter 6

FreeBSD kernel IPsec code Data Structure Documentation

6.1 _keystat Struct Reference

Data Fields

- u_long getspi_count

6.1.1 Detailed Description

Definition at line 362 of file key.c.

6.1.2 Field Documentation

6.1.2.1 u_long _keystat::getspi_count

Definition at line 363 of file key.c.

Referenced by key_do_getnewspi(), and key_init().

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/key.c

6.2 ah Struct Reference

`#include <ah.h>`

Data Fields

- u_int8_t ah_nxt
- u_int8_t ah_len
- u_int16_t ah_reserve
- u_int32_t ah_spi

6.2.1 Detailed Description

Definition at line 40 of file ah.h.

6.2.2 Field Documentation

6.2.2.1 u_int8_t ah::ah_len

Definition at line 42 of file ah.h.

6.2.2.2 u_int8_t ah::ah_nxt

Definition at line 41 of file ah.h.

6.2.2.3 u_int16_t ah::ah_reserve

Definition at line 43 of file ah.h.

6.2.2.4 u_int32_t ah::ah_spi

Definition at line 44 of file ah.h.

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/ah.h

6.3 ahstat Struct Reference

`#include <ah_var.h>`

Data Fields

- u_int32_t ahs_hdrops
- u_int32_t ahs_nopf
- u_int32_t ahs_notdb
- u_int32_t ahs_badkcr
- u_int32_t ahs_badauth
- u_int32_t ahs_noxform
- u_int32_t ahs_qfull
- u_int32_t ahs_wrap
- u_int32_t ahs_replay
- u_int32_t ahs_badauthl
- u_int32_t ahs_input
- u_int32_t ahs_output
- u_int32_t ahs_invalid
- u_int64_t ahs_ibytes
- u_int64_t ahs_obytes
- u_int32_t ahs_toobig
- u_int32_t ahs_pdrops
- u_int32_t ahs_crypto
- u_int32_t ahs_tunnel
- u_int32_t ahs_hist [AH_ALG_MAX]

6.3.1 Detailed Description

Definition at line 50 of file ah_var.h.

6.3.2 Field Documentation

6.3.2.1 u_int32_t ahstat::ahs_badauth

Definition at line 55 of file ah_var.h.

Referenced by ah_input_cb().

6.3.2.2 u_int32_t ahstat::ahs_badauthl

Definition at line 60 of file ah_var.h.

Referenced by ah_input().

6.3.2.3 u_int32_t ahstat::ahs_badkcr

Definition at line 54 of file ah_var.h.

6.3.2.4 u_int32_t ahstat::ahs_crypto

Definition at line 68 of file ah_var.h.

Referenced by ah_input(), ah_input_cb(), ah_output(), and ah_output_cb().

6.3.2.5 u_int32_t ahstat::ahs_hdrops

Definition at line 51 of file ah_var.h.

Referenced by ah_input(), ah_input_cb(), and ah_output().

6.3.2.6 u_int32_t ahstat::ahs_hist[AH_ALG_MAX]

Definition at line 70 of file ah_var.h.

Referenced by ah_input_cb(), and ah_output_cb().

6.3.2.7 u_int64_t ahstat::ahs_ibytes

Definition at line 64 of file ah_var.h.

Referenced by ah_input().

6.3.2.8 u_int32_t ahstat::ahs_input

Definition at line 61 of file ah_var.h.

6.3.2.9 u_int32_t ahstat::ahs_invalid

Definition at line 63 of file ah_var.h.

6.3.2.10 u_int32_t ahstat::ahs_nopf

Definition at line 52 of file ah_var.h.

Referenced by ah_output().

6.3.2.11 u_int32_t ahstat::ahs_notdb

Definition at line 53 of file ah_var.h.

Referenced by ah_input_cb(), and ah_output_cb().

6.3.2.12 u_int32_t ahstat::ahs_noxform

Definition at line 56 of file ah_var.h.

Referenced by ah_input_cb(), and ah_output_cb().

6.3.2.13 u_int64_t ahstat::ahs_obytes

Definition at line 65 of file ah_var.h.

Referenced by ah_output().

6.3.2.14 u_int32_t ahstat::ahs_output

Definition at line 62 of file ah_var.h.

Referenced by ah_output().

6.3.2.15 u_int32_t ahstat::ahs_pdrops

Definition at line 67 of file ah_var.h.

6.3.2.16 u_int32_t ahstat::ahs_qfull

Definition at line 57 of file ah_var.h.

6.3.2.17 u_int32_t ahstat::ahs_replay

Definition at line 59 of file ah_var.h.

Referenced by ah_input(), and ah_input_cb().

6.3.2.18 u_int32_t ahstat::ahs_toobig

Definition at line 66 of file ah_var.h.

Referenced by ah_output().

6.3.2.19 u_int32_t ahstat::ahs_tunnel

Definition at line 69 of file ah_var.h.

6.3.2.20 u_int32_t ahstat::ahs_wrap

Definition at line 58 of file ah_var.h.

Referenced by ah_output().

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/ah_var.h

6.4 esp Struct Reference

`#include <esp.h>`

Data Fields

- u_int32_t esp_spi

6.4.1 Detailed Description

Definition at line 40 of file esp.h.

6.4.2 Field Documentation

6.4.2.1 u_int32_t esp::esp_spi

Definition at line 41 of file esp.h.

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/esp.h

6.5 espstat Struct Reference

`#include <esp_var.h>`

Data Fields

- u_int32_t esps_hdrops
- u_int32_t esps_nopf
- u_int32_t esps_notdb
- u_int32_t esps_badkcr
- u_int32_t esps_qfull
- u_int32_t esps_noxform
- u_int32_t esps_badilen
- u_int32_t esps_wrap
- u_int32_t esps_badenc
- u_int32_t esps_badauth
- u_int32_t esps_replay
- u_int32_t esps_input
- u_int32_t esps_output
- u_int32_t esps_invalid
- u_int64_t esps_ibytes
- u_int64_t esps_obytes
- u_int32_t esps_toobig
- u_int32_t esps_pdrops
- u_int32_t esps_crypto
- u_int32_t esps_tunnel
- u_int32_t esps_hist [ESP_ALG_MAX]

6.5.1 Detailed Description

Definition at line 50 of file esp_var.h.

6.5.2 Field Documentation

6.5.2.1 u_int32_t espstat::esps_badauth

Definition at line 60 of file esp_var.h.

Referenced by esp_input_cb().

6.5.2.2 u_int32_t espstat::esps_badenc

Definition at line 59 of file esp_var.h.

Referenced by esp_input_cb().

6.5.2.3 u_int32_t espstat::esps_badilen

Definition at line 57 of file esp_var.h.

Referenced by esp_input(), and esp_input_cb().

6.5.2.4 u_int32_t espstat::esps_badkcr

Definition at line 54 of file esp_var.h.

6.5.2.5 u_int32_t espstat::esps_crypto

Definition at line 69 of file esp_var.h.

Referenced by esp_input(), esp_input_cb(), esp_output(), and esp_output_cb().

6.5.2.6 u_int32_t espstat::esps_hdrops

Definition at line 51 of file esp_var.h.

Referenced by esp_input_cb(), and esp_output().

6.5.2.7 u_int32_t espstat::esps_hist[ESP_ALG_MAX]

Definition at line 71 of file esp_var.h.

Referenced by esp_input_cb(), and esp_output_cb().

6.5.2.8 u_int64_t espstat::esps_ibytes

Definition at line 65 of file esp_var.h.

Referenced by esp_input().

6.5.2.9 u_int32_t espstat::esps_input

Definition at line 62 of file esp_var.h.

6.5.2.10 u_int32_t espstat::esps_invalid

Definition at line 64 of file esp_var.h.

6.5.2.11 u_int32_t espstat::esps_nopf

Definition at line 52 of file esp_var.h.

Referenced by esp_output().

6.5.2.12 u_int32_t espstat::esps_notdb

Definition at line 53 of file esp_var.h.

Referenced by esp_input_cb(), and esp_output_cb().

6.5.2.13 u_int32_t espstat::esps_noxform

Definition at line 56 of file esp_var.h.

Referenced by esp_input_cb(), and esp_output_cb().

6.5.2.14 u_int64_t espstat::esps_obytes

Definition at line 66 of file esp_var.h.

Referenced by esp_output().

6.5.2.15 u_int32_t espstat::esps_output

Definition at line 63 of file esp_var.h.

Referenced by esp_output().

6.5.2.16 u_int32_t espstat::esps_pdrops

Definition at line 68 of file esp_var.h.

6.5.2.17 u_int32_t espstat::esps_qfull

Definition at line 55 of file esp_var.h.

6.5.2.18 u_int32_t espstat::esps_replay

Definition at line 61 of file esp_var.h.

Referenced by esp_input(), and esp_input_cb().

6.5.2.19 u_int32_t espstat::esps_toobig

Definition at line 67 of file esp_var.h.

Referenced by esp_output().

6.5.2.20 u_int32_t espstat::esps_tunnel

Definition at line 70 of file esp_var.h.

6.5.2.21 u_int32_t espstat::esps_wrap

Definition at line 58 of file esp_var.h.

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/esp_var.h

6.6 esptail Struct Reference

`#include <esp.h>`

Data Fields

- u_int8_t esp_padlen
- u_int8_t esp_nxt

6.6.1 Detailed Description

Definition at line 62 of file esp.h.

6.6.2 Field Documentation

6.6.2.1 u_int8_t esptail::esp_nxt

Definition at line 64 of file esp.h.

6.6.2.2 u_int8_t esptail::esp_padlen

Definition at line 63 of file esp.h.

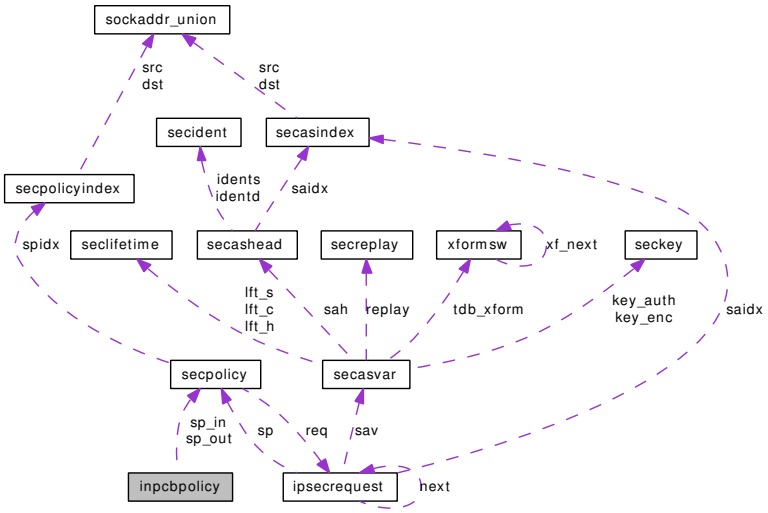
The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/esp.h

6.7 inpcbpolicy Struct Reference

`#include <ipsec.h>`

Collaboration diagram for inpcbpolicy:



Data Fields

- secpolicy * sp_in
- secpolicy * sp_out
- int priv

6.7.1 Detailed Description

Definition at line 136 of file ipsec.h.

6.7.2 Field Documentation

6.7.2.1 int inpcbpolicy::priv

Definition at line 139 of file ipsec.h.

Referenced by ipsec_getpolicybysock().

6.7.2.2 struct secpolicy* inpcbpolicy::sp_in

Definition at line 137 of file ipsec.h.

Referenced by ipsec_getpolicybysock().

6.7.2.3 struct secpolicy* inpcbpolicy::sp_out

Definition at line 138 of file ipsec.h.

Referenced by ipsec_getpolicybysock().

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/ipsec.h

6.8 ipcomp Struct Reference

`#include <ipcomp.h>`

Data Fields

- u_int8_t comp_nxt
- u_int8_t comp_flags
- u_int16_t comp_cpi

6.8.1 Detailed Description

Definition at line 40 of file ipcomp.h.

6.8.2 Field Documentation

6.8.2.1 u_int16_t ipcomp::comp_cpi

Definition at line 43 of file ipcomp.h.

Referenced by ipcomp_output().

6.8.2.2 u_int8_t ipcomp::comp_flags

Definition at line 42 of file ipcomp.h.

Referenced by ipcomp_output().

6.8.2.3 u_int8_t ipcomp::comp_nxt

Definition at line 41 of file ipcomp.h.

Referenced by ipcomp_output().

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/ipcomp.h

6.9 ipcompstat Struct Reference

`#include <ipcomp_var.h>`

Data Fields

- u_int32_t ipcomps_hdrops
- u_int32_t ipcomps_nopf
- u_int32_t ipcomps_notdb
- u_int32_t ipcomps_badkcr
- u_int32_t ipcomps_qfull
- u_int32_t ipcomps_noxform
- u_int32_t ipcomps_wrap
- u_int32_t ipcomps_input
- u_int32_t ipcomps_output
- u_int32_t ipcomps_invalid
- u_int64_t ipcomps_ibytes
- u_int64_t ipcomps_obytes
- u_int32_t ipcomps_toobig
- u_int32_t ipcomps_pdrops
- u_int32_t ipcomps_crypto
- u_int32_t ipcomps_hist [IPCOMP_ALG_MAX]

6.9.1 Detailed Description

Definition at line 44 of file ipcomp_var.h.

6.9.2 Field Documentation

6.9.2.1 u_int32_t ipcompstat::ipcomps_badkcr

Definition at line 48 of file ipcomp_var.h.

6.9.2.2 u_int32_t ipcompstat::ipcomps_crypto

Definition at line 59 of file ipcomp_var.h.

Referenced by ipcomp_input(), ipcomp_input_cb(), ipcomp_output(), and ipcomp_output_cb().

6.9.2.3 u_int32_t ipcompstat::ipcomps_hdrops

Definition at line 45 of file ipcomp_var.h.

Referenced by ipcomp_input_cb(), and ipcomp_output().

6.9.2.4 u_int32_t ipcompstat::ipcomps_hist[IPCOMP_ALG_MAX]

Definition at line 60 of file ipcomp_var.h.

Referenced by ipcomp_input_cb(), and ipcomp_output_cb().

6.9.2.5 u_int64_t ipcompstat::ipcomps_ibytes

Definition at line 55 of file ipcomp_var.h.

6.9.2.6 u_int32_t ipcompstat::ipcomps_input

Definition at line 52 of file ipcomp_var.h.

6.9.2.7 u_int32_t ipcompstat::ipcomps_invalid

Definition at line 54 of file ipcomp_var.h.

6.9.2.8 u_int32_t ipcompstat::ipcomps_nopf

Definition at line 46 of file ipcomp_var.h.

Referenced by ipcomp_output(), and ipcomp_output_cb().

6.9.2.9 u_int32_t ipcompstat::ipcomps_notdb

Definition at line 47 of file ipcomp_var.h.

Referenced by ipcomp_input_cb(), and ipcomp_output_cb().

6.9.2.10 u_int32_t ipcompstat::ipcomps_noxform

Definition at line 50 of file ipcomp_var.h.

Referenced by ipcomp_input_cb(), and ipcomp_output_cb().

6.9.2.11 u_int64_t ipcompstat::ipcomps_obytes

Definition at line 56 of file ipcomp_var.h.

Referenced by ipcomp_output().

6.9.2.12 u_int32_t ipcompstat::ipcomps_output

Definition at line 53 of file ipcomp_var.h.

Referenced by ipcomp_output().

6.9.2.13 u_int32_t ipcompstat::ipcomps_pdrops

Definition at line 58 of file ipcomp_var.h.

6.9.2.14 u_int32_t ipcompstat::ipcomps_qfull

Definition at line 49 of file ipcomp_var.h.

6.9.2.15 u_int32_t ipcompstat::ipcomps_toobig

Definition at line 57 of file ipcomp_var.h.

Referenced by ipcomp_output().

6.9.2.16 u_int32_t ipcompstat::ipcomps_wrap

Definition at line 51 of file ipcomp_var.h.

Referenced by ipcomp_output().

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/ipcomp_var.h

6.10 ipipstat Struct Reference

`#include <ipip_var.h>`

Data Fields

- u_int32_t ipips_ipackets
- u_int32_t ipips_opackets
- u_int32_t ipips_hdrops
- u_int32_t ipips_qfull
- u_int64_t ipips_ibytes
- u_int64_t ipips_obytes
- u_int32_t ipips_pdrops
- u_int32_t ipips_spoof
- u_int32_t ipips_family
- u_int32_t ipips_unspec

6.10.1 Detailed Description

Definition at line 47 of file ipip_var.h.

6.10.2 Field Documentation

6.10.2.1 u_int32_t ipipstat::ipips_family

Definition at line 57 of file ipip_var.h.

Referenced by _ipip_input(), and ipip_output().

6.10.2.2 u_int32_t ipipstat::ipips_hdrops

Definition at line 51 of file ipip_var.h.

Referenced by _ipip_input(), and ipip_output().

6.10.2.3 u_int64_t ipipstat::ipips_ibytes

Definition at line 53 of file ipip_var.h.

Referenced by _ipip_input().

6.10.2.4 u_int32_t ipipstat::ipips_ipackets

Definition at line 49 of file ipip_var.h.

Referenced by _ipip_input().

6.10.2.5 u_int64_t ipipstat::ipips_obytes

Definition at line 54 of file ipip_var.h.

Referenced by ipip_output().

6.10.2.6 u_int32_t ipipstat::ipips_opackets

Definition at line 50 of file ipip_var.h.

Referenced by ipip_output().

6.10.2.7 u_int32_t ipipstat::ipips_pdrops

Definition at line 55 of file ipip_var.h.

6.10.2.8 u_int32_t ipipstat::ipips_qfull

Definition at line 52 of file ipip_var.h.

Referenced by _ipip_input().

6.10.2.9 u_int32_t ipipstat::ipips_spoof

Definition at line 56 of file ipip_var.h.

Referenced by _ipip_input().

6.10.2.10 u_int32_t ipipstat::ipips_unspec

Definition at line 58 of file ipip_var.h.

Referenced by ipip_output().

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/ipip_var.h

6.11 ipsec_history Struct Reference

`#include <ipsec.h>`

Data Fields

- int ih_proto
- u_int32_t ih_spi

6.11.1 Detailed Description

Definition at line 327 of file ipsec.h.

6.11.2 Field Documentation

6.11.2.1 int ipsec_history::ih_proto

Definition at line 328 of file ipsec.h.

6.11.2.2 u_int32_t ipsec_history::ih_spi

Definition at line 329 of file ipsec.h.

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/ipsec.h

6.12 ipsec_output_state Struct Reference

`#include <ipsec.h>`

Data Fields

- mbuf ∗ m
- route ∗ ro
- sockaddr ∗ dst

6.12.1 Detailed Description

Definition at line 321 of file ipsec.h.

6.12.2 Field Documentation

6.12.2.1 struct sockaddr∗ ipsec_output_state::dst

Definition at line 324 of file ipsec.h.

6.12.2.2 struct mbuf∗ ipsec_output_state::m

Definition at line 322 of file ipsec.h.

6.12.2.3 struct route∗ ipsec_output_state::ro

Definition at line 323 of file ipsec.h.

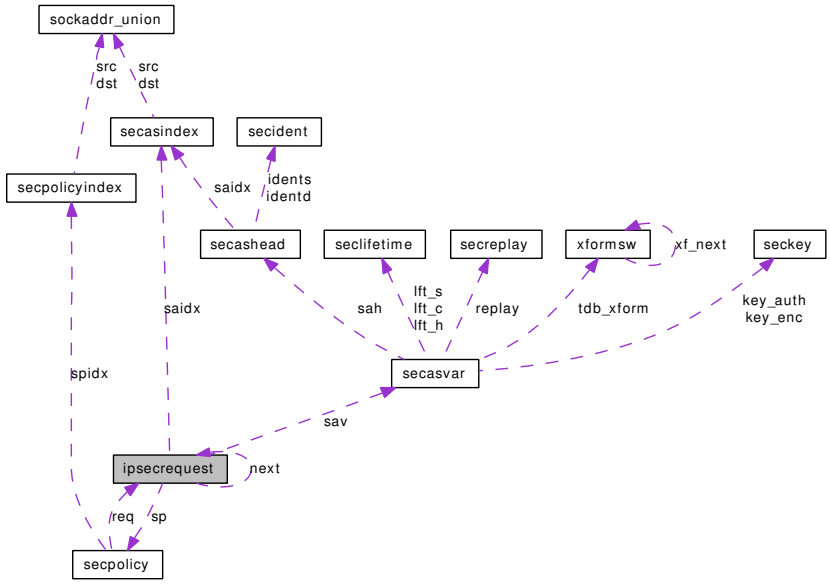
The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/ipsec.h

6.13 ipsecrequest Struct Reference

`#include <ipsec.h>`

Collaboration diagram for ipsecrequest:



Data Fields

- ipsecrequest * next
- secasindex saidx
- u_int level
- secasvar * sav
- secpolicy * sp
- mtx lock

6.13.1 Detailed Description

Definition at line 110 of file ipsec.h.

6.13.2 Field Documentation

6.13.2.1 u_int ipsecrequest::level

Definition at line 116 of file ipsec.h.

Referenced by ipsec_deepcopy_policy(), kdebug_secpolicy(), and key_sp2msg().

6.13.2.2 struct mtx ipsecrequest::lock

Definition at line 120 of file ipsec.h.

6.13.2.3 struct ipsecrequest* ipsecrequest::next

Definition at line 111 of file ipsec.h.

Referenced by ipsec_deepcopy_policy(), ipsec_hdrsiz(), ipsec_in_reject(), ipsec_process_done(), kdebug_secpolicy(), key_delsp(), key_getspreqmsglen(), key_gettunnel(), key_msg2sp(), and key_sp2msg().

6.13.2.4 struct secasindex ipsecrequest::saidx

Definition at line 114 of file ipsec.h.

Referenced by ipsec_deepcopy_policy(), ipsec_hdrsiz(), ipsec_in_reject(), ipsec_nextisr(), kdebug_secpolicy(), key_gettunnel(), key_sp2msg(), and key_spdadd().

6.13.2.5 struct secasvar* ipsecrequest::sav

Definition at line 118 of file ipsec.h.

Referenced by ah_output(), ah_output_cb(), esp_output(), esp_output_cb(), ipcomp_output(), ipcomp_output_cb(), ipip_output(), ipsec_hdrsiz(), ipsec_in_reject(), ipsec_nextisr(), ipsec_process_done(), kdebug_secpolicy(), key_allocsp2(), key_checkrequest(), and key_delsp().

6.13.2.6 struct secpolicy* ipsecrequest::sp

Definition at line 119 of file ipsec.h.

Referenced by ipsec_hdrsiz(), ipsec_in_reject(), key_checkrequest(), and key_delsp().

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/ipsec.h

6.14 ipsecstat Struct Reference

`#include <ipsec.h>`

Data Fields

- u_quad_t in_success
- u_quad_t in_polvio
- u_quad_t in_nosa
- u_quad_t in_inval
- u_quad_t in_nomem
- u_quad_t in_badspi
- u_quad_t in_ahreplay
- u_quad_t in_espreplay
- u_quad_t in_ahauthsucc
- u_quad_t in_ahauthfail
- u_quad_t in_espauthsucc
- u_quad_t in_espauthfail
- u_quad_t in_esphist [256]
- u_quad_t in_ahhist [256]
- u_quad_t in_comphist [256]
- u_quad_t out_success
- u_quad_t out_polvio
- u_quad_t out_nosa
- u_quad_t out_inval
- u_quad_t out_nomem
- u_quad_t out_noroute
- u_quad_t out_esphist [256]
- u_quad_t out_ahhist [256]
- u_quad_t out_comphist [256]
- u_quad_t spdcachelookup
- u_quad_t spdcachemiss

6.14.1 Detailed Description

Definition at line 209 of file ipsec.h.

6.14.2 Field Documentation

6.14.2.1 u_quad_t ipsecstat::in_ahauthfail

Definition at line 220 of file ipsec.h.

6.14.2.2 u_quad_t ipsecstat::in_ahauthsucc

Definition at line 219 of file ipsec.h.

6.14.2.3 u_quad_t ipsecstat::in_ahhist[256]

Definition at line 224 of file ipsec.h.

6.14.2.4 u_quad_t ipsecstat::in_ahreplay

Definition at line 217 of file ipsec.h.

6.14.2.5 u_quad_t ipsecstat::in_badspi

Definition at line 216 of file ipsec.h.

6.14.2.6 u_quad_t ipsecstat::in_comphist[256]

Definition at line 225 of file ipsec.h.

6.14.2.7 u_quad_t ipsecstat::in_espauthfail

Definition at line 222 of file ipsec.h.

6.14.2.8 u_quad_t ipsecstat::in_espauthsucc

Definition at line 221 of file ipsec.h.

6.14.2.9 u_quad_t ipsecstat::in_esphist[256]

Definition at line 223 of file ipsec.h.

6.14.2.10 u_quad_t ipsecstat::in_espreplay

Definition at line 218 of file ipsec.h.

6.14.2.11 u_quad_t ipsecstat::in_inval

Definition at line 214 of file ipsec.h.

6.14.2.12 u_quad_t ipsecstat::in_nomem

Definition at line 215 of file ipsec.h.

6.14.2.13 u_quad_t ipsecstat::in_nosa

Definition at line 213 of file ipsec.h.

6.14.2.14 u_quad_t ipsecstat::in_polvio

Definition at line 211 of file ipsec.h.

6.14.2.15 u_quad_t ipsecstat::in_success

Definition at line 210 of file ipsec.h.

6.14.2.16 u_quad_t ipsecstat::out_ahhist[256]

Definition at line 234 of file ipsec.h.

6.14.2.17 u_quad_t ipsecstat::out_comphist[256]

Definition at line 235 of file ipsec.h.

6.14.2.18 u_quad_t ipsecstat::out_esphist[256]

Definition at line 233 of file ipsec.h.

6.14.2.19 u_quad_t ipsecstat::out_inval

Definition at line 230 of file ipsec.h.

6.14.2.20 u_quad_t ipsecstat::out_nomem

Definition at line 231 of file ipsec.h.

6.14.2.21 u_quad_t ipsecstat::out_noroute

Definition at line 232 of file ipsec.h.

6.14.2.22 u_quad_t ipsecstat::out_nosa

Definition at line 229 of file ipsec.h.

6.14.2.23 u_quad_t ipsecstat::out_polvio

Definition at line 227 of file ipsec.h.

6.14.2.24 u_quad_t ipsecstat::out_success

Definition at line 226 of file ipsec.h.

6.14.2.25 u_quad_t ipsecstat::spdcachelookup

Definition at line 237 of file ipsec.h.

6.14.2.26 u_quad_t ipsecstat::spdcachemiss

Definition at line 238 of file ipsec.h.

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/ipsec.h

6.15 key_cb Struct Reference

Data Fields

- int key_count
- int any_count

6.15.1 Detailed Description

Definition at line 63 of file keysock.c.

6.15.2 Field Documentation

6.15.2.1 int key_cb::any_count

Definition at line 65 of file keysock.c.

Referenced by key_attach(), and key_detach().

6.15.2.2 int key_cb::key_count

Definition at line 64 of file keysock.c.

Referenced by key_attach(), and key_detach().

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/keysock.c

6.16 keycb Struct Reference

`#include <keysock.h>`

Data Fields

- rawcb kp_raw
- int kp_promisc
- int kp_registered

6.16.1 Detailed Description

Definition at line 66 of file keysock.h.

6.16.2 Field Documentation

6.16.2.1 int keycb::kp_promisc

Definition at line 68 of file keysock.h.

Referenced by key_promisc(), and key_sendup_mbuf().

6.16.2.2 struct rawcb keycb::kp_raw

Definition at line 67 of file keysock.h.

Referenced by key_detach().

6.16.2.3 int keycb::kp_registered

Definition at line 69 of file keysock.h.

Referenced by key_register(), and key_sendup_mbuf().

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/keysock.h

6.17 newah Struct Reference

`#include <ah.h>`

Data Fields

- u_int8_t ah_nxt
- u_int8_t ah_len
- u_int16_t ah_reserve
- u_int32_t ah_spi
- u_int32_t ah_seq

6.17.1 Detailed Description

Definition at line 48 of file ah.h.

6.17.2 Field Documentation

6.17.2.1 u_int8_t newah::ah_len

Definition at line 50 of file ah.h.

Referenced by ah_input(), and ah_output().

6.17.2.2 u_int8_t newah::ah_nxt

Definition at line 49 of file ah.h.

Referenced by ah_output().

6.17.2.3 u_int16_t newah::ah_reserve

Definition at line 51 of file ah.h.

Referenced by ah_output().

6.17.2.4 u_int32_t newah::ah_seq

Definition at line 53 of file ah.h.

Referenced by ah_input(), and ah_output().

6.17.2.5 u_int32_t newah::ah_spi

Definition at line 52 of file ah.h.

Referenced by ah_output().

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/ah.h

6.18 newesp Struct Reference

`#include <esp.h>`

Data Fields

- u_int32_t esp_spi
- u_int32_t esp_seq

6.18.1 Detailed Description

Definition at line 51 of file esp.h.

6.18.2 Field Documentation

6.18.2.1 u_int32_t newesp::esp_seq

Definition at line 53 of file esp.h.

Referenced by esp_input().

6.18.2.2 u_int32_t newesp::esp_spi

Definition at line 52 of file esp.h.

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/esp.h

6.19 newipsecstat Struct Reference

`#include <ipsec.h>`

Data Fields

- u_int32_t ips_in_polvio
- u_int32_t ips_out_polvio
- u_int32_t ips_out_nosa
- u_int32_t ips_out_nomem
- u_int32_t ips_out_noroute
- u_int32_t ips_out_inval
- u_int32_t ips_out_bundlesa
- u_int32_t ips_mbcoalesced
- u_int32_t ips_clcoalesced
- u_int32_t ips_clcopied
- u_int32_t ips_mbinserted
- u_int32_t ips_input_front
- u_int32_t ips_input_middle
- u_int32_t ips_input_end

6.19.1 Detailed Description

Definition at line 242 of file ipsec.h.

6.19.2 Field Documentation

6.19.2.1 u_int32_t newipsecstat::ips_clcoalesced

Definition at line 251 of file ipsec.h.

6.19.2.2 u_int32_t newipsecstat::ips_clcopied

Definition at line 252 of file ipsec.h.

6.19.2.3 u_int32_t newipsecstat::ips_in_polvio

Definition at line 243 of file ipsec.h.

Referenced by ipsec4_in_reject().

6.19.2.4 u_int32_t newipsecstat::ips_input_end

Definition at line 260 of file ipsec.h.

6.19.2.5 u_int32_t newipsecstat::ips_input_front

Definition at line 258 of file ipsec.h.

6.19.2.6 u_int32_t newipsecstat::ips_input_middle

Definition at line 259 of file ipsec.h.

6.19.2.7 u_int32_t newipsecstat::ips_mbcoalesced

Definition at line 250 of file ipsec.h.

6.19.2.8 u_int32_t newipsecstat::ips_mbinserted

Definition at line 253 of file ipsec.h.

6.19.2.9 u_int32_t newipsecstat::ips_out_bundlesa

Definition at line 249 of file ipsec.h.

6.19.2.10 u_int32_t newipsecstat::ips_out_inval

Definition at line 248 of file ipsec.h.

Referenced by ipsec4_checkpolicy().

6.19.2.11 u_int32_t newipsecstat::ips_out_nomem

Definition at line 246 of file ipsec.h.

6.19.2.12 u_int32_t newipsecstat::ips_out_noroute

Definition at line 247 of file ipsec.h.

6.19.2.13 u_int32_t newipsecstat::ips_out_nosa

Definition at line 245 of file ipsec.h.

6.19.2.14 u_int32_t newipsecstat::ips_out_polvio

Definition at line 244 of file ipsec.h.

Referenced by ipsec4_checkpolicy().

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/ipsec.h

6.20 pfkeystat Struct Reference

`#include <keysock.h>`

Data Fields

- u_quad_t out_total
- u_quad_t out_bytes
- u_quad_t out_msgtype [256]
- u_quad_t out_invlen
- u_quad_t out_invver
- u_quad_t out_invmsgtype
- u_quad_t out_tooshort
- u_quad_t out_nomem
- u_quad_t out_dupext
- u_quad_t out_invexttype
- u_quad_t out_invsatype
- u_quad_t out_invaddr
- u_quad_t in_total
- u_quad_t in_bytes
- u_quad_t in_msgtype [256]
- u_quad_t in_msgtarget [3]
- u_quad_t in_nomem
- u_quad_t sockerr

6.20.1 Detailed Description

Definition at line 37 of file keysock.h.

6.20.2 Field Documentation

6.20.2.1 u_quad_t pfkeystat::in_bytes

Definition at line 53 of file keysock.h.

Referenced by key_sendup(), and key_sendup_mbuf().

6.20.2.2 u_quad_t pfkeystat::in_msgtarget[3]

Definition at line 55 of file keysock.h.

Referenced by key_sendup_mbuf().

6.20.2.3 u_quad_t pfkeystat::in_msgtype[256]

Definition at line 54 of file keysock.h.

Referenced by key_sendup(), key_sendup0(), and key_sendup_mbuf().

6.20.2.4 u_quad_t pfkeystat::in_nomem

Definition at line 56 of file keysock.h.

Referenced by key_sendup(), key_sendup0(), and key_sendup_mbuf().

6.20.2.5 u_quad_t pfkeystat::in_total

Definition at line 52 of file keysock.h.

Referenced by key_sendup(), and key_sendup_mbuf().

6.20.2.6 u_quad_t pfkeystat::out_bytes

Definition at line 40 of file keysock.h.

Referenced by key_output().

6.20.2.7 u_quad_t pfkeystat::out_dupext

Definition at line 47 of file keysock.h.

6.20.2.8 u_quad_t pfkeystat::out_invaddr

Definition at line 50 of file keysock.h.

6.20.2.9 u_quad_t pfkeystat::out_invexttype

Definition at line 48 of file keysock.h.

6.20.2.10 u_quad_t pfkeystat::out_invlen

Definition at line 42 of file keysock.h.

Referenced by key_output().

6.20.2.11 u_quad_t pfkeystat::out_invmsgtype

Definition at line 44 of file keysock.h.

6.20.2.12 u_quad_t pfkeystat::out_invsatype

Definition at line 49 of file keysock.h.

6.20.2.13 u_quad_t pfkeystat::out_invver

Definition at line 43 of file keysock.h.

6.20.2.14 u_quad_t pfkeystat::out_msgtype[256]

Definition at line 41 of file keysock.h.

Referenced by key_output().

6.20.2.15 u_quad_t pfkeystat::out_nomem

Definition at line 46 of file keysock.h.

Referenced by key_output().

6.20.2.16 u_quad_t pfkeystat::out_tooshort

Definition at line 45 of file keysock.h.

Referenced by key_output().

6.20.2.17 u_quad_t pfkeystat::out_total

Definition at line 39 of file keysock.h.

Referenced by key_output().

6.20.2.18 u_quad_t pfkeystat::sockerr

Definition at line 58 of file keysock.h.

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/keysock.h

6.21 sadb_msghdr Struct Reference

Data Fields

- sadb_msg * msg
- sadb_ext * ext [SADB_EXT_MAX+1]
- int extoff [SADB_EXT_MAX+1]
- int extlen [SADB_EXT_MAX+1]

6.21.1 Detailed Description

Definition at line 366 of file key.c.

6.21.2 Field Documentation

6.21.2.1 struct sadb_ext* sadb_msghdr::ext[SADB_EXT_MAX+1]

Definition at line 368 of file key.c.

Referenced by key_acquire2(), key_add(), key_delete(), key_delete_all(), key_gather_mbuf(), key_get(), key_getspi(), key_newsav(), key_parse(), key_setident(), key_setsaval(), key_spdadd(), key_spddelete(), key_spddelete2(), key_spdget(), and key_update().

6.21.2.2 int sadb_msghdr::extlen[SADB_EXT_MAX+1]

Definition at line 370 of file key.c.

Referenced by key_acquire2(), key_add(), key_delete(), key_gather_mbuf(), key_get(), key_getspi(), key_setident(), key_setsaval(), key_spdadd(), key_spddelete(), key_spddelete2(), key_spdget(), and key_update().

6.21.2.3 int sadb_msghdr::extoff[SADB_EXT_MAX+1]

Definition at line 369 of file key.c.

Referenced by key_gather_mbuf(), and key_spddelete2().

6.21.2.4 struct sadb_msg* sadb_msghdr::msg

Definition at line 367 of file key.c.

Referenced by key_acquire2(), key_add(), key_delete(), key_dump(), key_flush(), key_gather_mbuf(), key_get(), key_getmsgbuf_x1(), key_getspi(), key_newsav(), key_parse(), key_promisc(), key_register(), key_setident(), key_setsaval(), key_spdadd(), key_spddelete(), key_spddelete2(), key_spddump(), key_spdflush(), key_spdget(), and key_update().

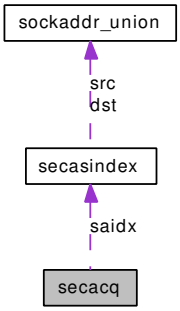
The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/key.c

6.22 secacq Struct Reference

`#include <keydb.h>`

Collaboration diagram for secacq:



Public Member Functions

- LIST_ENTRY (secacq) chain

Data Fields

- secasindex saidx
- u_int32_t seq
- time_t created
- int count

6.22.1 Detailed Description

Definition at line 181 of file keydb.h.

6.22.2 Member Function Documentation

6.22.2.1 secacq::LIST_ENTRY (secacq)

6.22.3 Field Documentation

6.22.3.1 int secacq::count

Definition at line 188 of file keydb.h.

Referenced by key_acquire(), key_acquire2(), key_getspi(), and key_newacq().

6.22.3.2 time_t secacq::created

Definition at line 187 of file keydb.h.

Referenced by key_acquire2(), key_flush_acq(), key_getspi(), and key_newacq().

6.22.3.3 struct secasindex secacq::saidx

Definition at line 184 of file keydb.h.

Referenced by key_acquire(), key_getacq(), and key_newacq().

6.22.3.4 u_int32_t secacq::seq

Definition at line 186 of file keydb.h.

Referenced by key_acquire(), key_getacqbyseq(), and key_newacq().

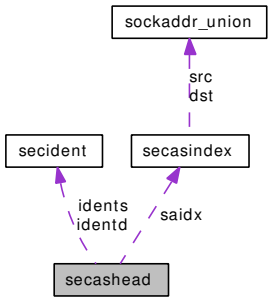
The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/keydb.h

6.23 secashead Struct Reference

`#include <keydb.h>`

Collaboration diagram for secashead:



Public Member Functions

- LIST_ENTRY (secashead) chain
- LIST_HEAD (_satree, secasvar) savtree[SADB_SASTATE_MAX+1]

Data Fields

- secasindex saidx
- secident * idents
- secident * identd
- u_int8_t state
- route sa_route

6.23.1 Detailed Description

Definition at line 89 of file keydb.h.

6.23.2 Member Function Documentation

6.23.2.1 secashead::LIST_ENTRY (secashead)

6.23.2.2 secashead::LIST_HEAD (_satree, secasvar)

6.23.3 Field Documentation

6.23.3.1 struct secident* secashead::identd

Definition at line 95 of file keydb.h.

6.23.3.2 struct secident* secashead::idents

Definition at line 94 of file keydb.h.

6.23.3.3 struct route secashead::sa_route

Definition at line 103 of file keydb.h.

Referenced by key_sa_routechange().

6.23.3.4 struct secasindex secashead::saidx

Definition at line 92 of file keydb.h.

Referenced by ah_input(), ah_input_cb(), ah_output(), esp_input_cb(), ipcomp_input(), ipcomp_input_cb(), ipcomp_output(), ipcomp_output_cb(), ipip_output(), ipsec_process_done(), key_acquire2(), key_allocsa_policy(), key_checkspidup(), key_delete(), key_delete_all(), key_do_allocsa_policy(), key_dump(), key_expire(), key_flush(), key_get(), key_getsah(), key_mature(), key_newsah(), and key_update().

6.23.3.5 u_int8_t secashead::state

Definition at line 98 of file keydb.h.

Referenced by key_acquire2(), key_allocsa_policy(), key_delete(), key_delete_all(), key_flush(), key_flush_sad(), key_get(), key_getsah(), and key_newsah().

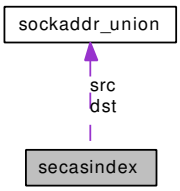
The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/keydb.h

6.24 secasindex Struct Reference

`#include <keydb.h>`

Collaboration diagram for secasindex:



Data Fields

- sockaddr_union src
- sockaddr_union dst
- u_int16_t proto
- u_int8_t mode
- u_int32_t reqid

6.24.1 Detailed Description

Definition at line 54 of file keydb.h.

6.24.2 Field Documentation

6.24.2.1 union sockaddr_union secasindex::dst

Definition at line 56 of file keydb.h.

Referenced by ah_input(), ah_input_cb(), ah_output(), esp_input_cb(), esp_output(), ipcomp_input(), ipcomp_input_cb(), ipcomp_output(), ipcomp_output_cb(), ipip_output(), ipsec_deepcopy_policy(), ipsec_hdrsiz(), ipsec_logsastr(), ipsec_nextisr(), ipsec_process_done(), key_acquire(), key_checkspidup(), key_cmpsaidx(), key_do_allocsa_policy(), key_expire(), key_gettunnel(), key_sp2msg(), and key_-spdadd().

6.24.2.2 u_int8_t secasindex::mode

Definition at line 58 of file keydb.h.

Referenced by ipsec_deepcopy_policy(), ipsec_hdrsiz(), ipsec_nextisr(), key_checkrequest(), key_-cmpsaidx(), key_expire(), key_gettunnel(), and key_sp2msg().

6.24.2.3 u_int16_t secasindex::proto

Definition at line 57 of file keydb.h.

Referenced by ah_input(), ah_output(), esp_output(), ipcomp_input(), ipcomp_output(), ipsec_deepcopy_policy(), ipsec_hdrsiz(), ipsec_in_reject(), ipsec_nextisr(), ipsec_process_done(), key_acquire(), key_cmpsaidx(), key_do_allocsa_policy(), key_do_getnewspi(), key_dump(), key_expire(), key_flush(), key_get(), key_mature(), key_sp2msg(), and key_update().

6.24.2.4 u_int32_t secasindex::reqid

Definition at line 59 of file keydb.h.

Referenced by ipsec_deepcopy_policy(), key_cmpsaidx(), key_expire(), and key_sp2msg().

6.24.2.5 union sockaddr_union secasindex::src

Definition at line 55 of file keydb.h.

Referenced by ipip_output(), ipsec_deepcopy_policy(), ipsec_logsastr(), ipsec_nextisr(), key_acquire(), key_cmpsaidx(), key_do_allocsa_policy(), key_expire(), key_gettunnel(), key_sp2msg(), and key_spdadd().

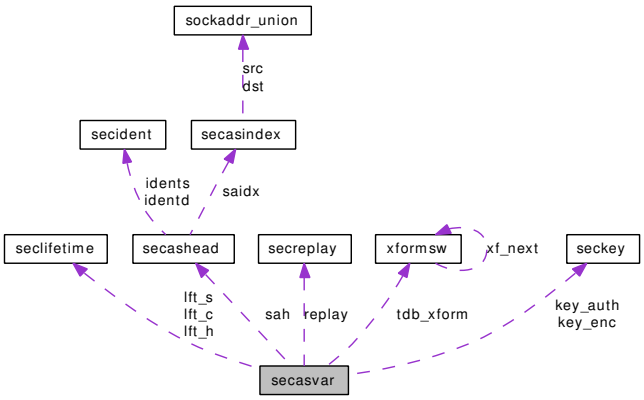
The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/keydb.h

6.25 secasvar Struct Reference

`#include <keydb.h>`

Collaboration diagram for secasvar:



Public Member Functions

- LIST_ENTRY (secasvar) chain

Data Fields

- mtx lock
- u_int refcnt
- u_int8_t state
- u_int8_t alg_auth
- u_int8_t alg_enc
- u_int8_t alg_comp
- u_int32_t spi
- u_int32_t flags
- seckey * key_auth
- seckey * key_enc
- caddr_t iv
- u_int ivlen
- void * sched
- size_t schedlen
- secreplay * replay
- time_t created
- seclifetime * lft_c
- seclifetime * lft_h
- seclifetime * lft_s
- u_int32_t seq
- pid_t pid
- secashead * sah
- xformsw * tdb_xform
- enc_xform * tdb_encalgxform

- auth_hash * tdb_authalgxform
- comp_algo * tdb_compalgxform
- u_int64_t tdb_cryptoid

6.25.1 Detailed Description

Definition at line 112 of file keydb.h.

6.25.2 Member Function Documentation

6.25.2.1 secasvar::LIST_ENTRY (secasvar)

6.25.3 Field Documentation

6.25.3.1 u_int8_t secasvar::alg_auth

Definition at line 119 of file keydb.h.

Referenced by ah_init0(), ah_input_cb(), ah_output_cb(), esp_init(), esp_input_cb(), esp_output_cb(), key_mature(), and tcpsignature_init().

6.25.3.2 u_int8_t secasvar::alg_comp

Definition at line 121 of file keydb.h.

Referenced by ipcomp_init(), ipcomp_input_cb(), and ipcomp_output_cb().

6.25.3.3 u_int8_t secasvar::alg_enc

Definition at line 120 of file keydb.h.

Referenced by esp_init(), esp_input_cb(), esp_output_cb(), ipcomp_init(), and key_mature().

6.25.3.4 time_t secasvar::created

Definition at line 133 of file keydb.h.

Referenced by key_flush_sad(), and key_newsav().

6.25.3.5 u_int32_t secasvar::flags

Definition at line 123 of file keydb.h.

Referenced by ah_init0(), ah_output(), esp_hdrsiz(), esp_init(), esp_input(), esp_input_cb(), esp_output(), ipsec_updatereplay(), and key_mature().

6.25.3.6 caddr_t secasvar::iv

Definition at line 127 of file keydb.h.

Referenced by esp_init(), esp_zeroize(), and key_cleansav().

6.25.3.7 u_int secasvar::ivlen

Definition at line 128 of file keydb.h.

Referenced by esp_init().

6.25.3.8 struct seckey* secasvar::key_auth

Definition at line 125 of file keydb.h.

Referenced by ah_init0(), ah_input(), ah_output(), ah_zeroize(), key_cleansav(), tcpsignature_init(), and tcpsignature_zeroize().

6.25.3.9 struct seckey* secasvar::key_enc

Definition at line 126 of file keydb.h.

Referenced by esp_init(), esp_zeroize(), and key_cleansav().

6.25.3.10 struct seclifetime* secasvar::lft_c

Definition at line 135 of file keydb.h.

Referenced by key_cleansav(), key_do_allocsa_policy(), key_expire(), and key_flush_sad().

6.25.3.11 struct seclifetime* secasvar::lft_h

Definition at line 136 of file keydb.h.

Referenced by key_cleansav(), and key_flush_sad().

6.25.3.12 struct seclifetime* secasvar::lft_s

Definition at line 137 of file keydb.h.

Referenced by key_cleansav(), key_expire(), and key_flush_sad().

6.25.3.13 struct mtx secasvar::lock

Definition at line 114 of file keydb.h.

6.25.3.14 pid_t secasvar::pid

Definition at line 140 of file keydb.h.

Referenced by key_newsav(), and key_update().

6.25.3.15 u_int secasvar::refcnt

Definition at line 116 of file keydb.h.

Referenced by key_allocsa(), key_do_allocsa_policy(), key_expire(), key_freesav(), sa_addref(), sa_-delref(), and sa_initref().

6.25.3.16 struct secreplay* secasvar::replay

Definition at line 132 of file keydb.h.

Referenced by ah_init0(), ah_input(), ah_input_cb(), ah_output(), esp_hdrsiz(), esp_input_cb(), ipsec_-chkreplay(), ipsec_updatereplay(), key_cleansav(), and key_expire().

6.25.3.17 struct secashead* secasvar::sah

Definition at line 142 of file keydb.h.

Referenced by ah_input(), ah_input_cb(), ah_output(), esp_input_cb(), ipcomp_input(), ipcomp_input_-cb(), ipcomp_output(), ipcomp_output_cb(), ipip_output(), ipsec_process_done(), key_allocsa(), key_-allocsa_policy(), key_checkrequest(), key_checkspidup(), key_do_allocsa_policy(), key_expire(), key_-flush(), key_flush_sad(), key_mature(), key_newsav(), and key_update().

6.25.3.18 void* secasvar::sched

Definition at line 129 of file keydb.h.

Referenced by key_cleansav().

6.25.3.19 size_t secasvar::schedlen

Definition at line 130 of file keydb.h.

Referenced by key_cleansav().

6.25.3.20 u_int32_t secasvar::seq

Definition at line 139 of file keydb.h.

Referenced by key_expire(), key_getspi(), and key_newsav().

6.25.3.21 u_int32_t secasvar::spi

Definition at line 122 of file keydb.h.

Referenced by ah_input(), ah_input_cb(), ah_output(), esp_input_cb(), ipcomp_input(), ipcomp_input_-cb(), ipcomp_output(), ipcomp_output_cb(), ipip_output(), ipsec_common_input(), ipsec_process_-done(), key_allocsp2(), key_freesav(), key_getspi(), key_mature(), key_newsav(), key_update(), and tcpsignature_init().

6.25.3.22 u_int8_t secasvar::state

Definition at line 117 of file keydb.h.

Referenced by key_allocsa(), key_checkrequest(), key_delete_all(), key_delsah(), key_do_allocsa_-policy(), key_flush_sad(), key_getsavbyspi(), and key_newsav().

6.25.3.23 struct auth_hash* secasvar::tdb_authalgxform

Definition at line 151 of file keydb.h.

Referenced by ah_hdrsiz(), ah_init0(), ah_input(), ah_input_cb(), ah_output(), ah_zeroize(), esp_hdrsiz(), esp_init(), esp_input(), esp_input_cb(), esp_output(), esp_output_cb(), ipsec_in_reject(), and tcpsignature_zeroize().

6.25.3.24 struct comp_algo* secasvar::tdb_compalgxform

Definition at line 152 of file keydb.h.

Referenced by ipcomp_init(), ipcomp_input(), and ipcomp_output().

6.25.3.25 u_int64_t secasvar::tdb_cryptoid

Definition at line 153 of file keydb.h.

Referenced by ah_init(), ah_input(), ah_input_cb(), ah_output(), ah_output_cb(), ah_zeroize(), esp_init(), esp_input_cb(), esp_output_cb(), ipcomp_init(), ipcomp_input(), ipcomp_input_cb(), ipcomp_output(), ipcomp_output_cb(), ipcomp_zeroize(), and tcpsignature_zeroize().

6.25.3.26 struct enc_xform* secasvar::tdb_encalgxform

Definition at line 150 of file keydb.h.

Referenced by esp_hdrsiz(), esp_init(), esp_input(), esp_input_cb(), esp_output(), and esp_zeroize().

6.25.3.27 struct xformsw* secasvar::tdb_xform

Definition at line 149 of file keydb.h.

Referenced by ah_init0(), ah_zeroize(), esp_init(), esp_zeroize(), ipcomp_init(), ipip_output(), ipsec_common_input(), ipsec_nextisr(), key_cleansav(), tcpsignature_zeroize(), and xform_init().

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/keydb.h

6.26 secident Struct Reference

`#include <keydb.h>`

Data Fields

- u_int16_t type
- u_int64_t id

6.26.1 Detailed Description

Definition at line 70 of file keydb.h.

6.26.2 Field Documentation

6.26.2.1 u_int64_t secident::id

Definition at line 72 of file keydb.h.

6.26.2.2 u_int16_t secident::type

Definition at line 71 of file keydb.h.

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/keydb.h

6.27 seckey Struct Reference

`#include <keydb.h>`

Data Fields

- u_int16_t bits
- char * key_data

6.27.1 Detailed Description

Definition at line 76 of file keydb.h.

6.27.2 Field Documentation

6.27.2.1 u_int16_t seckey::bits

Definition at line 77 of file keydb.h.

Referenced by key_dup_keymsg(), and key_setkey().

6.27.2.2 char* seckey::key_data

Definition at line 78 of file keydb.h.

Referenced by ah_init0(), ah_input(), ah_output(), ah_zeroize(), esp_init(), esp_zeroize(), key_cleansav(), key_dup_keymsg(), key_setkey(), and tcpsignature_zeroize().

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/keydb.h

6.28 seclifetime Struct Reference

`#include <keydb.h>`

Data Fields

- u_int32_t allocations
- u_int64_t bytes
- u_int64_t addtime
- u_int64_t usetime

6.28.1 Detailed Description

Definition at line 81 of file keydb.h.

6.28.2 Field Documentation

6.28.2.1 u_int64_t seclifetime::addtime

Definition at line 84 of file keydb.h.

Referenced by key_do_allocsa_policy(), key_dup_lifemsg(), key_expire(), key_flush_sad(), and key_-setlifetime().

6.28.2.2 u_int32_t seclifetime::allocations

Definition at line 82 of file keydb.h.

Referenced by key_dup_lifemsg(), key_expire(), and key_setlifetime().

6.28.2.3 u_int64_t seclifetime::bytes

Definition at line 83 of file keydb.h.

Referenced by key_dup_lifemsg(), key_expire(), key_flush_sad(), and key_setlifetime().

6.28.2.4 u_int64_t seclifetime::usetime

Definition at line 85 of file keydb.h.

Referenced by key_dup_lifemsg(), key_expire(), key_flush_sad(), and key_setlifetime().

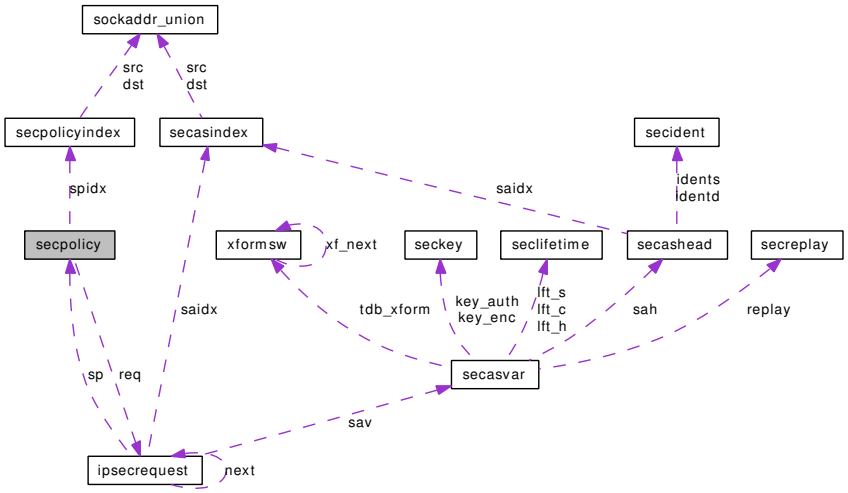
The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/keydb.h

6.29 secpolicy Struct Reference

`#include <ipsec.h>`

Collaboration diagram for secpolicy:



Public Member Functions

- LIST_ENTRY (secpolicy) chain

Data Fields

- mtx lock
- u_int refcnt
- secpolicyindex spidx
- u_int32_t id
- u_int state
- u_int16_t policy
- u_int16_t scangen
- ipsecrequest * req
- time_t created
- time_t lastused
- long lifetime
- long validtime

6.29.1 Detailed Description

Definition at line 73 of file ipsec.h.

6.29.2 Member Function Documentation

6.29.2.1 secpolicy::LIST_ENTRY (secpolicy)

6.29.3 Field Documentation

6.29.3.1 time_t secpolicy::created

Definition at line 96 of file ipsec.h.

Referenced by key_spdadd().

6.29.3.2 u_int32_t secpolicy::id

Definition at line 79 of file ipsec.h.

Referenced by _key_freesp(), key_acquire(), key_allocsp(), key_allocsp2(), key_getspbyid(), key_-gettunnel(), key_spdadd(), and key_spddelete().

6.29.3.3 time_t secpolicy::lastused

Definition at line 97 of file ipsec.h.

Referenced by key_allocsp(), key_allocsp2(), key_gettunnel(), and key_spdadd().

6.29.3.4 long secpolicy::lifetime

Definition at line 98 of file ipsec.h.

Referenced by key_spdadd().

6.29.3.5 struct mtx secpolicy::lock

Definition at line 75 of file ipsec.h.

6.29.3.6 u_int16_t secpolicy::policy

Definition at line 83 of file ipsec.h.

Referenced by ipsec4_checkpolicy(), ipsec_deepcopy_policy(), ipsec_getpolicybysock(), key_acquire(), key_allocsp_default(), key_init(), and key_msg2sp().

6.29.3.7 u_int secpolicy::refcnt

Definition at line 77 of file ipsec.h.

Referenced by _key_freesp(), ipsec_attach(), ipsec_getpolicybysock(), key_allocsp(), key_allocsp2(), key_allocsp_default(), key_delsp(), key_gettunnel(), key_init(), and key_spdadd().

6.29.3.8 struct ipsecrequest* secpolicy::req

Definition at line 85 of file ipsec.h.

Referenced by ipsec4_checkpolicy(), ipsec_deepcopy_policy(), key_allocsp2(), key_gettunnel(), key_-msg2sp(), and key_spdadd().

6.29.3.9 u_int16_t secpolicy::scangen

Definition at line 84 of file ipsec.h.

6.29.3.10 struct secpolicyindex secpolicy::spidx

Definition at line 78 of file ipsec.h.

Referenced by ipsec_getpolicybyaddr(), ipsec_getpolicybysock(), key_acquire(), key_allocsp(), key_-allocsp2(), key_getsp(), key_gettunnel(), key_msg2sp(), and key_spdadd().

6.29.3.11 u_int secpolicy::state

Definition at line 80 of file ipsec.h.

Referenced by ipsec_deepcopy_policy(), ipsec_set_policy(), key_allocsp(), key_allocsp2(), key_delsp(), key_getsp(), key_getspbyid(), key_gettunnel(), key_spdadd(), key_spddelete(), key_spddelete2(), and key_spdflush().

6.29.3.12 long secpolicy::validtime

Definition at line 99 of file ipsec.h.

Referenced by key_spdadd().

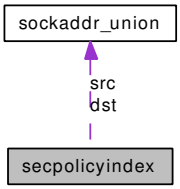
The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/ipsec.h

6.30 secpolicyindex Struct Reference

`#include <ipsec.h>`

Collaboration diagram for secpolicyindex:



Data Fields

- u_int8_t dir
- sockaddr_union src
- sockaddr_union dst
- u_int8_t prefs
- u_int8_t prefd
- u_int16_t ul_proto

6.30.1 Detailed Description

Definition at line 57 of file ipsec.h.

6.30.2 Field Documentation

6.30.2.1 u_int8_t secpolicyindex::dir

Definition at line 58 of file ipsec.h.

Referenced by ipsec_getpolicybyaddr(), key_acquire(), key_allocsp(), key_allocsp2(), key_getsp(), key_-msg2sp(), and key_spdadd().

6.30.2.2 union sockaddr_union secpolicyindex::dst

Definition at line 60 of file ipsec.h.

Referenced by ipsec4_get_ulp(), ipsec4_setspidx_ipaddr(), key_allocsp2(), key_cmpspidx_exactly(), key_cmpspidx_withmask(), and key_gettunnel().

6.30.2.3 u_int8_t secpolicyindex::prefd

Definition at line 62 of file ipsec.h.

Referenced by ipsec4_setspidx_ipaddr(), key_cmpspidx_exactly(), and key_cmpspidx_withmask().

6.30.2.4 u_int8_t secpolicyindex::prefs

Definition at line 61 of file ipsec.h.

Referenced by ipsec4_setspidx_ipaddr(), key_cmpspidx_exactly(), and key_cmpspidx_withmask().

6.30.2.5 union sockaddr_union secpolicyindex::src

Definition at line 59 of file ipsec.h.

Referenced by ipsec4_get_ulp(), ipsec4_setspidx_ipaddr(), key_cmpspidx_exactly(), key_cmpspidx_withmask(), and key_gettunnel().

6.30.2.6 u_int16_t secpolicyindex::ul_proto

Definition at line 63 of file ipsec.h.

Referenced by ipsec4_get_ulp(), key_allocsp2(), key_cmpspidx_exactly(), and key_cmpspidx_withmask().

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/ipsec.h

6.31 secreg Struct Reference

`#include <keydb.h>`

Public Member Functions

- LIST_ENTRY (secreg) chain

Data Fields

- socket * so

6.31.1 Detailed Description

Definition at line 174 of file keydb.h.

6.31.2 Member Function Documentation

6.31.2.1 secreg::LIST_ENTRY (secreg)

6.31.3 Field Documentation

6.31.3.1 struct socket* secreg::so

Definition at line 177 of file keydb.h.

Referenced by key_freereg(), and key_register().

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/keydb.h

6.32 secreplay Struct Reference

`#include <keydb.h>`

Data Fields

- u_int32_t count
- u_int wsize
- u_int32_t seq
- u_int32_t lastseq
- caddr_t bitmap
- int overflow

6.32.1 Detailed Description

Definition at line 164 of file keydb.h.

6.32.2 Field Documentation

6.32.2.1 caddr_t secreplay::bitmap

Definition at line 169 of file keydb.h.

Referenced by ipsec_chkreplay(), and ipsec_updatereplay().

6.32.2.2 u_int32_t secreplay::count

Definition at line 165 of file keydb.h.

Referenced by ah_output(), ipsec_chkreplay(), ipsec_updatereplay(), and key_expire().

6.32.2.3 u_int32_t secreplay::lastseq

Definition at line 168 of file keydb.h.

Referenced by ipsec_chkreplay(), and ipsec_updatereplay().

6.32.2.4 int secreplay::overflow

Definition at line 170 of file keydb.h.

Referenced by ipsec_updatereplay().

6.32.2.5 u_int32_t secreplay::seq

Definition at line 167 of file keydb.h.

6.32.2.6 u_int secreplay::wsize

Definition at line 166 of file keydb.h.

Referenced by ipsec_chkreplay(), and ipsec_updatereplay().

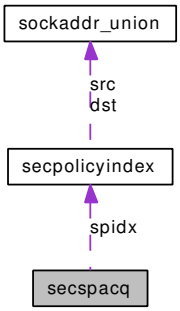
The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/keydb.h

6.33 secspacq Struct Reference

`#include <ipsec.h>`

Collaboration diagram for secspacq:



Public Member Functions

- LIST_ENTRY (secspacq) chain

Data Fields

- secpolicyindex spidx
- time_t created
- int count

6.33.1 Detailed Description

Definition at line 143 of file ipsec.h.

6.33.2 Member Function Documentation

6.33.2.1 secspacq::LIST_ENTRY (secspacq)

6.33.3 Field Documentation

6.33.3.1 int secspacq::count

Definition at line 149 of file ipsec.h.

Referenced by key_newspacq(), key_spdacquire(), and key_spdadd().

6.33.3.2 time_t secspacq::created

Definition at line 148 of file ipsec.h.

Referenced by key_flush_spacq(), key_newspacq(), and key_spdadd().

6.33.3.3 struct secpolicyindex secspacq::spidx

Definition at line 146 of file ipsec.h.

Referenced by key_getspacq(), key_newspacq(), and key_spdadd().

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/ipsec.h

6.34 sockaddr_union Union Reference

`#include <keydb.h>`

Data Fields

- sockaddr sa
- sockaddr_in sin
- sockaddr_in6 sin6

6.34.1 Detailed Description

Definition at line 45 of file keydb.h.

6.34.2 Field Documentation

6.34.2.1 struct sockaddr sockaddr_union::sa

Definition at line 46 of file keydb.h.

Referenced by ah_input(), ah_input_cb(), ah_output(), esp_input_cb(), esp_output(), ipcomp_input_cb(), ipcomp_output(), ipcomp_output_cb(), ipip_output(), ipsec_address(), ipsec_hdrsiz(), ipsec_logsastr(), ipsec_nextisr(), ipsec_process_done(), key_acquire(), key_allocsa(), key_allocsp2(), key_cmpsaidx(), key_cmpspidx_exactly(), key_cmpspidx_withmask(), key_do_allocsa_policy(), key_expire(), key_gettunnel(), key_sp2msg(), and key_spdadd().

6.34.2.2 struct sockaddr_in sockaddr_union::sin

Definition at line 47 of file keydb.h.

Referenced by ipip_output(), ipsec4_get_ulp(), ipsec4_setspidx_ipaddr(), ipsec_address(), ipsec_nextisr(), and key_cmpspidx_withmask().

6.34.2.3 struct sockaddr_in6 sockaddr_union::sin6

Definition at line 48 of file keydb.h.

Referenced by ipip_output(), ipsec_address(), ipsec_nextisr(), and key_cmpspidx_withmask().

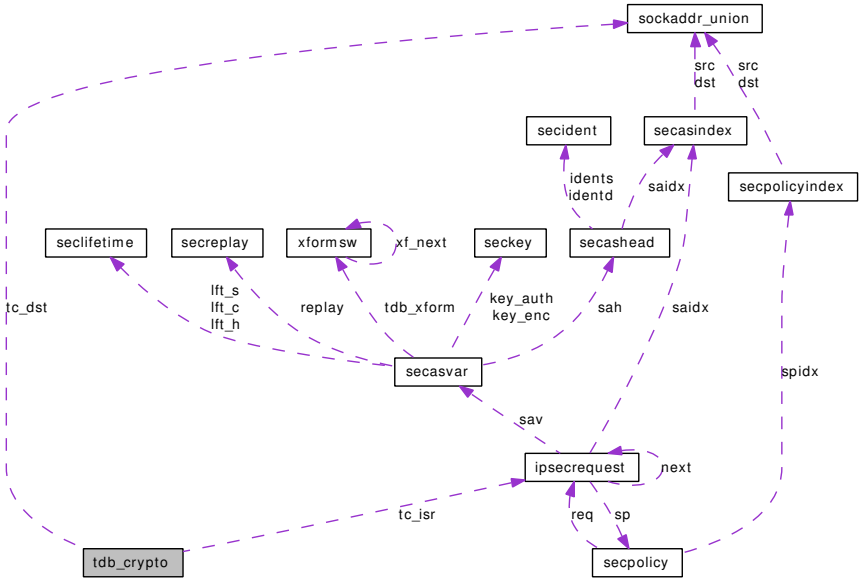
The documentation for this union was generated from the following file:

- /usr/src/sys/netipsec/keydb.h

6.35 tdb_crypto Struct Reference

`#include <xform.h>`

Collaboration diagram for tdb_crypto:



Data Fields

- ipsecrequest * tc_isr
- u_int32_t tc_spi
- sockaddr_union tc_dst
- u_int8_t tc_proto
- u_int8_t tc_nxt
- int tc_protoff
- int tc_skip
- caddr_t tc_ptr

6.35.1 Detailed Description

Definition at line 65 of file xform.h.

6.35.2 Field Documentation

6.35.2.1 union sockaddr_union tdb_crypto::tc_dst

Definition at line 68 of file xform.h.

Referenced by ah_input(), ah_input_cb(), ah_output(), ah_output_cb(), esp_input(), esp_input_cb(), esp_output(), esp_output_cb(), ipcomp_input_cb(), ipcomp_output(), and ipcomp_output_cb().

6.35.2.2 struct ipsecrequest* tdb_crypto::tc_isr

Definition at line 66 of file xform.h.

Referenced by ah_output(), ah_output_cb(), esp_output(), esp_output_cb(), ipcomp_output(), and ipcomp_output_cb().

6.35.2.3 u_int8_t tdb_crypto::tc_nxt

Definition at line 70 of file xform.h.

Referenced by ah_input(), and ah_input_cb().

6.35.2.4 u_int8_t tdb_crypto::tc_proto

Definition at line 69 of file xform.h.

Referenced by ah_input(), ah_input_cb(), ah_output(), ah_output_cb(), esp_input(), esp_input_cb(), esp_output(), esp_output_cb(), ipcomp_input_cb(), ipcomp_output(), and ipcomp_output_cb().

6.35.2.5 int tdb_crypto::tc_protoff

Definition at line 71 of file xform.h.

Referenced by ah_input(), ah_input_cb(), ah_output(), ah_output_cb(), esp_input(), esp_input_cb(), and ipcomp_input_cb().

6.35.2.6 caddr_t tdb_crypto::tc_ptr

Definition at line 73 of file xform.h.

Referenced by ah_input(), ah_input_cb(), esp_input(), esp_input_cb(), and ipcomp_input_cb().

6.35.2.7 int tdb_crypto::tc_skip

Definition at line 72 of file xform.h.

Referenced by ah_input(), ah_input_cb(), ah_output(), ah_output_cb(), esp_input(), esp_input_cb(), ipcomp_input_cb(), ipcomp_output(), and ipcomp_output_cb().

6.35.2.8 u_int32_t tdb_crypto::tc_spi

Definition at line 67 of file xform.h.

Referenced by ah_input(), ah_input_cb(), ah_output(), ah_output_cb(), esp_input(), esp_input_cb(), esp_output(), esp_output_cb(), ipcomp_input_cb(), ipcomp_output(), and ipcomp_output_cb().

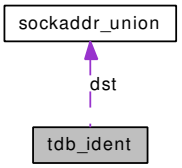
The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/xform.h

6.36 tdb_ident Struct Reference

`#include <xform.h>`

Collaboration diagram for tdb_ident:



Data Fields

- u_int32_t spi
- sockaddr_union dst
- u_int8_t proto

6.36.1 Detailed Description

Definition at line 56 of file xform.h.

6.36.2 Field Documentation

6.36.2.1 union sockaddr_union tdb_ident::dst

Definition at line 58 of file xform.h.

Referenced by ah_input(), esp_input(), ipsec_getpolicy(), and ipsec_process_done().

6.36.2.2 u_int8_t tdb_ident::proto

Definition at line 59 of file xform.h.

Referenced by ah_input(), esp_input(), ipsec_getpolicy(), and ipsec_process_done().

6.36.2.3 u_int32_t tdb_ident::spi

Definition at line 57 of file xform.h.

Referenced by ah_input(), esp_input(), ipsec_getpolicy(), and ipsec_process_done().

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/xform.h

6.37 xformsw Struct Reference

`#include <xform.h>`

Collaboration diagram for xformsw:

Data Fields

- u_short xf_type
- u_short xf_flags
- char * xf_name
- int(* xf_init)(struct secasvar *, struct xformsw *)
- int(* xf_zeroize)(struct secasvar *)
- int(* xf_input)(struct mbuf *, struct secasvar *, int, int)
- int(* xf_output)(struct mbuf *, struct ipsecrequest *, struct mbuf **, int, int)
- xformsw * xf_next

6.37.1 Detailed Description

Definition at line 79 of file xform.h.

6.37.2 Field Documentation

6.37.2.1 u_short xformsw::xf_flags

Definition at line 86 of file xform.h.

6.37.2.2 int(* xformsw::xf_init)(struct secasvar *, struct xformsw *)

Referenced by xform_init().

6.37.2.3 int(* xformsw::xf_input)(struct mbuf *, struct secasvar *,int, int)

Referenced by ipsec_common_input().

6.37.2.4 char* xformsw::xf_name

Definition at line 90 of file xform.h.

6.37.2.5 struct xformsw* xformsw::xf_next

Definition at line 97 of file xform.h.

Referenced by xform_init().

6.37.2.6 int(∗ xformsw::xf_output)(struct mbuf ∗,struct ipsecrequest ∗, struct mbuf ∗∗, int, int)

6.37.2.7 u_short xformsw::xf_type

Definition at line 80 of file xform.h.

Referenced by ipip_output(), and xform_init().

6.37.2.8 int(∗ xformsw::xf_zeroize)(struct secasvar ∗)

Referenced by key_cleansav().

The documentation for this struct was generated from the following file:

- /usr/src/sys/netipsec/xform.h

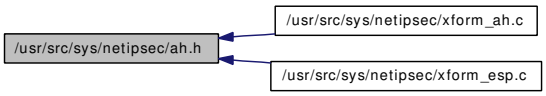
Chapter 7

FreeBSD kernel IPsec code File Documentation

7.1 notreviewed.dox File Reference

7.2 /usr/src/sys/netipsec/ah.h File Reference

This graph shows which files directly or indirectly include this file:

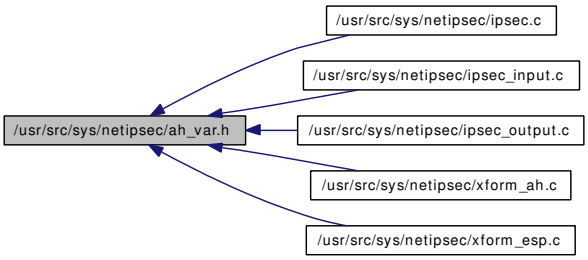


Data Structures

- struct ah
- struct newah

7.3 /usr/src/sys/netipsec/ah_var.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct ahstat

Defines

- #define AH_ALG_MAX 16

Variables

- int ah_enable
- int ah_cleartos
- ahstat ahstat

7.3.1 Define Documentation

7.3.1.1 #define AH_ALG_MAX 16

Definition at line 48 of file ah_var.h.

Referenced by ah_algorithm_lookup().

7.3.2 Variable Documentation

7.3.2.1 int ah_cleartos

Definition at line 91 of file xform_ah.c.

Referenced by ah_massage_headers().

7.3.2.2 int ah_enable

Definition at line 90 of file xform_ah.c.

Referenced by ipsec_common_input(), and ipsec_nextisr().

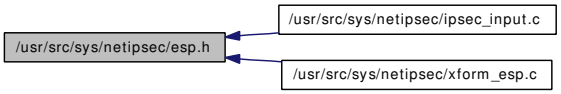
7.3.2.3 struct ahstat ahstat

Definition at line 92 of file xform_ah.c.

Referenced by ah_input(), ah_input_cb(), ah_output(), and ah_output_cb().

7.4 /usr/src/sys/netipsec/esp.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct esp
- struct newesp
- struct esptail

Defines

- #define ESP_ALEN 12

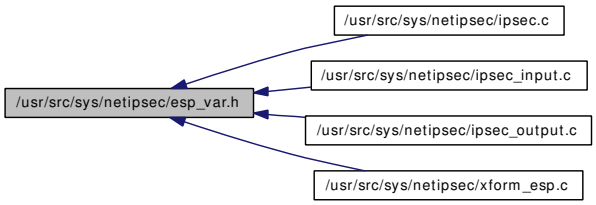
7.4.1 Define Documentation

7.4.1.1 #define ESP_ALEN 12

Definition at line 68 of file esp.h.

7.5 /usr/src/sys/netipsec/esp_var.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct espstat

Defines

- #define ESP_ALG_MAX 256

Variables

- int esp_enable
- espstat espstat

7.5.1 Define Documentation

7.5.1.1 #define ESP_ALG_MAX 256

Definition at line 48 of file esp_var.h.

Referenced by esp_algorithm_lookup().

7.5.2 Variable Documentation

7.5.2.1 int esp_enable

Definition at line 78 of file xform_esp.c.

Referenced by ipsec_common_input(), and ipsec_nextisr().

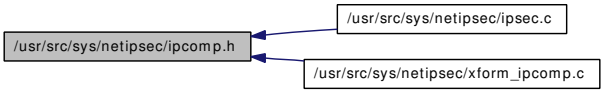
7.5.2.2 struct espstat espstat

Definition at line 79 of file xform_esp.c.

Referenced by esp_input(), esp_input_cb(), esp_output(), and esp_output_cb().

7.6 /usr/src/sys/netipsec/ipcomp.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct ipcomp

Defines

- #define IPCOMP_HLENGTH 4
- #define IPCOMP_OUI 1
- #define IPCOMP_DEFLATE 2
- #define IPCOMP_LZS 3
- #define IPCOMP_MAX 4
- #define IPCOMP_CPI_NEGOTIATE_MIN 256

7.6.1 Define Documentation

7.6.1.1 #define IPCOMP_CPI_NEGOTIATE_MIN 256

Definition at line 54 of file ipcomp.h.

7.6.1.2 #define IPCOMP_DEFLATE 2

Definition at line 50 of file ipcomp.h.

7.6.1.3 #define IPCOMP_HLENGTH 4

Definition at line 46 of file ipcomp.h.

Referenced by ipcomp_input(), ipcomp_input_cb(), and ipcomp_output().

7.6.1.4 #define IPCOMP_LZS 3

Definition at line 51 of file ipcomp.h.

7.6.1.5 #define IPCOMP_MAX 4

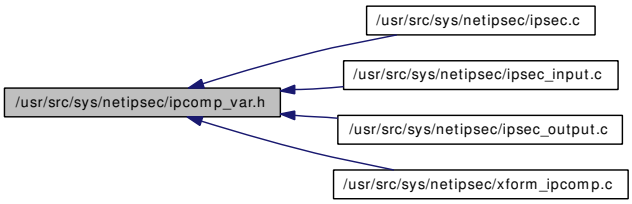
Definition at line 52 of file ipcomp.h.

7.6.1.6 #define IPCOMP_OUI 1

Definition at line 49 of file ipcomp.h.

7.7 /usr/src/sys/netipsec/ipcomp_var.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct ipcompstat

Defines

- #define IPCOMP_ALG_MAX 8

Variables

- int ipcomp_enable
- ipcompstat ipcompstat

7.7.1 Define Documentation

7.7.1.1 #define IPCOMP_ALG_MAX 8

Definition at line 42 of file ipcomp_var.h.

Referenced by ipcomp_algorithm_lookup().

7.7.2 Variable Documentation

7.7.2.1 int ipcomp_enable

Definition at line 69 of file xform_ipcomp.c.

Referenced by ipsec_common_input(), and ipsec_nextisr().

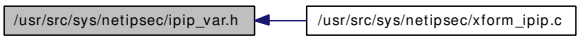
7.7.2.2 struct ipcompstat ipcompstat

Definition at line 70 of file xform_ipcomp.c.

Referenced by ipcomp_input(), ipcomp_input_cb(), ipcomp_output(), and ipcomp_output_cb().

7.8 /usr/src/sys/netipsec/ipip_var.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct ipipstat

Variables

- int ipip_allow
- ipipstat ipipstat

7.8.1 Variable Documentation

7.8.1.1 int ipip_allow

Definition at line 92 of file xform_ipip.c.

Referenced by _ipip_input().

7.8.1.2 struct ipipstat ipipstat

Definition at line 93 of file xform_ipip.c.

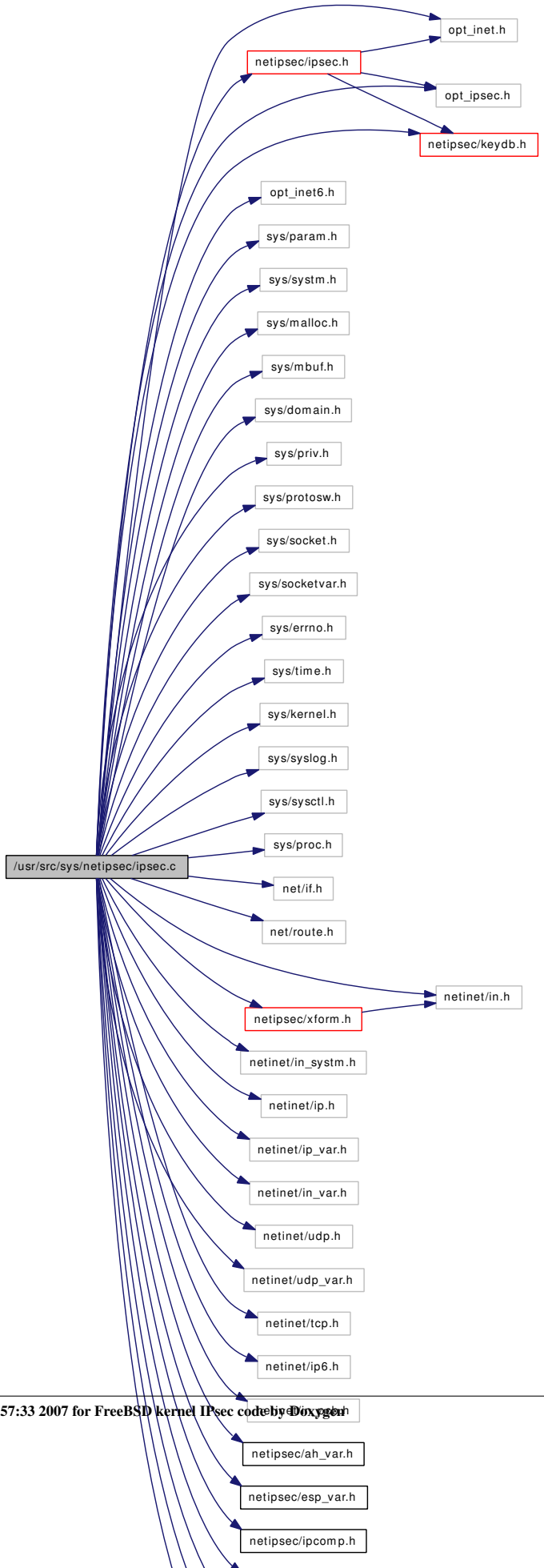
Referenced by _ipip_input(), and ipip_output().

7.9 /usr/src/sys/netipsec/ipsec.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include "opt_ipsec.h"
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/domain.h>
#include <sys/priv.h>
#include <sys/protosw.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/errno.h>
#include <sys/time.h>
#include <sys/kernel.h>
#include <sys/syslog.h>
#include <sys/sysctl.h>
#include <sys/proc.h>
#include <net/if.h>
#include <net/route.h>
#include <netinet/in.h>
#include <netinet/in_systm.h>
#include <netinet/ip.h>
#include <netinet/ip_var.h>
#include <netinet/in_var.h>
#include <netinet/udp.h>
#include <netinet/udp_var.h>
#include <netinet/tcp.h>
#include <netinet/ip6.h>
#include <netinet/in_pcb.h>
#include <netipsec/ipsec.h>
#include <netipsec/ah_var.h>
#include <netipsec/esp_var.h>
#include <netipsec/ipcomp.h>
#include <netipsec/ipcomp_var.h>
```

```
#include <netipsec/key.h>
#include <netipsec/keydb.h>
#include <netipsec/key_debug.h>
#include <netipsec/xform.h>
#include <machine/in_cksum.h>
```

Include dependency graph for ipsec.c:


opt_inet.h
netipsec/ipsec.h
opt_ipsec.h
netipsec/keydb.h
opt_inet6.h
sys/param.h
sys/systm.h
sys/malloc.h
sys/mbuf.h
sys/domain.h
sys/priv.h
sys/protosw.h
sys/socket.h
sys/socketvar.h
sys/errno.h
sys/time.h
sys/kernel.h
sys/syslog.h
sys/sysctl.h
sys/proc.h
/usr/src/sys/netipsec/ipsec.c
net/if.h
net/route.h
netinet/in.h
netipsec/xform.h
netinet/in_systm.h
netinet/ip.h
netinet/ip_var.h
netinet/in_var.h
netinet/udp.h
netinet/udp_var.h
netinet/tcp.h
netinet/ip6.h
netipsec/ah_var.h
netipsec/esp_var.h
netipsec/ipcomp.h

Defines

- #define KEY_ALLOCSP_DEFAULT() key_allocsp_default(__FILE__, __LINE__)
- #define IPSEC_CHECK_DEFAULT(lev)

Functions

- SYSCTL_DECL (_net_inet_ipsec)
- SYSCTL_INT (_net_inet_ipsec, IPSECCTL_DEF_POLICY, def_policy, CTLFLAG_RW,&ip4_-def_policy.policy, 0,"")
- SYSCTL_INT (_net_inet_ipsec, IPSECCTL_DEF_ESP_TRANSLEV, esp_trans_deflev, CTLFLAG_RW,&ip4_esp_trans_deflev, 0,"")
- SYSCTL_INT (_net_inet_ipsec, IPSECCTL_DEF_ESP_NETLEV, esp_net_deflev, CTLFLAG_-RW,&ip4_esp_net_deflev, 0,"")
- SYSCTL_INT (_net_inet_ipsec, IPSECCTL_DEF_AH_TRANSLEV, ah_trans_deflev, CTLFLAG_RW,&ip4_ah_trans_deflev, 0,"")
- SYSCTL_INT (_net_inet_ipsec, IPSECCTL_DEF_AH_NETLEV, ah_net_deflev, CTLFLAG_-RW,&ip4_ah_net_deflev, 0,"")
- SYSCTL_INT (_net_inet_ipsec, IPSECCTL_AH_CLEARTOS, ah_cleartos, CTLFLAG_-RW,&ah_cleartos, 0,"")
- SYSCTL_INT (_net_inet_ipsec, IPSECCTL_AH_OFFSETMASK, ah_offsetmask, CTLFLAG_-RW,&ip4_ah_offsetmask, 0,"")
- SYSCTL_INT (_net_inet_ipsec, IPSECCTL_DFBIT, dfbit, CTLFLAG_RW,&ip4_ipsec_dfbit, 0,"")
- SYSCTL_INT (_net_inet_ipsec, IPSECCTL_ECN, ecn, CTLFLAG_RW,&ip4_ipsec_ecn, 0,"")
- SYSCTL_INT (_net_inet_ipsec, IPSECCTL_DEBUG, debug, CTLFLAG_RW,&ipsec_debug, 0,"")
- SYSCTL_INT (_net_inet_ipsec, IPSECCTL_ESP_RANDPAD, esp_randpad, CTLFLAG_-RW,&ip4_esp_randpad, 0,"")
- SYSCTL_INT (_net_inet_ipsec, OID_AUTO, crypto_support, CTLFLAG_RW,&crypto_support, 0,"")
- SYSCTL_STRUCT (_net_inet_ipsec, OID_AUTO, ipsecstats, CTLFLAG_RD,&newipsecstat, newipsecstat,"")
- static int ipsec4_setspidx_inpcb __P ((struct mbuf *, struct inpcb *pcb))
- static int ipsec_setspidx __P ((struct mbuf *, struct secpolicyindex *, int))
- static void ipsec4_get_ulp __P ((struct mbuf *m, struct secpolicyindex *, int))
- static int ipsec4_setspidx_ipaddr __P ((struct mbuf *, struct secpolicyindex *))
- static void ipsec_delpcbpolicy __P ((struct inpcbpolicy *))
- static struct secpolicy *ipsec_deepcopy_policy __P ((struct secpolicy *src))
- static int ipsec_set_policy __P ((struct secpolicy **pcb_sp, int optname, caddr_t request, size_t len, int priv))
- static int ipsec_get_policy __P ((struct secpolicy *pcb_sp, struct mbuf **mp))
- static void vshiftl __P ((unsigned char *, int, int))
- static size_t ipsec_hdrsiz __P ((struct secpolicy *))
- MALLOC_DEFINE (M_IPSEC_INPCB,"inpcbpolicy","inpcb-resident ipsec policy")
- static struct secpolicy * key_allocsp_default (const char *where, int tag)
- secpolicy * ipsec_getpolicy (struct tdb_ident *tdbi, u_int dir)
- secpolicy * ipsec_getpolicybysock (struct mbuf *m, u_int dir, struct inpcb *inp, int *error)
- secpolicy * ipsec_getpolicybyaddr (struct mbuf *m, u_int dir, int flag, int *error)
- secpolicy * ipsec4_checkpolicy (struct mbuf *m, u_int dir, u_int flag, int *error, struct inpcb *inp)
- static int ipsec4_setspidx_inpcb (struct mbuf *m, struct inpcb *pcb)
- static int ipsec_setspidx (struct mbuf *m, struct secpolicyindex *spidx, int needport)
- static void ipsec4_get_ulp (struct mbuf *m, struct secpolicyindex *spidx, int needport)

- static int ipsec4_setspidx_ipaddr (struct mbuf *m, struct secpolicyindex *spidx)
- static void ipsec_delpcbpolicy (struct inpcbpolicy *p)
- int ipsec_init_policy (struct socket *so, struct inpcbpolicy **pcb_sp)
- int ipsec_copy_policy (struct inpcbpolicy *old, struct inpcbpolicy *new)
- ipsecrequest * ipsec_newisr (void)
- void ipsec_delisr (struct ipsecrequest *p)
- static struct secpolicy * ipsec_deepcopy_policy (struct secpolicy *src)
- static int ipsec_set_policy (struct secpolicy **pcb_sp, int optname, caddr_t request, size_t len, int priv)
- static int ipsec_get_policy (struct secpolicy *pcb_sp, struct mbuf **mp)
- int ipsec4_set_policy (struct inpcb *inp, int optname, caddr_t request, size_t len, int priv)
- int ipsec4_get_policy (struct inpcb *inp, caddr_t request, size_t len, struct mbuf **mp)
- int ipsec4_delete_pcbpolicy (struct inpcb *inp)
- u_int ipsec_get_reqlevel (struct ipsecrequest *isr)
- int ipsec_in_reject (struct secpolicy *sp, struct mbuf *m)
- int ipsec4_in_reject (struct mbuf *m, struct inpcb *inp)
- static size_t ipsec_hdrsiz (struct secpolicy *sp)
- size_t ipsec4_hdrsiz (struct mbuf *m, u_int dir, struct inpcb *inp)
- int ipsec_chkreplay (u_int32_t seq, struct secasvar *sav)
- int ipsec_updatereplay (u_int32_t seq, struct secasvar *sav)
- static void vshiftl (unsigned char *bitmap, int nbit, int wsize)
- static char * inet_ntoa4 (struct in_addr ina)
- char * ipsec_address (union sockaddr_union *sa)
- const char * ipsec_logsastr (struct secasvar *sav)
- void ipsec_dumpmbuf (struct mbuf *m)
- static void ipsec_attach (void)
- void xform_register (struct xformsw *xsp)
- int xform_init (struct secasvar *sav, int xftype)

Variables

- int ipsec_debug = 0
- newipsecstat newipsecstat
- int ip4_ah_offsetmask = 0
- int ip4_ipsec_dfbit = 0
- int ip4_esp_trans_deflev = IPSEC_LEVEL_USE
- int ip4_esp_net_deflev = IPSEC_LEVEL_USE
- int ip4_ah_trans_deflev = IPSEC_LEVEL_USE
- int ip4_ah_net_deflev = IPSEC_LEVEL_USE
- secpolicy ip4_def_policy
- int ip4_ipsec_ecn = 0
- int ip4_esp_randpad = -1
- int crypto_support = 0
- static struct xformsw * xforms = NULL

7.9.1 Define Documentation

7.9.1.1 #define IPSEC_CHECK_DEFAULT(lev)

Value:

```
(((lev) != IPSEC_LEVEL_USE && (lev) != IPSEC_LEVEL_REQUIRE            \
                        && (lev) != IPSEC_LEVEL_UNIQUE)                         \
                ? (ipsec_debug                                                  \
                        ? log(LOG_INFO, "fixed system default level " #lev ":%d->%d\n",\
                                (lev), IPSEC_LEVEL_REQUIRE)                     \
                        : 0),                                                   \
                        (lev) = IPSEC_LEVEL_REQUIRE,                            \
                        (lev)                                                   \
                : (lev))
```

Referenced by ipsec_get_reqlevel().

7.9.1.2 #define KEY_ALLOCSP_DEFAULT() key_allocsp_default(__FILE__, __LINE__)

Definition at line 248 of file ipsec.c.

Referenced by ipsec_getpolicy(), ipsec_getpolicybyaddr(), and ipsec_getpolicybysock().

7.9.2 Function Documentation

7.9.2.1 static size_t ipsec_hdrsiz __P ((struct secpolicy *)) [static]

7.9.2.2 static void vshiftl __P ((unsigned char *, int, int)) [static]

7.9.2.3 static int ipsec_get_policy __P ((struct secpolicy *pcb_sp, struct mbuf **mp)) [static]

7.9.2.4 static int ipsec_set_policy __P ((struct secpolicy **pcb_sp, int optname, caddr_t request, size_t len, int priv)) [static]

7.9.2.5 static struct secpolicy* ipsec_deepcopy_policy __P ((struct secpolicy *src)) [static]

7.9.2.6 static void ipsec_delpcbpolicy __P ((struct inpcbpolicy *)) [static]

7.9.2.7 static int ipsec4_setspidx_ipaddr __P ((struct mbuf *, struct secpolicyindex *)) [static]

7.9.2.8 static void ipsec4_get_ulp __P ((struct mbuf *m, struct secpolicyindex *, int)) [static]

7.9.2.9 static int ipsec_setspidx __P ((struct mbuf *, struct secpolicyindex *, int)) [static]

7.9.2.10 static int ipsec4_setspidx_inpcb __P ((struct mbuf *, struct inpcb *pcb)) [static]

7.9.2.11 static char* inet_ntoa4 (struct in_addr *ina*) [static]

Definition at line 1829 of file ipsec.c.

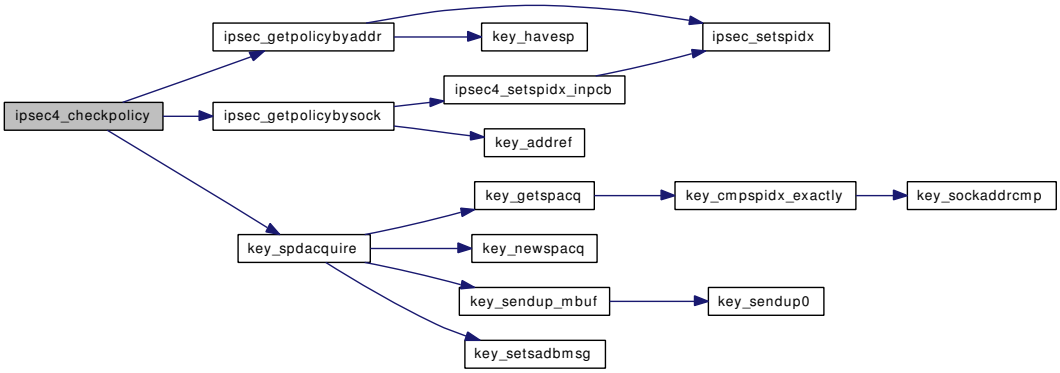
Referenced by ipsec_address().

7.9.2.12 struct secpolicy* ipsec4_checkpolicy (struct mbuf * *m*, u_int *dir*, u_int *flag*, int * *error*, struct inpcb * *inp*)

Definition at line 437 of file ipsec.c.

References newipsecstat::ips_out_inval, newipsecstat::ips_out_polvio, IPSEC_ASSERT, ipsec_getpolicybyaddr(), ipsec_getpolicybysock(), IPSEC_POLICY_BYPASS, IPSEC_POLICY_DISCARD, IPSEC_POLICY_ENTRUST, IPSEC_POLICY_IPSEC, IPSEC_POLICY_NONE, KEY_FREESP, key_spdacquire(), newipsecstat, secpolicy::policy, and secpolicy::req.

Here is the call graph for this function:



7.9.2.13 int ipsec4_delete_pcbpolicy (struct inpcb * *inp*)

Definition at line 1135 of file ipsec.c.

References IPSEC_ASSERT, ipsec_delpcbpolicy(), and KEY_FREESP.

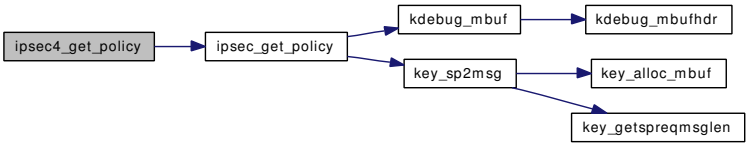
Here is the call graph for this function:

7.9.2.14 int ipsec4_get_policy (struct inpcb * *inp*, caddr_t *request*, size_t *len*, struct mbuf ** *mp*)

Definition at line 1099 of file ipsec.c.

References IPSEC_ASSERT, IPSEC_DIR_INBOUND, IPSEC_DIR_OUTBOUND, ipsec_get_policy(), and ipseclog.

Here is the call graph for this function:



7.9.2.15 static void ipsec4_get_ulp (struct mbuf * *m*, struct secpolicyindex * *spidx*, int *needport*) [static]

Definition at line 629 of file ipsec.c.

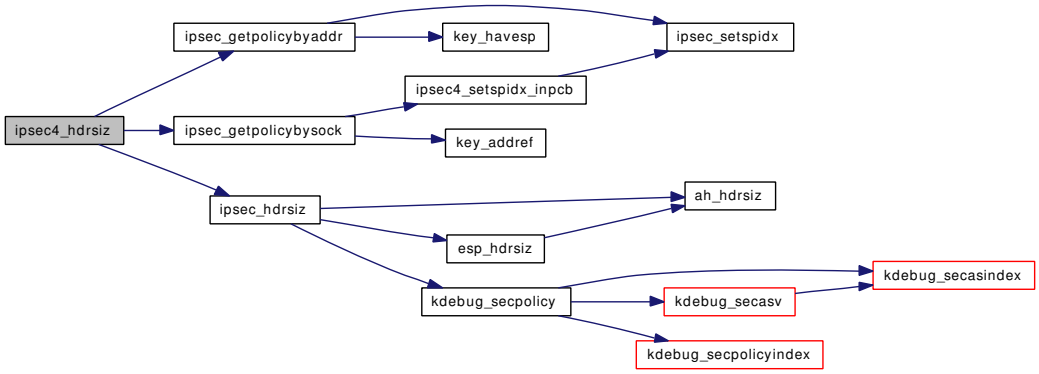
References secpolicyindex::dst, IPSEC_ASSERT, IPSEC_PORT_ANY, IPSEC_ULPROTO_ANY, sockaddr_union::sin, secpolicyindex::src, and secpolicyindex::ul_proto.

7.9.2.16 size_t ipsec4_hdrsiz (struct mbuf * *m*, u_int *dir*, struct inpcb * *inp*)

Definition at line 1567 of file ipsec.c.

References IPSEC_ASSERT, ipsec_getpolicybyaddr(), ipsec_getpolicybysock(), ipsec_hdrsiz(), KEY_FREESP, KEYDEBUG, and KEYDEBUG_IPSEC_DATA.

Here is the call graph for this function:

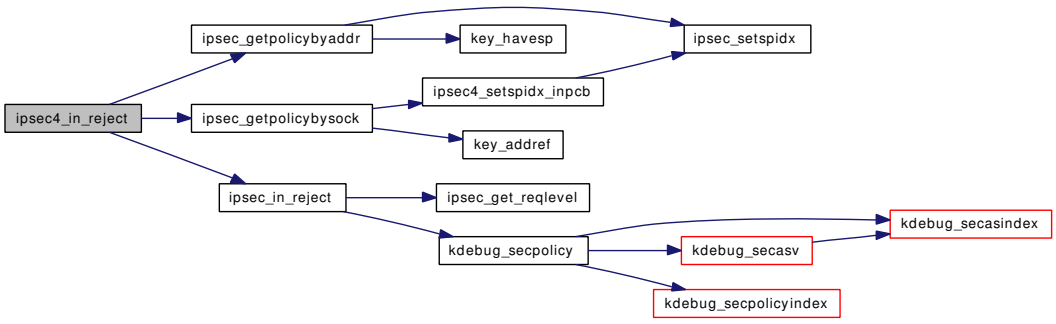


7.9.2.17 int ipsec4_in_reject (struct mbuf * *m*, struct inpcb * *inp*)

Definition at line 1431 of file ipsec.c.

References newipsecstat::ips_in_polvio, IPSEC_ASSERT, IPSEC_DIR_INBOUND, ipsec_getpolicybyaddr(), ipsec_getpolicybysock(), ipsec_in_reject(), KEY_FREESP, and newipsecstat.

Here is the call graph for this function:

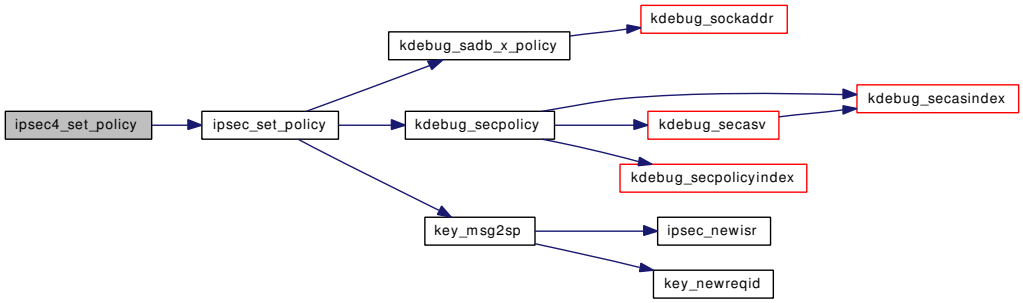


7.9.2.18 int ipsec4_set_policy (struct inpcb * *inp*, int *optname*, caddr_t *request*, size_t *len*, int *priv*)

Definition at line 1064 of file ipsec.c.

References IPSEC_DIR_INBOUND, IPSEC_DIR_OUTBOUND, ipsec_set_policy(), and ipseclog.

Here is the call graph for this function:



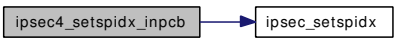
7.9.2.19 static int ipsec4_setspidx_inpcb (struct mbuf * *m*, struct inpcb * *pcb*) [static]

Definition at line 483 of file ipsec.c.

References IPSEC_ASSERT, IPSEC_DIR_INBOUND, IPSEC_DIR_OUTBOUND, and ipsec_setspidx().

Referenced by ipsec_getpolicybysock().

Here is the call graph for this function:



7.9.2.20 static int ipsec4_setspidx_ipaddr (struct mbuf * *m*, struct secpolicyindex * *spidx*) [static]

Definition at line 713 of file ipsec.c.

References secpolicyindex::dst, secpolicyindex::prefd, secpolicyindex::prefs, sockaddr_union::sin, and secpolicyindex::src.

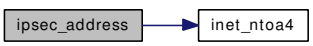
7.9.2.21 char* ipsec_address (union sockaddr_union * *sa*)

Definition at line 1844 of file ipsec.c.

References inet_ntoa4(), sockaddr_union::sa, sockaddr_union::sin, and sockaddr_union::sin6.

Referenced by ah_input(), ah_input_cb(), ah_output(), esp_input(), esp_input_cb(), esp_output(), esp_-output_cb(), ipcomp_input_cb(), ipcomp_output(), ipcomp_output_cb(), ipip_output(), ipsec_common_-input(), and ipsec_logsastr().

Here is the call graph for this function:



7.9.2.22 static void ipsec_attach (void) [static]

Definition at line 1917 of file ipsec.c.

References ip4_def_policy, secpolicy::refcnt, and SECPOLICY_LOCK_INIT.

7.9.2.23 int ipsec_chkreplay (u_int32_t *seq*, struct secasvar * *sav*)

Definition at line 1647 of file ipsec.c.

References secreplay::bitmap, secreplay::count, IPSEC_ASSERT, IPSEC_SPLASSERT_SOFTNET, secreplay::lastseq, secasvar::replay, and secreplay::wsize.

Referenced by ah_input(), and esp_input().

7.9.2.24 int ipsec_copy_policy (struct inpcbpolicy * *old*, struct inpcbpolicy * *new*)

Definition at line 893 of file ipsec.c.

References ipsec_deepcopy_policy(), and KEY_FREESP.

Here is the call graph for this function:

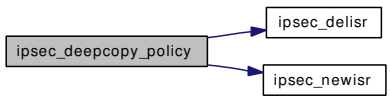
7.9.2.25 static struct secpolicy* ipsec_deepcopy_policy (struct secpolicy * *src*) [static]

Definition at line 937 of file ipsec.c.

References secasindex::dst, ipsec_delisr(), ipsec_newisr(), KEY_NEWSP, ipsecrequest::level, secasindex::mode, ipsecrequest::next, secpolicy::policy, secasindex::proto, secpolicy::req, secasindex::reqid, ipsecrequest::saidx, secasindex::src, and secpolicy::state.

Referenced by ipsec_copy_policy().

Here is the call graph for this function:



7.9.2.26 void ipsec_delisr (struct ipsecrequest * *p*)

Definition at line 929 of file ipsec.c.

References IPSECREQUEST_LOCK_DESTROY.

Referenced by ipsec_deepcopy_policy(), and key_delsp().

7.9.2.27 static void ipsec_delpcbpolicy (struct inpcbpolicy * *p*) [static]

Definition at line 844 of file ipsec.c.

Referenced by ipsec4_delete_pcbpolicy(), and ipsec_init_policy().

7.9.2.28 void ipsec_dumpmbuf (struct mbuf * *m*)

Definition at line 1892 of file ipsec.c.

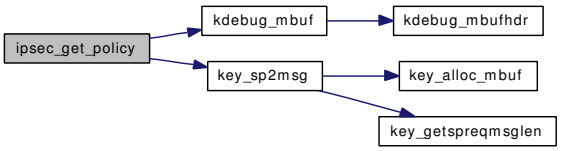
7.9.2.29 static int ipsec_get_policy (struct secpolicy * *pcb_sp*, struct mbuf ** *mp*) [static]

Definition at line 1041 of file ipsec.c.

References ipseclog, kdebug_mbuf(), key_sp2msg(), and KEYDEBUG.

Referenced by ipsec4_get_policy().

Here is the call graph for this function:



7.9.2.30 u_int ipsec_get_reqlevel (struct ipsecrequest * *isr*)

Definition at line 1253 of file ipsec.c.

References ip4_ah_net_deflev, ip4_ah_trans_deflev, ip4_esp_net_deflev, ip4_esp_trans_deflev, ip6_ah_net_deflev, ip6_ah_trans_deflev, ip6_esp_net_deflev, ip6_esp_trans_deflev, IPSEC_ASSERT, IPSEC_CHECK_DEFAULT, IPSEC_LEVEL_DEFAULT, IPSEC_LEVEL_REQUIRE, IPSEC_LEVEL_UNIQUE, IPSEC_LEVEL_USE, and IPSEC_MODE_TUNNEL.

Referenced by ipsec_in_reject(), ipsec_nextisr(), and key_checkrequest().

7.9.2.31 struct secpolicy* ipsec_getpolicy (struct tdb_ident * *tdbi*, u_int *dir*)

Definition at line 264 of file ipsec.c.

References tdb_ident::dst, IPSEC_ASSERT, IPSEC_DIR_INBOUND, IPSEC_DIR_OUTBOUND, KEY_ALLOCSP2, KEY_ALLOCSP_DEFAULT, tdb_ident::proto, and tdb_ident::spi.

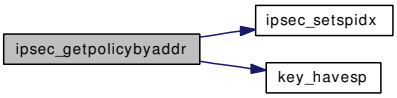
7.9.2.32 struct secpolicy* ipsec_getpolicybyaddr (struct mbuf * *m*, u_int *dir*, int *flag*, int * *error*)

Definition at line 402 of file ipsec.c.

References secpolicyindex::dir, DPRINTF, IPSEC_ASSERT, IPSEC_DIR_INBOUND, IPSEC_DIR_OUTBOUND, ipsec_setspidx(), KEY_ALLOCSP, KEY_ALLOCSP_DEFAULT, key_havesp(), and secpolicy::spidx.

Referenced by ipsec4_checkpolicy(), ipsec4_hdrsiz(), and ipsec4_in_reject().

Here is the call graph for this function:



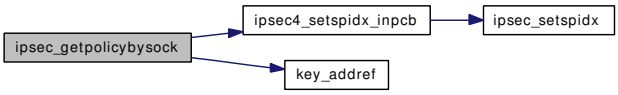
7.9.2.33 struct secpolicy* ipsec_getpolicybysock (struct mbuf * *m*, u_int *dir*, struct inpcb * *inp*, int * *error*)

Definition at line 292 of file ipsec.c.

References ipsec4_setspidx_inpcb(), IPSEC_ASSERT, IPSEC_DIR_INBOUND, IPSEC_DIR_-OUTBOUND, IPSEC_POLICY_BYPASS, IPSEC_POLICY_ENTRUST, IPSEC_POLICY_-IPSEC, ipseclog, key_addref(), KEY_ALLOCSP, KEY_ALLOCSP_DEFAULT, KEYDEBUG, KEYDEBUG_IPSEC_STAMP, secpolicy::policy, inpcbpolicy::priv, secpolicy::refcnt, inpcbpolicy::sp_in, inpcbpolicy::sp_out, and secpolicy::spidx.

Referenced by ipsec4_checkpolicy(), ipsec4_hdrsiz(), and ipsec4_in_reject().

Here is the call graph for this function:



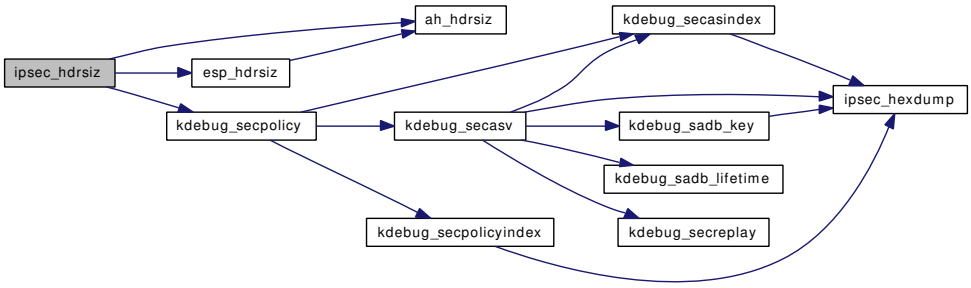
7.9.2.34 static size_t ipsec_hdrsiz (struct secpolicy * *sp*) [static]

Definition at line 1508 of file ipsec.c.

References ah_hdrsiz(), secasindex::dst, esp_hdrsiz(), IPSEC_ASSERT, IPSEC_MODE_TUNNEL, IPSEC_POLICY_BYPASS, IPSEC_POLICY_DISCARD, IPSEC_POLICY_IPSEC, IPSEC_POLICY_-NONE, ipseclog, kdebug_secpolicy(), KEYDEBUG, KEYDEBUG_IPSEC_DATA, secasindex::mode, ipsecrequest::next, secasindex::proto, sockaddr_union::sa, ipsecrequest::saidx, ipsecrequest::sav, and ipsecrequest::sp.

Referenced by ipsec4_hdrsiz().

Here is the call graph for this function:



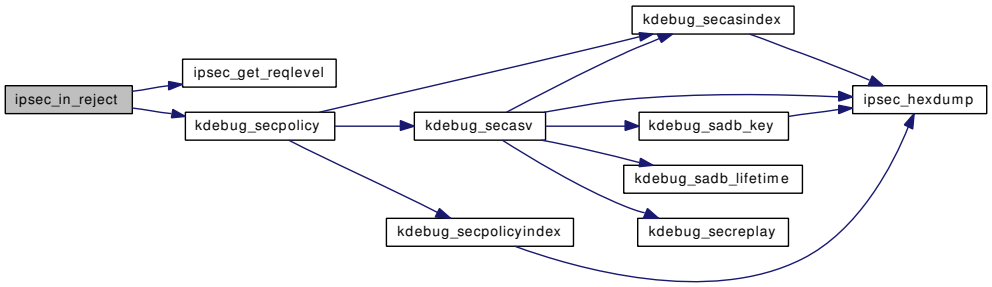
7.9.2.35 int ipsec_in_reject (struct secpolicy * *sp*, struct mbuf * *m*)

Definition at line 1358 of file ipsec.c.

References IPSEC_ASSERT, ipsec_get_reqlevel(), IPSEC_LEVEL_REQUIRE, IPSEC_POLICY_BYPASS, IPSEC_POLICY_DISCARD, IPSEC_POLICY_IPSEC, IPSEC_POLICY_NONE, kdebug_secpolicy(), KEYDEBUG, KEYDEBUG_IPSEC_DATA, ipsecrequest::next, secasindex::proto, ipsecrequest::saidx, ipsecrequest::sav, ipsecrequest::sp, and secasvar::tdb_authalgxform.

Referenced by ipsec4_in_reject().

Here is the call graph for this function:

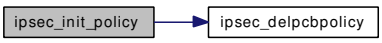


7.9.2.36 int ipsec_init_policy (struct socket * *so*, struct inpcbpolicy ** *pcb_sp*)

Definition at line 852 of file ipsec.c.

References ipsec_delpcbpolicy(), IPSEC_IS_PRIVILEGED_SO, IPSEC_POLICY_ENTRUST, IPSEC_SPSTATE_ALIVE, ipseclog, KEY_FREESP, and KEY_NEWSP.

Here is the call graph for this function:



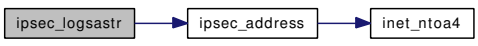
7.9.2.37 const char* ipsec_logsastr (struct secasvar * *sav*)

Definition at line 1866 of file ipsec.c.

References secasindex::dst, ipsec_address(), IPSEC_ASSERT, sockaddr_union::sa, and secasindex::src.

Referenced by ah_input(), esp_input(), esp_input_cb(), and ipsec_updatereplay().

Here is the call graph for this function:



7.9.2.38 struct ipsecrequest* ipsec_newisr (void)

Definition at line 918 of file ipsec.c.

References IPSECREQUEST_LOCK_INIT.

Referenced by ipsec_deepcopy_policy(), and key_msg2sp().

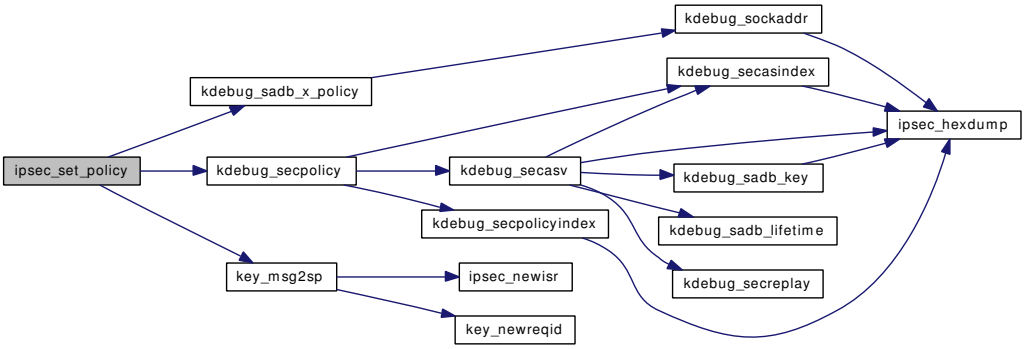
7.9.2.39 static int ipsec_set_policy (struct secpolicy ** *pcb_sp*, int *optname*, caddr_t *request*, size_t *len*, int *priv*) [static]

Definition at line 992 of file ipsec.c.

References IPSEC_POLICY_BYPASS, IPSEC_POLICY_DISCARD, IPSEC_POLICY_NONE, IPSEC_-SPSTATE_ALIVE, kdebug_sadb_x_policy(), kdebug_secpolicy(), KEY_FREESP, key_msg2sp(), KEY-DEBUG, and secpolicy::state.

Referenced by ipsec4_set_policy().

Here is the call graph for this function:



7.9.2.40 static int ipsec_setspidx (struct mbuf * *m*, struct secpolicyindex * *spidx*, int *needport*) [static]

Definition at line 552 of file ipsec.c.

References IPSEC_ASSERT, KEYDEBUG, and KEYDEBUG_IPSEC_DUMP.

Referenced by ipsec4_setspidx_inpcb(), and ipsec_getpolicybyaddr().

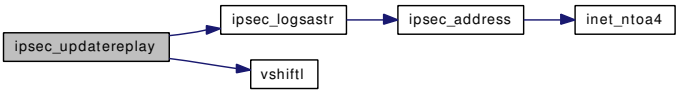
7.9.2.41 int ipsec_updatereplay (u_int32_t *seq*, struct secasvar * *sav*)

Definition at line 1707 of file ipsec.c.

References secreplay::bitmap, secreplay::count, secasvar::flags, IPSEC_ASSERT, ipsec_logsastr(), IPSEC_SPLASSERT_SOFTNET, ipseclog, secreplay::lastseq, secreplay::overflow, secasvar::replay, vshiftl(), and secreplay::wsize.

Referenced by ah_input_cb(), and esp_input_cb().

Here is the call graph for this function:

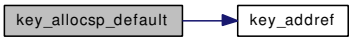


7.9.2.42 static struct secpolicy* key_allocsp_default (const char * *where*, int *tag*) [static]

Definition at line 227 of file ipsec.c.

References ip4_def_policy, IPSEC_POLICY_DISCARD, IPSEC_POLICY_NONE, ipseclog, key_addref(), KEYDEBUG, KEYDEBUG_IPSEC_STAMP, secpolicy::policy, and secpolicy::refcnt.

Here is the call graph for this function:



7.9.2.43 MALLOC_DEFINE (M_IPSEC_INPCB, "inpcbpolicy", "inpcb-resident ipsec policy")

7.9.2.44 SYSCTL_DECL (_net_inet_ipsec)

7.9.2.45 SYSCTL_INT (_net_inet_ipsec, OID_AUTO, crypto_support, CTLFLAG_RW, & *crypto_support*, 0, "")

7.9.2.46 SYSCTL_INT (_net_inet_ipsec, IPSECCTL_ESP_RANDPAD, esp_randpad, CTLFLAG_RW, & *ip4_esp_randpad*, 0, "")

7.9.2.47 SYSCTL_INT (_net_inet_ipsec, IPSECCTL_DEBUG, debug, CTLFLAG_RW, & *ipsec_debug*, 0, "")

7.9.2.48 SYSCTL_INT (_net_inet_ipsec, IPSECCTL_ECN, ecn, CTLFLAG_RW, & *ip4_ipsec_ecn*, 0, "")

7.9.2.49 SYSCTL_INT (_net_inet_ipsec, IPSECCTL_DFBIT, dfbit, CTLFLAG_RW, & *ip4_ipsec_dfbit*, 0, "")

7.9.2.50 SYSCTL_INT (_net_inet_ipsec, IPSECCTL_AH_OFFSETMASK, ah_offsetmask, CTLFLAG_RW, & *ip4_ah_offsetmask*, 0, "")

7.9.2.51 SYSCTL_INT (_net_inet_ipsec, IPSECCTL_AH_CLEARTOS, ah_cleartos, CTLFLAG_RW, & *ah_cleartos*, 0, "")

7.9.2.52 SYSCTL_INT (_net_inet_ipsec, IPSECCTL_DEF_AH_NETLEV, ah_net_deflev, CTLFLAG_RW, & *ip4_ah_net_deflev*, 0, "")

7.9.2.53 SYSCTL_INT (_net_inet_ipsec, IPSECCTL_DEF_AH_TRANSLEV, ah_trans_deflev, CTLFLAG_RW, & *ip4_ah_trans_deflev*, 0, "")

7.9.2.54 SYSCTL_INT (_net_inet_ipsec, IPSECCTL_DEF_ESP_NETLEV, esp_net_deflev, CTLFLAG_RW, & *ip4_esp_net_deflev*, 0, "")

7.9.2.55 SYSCTL_INT (_net_inet_ipsec, IPSECCTL_DEF_ESP_TRANSLEV, esp_trans_deflev, CTLFLAG_RW, & *ip4_esp_trans_deflev*, 0, "")

7.9.2.56 SYSCTL_INT (_net_inet_ipsec, IPSECCTL_DEF_POLICY, def_policy, CTLFLAG_RW, &ip4_def_policy. *policy*, 0, "")

7.9.2.57 SYSCTL_STRUCT (_net_inet_ipsec, OID_AUTO, ipsecstats, CTLFLAG_RD, & *newipsecstat*, newipsecstat, "")

7.9.2.58 static void vshiftl (unsigned char * *bitmap*, int *nbit*, int *wsize*) [static]

Definition at line 1807 of file ipsec.c.

Referenced by ipsec_updatereplay().

7.9.2.59 int xform_init (struct secasvar * *sav*, int *xftype*)

Definition at line 1943 of file ipsec.c.

References secasvar::tdb_xform, xformsw::xf_init, xformsw::xf_next, xformsw::xf_type, and xforms.

Referenced by key_mature(), and key_setsaval().

7.9.2.60 void xform_register (struct xformsw * *xsp*)

Definition at line 1933 of file ipsec.c.

References xforms.

Referenced by ah_attach(), esp_attach(), ipcomp_attach(), and tcpsignature_attach().

7.9.3 Variable Documentation

7.9.3.1 int crypto_support = 0

Definition at line 120 of file ipsec.c.

Referenced by ah_init(), esp_init(), and ipcomp_init().

7.9.3.2 int ip4_ah_net_deflev = IPSEC_LEVEL_USE

Definition at line 109 of file ipsec.c.

Referenced by ipsec_get_reqlevel().

7.9.3.3 int ip4_ah_offsetmask = 0

Definition at line 104 of file ipsec.c.

7.9.3.4 int ip4_ah_trans_deflev = IPSEC_LEVEL_USE

Definition at line 108 of file ipsec.c.

Referenced by ipsec_get_reqlevel().

7.9.3.5 struct secpolicy ip4_def_policy

Definition at line 110 of file ipsec.c.

Referenced by ipsec_attach(), key_allocsp_default(), and key_init().

7.9.3.6 int ip4_esp_net_deflev = IPSEC_LEVEL_USE

Definition at line 107 of file ipsec.c.

Referenced by ipsec_get_reqlevel().

7.9.3.7 int ip4_esp_randpad = -1

Definition at line 112 of file ipsec.c.

7.9.3.8 int ip4_esp_trans_deflev = IPSEC_LEVEL_USE

Definition at line 106 of file ipsec.c.

Referenced by ipsec_get_reqlevel().

7.9.3.9 int ip4_ipsec_dfbit = 0

Definition at line 105 of file ipsec.c.

7.9.3.10 int ip4_ipsec_ecn = 0

Definition at line 111 of file ipsec.c.

Referenced by _ipip_input().

7.9.3.11 int ipsec_debug = 0

Definition at line 99 of file ipsec.c.

7.9.3.12 struct newipsecstat newipsecstat

Definition at line 103 of file ipsec.c.

Referenced by ipsec4_checkpolicy(), and ipsec4_in_reject().

7.9.3.13 struct xformsw* xforms = NULL [static]

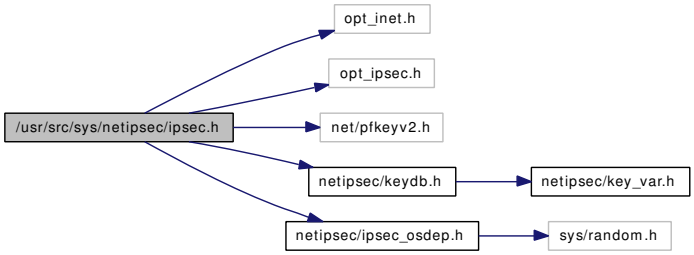
Definition at line 1927 of file ipsec.c.

Referenced by xform_init(), and xform_register().

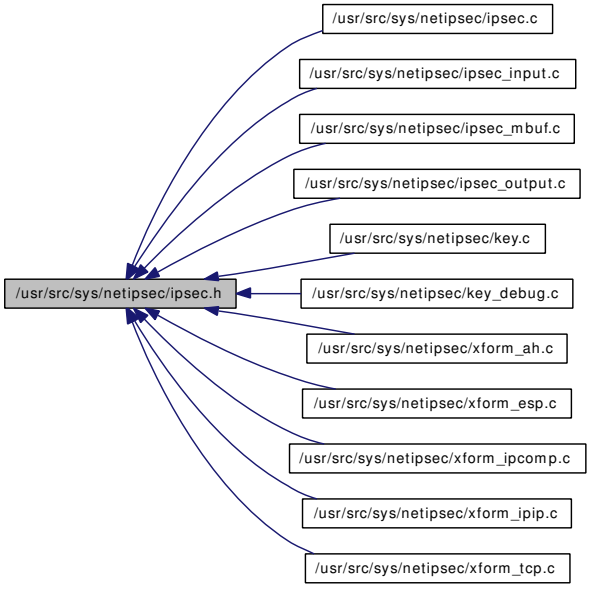
7.10 /usr/src/sys/netipsec/ipsec.h File Reference

```
#include "opt_inet.h"
#include "opt_ipsec.h"
#include <net/pfkeyv2.h>
#include <netipsec/keydb.h>
#include <netipsec/ipsec_osdep.h>
```

Include dependency graph for ipsec.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct secpolicyindex
- struct secpolicy
- struct ipsecrequest
- struct inpcbpolicy
- struct secspacq
- struct ipsecstat

- struct newipsecstat
- struct ipsec_output_state
- struct ipsec_history

Defines

- #define IPSEC_SPSTATE_DEAD 0
- #define IPSEC_SPSTATE_ALIVE 1
- #define SECPOLICY_LOCK_INIT(_sp) mtx_init(&(_sp) → lock, "ipsec policy", NULL, MTX_-DEF)
- #define SECPOLICY_LOCK(_sp) mtx_lock(&(_sp) → lock)
- #define SECPOLICY_UNLOCK(_sp) mtx_unlock(&(_sp) → lock)
- #define SECPOLICY_LOCK_DESTROY(_sp) mtx_destroy(&(_sp) → lock)
- #define SECPOLICY_LOCK_ASSERT(_sp) mtx_assert(&(_sp) → lock, MA_OWNED)
- #define IPSECREQUEST_LOCK_INIT(_isr) mtx_init(&(_isr) → lock, "ipsec request", NULL, MTX_DEF | MTX_RECURSE)
- #define IPSECREQUEST_LOCK(_isr) mtx_lock(&(_isr) → lock)
- #define IPSECREQUEST_UNLOCK(_isr) mtx_unlock(&(_isr) → lock)
- #define IPSECREQUEST_LOCK_DESTROY(_isr) mtx_destroy(&(_isr) → lock)
- #define IPSECREQUEST_LOCK_ASSERT(_isr) mtx_assert(&(_isr) → lock, MA_OWNED)
- #define IPSEC_PORT_ANY 0
- #define IPSEC_ULPROTO_ANY 255
- #define IPSEC_PROTO_ANY 255
- #define IPSEC_MODE_ANY 0
- #define IPSEC_MODE_TRANSPORT 1
- #define IPSEC_MODE_TUNNEL 2
- #define IPSEC_MODE_TCPMD5 3
- #define IPSEC_DIR_ANY 0
- #define IPSEC_DIR_INBOUND 1
- #define IPSEC_DIR_OUTBOUND 2
- #define IPSEC_DIR_MAX 3
- #define IPSEC_DIR_INVALID 4
- #define IPSEC_POLICY_DISCARD 0
- #define IPSEC_POLICY_NONE 1
- #define IPSEC_POLICY_IPSEC 2
- #define IPSEC_POLICY_ENTRUST 3
- #define IPSEC_POLICY_BYPASS 4
- #define IPSEC_LEVEL_DEFAULT 0
- #define IPSEC_LEVEL_USE 1
- #define IPSEC_LEVEL_REQUIRE 2
- #define IPSEC_LEVEL_UNIQUE 3
- #define IPSEC_MANUAL_REQID_MAX 0x3fff
- #define IPSEC_REPLAYWSIZE 32
- #define IPSECCTL_STATS 1
- #define IPSECCTL_DEF_POLICY 2
- #define IPSECCTL_DEF_ESP_TRANSLEV 3
- #define IPSECCTL_DEF_ESP_NETLEV 4
- #define IPSECCTL_DEF_AH_TRANSLEV 5
- #define IPSECCTL_DEF_AH_NETLEV 6
- #define IPSECCTL_AH_CLEARTOS 8

- #define IPSECCTL_AH_OFFSETMASK 9
- #define IPSECCTL_DFBIT 10
- #define IPSECCTL_ECN 11
- #define IPSECCTL_DEBUG 12
- #define IPSECCTL_ESP_RANDPAD 13
- #define IPSECCTL_MAXID 14
- #define IPSECCTL_NAMES
- #define IPSEC6CTL_NAMES
- #define ipseclog(x) do { if (ipsec_debug) log x; } while (0)
- #define DPRINTF(x) do { if (ipsec_debug) printf x; } while (0)
- #define ipsec_pcbconn(_x)
- #define ipsec_pcbdisconn(_x)

Functions

- ipsecrequest * ipsec_newisr (void)
- void ipsec_delisr (struct ipsecrequest *)
- secpolicy *ipsec_getpolicy __P ((struct tdb_ident *, u_int))
- secpolicy *ipsec4_checkpolicy __P ((struct mbuf *, u_int, u_int, int *, struct inpcb *))
- secpolicy * ipsec_getpolicybysock (struct mbuf *, u_int, struct inpcb *, int *)
- secpolicy * ipsec_getpolicybyaddr (struct mbuf *, u_int, int, int *)
- int ipsec_init_policy __P ((struct socket *so, struct inpcbpolicy **))
- int ipsec_copy_policy __P ((struct inpcbpolicy *, struct inpcbpolicy *))
- u_int ipsec_get_reqlevel __P ((struct ipsecrequest *))
- int ipsec_in_reject __P ((struct secpolicy *, struct mbuf *))
- int ipsec4_set_policy __P ((struct inpcb *inp, int optname, caddr_t request, size_t len, int priv))
- int ipsec4_get_policy __P ((struct inpcb *inpcb, caddr_t request, size_t len, struct mbuf **mp))
- int ipsec4_delete_pcbpolicy __P ((struct inpcb *))
- int ipsec4_in_reject __P ((struct mbuf *, struct inpcb *))
- int ipsec_chkreplay __P ((u_int32_t, struct secasvar *))
- size_t ipsec4_hdrsiz __P ((struct mbuf *, u_int, struct inpcb *))
- size_t ipsec_hdrsiz_tcp __P ((struct tcpcb *))
- char * ipsec_address (union sockaddr_union *sa)
- const char *ipsec_logsastr __P ((struct secasvar *))
- void ipsec_dumpmbuf __P ((struct mbuf *))
- void ah4_input (struct mbuf *m, int off)
- void ah4_ctlinput (int cmd, struct sockaddr *sa, void *)
- void esp4_input (struct mbuf *m, int off)
- void esp4_ctlinput (int cmd, struct sockaddr *sa, void *)
- void ipcomp4_input (struct mbuf *m, int off)
- int ipsec4_common_input (struct mbuf *m,...)
- int ipsec4_common_input_cb (struct mbuf *m, struct secasvar *sav, int skip, int protoff, struct m_tag *mt)
- int ipsec4_process_packet __P ((struct mbuf *, struct ipsecrequest *, int, int))
- int ipsec_process_done __P ((struct mbuf *, struct ipsecrequest *))
- void m_checkalignment (const char *where, struct mbuf *m0, int off, int len)
- mbuf * m_makespace (struct mbuf *m0, int skip, int hlen, int *off)
- caddr_t m_pad (struct mbuf *m, int n)
- int m_striphdr (struct mbuf *m, int skip, int hlen)
- int ipsec_filter (struct mbuf **, int)
- void ipsec_bpf (struct mbuf *, struct secasvar *, int)

Variables

- int ipsec_debug
- newipsecstat newipsecstat
- secpolicy ip4_def_policy
- int ip4_esp_trans_deflev
- int ip4_esp_net_deflev
- int ip4_ah_trans_deflev
- int ip4_ah_net_deflev
- int ip4_ah_cleartos
- int ip4_ah_offsetmask
- int ip4_ipsec_dfbit
- int ip4_ipsec_ecn
- int ip4_esp_randpad
- int crypto_support

7.10.1 Define Documentation

7.10.1.1 #define DPRINTF(x) do { if (ipsec_debug) printf x; } while (0)

Definition at line 353 of file ipsec.h.

Referenced by _ipip_input(), ah_init0(), ah_input(), ah_input_cb(), ah_massage_headers(), ah_output(), ah_output_cb(), esp_init(), esp_input(), esp_input_cb(), esp_output(), esp_output_cb(), ipcomp_init(), ipcomp_input(), ipcomp_input_cb(), ipcomp_output(), ipcomp_output_cb(), ipip_output(), ipsec_common_input(), ipsec_getpolicybyaddr(), ipsec_nextisr(), ipsec_process_done(), key_getsizes_ah(), m_pad(), and tcpsignature_init().

7.10.1.2 #define IPSEC6CTL_NAMES

Value:

```
{ \
        { 0, 0 }, \
        { 0, 0 }, \
        { "def_policy", CTLTYPE_INT }, \
        { "esp_trans_deflev", CTLTYPE_INT }, \
        { "esp_net_deflev", CTLTYPE_INT }, \
        { "ah_trans_deflev", CTLTYPE_INT }, \
        { "ah_net_deflev", CTLTYPE_INT }, \
        { 0, 0 }, \
        { 0, 0 }, \
        { 0, 0 }, \
        { 0, 0 }, \
        { "ecn", CTLTYPE_INT }, \
        { "debug", CTLTYPE_INT }, \
        { "esp_randpad", CTLTYPE_INT }, \
}
```

Definition at line 303 of file ipsec.h.

7.10.1.3 #define IPSEC_DIR_ANY 0

Definition at line 171 of file ipsec.h.

7.10.1.4 #define IPSEC_DIR_INBOUND 1

Definition at line 172 of file ipsec.h.

Referenced by ipsec4_get_policy(), ipsec4_in_reject(), ipsec4_set_policy(), ipsec4_setspidx_inpcb(), ipsec_getpolicy(), ipsec_getpolicybyaddr(), ipsec_getpolicybysock(), key_allocsp(), key_allocsp2(), key_-getspbyid(), key_gettunnel(), key_havesp(), key_spdadd(), and key_spddelete().

7.10.1.5 #define IPSEC_DIR_INVALID 4

Definition at line 175 of file ipsec.h.

7.10.1.6 #define IPSEC_DIR_MAX 3

Definition at line 174 of file ipsec.h.

7.10.1.7 #define IPSEC_DIR_OUTBOUND 2

Definition at line 173 of file ipsec.h.

Referenced by ipsec4_get_policy(), ipsec4_set_policy(), ipsec4_setspidx_inpcb(), ipsec_getpolicy(), ipsec_getpolicybyaddr(), ipsec_getpolicybysock(), key_allocsp(), key_allocsp2(), key_getspbyid(), key_-havesp(), key_spdadd(), and key_spddelete().

7.10.1.8 #define IPSEC_LEVEL_DEFAULT 0

Definition at line 190 of file ipsec.h.

Referenced by ipsec_get_reqlevel(), and key_msg2sp().

7.10.1.9 #define IPSEC_LEVEL_REQUIRE 2

Definition at line 192 of file ipsec.h.

Referenced by ipsec_get_reqlevel(), ipsec_in_reject(), key_checkrequest(), and key_msg2sp().

7.10.1.10 #define IPSEC_LEVEL_UNIQUE 3

Definition at line 193 of file ipsec.h.

Referenced by ipsec_get_reqlevel(), and key_msg2sp().

7.10.1.11 #define IPSEC_LEVEL_USE 1

Definition at line 191 of file ipsec.h.

Referenced by ipsec_get_reqlevel(), ipsec_nextisr(), and key_msg2sp().

7.10.1.12 #define IPSEC_MANUAL_REQID_MAX 0x3fff

Definition at line 195 of file ipsec.h.

Referenced by key_msg2sp(), and key_newreqid().

7.10.1.13 #define IPSEC_MODE_ANY 0

Definition at line 161 of file ipsec.h.

Referenced by key_acquire2(), key_add(), key_cmpsaidx(), key_delete(), key_delete_all(), key_get(), key_getspi(), key_msg2sp(), and key_update().

7.10.1.14 #define IPSEC_MODE_TCPMD5 3

Definition at line 164 of file ipsec.h.

7.10.1.15 #define IPSEC_MODE_TRANSPORT 1

Definition at line 162 of file ipsec.h.

Referenced by ipsec_nextisr(), key_checkrequest(), and key_msg2sp().

7.10.1.16 #define IPSEC_MODE_TUNNEL 2

Definition at line 163 of file ipsec.h.

Referenced by ipsec_get_reqlevel(), ipsec_hdrsiz(), key_checkrequest(), key_gettunnel(), and key_-msg2sp().

7.10.1.17 #define ipsec_pcbconn(_x)

Definition at line 356 of file ipsec.h.

7.10.1.18 #define ipsec_pcbdisconn(_x)

Definition at line 357 of file ipsec.h.

7.10.1.19 #define IPSEC_POLICY_BYPASS 4

Definition at line 187 of file ipsec.h.

Referenced by ipsec4_checkpolicy(), ipsec_getpolicybysock(), ipsec_hdrsiz(), ipsec_in_reject(), ipsec_-set_policy(), kdebug_secpolicy(), key_freesp_so(), key_msg2sp(), and key_spdadd().

7.10.1.20 #define IPSEC_POLICY_DISCARD 0

Definition at line 183 of file ipsec.h.

Referenced by ipsec4_checkpolicy(), ipsec_hdrsiz(), ipsec_in_reject(), ipsec_set_policy(), kdebug_-secpolicy(), key_allocsp_default(), and key_msg2sp().

7.10.1.21 #define IPSEC_POLICY_ENTRUST 3

Definition at line 186 of file ipsec.h.

Referenced by ipsec4_checkpolicy(), ipsec_getpolicybysock(), ipsec_init_policy(), kdebug_secpolicy(), key_freesp_so(), key_msg2sp(), and key_spdadd().

7.10.1.22 #define IPSEC_POLICY_IPSEC 2

Definition at line 185 of file ipsec.h.

Referenced by ipsec4_checkpolicy(), ipsec_getpolicybysock(), ipsec_hdrsiz(), ipsec_in_reject(), kdebug_sadb_x_policy(), kdebug_secpolicy(), key_freesp_so(), key_getspreqmsglen(), key_msg2sp(), key_sp2msg(), key_spdacquire(), and key_spdadd().

7.10.1.23 #define IPSEC_POLICY_NONE 1

Definition at line 184 of file ipsec.h.

Referenced by ipsec4_checkpolicy(), ipsec_hdrsiz(), ipsec_in_reject(), ipsec_set_policy(), kdebug_secpolicy(), key_allocsp_default(), key_init(), and key_msg2sp().

7.10.1.24 #define IPSEC_PORT_ANY 0

Definition at line 155 of file ipsec.h.

Referenced by ipsec4_get_ulp(), ipsec_nextisr(), and key_cmpspidx_withmask().

7.10.1.25 #define IPSEC_PROTO_ANY 255

Definition at line 157 of file ipsec.h.

Referenced by key_satype2proto().

7.10.1.26 #define IPSEC_REPLAYWSIZE 32

Definition at line 206 of file ipsec.h.

7.10.1.27 #define IPSEC_SPSTATE_ALIVE 1

Definition at line 82 of file ipsec.h.

Referenced by ipsec_init_policy(), ipsec_set_policy(), and key_spdadd().

7.10.1.28 #define IPSEC_SPSTATE_DEAD 0

Definition at line 81 of file ipsec.h.

Referenced by key_allocsp(), key_allocsp2(), key_delsp(), key_flush_spd(), key_getsp(), key_getspbyid(), key_gettunnel(), key_spdadd(), key_spddelete(), key_spddelete2(), and key_spdflush().

7.10.1.29 #define IPSEC_ULPROTO_ANY 255

Definition at line 156 of file ipsec.h.

Referenced by ipsec4_get_ulp(), key_acquire(), key_cmpspidx_withmask(), key_do_allocsa_policy(), key_expire(), and key_setdumpsa().

7.10.1.30 #define IPSECCTL_AH_CLEARTOS 8

Definition at line 278 of file ipsec.h.

7.10.1.31 #define IPSECCTL_AH_OFFSETMASK 9

Definition at line 279 of file ipsec.h.

7.10.1.32 #define IPSECCTL_DEBUG 12

Definition at line 282 of file ipsec.h.

7.10.1.33 #define IPSECCTL_DEF_AH_NETLEV 6

Definition at line 274 of file ipsec.h.

7.10.1.34 #define IPSECCTL_DEF_AH_TRANSLEV 5

Definition at line 273 of file ipsec.h.

7.10.1.35 #define IPSECCTL_DEF_ESP_NETLEV 4

Definition at line 272 of file ipsec.h.

7.10.1.36 #define IPSECCTL_DEF_ESP_TRANSLEV 3

Definition at line 271 of file ipsec.h.

7.10.1.37 #define IPSECCTL_DEF_POLICY 2

Definition at line 270 of file ipsec.h.

7.10.1.38 #define IPSECCTL_DFBIT 10

Definition at line 280 of file ipsec.h.

7.10.1.39 #define IPSECCTL_ECN 11

Definition at line 281 of file ipsec.h.

7.10.1.40 #define IPSECCTL_ESP_RANDPAD 13

Definition at line 283 of file ipsec.h.

7.10.1.41 #define IPSECCTL_MAXID 14

Definition at line 284 of file ipsec.h.

7.10.1.42 #define IPSECCTL_NAMES

Value:

```
{ \
        { 0, 0 }, \
        { 0, 0 }, \
        { "def_policy", CTLTYPE_INT }, \
        { "esp_trans_deflev", CTLTYPE_INT }, \
        { "esp_net_deflev", CTLTYPE_INT }, \
        { "ah_trans_deflev", CTLTYPE_INT }, \
        { "ah_net_deflev", CTLTYPE_INT }, \
        { 0, 0 }, \
        { "ah_cleartos", CTLTYPE_INT }, \
        { "ah_offsetmask", CTLTYPE_INT }, \
        { "dfbit", CTLTYPE_INT }, \
        { "ecn", CTLTYPE_INT }, \
        { "debug", CTLTYPE_INT }, \
        { "esp_randpad", CTLTYPE_INT }, \
}
```

Definition at line 286 of file ipsec.h.

7.10.1.43 #define IPSECCTL_STATS 1

Definition at line 269 of file ipsec.h.

7.10.1.44 #define ipseclog(x) do { if (ipsec_debug) log x; } while (0)

Definition at line 351 of file ipsec.h.

Referenced by ipsec4_get_policy(), ipsec4_set_policy(), ipsec_get_policy(), ipsec_getpolicybysock(), ipsec_hdrsiz(), ipsec_init_policy(), ipsec_updatereplay(), key_acquire2(), key_add(), key_align(), key_allocsp_default(), key_checkrequest(), key_checkspidup(), key_delete(), key_delete_all(), key_do_getnewspi(), key_dump(), key_dup_keymsg(), key_dup_lifemsg(), key_flush(), key_flush_sad(), key_freeso(), key_get(), key_getnewspid(), key_getsavbyspi(), key_getspi(), key_gettunnel(), key_mature(), key_msg2sp(), key_newacq(), key_newsav(), key_newspacq(), key_parse(), key_register(), key_setident(), key_setsaval(), key_spdadd(), key_spddelete(), key_spddelete2(), key_spdflush(), key_spdget(), and key_update().

7.10.1.45 #define IPSECREQUEST_LOCK(_isr) mtx_lock(&(_isr) → lock)

Definition at line 130 of file ipsec.h.

Referenced by ah_output_cb(), esp_output_cb(), ipcomp_output_cb(), and ipsec_nextisr().

7.10.1.46 #define IPSECREQUEST_LOCK_ASSERT(_isr) mtx_assert(&(_isr) → lock, MA_OWNED)

Definition at line 133 of file ipsec.h.

Referenced by ipsec_nextisr(), and key_checkrequest().

7.10.1.47 #define IPSECREQUEST_LOCK_DESTROY(_isr) mtx_destroy(&(_isr) → lock)

Definition at line 132 of file ipsec.h.

Referenced by ipsec_delisr().

7.10.1.48 #define IPSECREQUEST_LOCK_INIT(_isr) mtx_init(&(_isr) → lock, "ipsec request", NULL, MTX_DEF | MTX_RECURSE)

Definition at line 128 of file ipsec.h.

Referenced by ipsec_newisr().

7.10.1.49 #define IPSECREQUEST_UNLOCK(_isr) mtx_unlock(&(_isr) → lock)

Definition at line 131 of file ipsec.h.

Referenced by ah_output_cb(), esp_output_cb(), ipcomp_output_cb(), and ipsec_nextisr().

7.10.1.50 #define SECPOLICY_LOCK(_sp) mtx_lock(&(_sp) → lock)

Definition at line 104 of file ipsec.h.

7.10.1.51 #define SECPOLICY_LOCK_ASSERT(_sp) mtx_assert(&(_sp) → lock, MA_OWNED)

Definition at line 107 of file ipsec.h.

7.10.1.52 #define SECPOLICY_LOCK_DESTROY(_sp) mtx_destroy(&(_sp) → lock)

Definition at line 106 of file ipsec.h.

Referenced by _key_delsp().

7.10.1.53 #define SECPOLICY_LOCK_INIT(_sp) mtx_init(&(_sp) → lock, "ipsec policy", NULL, MTX_DEF)

Definition at line 102 of file ipsec.h.

Referenced by ipsec_attach(), and key_newsp().

7.10.1.54 #define SECPOLICY_UNLOCK(_sp) mtx_unlock(&(_sp) → lock)

Definition at line 105 of file ipsec.h.

7.10.2 Function Documentation

7.10.2.1 int ipsec_process_done __P ((struct mbuf *, struct ipsecrequest *))

7.10.2.2 int ipsec4_process_packet __P ((struct mbuf *, struct ipsecrequest *, int, int))

7.10.2.3 void kdebug_mbuf __P ((struct mbuf *))

7.10.2.4 void keydb_freesecasvar __P ((struct secasvar *))

7.10.2.5 size_t ipsec_hdrsiz_tcp __P ((struct tcpcb *))

7.10.2.6 size_t ipsec4_hdrsiz __P ((struct mbuf *, u_int, struct inpcb *))

7.10.2.7 int ipsec_updatereplay __P ((u_int32_t, struct secasvar *))

7.10.2.8 int ipsec4_in_reject __P ((struct mbuf *, struct inpcb *))

7.10.2.9 int ipsec4_delete_pcbpolicy __P ((struct inpcb *))

7.10.2.10 int ipsec4_get_policy __P ((struct inpcb *inpcb, caddr_t request, size_t len, struct mbuf **mp))

7.10.2.11 int ipsec4_set_policy __P ((struct inpcb *inp, int optname, caddr_t request, size_t len, int priv))

7.10.2.12 int ipsec_in_reject __P ((struct secpolicy *, struct mbuf *))

7.10.2.13 u_int ipsec_get_reqlevel __P ((struct ipsecrequest *))

7.10.2.14 int ipsec_copy_policy __P ((struct inpcbpolicy *, struct inpcbpolicy *))

7.10.2.15 int ipsec_init_policy __P ((struct socket *so, struct inpcbpolicy **))

7.10.2.16 struct secpolicy* ipsec4_checkpolicy __P ((struct mbuf *, u_int, u_int, int *, struct inpcb *))

7.10.2.17 struct secpolicy* ipsec_getpolicy __P ((struct tdb_ident *, u_int))

7.10.2.18 void ah4_ctlinput (int *cmd*, struct sockaddr * *sa*, void *)

7.10.2.19 void ah4_input (struct mbuf * *m*, int *off*)

7.10.2.20 void esp4_ctlinput (int *cmd*, struct sockaddr * *sa*, void *)

7.10.2.21 void esp4_input (struct mbuf * *m*, int *off*)

7.10.2.22 void ipcomp4_input (struct mbuf * *m*, int *off*)

7.10.2.23 int ipsec4_common_input (struct mbuf * *m*, ...)

7.10.2.24 int ipsec4_common_input_cb (struct mbuf * *m*, struct secasvar * *sav*, int *skip*, int *protoff*, struct m_tag * *mt*)

7.10.2.25 char* ipsec_address (union sockaddr_union * *sa*)

Definition at line 1844 of file ipsec.c.

References inet_ntoa4(), sockaddr_union::sa, sockaddr_union::sin, and sockaddr_union::sin6.

Referenced by ah_input(), ah_input_cb(), ah_output(), esp_input(), esp_input_cb(), esp_output(), esp_-output_cb(), ipcomp_input_cb(), ipcomp_output(), ipcomp_output_cb(), ipip_output(), ipsec_common_-input(), and ipsec_logsastr().

Here is the call graph for this function:

7.10.2.26 void ipsec_bpf (struct mbuf *, struct secasvar *, int)

7.10.2.27 void ipsec_delisr (struct ipsecrequest *)

Definition at line 929 of file ipsec.c.

References IPSECREQUEST_LOCK_DESTROY.

Referenced by ipsec_deepcopy_policy(), and key_delsp().

7.10.2.28 int ipsec_filter (struct mbuf **, int)

Referenced by _ipip_input().

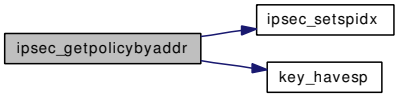
7.10.2.29 struct secpolicy* ipsec_getpolicybyaddr (struct mbuf *, u_int, int, int *)

Definition at line 402 of file ipsec.c.

References secpolicyindex::dir, DPRINTF, IPSEC_ASSERT, IPSEC_DIR_INBOUND, IPSEC_DIR_-OUTBOUND, ipsec_setspidx(), KEY_ALLOCSP, KEY_ALLOCSP_DEFAULT, key_havesp(), and secpolicy::spidx.

Referenced by ipsec4_checkpolicy(), ipsec4_hdrsiz(), and ipsec4_in_reject().

Here is the call graph for this function:



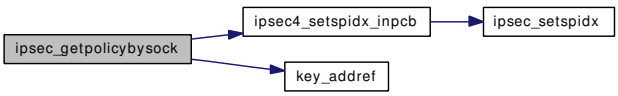
7.10.2.30 struct secpolicy* ipsec_getpolicybysock (struct mbuf *, u_int, struct inpcb *, int *)

Definition at line 292 of file ipsec.c.

References ipsec4_setspidx_inpcb(), IPSEC_ASSERT, IPSEC_DIR_INBOUND, IPSEC_DIR_-OUTBOUND, IPSEC_POLICY_BYPASS, IPSEC_POLICY_ENTRUST, IPSEC_POLICY_-IPSEC, ipseclog, key_addref(), KEY_ALLOCSP, KEY_ALLOCSP_DEFAULT, KEYDEBUG, KEYDEBUG_IPSEC_STAMP, secpolicy::policy, inpcbpolicy::priv, secpolicy::refcnt, inpcbpolicy::sp_in, inpcbpolicy::sp_out, and secpolicy::spidx.

Referenced by ipsec4_checkpolicy(), ipsec4_hdrsiz(), and ipsec4_in_reject().

Here is the call graph for this function:



7.10.2.31 struct ipsecrequest* ipsec_newisr (void)

Definition at line 918 of file ipsec.c.

References IPSECREQUEST_LOCK_INIT.

Referenced by ipsec_deepcopy_policy(), and key_msg2sp().

7.10.2.32 void m_checkalignment (const char * *where*, struct mbuf * *m0*, int *off*, int *len*)

Definition at line 293 of file ipsec_mbuf.c.

7.10.2.33 struct mbuf* m_makespace (struct mbuf * *m0*, int *skip*, int *hlen*, int * *off*)

Definition at line 54 of file ipsec_mbuf.c.

References IPSEC_ASSERT.

Referenced by ah_output(), esp_output(), and ipcomp_output().

7.10.2.34 caddr_t m_pad (struct mbuf * *m*, int *n*)

Definition at line 156 of file ipsec_mbuf.c.

References DPRINTF.

Referenced by esp_output().

7.10.2.35 int m_striphdr (struct mbuf * *m*, int *skip*, int *hlen*)

Definition at line 228 of file ipsec_mbuf.c.

Referenced by ah_input_cb(), esp_input_cb(), and ipcomp_input_cb().

7.10.3 Variable Documentation

7.10.3.1 int crypto_support

Definition at line 120 of file ipsec.c.

Referenced by ah_init(), esp_init(), and ipcomp_init().

7.10.3.2 int ip4_ah_cleartos

7.10.3.3 int ip4_ah_net_deflev

Definition at line 109 of file ipsec.c.

Referenced by ipsec_get_reqlevel().

7.10.3.4 int ip4_ah_offsetmask

Definition at line 104 of file ipsec.c.

7.10.3.5 int ip4_ah_trans_deflev

Definition at line 108 of file ipsec.c.

Referenced by ipsec_get_reqlevel().

7.10.3.6 struct secpolicy ip4_def_policy

Definition at line 110 of file ipsec.c.

Referenced by ipsec_attach(), key_allocsp_default(), and key_init().

7.10.3.7 int ip4_esp_net_deflev

Definition at line 107 of file ipsec.c.

Referenced by ipsec_get_reqlevel().

7.10.3.8 int ip4_esp_randpad

Definition at line 112 of file ipsec.c.

7.10.3.9 int ip4_esp_trans_deflev

Definition at line 106 of file ipsec.c.

Referenced by ipsec_get_reqlevel().

7.10.3.10 int ip4_ipsec_dfbit

Definition at line 105 of file ipsec.c.

7.10.3.11 int ip4_ipsec_ecn

Definition at line 111 of file ipsec.c.

Referenced by _ipip_input().

7.10.3.12 int ipsec_debug

Definition at line 99 of file ipsec.c.

7.10.3.13 struct newipsecstat newipsecstat

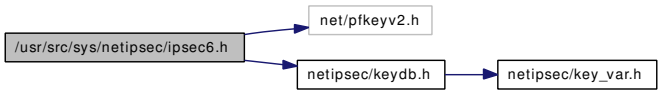
Definition at line 103 of file ipsec.c.

Referenced by ipsec4_checkpolicy(), and ipsec4_in_reject().

7.11 /usr/src/sys/netipsec/ipsec6.h File Reference

```
#include <net/pfkeyv2.h>
#include <netipsec/keydb.h>
```

Include dependency graph for ipsec6.h:



Defines

- #define ipsec6_getpolicybyaddr ipsec_getpolicybyaddr
- #define ipsec6_getpolicybysock ipsec_getpolicybysock
- #define ipsec6stat newipsecstat
- #define out_inval ips_out_inval
- #define in_polvio ips_in_polvio
- #define out_polvio ips_out_polvio
- #define key_freesp(_x) KEY_FREESP(&_x)

Functions

- int ipsec6_delete_pcbpolicy __P ((struct inpcb *))
- int ipsec6_set_policy __P ((struct inpcb *inp, int optname, caddr_t request, size_t len, int priv))
- int ipsec6_get_policy __P ((struct inpcb *inp, caddr_t request, size_t len, struct mbuf **mp))
- int ipsec6_in_reject __P ((struct mbuf *, struct inpcb *))
- size_t ipsec6_hdrsiz __P ((struct mbuf *, u_int, struct inpcb *))
- const char *ipsec6_logpacketstr __P ((struct ip6_hdr *, u_int32_t))
- int ipsec6_common_input (struct mbuf **mp, int *offp, int proto)
- int ipsec6_common_input_cb (struct mbuf *m, struct secasvar *sav, int skip, int protoff, struct m_tag *mt)
- void esp6_ctlinput (int, struct sockaddr *, void *)
- int ipsec6_output_trans __P ((struct ipsec_output_state *, u_char *, struct mbuf *, struct secpolicy *, int, int *))
- int ipsec6_output_tunnel __P ((struct ipsec_output_state *, struct secpolicy *, int))

Variables

- int ip6_esp_trans_deflev
- int ip6_esp_net_deflev
- int ip6_ah_trans_deflev
- int ip6_ah_net_deflev
- int ip6_ipsec_ecn
- int ip6_esp_randpad

7.11.1 Define Documentation

7.11.1.1 #define in_polvio ips_in_polvio

Definition at line 58 of file ipsec6.h.

7.11.1.2 #define ipsec6_getpolicybyaddr ipsec_getpolicybyaddr

Definition at line 54 of file ipsec6.h.

7.11.1.3 #define ipsec6_getpolicybysock ipsec_getpolicybysock

Definition at line 55 of file ipsec6.h.

7.11.1.4 #define ipsec6stat newipsecstat

Definition at line 56 of file ipsec6.h.

7.11.1.5 #define key_freesp(_x) KEY_FREESP(&_x)

Definition at line 60 of file ipsec6.h.

7.11.1.6 #define out_inval ips_out_inval

Definition at line 57 of file ipsec6.h.

7.11.1.7 #define out_polvio ips_out_polvio

Definition at line 59 of file ipsec6.h.

7.11.2 Function Documentation

7.11.2.1 int ipsec6_output_tunnel __P ((struct ipsec_output_state *, struct secpolicy *, int))

7.11.2.2 int ipsec6_output_trans __P ((struct ipsec_output_state *, u_char *, struct mbuf *, struct secpolicy *, int, int *))

7.11.2.3 const char* ipsec6_logpacketstr __P ((struct ip6_hdr *, u_int32_t))

7.11.2.4 size_t ipsec6_hdrsiz __P ((struct mbuf *, u_int, struct inpcb *))

7.11.2.5 int ipsec6_in_reject __P ((struct mbuf *, struct inpcb *))

7.11.2.6 int ipsec6_get_policy __P ((struct inpcb *inp, caddr_t request, size_t len, struct mbuf **mp))

7.11.2.7 int ipsec6_set_policy __P ((struct inpcb *inp, int optname, caddr_t request, size_t len, int priv))

7.11.2.8 int ipsec6_delete_pcbpolicy __P ((struct inpcb *))

7.11.2.9 void esp6_ctlinput (int, struct sockaddr *, void *)

7.11.2.10 int ipsec6_common_input (struct mbuf ** *mp*, int * *offp*, int *proto*)

7.11.2.11 int ipsec6_common_input_cb (struct mbuf * *m*, struct secasvar * *sav*, int *skip*, int *protoff*, struct m_tag * *mt*)

7.11.3 Variable Documentation

7.11.3.1 int ip6_ah_net_deflev

Referenced by ipsec_get_reqlevel().

7.11.3.2 int ip6_ah_trans_deflev

Referenced by ipsec_get_reqlevel().

7.11.3.3 int ip6_esp_net_deflev

Referenced by ipsec_get_reqlevel().

7.11.3.4 int ip6_esp_randpad

7.11.3.5 int ip6_esp_trans_deflev

Referenced by ipsec_get_reqlevel().

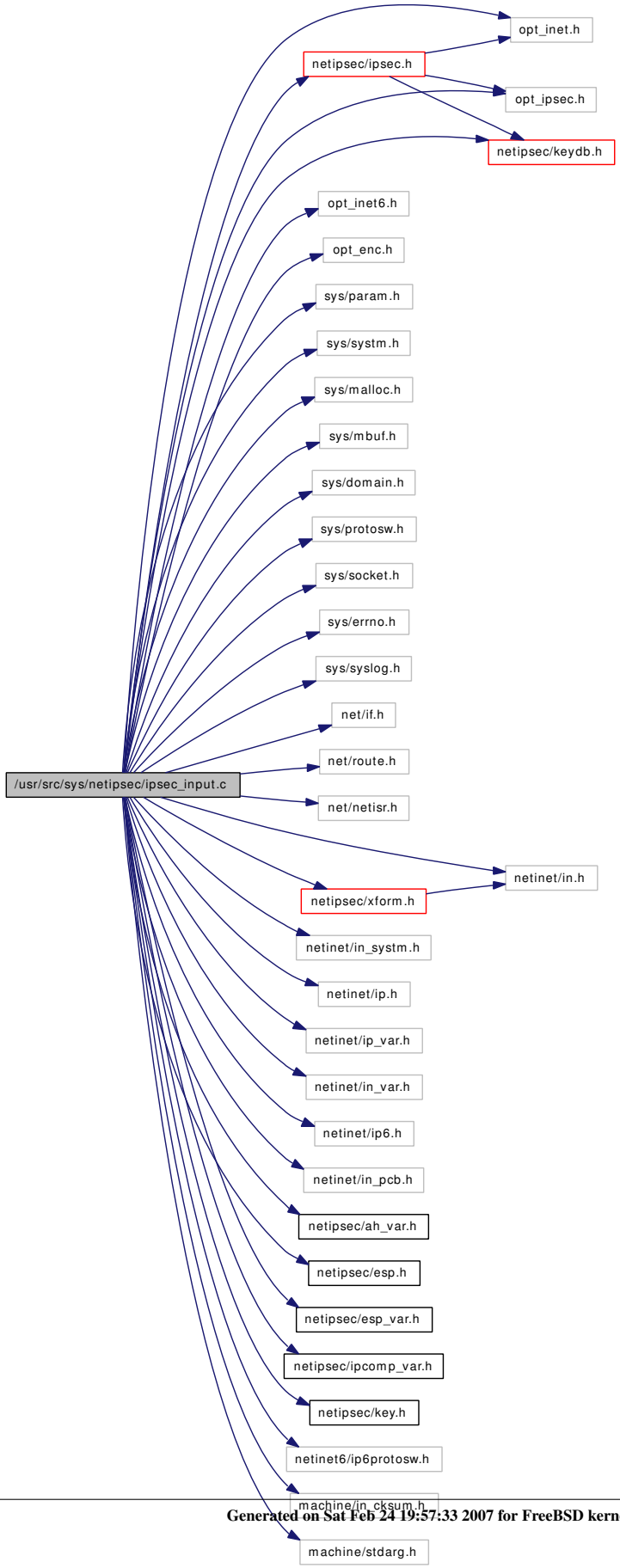
7.11.3.6 int ip6_ipsec_ecn

Referenced by _ipip_input().

7.12 /usr/src/sys/netipsec/ipsec_input.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include "opt_ipsec.h"
#include "opt_enc.h"
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/domain.h>
#include <sys/protosw.h>
#include <sys/socket.h>
#include <sys/errno.h>
#include <sys/syslog.h>
#include <net/if.h>
#include <net/route.h>
#include <net/netisr.h>
#include <netinet/in.h>
#include <netinet/in_systm.h>
#include <netinet/ip.h>
#include <netinet/ip_var.h>
#include <netinet/in_var.h>
#include <netinet/ip6.h>
#include <netinet/in_pcb.h>
#include <netipsec/ipsec.h>
#include <netipsec/ah_var.h>
#include <netipsec/esp.h>
#include <netipsec/esp_var.h>
#include <netipsec/ipcomp_var.h>
#include <netipsec/key.h>
#include <netipsec/keydb.h>
#include <netipsec/xform.h>
#include <netinet6/ip6protosw.h>
#include <machine/in_cksum.h>
#include <machine/stdarg.h>
```

Include dependency graph for ipsec_input.c:


opt_inet.h
netipsec/ipsec.h
opt_ipsec.h
netipsec/keydb.h
opt_inet6.h
opt_enc.h
sys/param.h
sys/systm.h
sys/malloc.h
sys/mbuf.h
sys/domain.h
sys/protosw.h
sys/socket.h
sys/errno.h
sys/syslog.h
net/if.h
net/route.h
net/netisr.h
/usr/src/sys/netipsec/ipsec_input.c
netinet/in.h
netipsec/xform.h
netinet/in_systm.h
netinet/ip.h
netinet/ip_var.h
netinet/in_var.h
netinet/ip6.h
netinet/in_pcb.h
netipsec/ah_var.h
netipsec/esp.h
netipsec/esp_var.h
netipsec/ipcomp_var.h
netipsec/key.h
netinet6/ip6protosw.h
machine/in_cksum.h
machine/stdarg.h

Defines

- #define IPSEC_ISTAT(p, x, y, z)

Functions

- static void ipsec4_common_ctlinput (int, struct sockaddr ∗, void ∗, int)
- static int ipsec_common_input (struct mbuf ∗m, int skip, int protoff, int af, int sproto)

7.12.1 Define Documentation

7.12.1.1 #define IPSEC_ISTAT(p, x, y, z)

Value:

```
((p) == IPPROTO_ESP ? (x)++ : \
                            (p) == IPPROTO_AH ? (y)++ : (z)++)
```

Definition at line 95 of file ipsec_input.c.

Referenced by ipsec_common_input().

7.12.2 Function Documentation

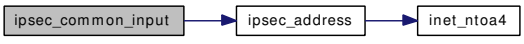
7.12.2.1 static void ipsec4_common_ctlinput (int, struct sockaddr ∗, void ∗, int) [static]

7.12.2.2 static int ipsec_common_input (struct mbuf ∗ *m*, int *skip*, int *protoff*, int *af*, int *sproto*) [static]

Definition at line 107 of file ipsec_input.c.

References ah_enable, DPRINTF, esp_enable, ipcomp_enable, ipsec_address(), IPSEC_ASSERT, IPSEC_ISTAT, KEY_ALLOCSA, KEY_FREESAV, secasvar::spi, secasvar::tdb_xform, and xformsw::xf_input.

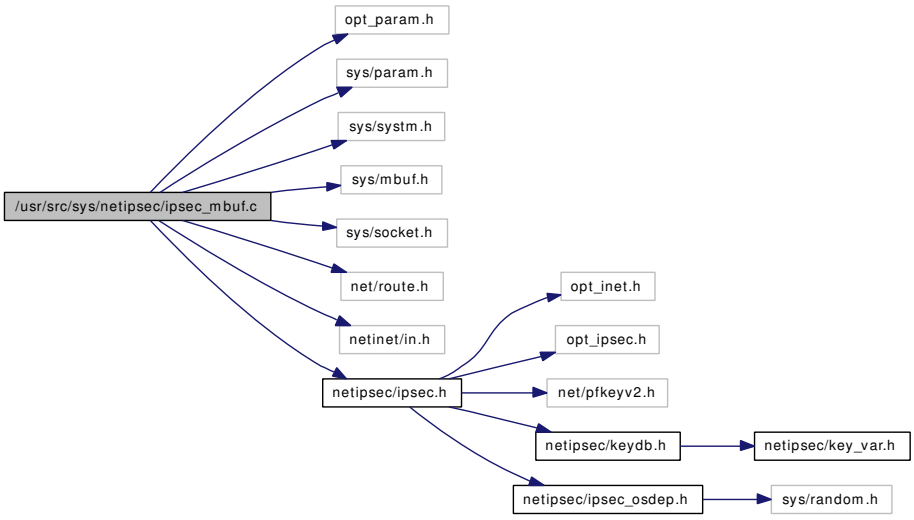
Here is the call graph for this function:



7.13 /usr/src/sys/netipsec/ipsec_mbuf.c File Reference

```
#include "opt_param.h"
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/mbuf.h>
#include <sys/socket.h>
#include <net/route.h>
#include <netinet/in.h>
#include <netipsec/ipsec.h>
```

Include dependency graph for ipsec_mbuf.c:



Functions

- mbuf ∗ m_makespace (struct mbuf ∗m0, int skip, int hlen, int ∗off)
- caddr_t m_pad (struct mbuf ∗m, int n)
- int m_striphdr (struct mbuf ∗m, int skip, int hlen)
- void m_checkalignment (const char ∗where, struct mbuf ∗m0, int off, int len)

7.13.1 Function Documentation

7.13.1.1 void m_checkalignment (const char ∗ *where*, struct mbuf ∗ *m0*, int *off*, int *len*)

Definition at line 293 of file ipsec_mbuf.c.

7.13.1.2 struct mbuf∗ m_makespace (struct mbuf ∗ *m0*, int *skip*, int *hlen*, int ∗ *off*)

Definition at line 54 of file ipsec_mbuf.c.

References IPSEC_ASSERT.

Referenced by ah_output(), esp_output(), and ipcomp_output().

7.13.1.3 caddr_t m_pad (struct mbuf * *m*, int *n*)

Definition at line 156 of file ipsec_mbuf.c.

References DPRINTF.

Referenced by esp_output().

7.13.1.4 int m_striphdr (struct mbuf * *m*, int *skip*, int *hlen*)

Definition at line 228 of file ipsec_mbuf.c.

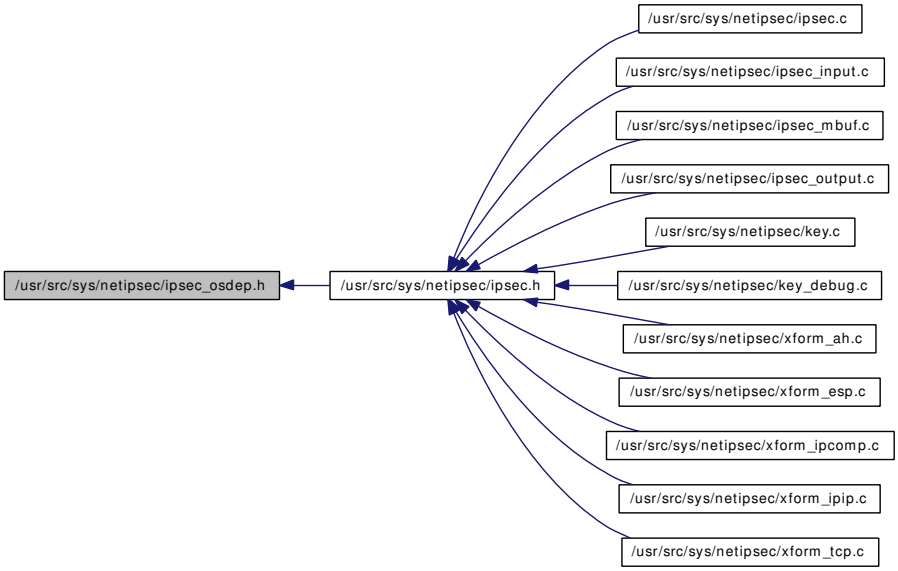
Referenced by ah_input_cb(), esp_input_cb(), and ipcomp_input_cb().

7.14 /usr/src/sys/netipsec/ipsec_osdep.h File Reference

```
#include <sys/random.h>
```

Include dependency graph for ipsec_osdep.h:

This graph shows which files directly or indirectly include this file:



Defines

- #define IPSEC_SPLASSERT(_x, _y) SPLASSERT(_x, _y)
- #define IPSEC_SPLASSERT_SOFTNET(_m) IPSEC_SPLASSERT(net,_m)
- #define IPSEC_ASSERT(_c, _m) KASSERT(_c, _m)
- #define IPSEC_IS_PRIVILEGED_SO(_so)
- #define rcb_list list

7.14.1 Define Documentation

7.14.1.1 #define IPSEC_ASSERT(_c, _m) KASSERT(_c, _m)

Definition at line 70 of file ipsec_osdep.h.

Referenced by _key_freesp(), ah_hdrsiz(), ah_input(), ah_input_cb(), ah_output(), ah_output_cb(), esp_hdrsiz(), esp_input(), esp_input_cb(), esp_output(), esp_output_cb(), ipcomp_input_cb(), ipcomp_output(), ipcomp_output_cb(), ipip_output(), ipsec4_checkpolicy(), ipsec4_delete_pcbpolicy(), ipsec4_get_policy(), ipsec4_get_ulp(), ipsec4_hdrsiz(), ipsec4_in_reject(), ipsec4_setspidx_inpcb(), ipsec_chkreplay(), ipsec_common_input(), ipsec_get_reqlevel(), ipsec_getpolicy(), ipsec_getpolicybyaddr(),

ipsec_getpolicybysock(), ipsec_hdrsiz(), ipsec_in_reject(), ipsec_logsastr(), ipsec_nextisr(), ipsec_process_done(), ipsec_setspidx(), ipsec_updatereplay(), key_acquire(), key_acquire2(), key_add(), key_align(), key_allocsa(), key_allocsp(), key_allocsp2(), key_checkrequest(), key_checktunnelsanity(), key_delete(), key_delsah(), key_delsav(), key_delsp(), key_do_allocsa_policy(), key_dump(), key_expire(), key_flush(), key_freereg(), key_freesav(), key_freeso(), key_freesp_so(), key_gather_mbuf(), key_get(), key_getcomb_ah(), key_getcomb_esp(), key_getcomb_ipcomp(), key_getmsgbuf_x1(), key_getsp(), key_getspi(), key_ismyaddr(), key_msg2sp(), key_newsah(), key_newsav(), key_parse(), key_promisc(), key_register(), key_sa_chgstate(), key_sa_recordxfer(), key_sa_stir_iv(), key_senderror(), key_setident(), key_setsaval(), key_sp2msg(), key_spdacquire(), key_spdadd(), key_spddelete(), key_spddelete2(), key_spddump(), key_spdexpire(), key_spdflush(), key_spdget(), key_update(), m_makespace(), sa_addref(), and sa_delref().

7.14.1.2 #define IPSEC_IS_PRIVILEGED_SO(_so)

Value:

```
((_so)->so_cred != NULL && \
         priv_check_cred((_so)->so_cred, PRIV_NETINET_IPSEC, 0) \
         == 0)
```

Definition at line 221 of file ipsec_osdep.h.

Referenced by ipsec_init_policy().

7.14.1.3 #define IPSEC_SPLASSERT(_x, _y) SPLASSERT(_x, _y)

Definition at line 65 of file ipsec_osdep.h.

7.14.1.4 #define IPSEC_SPLASSERT_SOFTNET(_m) IPSEC_SPLASSERT(net,_m)

Definition at line 69 of file ipsec_osdep.h.

Referenced by ah_input(), ah_output(), esp_input(), esp_output(), ipcomp_input(), ipcomp_output(), ipip_output(), ipsec_chkreplay(), ipsec_nextisr(), ipsec_process_done(), and ipsec_updatereplay().

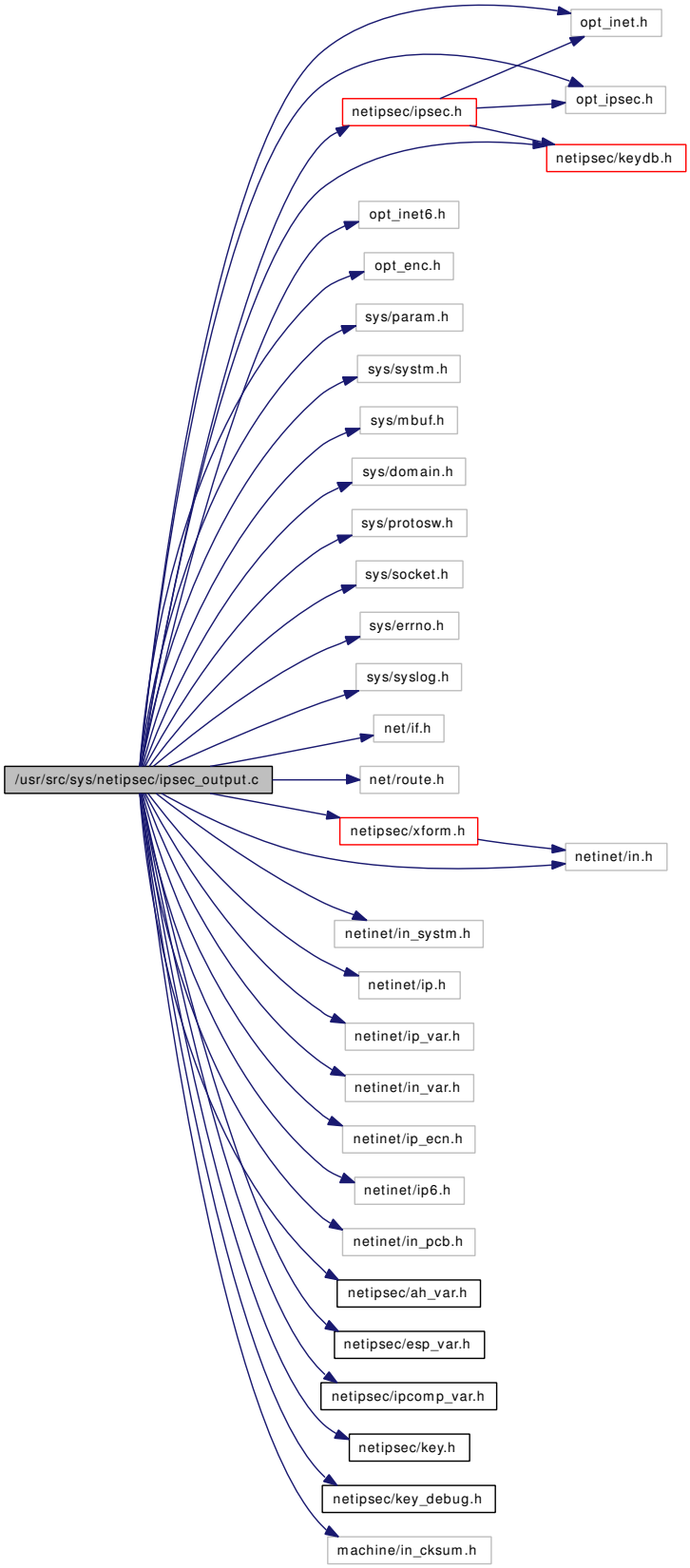
7.14.1.5 #define rcb_list list

Definition at line 241 of file ipsec_osdep.h.

7.15 /usr/src/sys/netipsec/ipsec_output.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include "opt_ipsec.h"
#include "opt_enc.h"
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/mbuf.h>
#include <sys/domain.h>
#include <sys/protosw.h>
#include <sys/socket.h>
#include <sys/errno.h>
#include <sys/syslog.h>
#include <net/if.h>
#include <net/route.h>
#include <netinet/in.h>
#include <netinet/in_systm.h>
#include <netinet/ip.h>
#include <netinet/ip_var.h>
#include <netinet/in_var.h>
#include <netinet/ip_ecn.h>
#include <netinet/ip6.h>
#include <netinet/in_pcb.h>
#include <netipsec/ipsec.h>
#include <netipsec/ah_var.h>
#include <netipsec/esp_var.h>
#include <netipsec/ipcomp_var.h>
#include <netipsec/xform.h>
#include <netipsec/key.h>
#include <netipsec/keydb.h>
#include <netipsec/key_debug.h>
#include <machine/in_cksum.h>
```

Include dependency graph for ipsec_output.c:


opt_inet.h
opt_ipsec.h
netipsec/ipsec.h
netipsec/keydb.h
opt_inet6.h
opt_enc.h
sys/param.h
sys/systm.h
sys/mbuf.h
sys/domain.h
sys/protosw.h
sys/socket.h
sys/errno.h
sys/syslog.h
net/if.h
net/route.h
/usr/src/sys/netipsec/ipsec_output.c
netipsec/xform.h
netinet/in.h
netinet/in_systm.h
netinet/ip.h
netinet/ip_var.h
netinet/in_var.h
netinet/ip_ecn.h
netinet/ip6.h
netinet/in_pcb.h
netipsec/ah_var.h
netipsec/esp_var.h
netipsec/ipcomp_var.h
netipsec/key.h
netipsec/key_debug.h
machine/in_cksum.h

Defines

- #define IPSEC_OSTAT(x, y, z)

Functions

- int ipsec_process_done (struct mbuf *m, struct ipsecrequest *isr)
- static struct ipsecrequest * ipsec_nextisr (struct mbuf *m, struct ipsecrequest *isr, int af, struct secasindex *saidx, int *error)

7.15.1 Define Documentation

7.15.1.1 #define IPSEC_OSTAT(x, y, z)

Value:

```
(isr->saidx.proto == IPPROTO_ESP ? (x)++ : \
                            isr->saidx.proto == IPPROTO_AH ? (y)++ : (z)++)
```

Referenced by ipsec_nextisr().

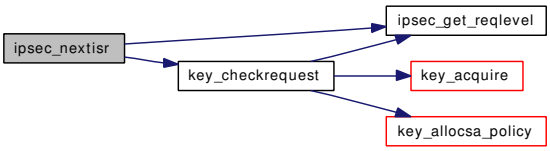
7.15.2 Function Documentation

7.15.2.1 static struct ipsecrequest* ipsec_nextisr (struct mbuf * *m*, struct ipsecrequest * *isr*, int *af*, struct secasindex * *saidx*, int * *error*) [static]

Definition at line 195 of file ipsec_output.c.

References ah_enable, DPRINTF, secasindex::dst, esp_enable, ipcomp_enable, IPSEC_ASSERT, ipsec_get_reqlevel(), IPSEC_LEVEL_USE, IPSEC_MODE_TRANSPORT, IPSEC_OSTAT, IPSEC_PORT_ANY, IPSEC_SPLASSERT_SOFTNET, IPSECREQUEST_LOCK, IPSECREQUEST_LOCK_ASSERT, IPSECREQUEST_UNLOCK, key_checkrequest(), secasindex::mode, secasindex::proto, sockaddr_union::sa, ipsecrequest::saidx, ipsecrequest::sav, sockaddr_union::sin, sockaddr_union::sin6, secasindex::src, and secasvar::tdb_xform.

Here is the call graph for this function:



7.15.2.2 int ipsec_process_done (struct mbuf * *m*, struct ipsecrequest * *isr*)

Definition at line 85 of file ipsec_output.c.

References DPRINTF, tdb_ident::dst, secasindex::dst, IPSEC_ASSERT, IPSEC_SPLASSERT_SOFTNET, KEY_FREESAV, key_sa_recordxfer(), ipsecrequest::next, secasindex::proto, tdb_ident::proto, sockaddr_union::sa, secasvar::sah, secashead::saidx, ipsecrequest::sav, secasvar::spi, and tdb_ident::spi.

Referenced by ah_output_cb(), esp_output_cb(), and ipcomp_output_cb().

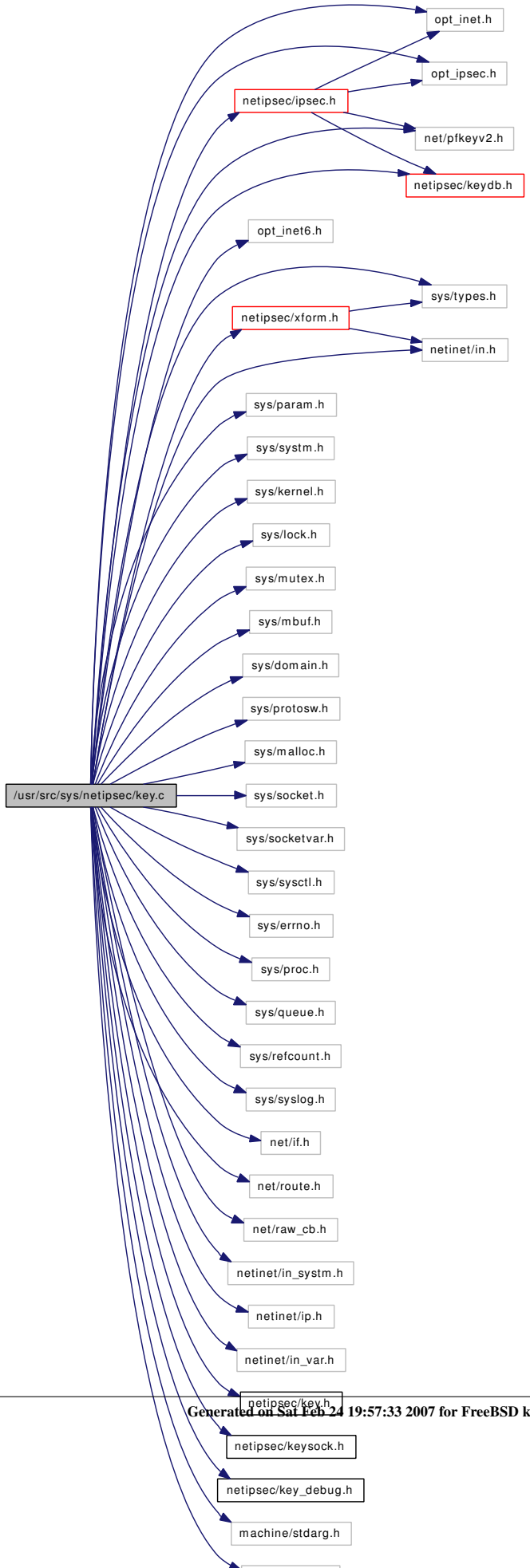
Here is the call graph for this function:

7.16 /usr/src/sys/netipsec/key.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include "opt_ipsec.h"
#include <sys/types.h>
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/kernel.h>
#include <sys/lock.h>
#include <sys/mutex.h>
#include <sys/mbuf.h>
#include <sys/domain.h>
#include <sys/protosw.h>
#include <sys/malloc.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/sysctl.h>
#include <sys/errno.h>
#include <sys/proc.h>
#include <sys/queue.h>
#include <sys/refcount.h>
#include <sys/syslog.h>
#include <net/if.h>
#include <net/route.h>
#include <net/raw_cb.h>
#include <netinet/in.h>
#include <netinet/in_systm.h>
#include <netinet/ip.h>
#include <netinet/in_var.h>
#include <net/pfkeyv2.h>
#include <netipsec/keydb.h>
#include <netipsec/key.h>
#include <netipsec/keysock.h>
#include <netipsec/key_debug.h>
#include <netipsec/ipsec.h>
#include <netipsec/xform.h>
```

```
#include <machine/stdarg.h>
#include <sys/random.h>
```

Include dependency graph for key.c:


opt_inet.h
opt_ipsec.h
netipsec/ipsec.h
net/pfkeyv2.h
netipsec/keydb.h
opt_inet6.h
sys/types.h
netipsec/xform.h
netinet/in.h
sys/param.h
sys/systm.h
sys/kernel.h
sys/lock.h
sys/mutex.h
sys/mbuf.h
sys/domain.h
sys/protosw.h
sys/malloc.h
/usr/src/sys/netipsec/key.c
sys/socket.h
sys/socketvar.h
sys/sysctl.h
sys/errno.h
sys/proc.h
sys/queue.h
sys/refcount.h
sys/syslog.h
net/if.h
net/route.h
net/raw_cb.h
netinet/in_systm.h
netinet/ip.h
netinet/in_var.h
netipsec/key.h
netipsec/keysock.h
netipsec/key_debug.h
machine/stdarg.h

Data Structures

- struct _keystat
- struct sadb_msghdr

Defines

- #define FULLMASK 0xff
- #define _BITS(bytes) ((bytes) << 3)
- #define SPTREE_LOCK_INIT()
- #define SPTREE_LOCK_DESTROY() mtx_destroy(&sptree_lock)
- #define SPTREE_LOCK() mtx_lock(&sptree_lock)
- #define SPTREE_UNLOCK() mtx_unlock(&sptree_lock)
- #define SPTREE_LOCK_ASSERT() mtx_assert(&sptree_lock, MA_OWNED)
- #define SAHTREE_LOCK_INIT()
- #define SAHTREE_LOCK_DESTROY() mtx_destroy(&sahtree_lock)
- #define SAHTREE_LOCK() mtx_lock(&sahtree_lock)
- #define SAHTREE_UNLOCK() mtx_unlock(&sahtree_lock)
- #define SAHTREE_LOCK_ASSERT() mtx_assert(&sahtree_lock, MA_OWNED)
- #define REGTREE_LOCK_INIT() mtx_init(&regtree_lock, "regtree", "fast ipsec regtree", MTX_-DEF)
- #define REGTREE_LOCK_DESTROY() mtx_destroy(&regtree_lock)
- #define REGTREE_LOCK() mtx_lock(&regtree_lock)
- #define REGTREE_UNLOCK() mtx_unlock(&regtree_lock)
- #define REGTREE_LOCK_ASSERT() mtx_assert(&regtree_lock, MA_OWNED)
- #define ACQ_LOCK_INIT() mtx_init(&acq_lock, "acqtree", "fast ipsec acquire list", MTX_DEF)
- #define ACQ_LOCK_DESTROY() mtx_destroy(&acq_lock)
- #define ACQ_LOCK() mtx_lock(&acq_lock)
- #define ACQ_UNLOCK() mtx_unlock(&acq_lock)
- #define ACQ_LOCK_ASSERT() mtx_assert(&acq_lock, MA_OWNED)
- #define SPACQ_LOCK_INIT()
- #define SPACQ_LOCK_DESTROY() mtx_destroy(&spacq_lock)
- #define SPACQ_LOCK() mtx_lock(&spacq_lock)
- #define SPACQ_UNLOCK() mtx_unlock(&spacq_lock)
- #define SPACQ_LOCK_ASSERT() mtx_assert(&spacq_lock, MA_OWNED)
- #define __LIST_CHAINED(elm) (!((elm) → chain.le_next == NULL && (elm) → chain.le_prev == NULL))
- #define LIST_INSERT_TAIL(head, elm, type, field)
- #define KEY_CHKSASTATE(head, sav, name)
- #define KEY_CHKSPDIR(head, sp, name)
- #define KEY_SETSECSPIDX(_dir, s, d, ps, pd, ulp, idx)
- #define KEY_SETSECASIDX(p, m, r, s, d, idx)
- #define KEY_NEWSAV(m, sadb, sah, e) key_newsav(m, sadb, sah, e, __FILE__, __LINE__)
- #define CMP_HEAD 1
- #define CMP_MODE_REQID 2
- #define CMP_REQID 3
- #define CMP_EXACTLY 4
- #define SP_ADDREF(p)
- #define SP_DELREF(p)
- #define N(a) _ARRAYLEN(a)
- #define satosin(s) ((const struct sockaddr_in *)s)
- #define satosin6(s) ((const struct sockaddr_in6 *)s)

Functions

- static LIST_HEAD (_sptree, secpolicy)
- SYSCTL_INT (_net_key, KEYCTL_DEBUG_LEVEL, debug, CTLFLAG_RW,&key_debug_level, 0,"")
- SYSCTL_INT (_net_key, KEYCTL_SPI_TRY, spi_trycnt, CTLFLAG_RW,&key_spi_trycnt, 0,"")
- SYSCTL_INT (_net_key, KEYCTL_SPI_MIN_VALUE, spi_minval, CTLFLAG_RW,&key_spi_minval, 0,"")
- SYSCTL_INT (_net_key, KEYCTL_SPI_MAX_VALUE, spi_maxval, CTLFLAG_RW,&key_spi_maxval, 0,"")
- SYSCTL_INT (_net_key, KEYCTL_RANDOM_INT, int_random, CTLFLAG_RW,&key_int_random, 0,"")
- SYSCTL_INT (_net_key, KEYCTL_LARVAL_LIFETIME, larval_lifetime, CTLFLAG_RW,&key_larval_lifetime, 0,"")
- SYSCTL_INT (_net_key, KEYCTL_BLOCKACQ_COUNT, blockacq_count, CTLFLAG_RW,&key_blockacq_count, 0,"")
- SYSCTL_INT (_net_key, KEYCTL_BLOCKACQ_LIFETIME, blockacq_lifetime, CTLFLAG_RW,&key_blockacq_lifetime, 0,"")
- SYSCTL_INT (_net_key, KEYCTL_ESP_AUTH, esp_auth, CTLFLAG_RW,&ipsec_esp_auth, 0,"")
- SYSCTL_INT (_net_key, KEYCTL_ESP_KEYMIN, esp_keymin, CTLFLAG_RW,&ipsec_esp_keymin, 0,"")
- SYSCTL_INT (_net_key, KEYCTL_AH_KEYMIN, ah_keymin, CTLFLAG_RW,&ipsec_ah_keymin, 0,"")
- SYSCTL_INT (_net_key, KEYCTL_PREFERED_OLDSA, preferred_oldsa, CTLFLAG_RW,&key_preferred_oldsa, 0,"")
- MALLOC_DEFINE (M_IPSEC_SA,"secasvar","ipsec security association")
- MALLOC_DEFINE (M_IPSEC_SAH,"sahead","ipsec sa head")
- MALLOC_DEFINE (M_IPSEC_SP,"ipsecpolicy","ipsec security policy")
- MALLOC_DEFINE (M_IPSEC_SR,"ipsecrequest","ipsec security request")
- MALLOC_DEFINE (M_IPSEC_MISC,"ipsec-misc","ipsec miscellaneous")
- MALLOC_DEFINE (M_IPSEC_SAQ,"ipsec-saq","ipsec sa acquire")
- MALLOC_DEFINE (M_IPSEC_SAR,"ipsec-reg","ipsec sa acquire")
- static struct secasvar *key_allocsa_policy __P ((const struct secasindex *))
- static void key_freesp_so __P ((struct secpolicy **))
- static struct secasvar *key_do_allocsa_policy __P ((struct secashead *, u_int))
- static void key_delsp __P ((struct secpolicy *))
- static struct secpolicy *key_getsp __P ((struct secpolicyindex *))
- static void _key_delsp (struct secpolicy *sp)
- static struct secpolicy *key_getspbyid __P ((u_int32_t))
- static u_int32_t key_newreqid __P ((void))
- static struct mbuf *key_gather_mbuf __P ((struct mbuf *, const struct sadb_msghdr *, int, int,...))
- static int key_spdadd __P ((struct socket *, struct mbuf *, const struct sadb_msghdr *))
- static struct mbuf *key_setdumpsp __P ((struct secpolicy *, u_int8_t, u_int32_t, u_int32_t))
- static struct secashead *key_newsah __P ((struct secasindex *))
- static void key_delsah __P ((struct secashead *))
- static struct secasvar *key_newsav __P ((struct mbuf *, const struct sadb_msghdr *, struct secashead *, int *, const char *, int))
- static void key_delsav __P ((struct secasvar *))
- static struct secasvar *key_checkspidup __P ((struct secasindex *, u_int32_t))
- static struct secasvar *key_getsavbyspi __P ((struct secashead *, u_int32_t))

- static int key_setsaval __P ((struct secasvar *, struct mbuf *, const struct sadb_msghdr *))
- static struct mbuf *key_setdumpsa __P ((struct secasvar *, u_int8_t, u_int8_t, u_int32_t, u_int32_t))
- static struct mbuf *key_setsadbmsg __P ((u_int8_t, u_int16_t, u_int8_t, u_int32_t, pid_t, u_int16_t))
- static struct mbuf *key_setsadbaddr __P ((u_int16_t, const struct sockaddr *, u_int8_t, u_int16_t))
- static struct mbuf *key_setsadbxsa2 __P ((u_int8_t, u_int32_t, u_int32_t))
- static struct mbuf *key_setsadbxpolicy __P ((u_int16_t, u_int8_t, u_int32_t))
- static struct seckey * key_dup_keymsg (const struct sadb_key *, u_int, struct malloc_type *)
- static struct seclifetime * key_dup_lifemsg (const struct sadb_lifetime *src, struct malloc_type *type)
- static int key_cmpsaidx __P ((const struct secasindex *, const struct secasindex *, int))
- static int key_cmpspidx_exactly __P ((struct secpolicyindex *, struct secpolicyindex *))
- static int key_sockaddrcmp __P ((const struct sockaddr *, const struct sockaddr *, int))
- static int key_bbcmp __P ((const void *, const void *, u_int))
- static u_int16_t key_satype2proto __P ((u_int8_t))
- static u_int8_t key_proto2satype __P ((u_int16_t))
- static u_int32_t key_do_getnewspi __P ((struct sadb_spirange *, struct secasindex *))
- static int key_setident __P ((struct secashead *, struct mbuf *, const struct sadb_msghdr *))
- static struct mbuf *key_getmsgbuf_x1 __P ((struct mbuf *, const struct sadb_msghdr *))
- static void key_getcomb_setlifetime __P ((struct sadb_comb *))
- static int key_acquire __P ((const struct secasindex *, struct secpolicy *))
- static int key_senderror __P ((struct socket *, struct mbuf *, int))
- static int key_validate_ext __P ((const struct sadb_ext *, int))
- static int key_align __P ((struct mbuf *, struct sadb_msghdr *))
- static struct mbuf * key_setlifetime (struct seclifetime *src, u_int16_t exttype)
- static struct mbuf * key_setkey (struct seckey *src, u_int16_t exttype)
- static void key_sa_chgstate __P ((struct secasvar *, u_int8_t))
- static struct mbuf *key_alloc_mbuf __P ((int))
- static __inline void sa_initref (struct secasvar *sav)
- static __inline void sa_addref (struct secasvar *sav)
- static __inline int sa_delref (struct secasvar *sav)
- void key_addref (struct secpolicy *sp)
- int key_havesp (u_int dir)
- secpolicy * key_allocsp (struct secpolicyindex *spidx, u_int dir, const char *where, int tag)
- secpolicy * key_allocsp2 (u_int32_t spi, union sockaddr_union *dst, u_int8_t proto, u_int dir, const char *where, int tag)
- secpolicy * key_gettunnel (const struct sockaddr *osrc, const struct sockaddr *odst, const struct sockaddr *isrc, const struct sockaddr *idst, const char *where, int tag)
- int key_checkrequest (struct ipsecrequest *isr, const struct secasindex *saidx)
- static struct secasvar * key_allocsa_policy (const struct secasindex *saidx)
- static struct secasvar * key_do_allocsa_policy (struct secashead *sah, u_int state)
- secasvar * key_allocsa (union sockaddr_union *dst, u_int proto, u_int32_t spi, const char *where, int tag)
- void _key_freesp (struct secpolicy **spp, const char *where, int tag)
- void key_freeso (struct socket *so)
- static void key_freesp_so (struct secpolicy **sp)
- void key_freesav (struct secasvar **psav, const char *where, int tag)
- static void key_delsp (struct secpolicy *sp)
- static struct secpolicy * key_getsp (struct secpolicyindex *spidx)
- static struct secpolicy * key_getspbyid (u_int32_t id)
- secpolicy * key_newsp (const char *where, int tag)

- secpolicy * key_msg2sp (struct sadb_x_policy *xpl0, size_t len, int *error)
- static u_int32_t key_newreqid ()
- mbuf * key_sp2msg (struct secpolicy *sp)
- static struct mbuf * key_gather_mbuf (struct mbuf *m, const struct sadb_msghdr *mhp, int ndeep, int nitem, va_alist)
- static int key_spdadd (struct socket *so, struct mbuf *m, const struct sadb_msghdr *mhp)
- static u_int32_t key_getnewspid ()
- static int key_spddelete (struct socket *so, struct mbuf *m, const struct sadb_msghdr *mhp)
- static int key_spddelete2 (struct socket *so, struct mbuf *m, const struct sadb_msghdr *mhp)
- static int key_spdget (struct socket *so, struct mbuf *m, const struct sadb_msghdr *mhp)
- int key_spdacquire (struct secpolicy *sp)
- static int key_spdflush (struct socket *so, struct mbuf *m, const struct sadb_msghdr *mhp)
- static int key_spddump (struct socket *so, struct mbuf *m, const struct sadb_msghdr *mhp)
- static struct mbuf * key_setdumpsp (struct secpolicy *sp, u_int8_t type, u_int32_t seq, u_int32_t pid)
- static u_int key_getspreqmsglen (struct secpolicy *sp)
- static int key_spdexpire (struct secpolicy *sp)
- static struct secashead * key_newsah (struct secasindex *saidx)
- static void key_delsah (struct secashead *sah)
- static struct secasvar * key_newsav (struct mbuf *m, const struct sadb_msghdr *mhp, struct secashead *sah, int *errp, const char *where, int tag)
- static void key_cleansav (struct secasvar *sav)
- static void key_delsav (struct secasvar *sav)
- static struct secashead * key_getsah (struct secasindex *saidx)
- static struct secasvar * key_checkspidup (struct secasindex *saidx, u_int32_t spi)
- static struct secasvar * key_getsavbyspi (struct secashead *sah, u_int32_t spi)
- static int key_setsaval (struct secasvar *sav, struct mbuf *m, const struct sadb_msghdr *mhp)
- static int key_mature (struct secasvar *sav)
- static struct mbuf * key_setdumpsa (struct secasvar *sav, u_int8_t type, u_int8_t satype, u_int32_t seq, u_int32_t pid)
- static struct mbuf * key_setsadbmsg (u_int8_t type, u_int16_t tlen, u_int8_t satype, u_int32_t seq, pid_t pid, u_int16_t reserved)
- static struct mbuf * key_setsadbsa (struct secasvar *sav)
- static struct mbuf * key_setsadbaddr (u_int16_t exttype, const struct sockaddr *saddr, u_int8_t prefixlen, u_int16_t ul_proto)
- static struct mbuf * key_setsadbxsa2 (u_int8_t mode, u_int32_t seq, u_int32_t reqid)
- static struct mbuf * key_setsadbxpolicy (u_int16_t type, u_int8_t dir, u_int32_t id)
- int key_ismyaddr (struct sockaddr *sa)
- static int key_cmpsaidx (const struct secasindex *saidx0, const struct secasindex *saidx1, int flag)
- static int key_cmpspidx_exactly (struct secpolicyindex *spidx0, struct secpolicyindex *spidx1)
- static int key_cmpspidx_withmask (struct secpolicyindex *spidx0, struct secpolicyindex *spidx1)
- static int key_sockaddrcmp (const struct sockaddr *sa1, const struct sockaddr *sa2, int port)
- static int key_bbcmp (const void *a1, const void *a2, u_int bits)
- static void key_flush_spd (time_t now)
- static void key_flush_sad (time_t now)
- static void key_flush_acq (time_t now)
- static void key_flush_spacq (time_t now)
- void key_timehandler (void)
- u_long key_random ()
- void key_randomfill (void *p, size_t l)

- static u_int16_t key_satype2proto (u_int8_t satype)
- static u_int8_t key_proto2satype (u_int16_t proto)
- static int key_getspi (struct socket *so, struct mbuf *m, const struct sadb_msghdr *mhp)
- static u_int32_t key_do_getnewspi (struct sadb_spirange *spirange, struct secasindex *saidx)
- static int key_update (struct socket *so, struct mbuf *m, const struct sadb_msghdr *mhp)
- static int key_add (struct socket *so, struct mbuf *m, const struct sadb_msghdr *mhp)
- static int key_setident (struct secashead *sah, struct mbuf *m, const struct sadb_msghdr *mhp)
- static struct mbuf * key_getmsgbuf_x1 (struct mbuf *m, const struct sadb_msghdr *mhp)
- static int key_delete_all __P ((struct socket *, struct mbuf *, const struct sadb_msghdr *, u_int16_t))
- static int key_delete (struct socket *so, struct mbuf *m, const struct sadb_msghdr *mhp)
- static int key_delete_all (struct socket *so, struct mbuf *m, const struct sadb_msghdr *mhp, u_int16_t proto)
- static int key_get (struct socket *so, struct mbuf *m, const struct sadb_msghdr *mhp)
- static void key_getcomb_setlifetime (struct sadb_comb *comb)
- static struct mbuf * key_getcomb_esp ()
- static void key_getsizes_ah (const struct auth_hash *ah, int alg, u_int16_t *min, u_int16_t *max)
- static struct mbuf * key_getcomb_ah ()
- static struct mbuf * key_getcomb_ipcomp ()
- static struct mbuf * key_getprop (struct secasindex *saidx) const
- static int key_acquire (const struct secasindex *saidx, struct secpolicy *sp)
- static struct secacq * key_newacq (const struct secasindex *saidx)
- static struct secacq * key_getacq (const struct secasindex *saidx)
- static struct secacq * key_getacqbyseq (u_int32_t seq)
- static struct secspacq * key_newspacq (struct secpolicyindex *spidx)
- static struct secspacq * key_getspacq (struct secpolicyindex *spidx)
- static int key_acquire2 (struct socket *so, struct mbuf *m, const struct sadb_msghdr *mhp)
- static int key_register (struct socket *so, struct mbuf *m, const struct sadb_msghdr *mhp)
- void key_freereg (struct socket *so)
- static int key_expire (struct secasvar *sav)
- static int key_flush (struct socket *so, struct mbuf *m, const struct sadb_msghdr *mhp)
- static int key_dump (struct socket *so, struct mbuf *m, const struct sadb_msghdr *mhp)
- static int key_promisc (struct socket *so, struct mbuf *m, const struct sadb_msghdr *mhp)
- int key_parse (struct mbuf *m, struct socket *so)
- static int key_senderror (struct socket *so, struct mbuf *m, int code)
- static int key_align (struct mbuf *m, struct sadb_msghdr *mhp)
- static int key_validate_ext (struct sadb_ext *ext, int len) const
- void key_init ()
- int key_checktunnelsanity (struct secasvar *sav, u_int family, caddr_t src, caddr_t dst)
- void key_sa_recordxfer (struct secasvar *sav, struct mbuf *m)
- void key_sa_routechange (struct sockaddr *dst)
- static void key_sa_chgstate (struct secasvar *sav, u_int8_t state)
- void key_sa_stir_iv (struct secasvar *sav)
- static struct mbuf * key_alloc_mbuf (int l)

Variables

- u_int32_t key_debug_level = 0
- static u_int key_spi_trycnt = 1000
- static u_int32_t key_spi_minval = 0x100
- static u_int32_t key_spi_maxval = 0x0fffffff
- static u_int32_t policy_id = 0
- static u_int key_int_random = 60
- static u_int key_larval_lifetime = 30
- static int key_blockacq_count = 10
- static int key_blockacq_lifetime = 20
- static int key_preferred_oldsa = 1
- static u_int32_t acq_seq = 0
- static const u_int saorder_state_valid_prefer_new [ ]
- static u_int saorder_state_alive [ ]
- static u_int saorder_state_any [ ]
- static const int minsize [ ]
- static const int maxsize [ ]
- static int ipsec_esp_keymin = 256
- static int ipsec_esp_auth = 0
- static int ipsec_ah_keymin = 128
- _keystat keystat

7.16.1 Define Documentation

7.16.1.1 #define __LIST_CHAINED(elm) (!((elm) → chain.le_next == NULL && (elm) → chain.le_prev == NULL))

Definition at line 292 of file key.c.

Referenced by key_delsah(), key_delsav(), key_delsp(), key_flush_acq(), key_flush_spacq(), key_freereg(), and key_sa_chgstate().

7.16.1.2 #define _BITS(bytes) ((bytes) << 3)

Definition at line 101 of file key.c.

Referenced by key_getcomb_ah(), key_getcomb_esp(), and key_register().

7.16.1.3 #define ACQ_LOCK() mtx_lock(&acq_lock)

Referenced by key_flush_acq(), key_getacq(), key_getacqbyseq(), and key_newacq().

7.16.1.4 #define ACQ_LOCK_ASSERT() mtx_assert(&acq_lock, MA_OWNED)

7.16.1.5 #define ACQ_LOCK_DESTROY() mtx_destroy(&acq_lock)

7.16.1.6 #define ACQ_LOCK_INIT() mtx_init(&acq_lock, "acqtree", "fast ipsec acquire list", MTX_DEF)

Referenced by key_init().

7.16.1.7 #define ACQ_UNLOCK() mtx_unlock(&acq_lock)

Referenced by key_flush_acq(), key_getacq(), key_getacqbyseq(), and key_newacq().

7.16.1.8 #define CMP_EXACTLY 4

Definition at line 436 of file key.c.

Referenced by key_cmpsaidx(), and key_getacq().

7.16.1.9 #define CMP_HEAD 1

Definition at line 433 of file key.c.

Referenced by key_delete(), key_delete_all(), and key_get().

7.16.1.10 #define CMP_MODE_REQID 2

Definition at line 434 of file key.c.

Referenced by key_acquire2(), key_allocsa_policy(), and key_cmpsaidx().

7.16.1.11 #define CMP_REQID 3

Definition at line 435 of file key.c.

Referenced by key_cmpsaidx(), and key_getsah().

7.16.1.12 #define FULLMASK 0xff

Definition at line 100 of file key.c.

Referenced by key_acquire(), key_expire(), key_setdumpsa(), and key_setsadbaddr().

7.16.1.13 #define KEY_CHKSASTATE(head, sav, name)

Value:

```
do { \
        if ((head) != (sav)) {                                          \
                ipseclog((LOG_DEBUG, "%s: state mismatched (TREE=%d SA=%d)\n", \
                        (name), (head), (sav)));                        \
                continue;                                               \
        }                                                               \
} while (0)
```

Definition at line 306 of file key.c.

Referenced by key_allocsa(), key_delsah(), and key_do_allocsa_policy().

7.16.1.14 #define KEY_CHKSPDIR(head, sp, name)

Value:

```
do { \
        if ((head) != (sp)) {                                           \
                ipseclog((LOG_DEBUG, "%s: direction mismatched (TREE=%d SP=%d), " \
                        "anyway continue.\n",                           \
                        (name), (head), (sp)));                         \
        }                                                               \
} while (0)
```

Definition at line 315 of file key.c.

Referenced by key_allocsp(), and key_allocsp2().

7.16.1.15 #define KEY_NEWSAV(m, sadb, sah, e) key_newsav(m, sadb, sah, e, __FILE__, __LINE__)

Definition at line 405 of file key.c.

Referenced by key_add(), and key_getspi().

7.16.1.16 #define KEY_SETSECASIDX(p, m, r, s, d, idx)

Value:

```
do { \
        bzero((idx), sizeof(struct secasindex));                        \
        (idx)->proto = (p);                                             \
        (idx)->mode = (m);                                              \
        (idx)->reqid = (r);                                             \
        bcopy((s), &(idx)->src, ((const struct sockaddr *)(s))->sa_len);    \
        bcopy((d), &(idx)->dst, ((const struct sockaddr *)(d))->sa_len);    \
} while (0)
```

Definition at line 351 of file key.c.

Referenced by key_acquire2(), key_add(), key_delete(), key_delete_all(), key_get(), key_getspi(), and key_update().

7.16.1.17 #define KEY_SETSECSPIDX(_dir, s, d, ps, pd, ulp, idx)

Value:

```
do { \
        bzero((idx), sizeof(struct secpolicyindex));                    \
        (idx)->dir = (_dir);                                            \
        (idx)->prefs = (ps);                                            \
        (idx)->prefd = (pd);                                            \
        (idx)->ul_proto = (ulp);                                        \
        bcopy((s), &(idx)->src, ((const struct sockaddr *)(s))->sa_len);    \
        bcopy((d), &(idx)->dst, ((const struct sockaddr *)(d))->sa_len);    \
} while (0)
```

Definition at line 336 of file key.c.

Referenced by key_spdadd(), and key_spddelete().

7.16.1.18 #define LIST_INSERT_TAIL(head, elm, type, field)

Value:

```
do {\
        struct type *curelm = LIST_FIRST(head); \
        if (curelm == NULL) {\
                LIST_INSERT_HEAD(head, elm, field); \
        } else { \
                while (LIST_NEXT(curelm, field)) \
                        curelm = LIST_NEXT(curelm, field);\
                LIST_INSERT_AFTER(curelm, elm, field);\
        }\
} while (0)
```

Definition at line 294 of file key.c.

Referenced by key_newsav(), and key_spdadd().

7.16.1.19 #define N(a) _ARRAYLEN(a)

Referenced by key_allocsa_policy().

7.16.1.20 #define REGTREE_LOCK() mtx_lock(&regtree_lock)

Referenced by key_freereg(), and key_register().

7.16.1.21 #define REGTREE_LOCK_ASSERT() mtx_assert(&regtree_lock, MA_OWNED)

7.16.1.22 #define REGTREE_LOCK_DESTROY() mtx_destroy(&regtree_lock)

7.16.1.23 #define REGTREE_LOCK_INIT() mtx_init(&regtree_lock, "regtree", "fast ipsec regtree", MTX_DEF)

Referenced by key_init().

7.16.1.24 #define REGTREE_UNLOCK() mtx_unlock(&regtree_lock)

Referenced by key_freereg(), and key_register().

7.16.1.25 #define SAHTREE_LOCK() mtx_lock(&sahtree_lock)

Referenced by key_acquire2(), key_add(), key_allocsa(), key_allocsa_policy(), key_checkspidup(), key_-delete(), key_delete_all(), key_do_allocsa_policy(), key_dump(), key_flush(), key_flush_sad(), key_get(), key_getsah(), key_mature(), key_newsah(), key_sa_routechange(), and key_update().

7.16.1.26 #define SAHTREE_LOCK_ASSERT() mtx_assert(&sahtree_lock, MA_OWNED)

Referenced by key_delsah(), key_getsavbyspi(), and key_sa_chgstate().

7.16.1.27 #define SAHTREE_LOCK_DESTROY() mtx_destroy(&sahtree_lock)

7.16.1.28 #define SAHTREE_LOCK_INIT()

Value:

```
mtx_init(&sahtree_lock, "sahtree", \
                "fast ipsec security association database", MTX_DEF)
```

Referenced by key_init().

7.16.1.29 #define SAHTREE_UNLOCK() mtx_unlock(&sahtree_lock)

Referenced by key_acquire2(), key_add(), key_allocsa(), key_allocsa_policy(), key_checkspidup(), key_-delete(), key_delete_all(), key_do_allocsa_policy(), key_dump(), key_flush(), key_flush_sad(), key_get(), key_getsah(), key_mature(), key_newsah(), key_sa_routechange(), and key_update().

7.16.1.30 #define satosin(s) ((const struct sockaddr_in *)s)

Referenced by key_sockaddrcmp().

7.16.1.31 #define satosin6(s) ((const struct sockaddr_in6 *)s)

Referenced by key_sockaddrcmp().

7.16.1.32 #define SP_ADDREF(p)

Value:

```
do {                                            \
        (p)->refcnt++;                                                  \
        IPSEC_ASSERT((p)->refcnt != 0, ("SP refcnt overflow"));         \
} while (0)
```

Definition at line 527 of file key.c.

Referenced by key_addref(), key_allocsp(), key_allocsp2(), key_getsp(), key_getspbyid(), and key_-gettunnel().

7.16.1.33 #define SP_DELREF(p)

Value:

```
do {                                            \
        IPSEC_ASSERT((p)->refcnt > 0, ("SP refcnt underflow"));         \
        (p)->refcnt--;                                                  \
} while (0)
```

Definition at line 531 of file key.c.

Referenced by _key_freesp().

7.16.1.34 #define SPACQ_LOCK() mtx_lock(&spacq_lock)

Referenced by key_flush_spacq(), key_getspacq(), and key_newspacq().

7.16.1.35 #define SPACQ_LOCK_ASSERT() mtx_assert(&spacq_lock, MA_OWNED)

7.16.1.36 #define SPACQ_LOCK_DESTROY() mtx_destroy(&spacq_lock)

7.16.1.37 #define SPACQ_LOCK_INIT()

Value:

```
mtx_init(&spacq_lock, "spacqtree", \
                "fast ipsec security policy acquire list", MTX_DEF)
```

Referenced by key_init().

7.16.1.38 #define SPACQ_UNLOCK() mtx_unlock(&spacq_lock)

Referenced by key_flush_spacq(), key_getspacq(), key_newspacq(), key_spdacquire(), and key_spdadd().

7.16.1.39 #define SPTREE_LOCK() mtx_lock(&sptree_lock)

Referenced by _key_freesp(), key_addref(), key_allocsp(), key_allocsp2(), key_flush_spd(), key_getsp(), key_getspbyid(), key_gettunnel(), and key_spdflush().

7.16.1.40 #define SPTREE_LOCK_ASSERT() mtx_assert(&sptree_lock, MA_OWNED)

Referenced by key_delsp().

7.16.1.41 #define SPTREE_LOCK_DESTROY() mtx_destroy(&sptree_lock)

7.16.1.42 #define SPTREE_LOCK_INIT()

Value:

```
mtx_init(&sptree_lock, "sptree", \
                "fast ipsec security policy database", MTX_DEF)
```

Referenced by key_init().

7.16.1.43 #define SPTREE_UNLOCK() mtx_unlock(&sptree_lock)

Referenced by _key_freesp(), key_addref(), key_allocsp(), key_allocsp2(), key_flush_spd(), key_getsp(), key_getspbyid(), key_gettunnel(), and key_spdflush().

7.16.2 Function Documentation

7.16.2.1 static int key_delete_all __P ((struct socket *, struct mbuf *, const struct sadb_msghdr *, u_int16_t)) [static]

7.16.2.2 static struct mbuf* key_alloc_mbuf __P ((int)) [static]

7.16.2.3 static void key_sa_chgstate __P ((struct secasvar *, u_int8_t)) [static]

7.16.2.4 static int key_align __P ((struct mbuf *, struct sadb_msghdr *)) [static]

7.16.2.5 static int key_validate_ext __P ((const struct sadb_ext *, int)) [static]

7.16.2.6 static int key_senderror __P ((struct socket *, struct mbuf *, int)) [static]

7.16.2.7 static int key_acquire __P ((const struct secasindex *, struct secpolicy *)) [static]

7.16.2.8 static void key_getcomb_setlifetime __P ((struct sadb_comb *)) [static]

7.16.2.9 static struct mbuf* key_getmsgbuf_x1 __P ((struct mbuf *, const struct sadb_msghdr *)) [static]

7.16.2.10 static int key_setident __P ((struct secashead *, struct mbuf *, const struct sadb_msghdr *)) [static]

7.16.2.11 static u_int32_t key_do_getnewspi __P ((struct sadb_spirange *, struct secasindex *)) [static]

7.16.2.12 static u_int8_t key_proto2satype __P ((u_int16_t)) [static]

7.16.2.13 static u_int16_t key_satype2proto __P ((u_int8_t)) [static]

7.16.2.14 static int key_bbcmp __P ((const void *, const void *, u_int)) [static]

7.16.2.15 static int key_sockaddrcmp __P ((const struct sockaddr *, const struct sockaddr *, int)) [static]

7.16.2.16 static int key_cmpspidx_withmask __P ((struct secpolicyindex *, struct secpolicyindex *)) [static]

7.16.2.17 static int key_cmpsaidx __P ((const struct secasindex *, const struct secasindex *, int)) [static]

7.16.2.18 static struct mbuf* key_setsadbxpolicy __P ((u_int16_t, u_int8_t, u_int32_t)) [static]

7.16.2.19 static struct mbuf* key_setsadbxsa2 __P ((u_int8_t, u_int32_t, u_int32_t)) [static]

7.16.2.20 static struct mbuf* key_setsadbaddr __P ((u_int16_t, const struct sockaddr *, u_int8_t, u_int16_t)) [static]

7.16.2.21 static struct mbuf* key_setsadbmsg __P ((u_int8_t, u_int16_t, u_int8_t, u_int32_t, pid_t, u_int16_t)) [static]

7.16.2.22 static struct mbuf* key_setdumpsa __P ((struct secasvar *, u_int8_t, u_int8_t, u_int32_t, u_int32_t)) [static]

7.16.2.23 static int key_setsaval __P ((struct secasvar *, struct mbuf *, const struct sadb_msghdr *)) [static]

7.16.2.24 static struct secasvar* key_getsavbyspi __P ((struct secashead *, u_int32_t))

```
{
        NULL,
        key_getspi,
        key_update,
        key_add,
        key_delete,
        key_get,
        key_acquire2,
        key_register,
        NULL,
        key_flush,
        key_dump,
        key_promisc,
        NULL,
        key_spdadd,
        key_spdadd,
        key_spddelete,
        key_spdget,
        NULL,
        key_spddump,
        key_spdflush,
        key_spdadd,
        NULL,
        key_spddelete2,
}
```

7.16.2.32 static struct mbuf* key_gather_mbuf __P ((struct mbuf *, const struct sadb_msghdr *, int, int,...)) [static]

7.16.2.33 static struct mbuf *key_getcomb_ipcomp __P ((void)) [static]

7.16.2.34 static struct secacq *key_getacqbyseq __P ((u_int32_t)) [static]

7.16.2.35 static struct secspacq *key_getspacq __P ((struct secpolicyindex *)) [static]

7.16.2.36 static int key_spdexpire __P ((struct secpolicy *)) [static]

7.16.2.37 static struct secasvar* key_do_allocsa_policy __P ((struct secashead *, u_int)) [static]

7.16.2.38 static void key_freesp_so __P ((struct secpolicy **)) [static]

7.16.2.39 static struct secacq *key_getacq __P ((const struct secasindex *)) [static]

7.16.2.40 static void _key_delsp (struct secpolicy * *sp*) [static]

Definition at line 1340 of file key.c.

References SECPOLICY_LOCK_DESTROY.

Referenced by key_delsp(), and key_spdadd().

7.16.2.41 void _key_freesp (struct secpolicy ** *spp*, const char * *where*, int *tag*)

Definition at line 1115 of file key.c.

References secpolicy::id, IPSEC_ASSERT, key_delsp(), KEYDEBUG, KEYDEBUG_IPSEC_STAMP, secpolicy::refcnt, SP_DELREF, SPTREE_LOCK, and SPTREE_UNLOCK.

Here is the call graph for this function:

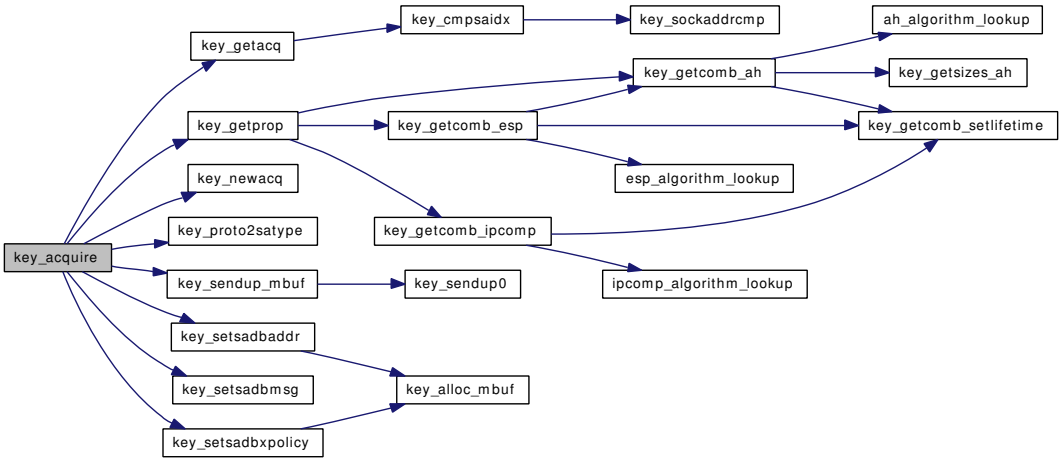
7.16.2.42 static int key_acquire (const struct secasindex * *saidx*, struct secpolicy * *sp*) [static]

Definition at line 5677 of file key.c.

References secacq::count, secpolicyindex::dir, secasindex::dst, FULLMASK, secpolicy::id, IPSEC_-ASSERT, IPSEC_ULPROTO_ANY, key_getacq(), key_getprop(), key_newacq(), key_proto2satype(), key_sendup_mbuf(), KEY_SENDUP_REGISTERED, key_setsadbaddr(), key_setsadbmsg(), key_-setsadbxpolicy(), secpolicy::policy, secasindex::proto, sockaddr_union::sa, secacq::saidx, secacq::seq, secpolicy::spidx, and secasindex::src.

Referenced by key_acquire2(), and key_checkrequest().

Here is the call graph for this function:

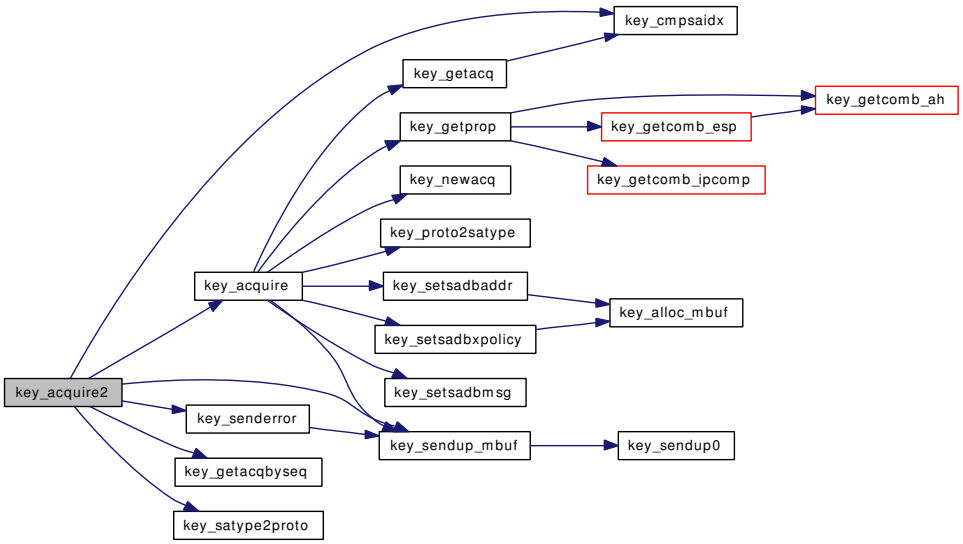


7.16.2.43 static int key_acquire2 (struct socket * *so*, struct mbuf * *m*, const struct sadb_msghdr * *mhp*) [static]

Definition at line 5957 of file key.c.

References CMP_MODE_REQID, secacq::count, secacq::created, sadb_msghdr::ext, sadb_-msghdr::extlen, IPSEC_ASSERT, IPSEC_MODE_ANY, ipseclog, key_acquire(), key_cmpsaidx(), key_getacqbyseq(), key_satype2proto(), key_senderror(), key_sendup_mbuf(), KEY_SENDUP_-REGISTERED, KEY_SETSECASIDX, sadb_msghdr::msg, SAHTREE_LOCK, SAHTREE_UNLOCK, secashead::saidx, and secashead::state.

Here is the call graph for this function:

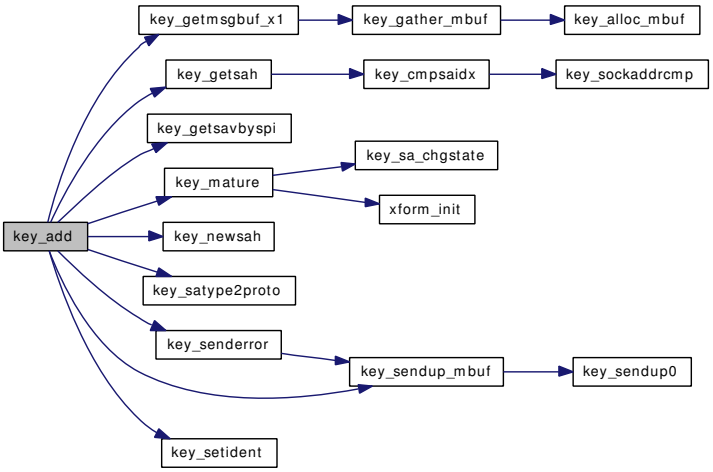


7.16.2.44 static int key_add (struct socket * *so*, struct mbuf * *m*, const struct sadb_msghdr * *mhp*) [static]

Definition at line 4866 of file key.c.

References sadb_msghdr::ext, sadb_msghdr::extlen, IPSEC_ASSERT, IPSEC_MODE_ANY, ipseclog, KEY_FREESAV, key_getmsgbuf_x1(), key_getsah(), key_getsavbyspi(), key_mature(), key_newsah(), KEY_NEWSAV, key_satype2proto(), key_senderror(), KEY_SENDUP_ALL, key_sendup_mbuf(), key_setident(), KEY_SETSECASIDX, sadb_msghdr::msg, SAHTREE_LOCK, and SAHTREE_UNLOCK.

Here is the call graph for this function:



7.16.2.45 void key_addref (struct secpolicy * *sp*)

Definition at line 541 of file key.c.

References SP_ADDREF, SPTREE_LOCK, and SPTREE_UNLOCK.

Referenced by ipsec_getpolicybysock(), and key_allocsp_default().

7.16.2.46 static int key_align (struct mbuf * *m*, struct sadb_msghdr * *mhp*) [static]

Definition at line 6933 of file key.c.

References IPSEC_ASSERT, and ipseclog.

Referenced by key_parse().

7.16.2.47 static struct mbuf* key_alloc_mbuf (int *l*) [static]

Definition at line 7243 of file key.c.

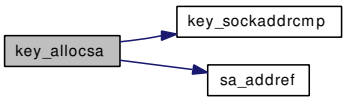
Referenced by key_expire(), key_gather_mbuf(), key_setkey(), key_setlifetime(), key_setsadbaddr(), key_setsadbsa(), key_setsadbxpolicy(), key_setsadbxsa2(), key_sp2msg(), and key_spdexpire().

7.16.2.48 struct secasvar* key_allocsa (union sockaddr_union * *dst*, u_int *proto*, u_int32_t *spi*, const char * *where*, int *tag*)

Definition at line 1042 of file key.c.

References _ARRAYLEN, IPSEC_ASSERT, KEY_CHKSASTATE, key_preferred_oldsa, key_sockaddrcmp(), KEYDEBUG, KEYDEBUG_IPSEC_STAMP, secasvar::refcnt, sockaddr_union::sa, sa_addref(), secasvar::sah, SAHTREE_LOCK, SAHTREE_UNLOCK, and secasvar::state.

Here is the call graph for this function:



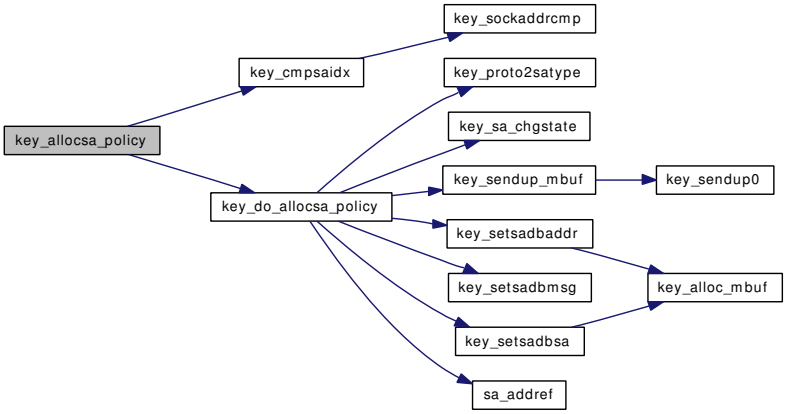
7.16.2.49 static struct secasvar* key_allocsa_policy (const struct secasindex * *saidx*) [static]

Definition at line 853 of file key.c.

References CMP_MODE_REQID, key_cmpsaidx(), key_do_allocsa_policy(), key_preferred_oldsa, N, secasvar::sah, SAHTREE_LOCK, SAHTREE_UNLOCK, secashead::saidx, and secashead::state.

Referenced by key_checkrequest().

Here is the call graph for this function:

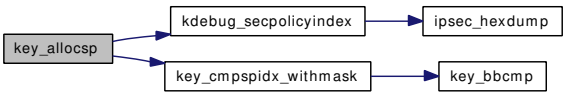


7.16.2.50 struct secpolicy* key_allocsp (struct secpolicyindex * *spidx*, u_int *dir*, const char * *where*, int *tag*)

Definition at line 568 of file key.c.

References secpolicyindex::dir, secpolicy::id, IPSEC_ASSERT, IPSEC_DIR_INBOUND, IPSEC_DIR_OUTBOUND, IPSEC_SPSTATE_DEAD, kdebug_secpolicyindex(), KEY_CHKSPDIR, key_cmpspidx_withmask(), KEYDEBUG, KEYDEBUG_IPSEC_DATA, KEYDEBUG_IPSEC_STAMP, secpolicy::lastused, secpolicy::refcnt, SP_ADDREF, secpolicy::spidx, SPTREE_LOCK, SPTREE_UNLOCK, and secpolicy::state.

Here is the call graph for this function:

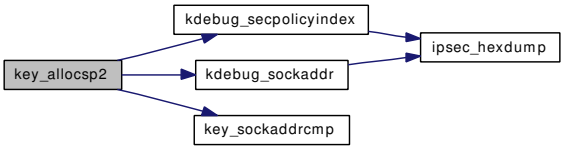


7.16.2.51 struct secpolicy* key_allocsp2 (u_int32_t *spi*, union sockaddr_union * *dst*, u_int8_t *proto*, u_int *dir*, const char * *where*, int *tag*)

Definition at line 620 of file key.c.

References secpolicyindex::dir, secpolicyindex::dst, secpolicy::id, IPSEC_ASSERT, IPSEC_DIR_INBOUND, IPSEC_DIR_OUTBOUND, IPSEC_SPSTATE_DEAD, kdebug_secpolicyindex(), kdebug_sockaddr(), KEY_CHKSPDIR, key_sockaddrcmp(), KEYDEBUG, KEYDEBUG_IPSEC_DATA, KEYDEBUG_IPSEC_STAMP, secpolicy::lastused, secpolicy::refcnt, secpolicy::req, sockaddr_union::sa, ipsecrequest::sav, SP_ADDREF, secasvar::spi, secpolicy::spidx, SPTREE_LOCK, SPTREE_UNLOCK, secpolicy::state, and secpolicyindex::ul_proto.

Here is the call graph for this function:



7.16.2.52 static int key_bbcmp (const void * *a1*, const void * *a2*, u_int *bits*) [static]

Definition at line 4032 of file key.c.

Referenced by key_cmpspidx_withmask().

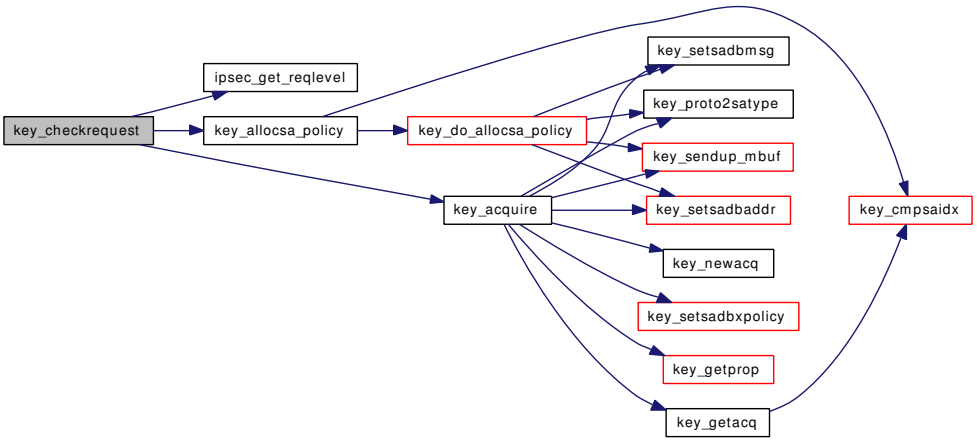
7.16.2.53 int key_checkrequest (struct ipsecrequest * *isr*, const struct secasindex * *saidx*)

Definition at line 759 of file key.c.

References IPSEC_ASSERT, ipsec_get_reqlevel(), IPSEC_LEVEL_REQUIRE, IPSEC_MODE_TRANSPORT, IPSEC_MODE_TUNNEL, ipseclog, IPSECREQUEST_LOCK_ASSERT, key_acquire(), key_allocsa_policy(), KEY_FREESAV, secasindex::mode, secasvar::sah, ipsecrequest::sav, ipsecrequest::sp, and secasvar::state.

Referenced by ipsec_nextisr().

Here is the call graph for this function:



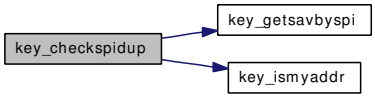
7.16.2.54 static struct secasvar* key_checkspidup (struct secasindex * *saidx*, u_int32_t *spi*) [static]

Definition at line 2899 of file key.c.

References secasindex::dst, ipseclog, key_getsavbyspi(), key_ismyaddr(), secasvar::sah, SAHTREE_LOCK, SAHTREE_UNLOCK, and secashead::saidx.

Referenced by key_do_getnewspi().

Here is the call graph for this function:



7.16.2.55 int key_checktunnelsanity (struct secasvar * *sav*, u_int *family*, caddr_t *src*, caddr_t *dst*)

Definition at line 7140 of file key.c.

References IPSEC_ASSERT.

7.16.2.56 static void key_cleansav (struct secasvar * *sav*) [static]

Definition at line 2798 of file key.c.

References _KEYLEN, secasvar::iv, secasvar::key_auth, seckey::key_data, secasvar::key_enc, secasvar::lft_c, secasvar::lft_h, secasvar::lft_s, secasvar::replay, secasvar::sched, secasvar::schedlen, secasvar::tdb_xform, and xformsw::xf_zeroize.

Referenced by key_delsav(), and key_setsaval().

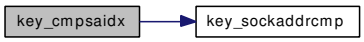
7.16.2.57 static int key_cmpsaidx (const struct secasindex * *saidx0*, const struct secasindex * *saidx1*, int *flag*) [static]

Definition at line 3782 of file key.c.

References CMP_EXACTLY, CMP_MODE_REQID, CMP_REQID, secasindex::dst, IPSEC_MODE_ANY, key_sockaddrcmp(), secasindex::mode, secasindex::proto, secasindex::reqid, sockaddr_union::sa, and secasindex::src.

Referenced by key_acquire2(), key_allocsa_policy(), key_delete(), key_delete_all(), key_get(), key_getacq(), and key_getsah().

Here is the call graph for this function:



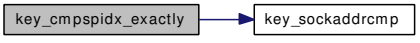
7.16.2.58 static int key_cmpspidx_exactly (struct secpolicyindex * *spidx0*, struct secpolicyindex * *spidx1*) [static]

Definition at line 3845 of file key.c.

References secpolicyindex::dst, key_sockaddrcmp(), secpolicyindex::prefd, secpolicyindex::prefs, sockaddr_union::sa, secpolicyindex::src, and secpolicyindex::ul_proto.

Referenced by key_getsp(), and key_getspacq().

Here is the call graph for this function:



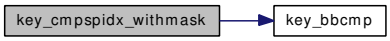
7.16.2.59 static int key_cmpspidx_withmask (struct secpolicyindex * *spidx0*, struct secpolicyindex * *spidx1*) [static]

Definition at line 3875 of file key.c.

References secpolicyindex::dst, IPSEC_PORT_ANY, IPSEC_ULPROTO_ANY, key_bbcmp(), secpolicyindex::prefd, secpolicyindex::prefs, sockaddr_union::sa, sockaddr_union::sin, sockaddr_union::sin6, secpolicyindex::src, and secpolicyindex::ul_proto.

Referenced by key_allocsp(), and key_gettunnel().

Here is the call graph for this function:

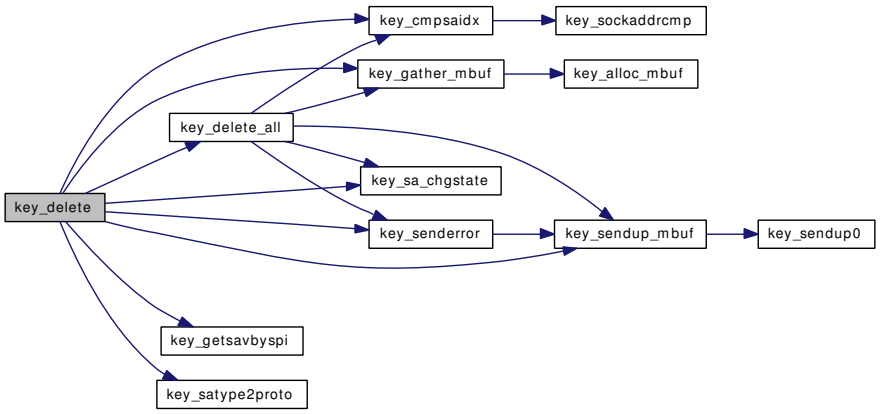


7.16.2.60 static int key_delete (struct socket * *so*, struct mbuf * *m*, const struct sadb_msghdr * *mhp*) [static]

Definition at line 5111 of file key.c.

References CMP_HEAD, sadb_msghdr::ext, sadb_msghdr::extlen, IPSEC_ASSERT, IPSEC_MODE_ANY, ipseclog, key_cmpsaidx(), key_delete_all(), KEY_FREESAV, key_gather_mbuf(), key_getsavbyspi(), key_sa_chgstate(), key_satype2proto(), key_senderror(), KEY_SENDUP_ALL, key_sendup_mbuf(), KEY_SETSECASIDX, sadb_msghdr::msg, SAHTREE_LOCK, SAHTREE_UNLOCK, secashead::saidx, and secashead::state.

Here is the call graph for this function:



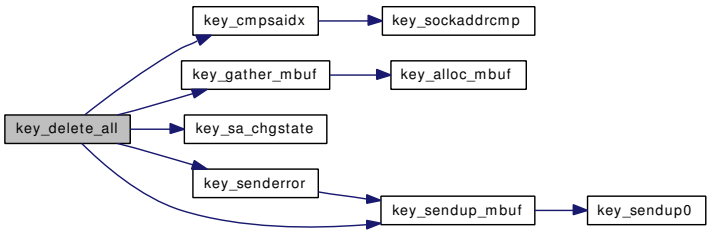
7.16.2.61 static int key_delete_all (struct socket * *so*, struct mbuf * *m*, const struct sadb_msghdr * *mhp*, u_int16_t *proto*) [static]

Definition at line 5221 of file key.c.

References _ARRAYLEN, CMP_HEAD, sadb_msghdr::ext, IPSEC_MODE_ANY, ipseclog, key_cmpsaidx(), KEY_FREESAV, key_gather_mbuf(), key_sa_chgstate(), key_senderror(), KEY_SENDUP_ALL, key_sendup_mbuf(), KEY_SETSECASIDX, SAHTREE_LOCK, SAHTREE_UNLOCK, secashead::saidx, secasvar::state, and secashead::state.

Referenced by key_delete().

Here is the call graph for this function:



7.16.2.62 static void key_delsah (struct secashead * *sah*) [static]

Definition at line 2652 of file key.c.

References __LIST_CHAINED, _ARRAYLEN, IPSEC_ASSERT, KEY_CHKSASTATE, KEY_-FREESAV, SAHTREE_LOCK_ASSERT, and secasvar::state.

Referenced by key_flush_sad().

7.16.2.63 static void key_delsav (struct secasvar * *sav*) [static]

Definition at line 2853 of file key.c.

References __LIST_CHAINED, IPSEC_ASSERT, key_cleansav(), and SECASVAR_LOCK_DESTROY.

Referenced by key_freesav().

Here is the call graph for this function:

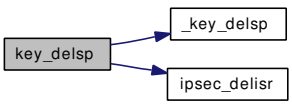
7.16.2.64 static void key_delsp (struct secpolicy * *sp*) [static]

Definition at line 1232 of file key.c.

References __LIST_CHAINED, _key_delsp(), IPSEC_ASSERT, ipsec_delisr(), IPSEC_SPSTATE_-DEAD, KEY_FREESAV, ipsecrequest::next, secpolicy::refcnt, ipsecrequest::sav, ipsecrequest::sp, SPTREE_LOCK_ASSERT, and secpolicy::state.

Referenced by _key_freesp().

Here is the call graph for this function:



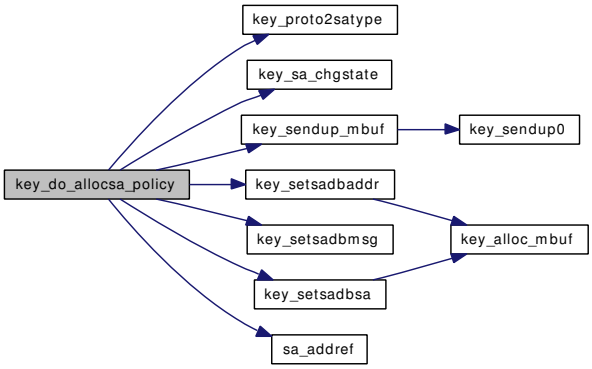
7.16.2.65 static struct secasvar* key_do_allocsa_policy (struct secashead * *sah*, u_int *state*) [static]

Definition at line 901 of file key.c.

References seclifetime::addtime, secasindex::dst, IPSEC_ASSERT, IPSEC_ULPROTO_ANY, KEY_CHKSASTATE, KEY_FREESAV, key_preferred_oldsa, key_proto2satype(), key_sa_chgstate(), key_sendup_mbuf(), KEY_SENDUP_REGISTERED, key_setsadbaddr(), key_setsadbmsg(), key_setsadbsa(), KEYDEBUG, KEYDEBUG_IPSEC_STAMP, secasvar::lft_c, secasindex::proto, secasvar::refcnt, sockaddr_union::sa, sa_addref(), secasvar::sah, SAHTREE_LOCK, SAHTREE_UNLOCK, secashead::saidx, secasindex::src, and secasvar::state.

Referenced by key_allocsa_policy().

Here is the call graph for this function:



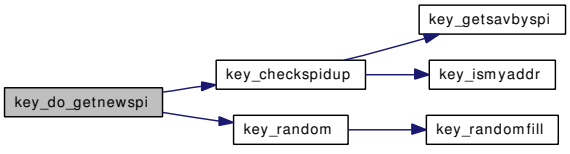
7.16.2.66 static u_int32_t key_do_getnewspi (struct sadb_spirange * *spirange*, struct secasindex * *saidx*) [static]

Definition at line 4588 of file key.c.

References _keystat::getspi_count, ipseclog, key_checkspidup(), key_random(), key_spi_maxval, key_spi_minval, key_spi_trycnt, keystat, and secasindex::proto.

Referenced by key_getspi().

Here is the call graph for this function:

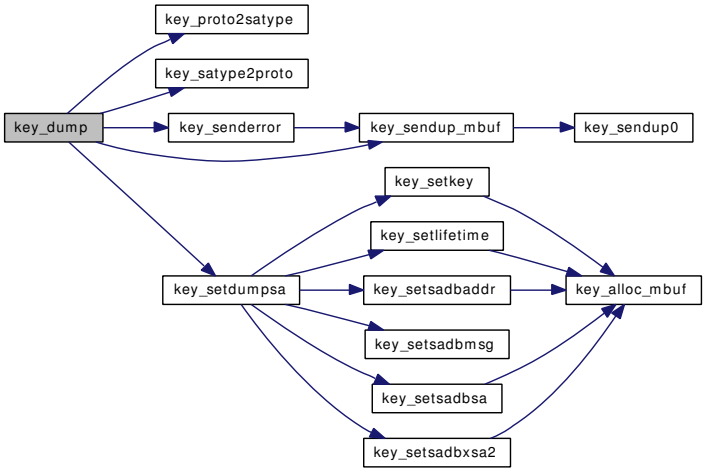


7.16.2.67 static int key_dump (struct socket * *so*, struct mbuf * *m*, const struct sadb_msghdr * *mhp*) [static]

Definition at line 6481 of file key.c.

References _ARRAYLEN, IPSEC_ASSERT, ipseclog, key_proto2satype(), key_satype2proto(), key_senderror(), key_sendup_mbuf(), KEY_SENDUP_ONE, key_setdumpsa(), sadb_msghdr::msg, secasindex::proto, SAHTREE_LOCK, SAHTREE_UNLOCK, and secashead::saidx.

Here is the call graph for this function:



7.16.2.68 struct seckey * key_dup_keymsg (const struct sadb_key *, u_int, struct malloc_type *) [static]

Definition at line 3636 of file key.c.

References seckey::bits, ipseclog, and seckey::key_data.

Referenced by key_setsaval().

7.16.2.69 static struct seclifetime * key_dup_lifemsg (const struct sadb_lifetime * *src*, struct malloc_type * *type*) [static]

Definition at line 3669 of file key.c.

References seclifetime::addtime, seclifetime::allocations, seclifetime::bytes, ipseclog, and seclifetime::usetime.

Referenced by key_setsaval().

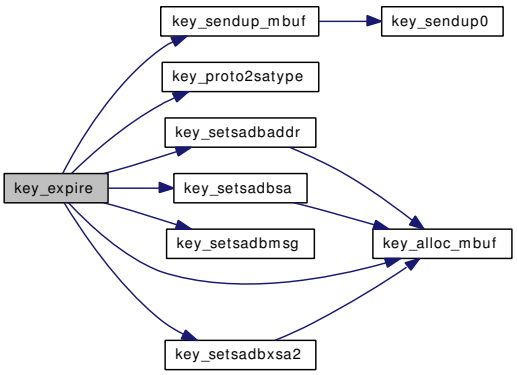
7.16.2.70 static int key_expire (struct secasvar * *sav*) [static]

Definition at line 6271 of file key.c.

References seclifetime::addtime, seclifetime::allocations, seclifetime::bytes, secreplay::count, secasindex::dst, FULLMASK, IPSEC_ASSERT, IPSEC_ULPROTO_ANY, key_alloc_mbuf(), key_proto2satype(), key_sendup_mbuf(), KEY_SENDUP_REGISTERED, key_setsadbaddr(), key_setsadbmsg(), key_setsadbsa(), key_setsadbxsa2(), secasvar::lft_c, secasvar::lft_s, secasindex::mode, secasindex::proto, secasvar::refcnt, secasvar::replay, secasindex::reqid, sockaddr_union::sa, secasvar::sah, secashead::saidx, secasvar::seq, secasindex::src, and seclifetime::usetime.

Referenced by key_flush_sad().

Here is the call graph for this function:

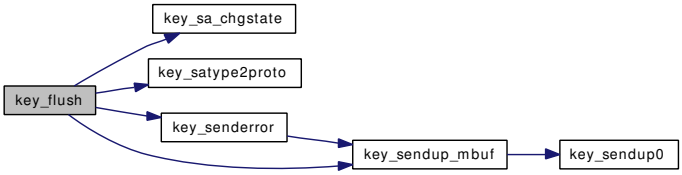


7.16.2.71 static int key_flush (struct socket * *so*, struct mbuf * *m*, const struct sadb_msghdr * *mhp*) [static]

Definition at line 6398 of file key.c.

References _ARRAYLEN, IPSEC_ASSERT, ipseclog, KEY_FREESAV, key_sa_chgstate(), key_satype2proto(), key_senderror(), KEY_SENDUP_ALL, key_sendup_mbuf(), sadb_msghdr::msg, secasindex::proto, secasvar::sah, SAHTREE_LOCK, SAHTREE_UNLOCK, secashead::saidx, and secashead::state.

Here is the call graph for this function:



7.16.2.72 static void key_flush_acq (time_t *now*) [static]

Definition at line 4234 of file key.c.

References __LIST_CHAINED, ACQ_LOCK, ACQ_UNLOCK, secacq::created, and key_blockacq_lifetime.

Referenced by key_timehandler().

7.16.2.73 static void key_flush_sad (time_t *now*) [static]

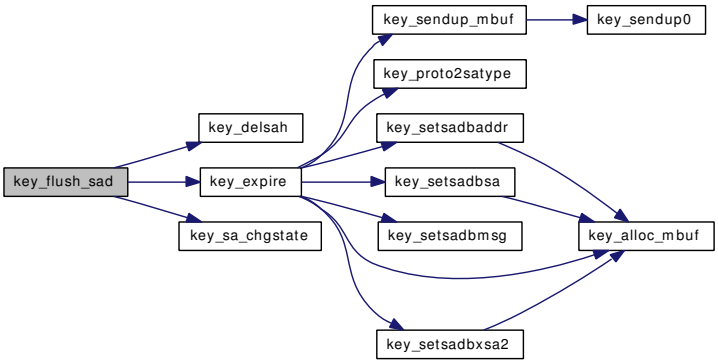
Definition at line 4096 of file key.c.

References seclifetime::addtime, seclifetime::bytes, secasvar::created, ipseclog, key_delsah(), key_expire(), KEY_FREESAV, key_larval_lifetime, key_sa_chgstate(), secasvar::lft_c, secasvar::lft_h,

secasvar::lft_s, secasvar::sah, SAHTREE_LOCK, SAHTREE_UNLOCK, secasvar::state, secashead::state, and seclifetime::usetime.

Referenced by key_timehandler().

Here is the call graph for this function:



7.16.2.74 static void key_flush_spacq (time_t *now*) [static]

Definition at line 4252 of file key.c.

References __LIST_CHAINED, secspacq::created, key_blockacq_lifetime, SPACQ_LOCK, and SPACQ_UNLOCK.

Referenced by key_timehandler().

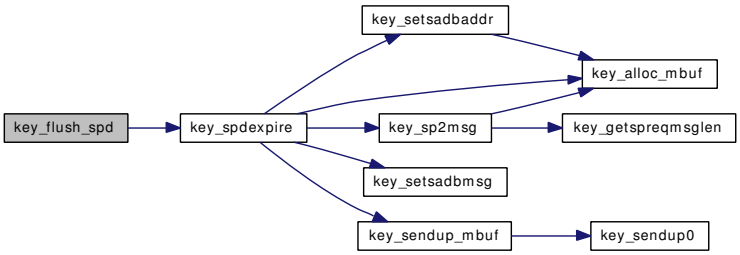
7.16.2.75 static void key_flush_spd (time_t *now*) [static]

Definition at line 4059 of file key.c.

References IPSEC_SPSTATE_DEAD, KEY_FREESP, key_spdexpire(), SPTREE_LOCK, and SPTREE_-UNLOCK.

Referenced by key_timehandler().

Here is the call graph for this function:



7.16.2.76 void key_freereg (struct socket * *so*)

Definition at line 6235 of file key.c.

References __LIST_CHAINED, IPSEC_ASSERT, REGTREE_LOCK, REGTREE_UNLOCK, and secreg::so.

Referenced by key_detach().

7.16.2.77 void key_freesav (struct secasvar ** *psav*, const char * *where*, int *tag*)

Definition at line 1208 of file key.c.

References IPSEC_ASSERT, key_delsav(), KEYDEBUG, KEYDEBUG_IPSEC_STAMP, secasvar::refcnt, sa_delref(), and secasvar::spi.

Here is the call graph for this function:

7.16.2.78 void key_freeso (struct socket * *so*)

Definition at line 1140 of file key.c.

References IPSEC_ASSERT, ipseclog, and key_freesp_so().

Here is the call graph for this function:

7.16.2.79 static void key_freesp_so (struct secpolicy ** *sp*) [static]

Definition at line 1189 of file key.c.

References IPSEC_ASSERT, IPSEC_POLICY_BYPASS, IPSEC_POLICY_ENTRUST, IPSEC_POLICY_IPSEC, and KEY_FREESP.

Referenced by key_freeso().

7.16.2.80 static struct mbuf* key_gather_mbuf (struct mbuf * *m*, const struct sadb_msghdr * *mhp*, int *ndeep*, int *nitem*, va_alist) [static]

Definition at line 1654 of file key.c.

References sadb_msghdr::ext, sadb_msghdr::extlen, sadb_msghdr::extoff, IPSEC_ASSERT, key_alloc_mbuf(), and sadb_msghdr::msg.

Referenced by key_delete(), key_delete_all(), key_getmsgbuf_x1(), key_getspi(), key_spdadd(), and key_spddelete().

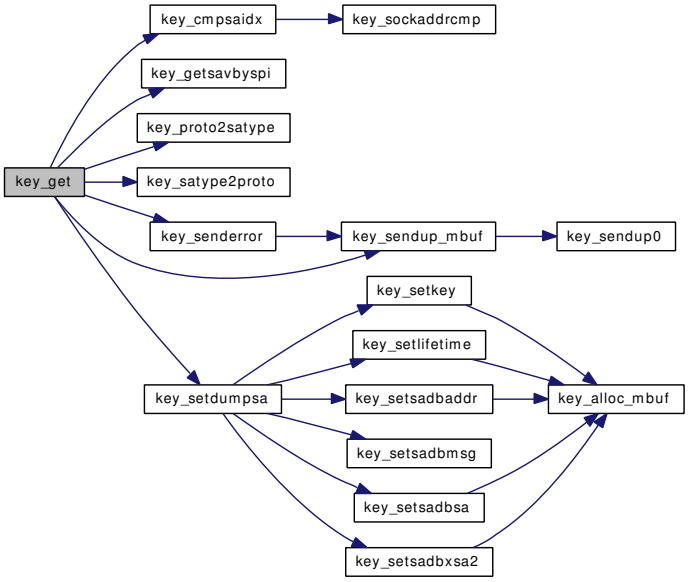
Here is the call graph for this function:

7.16.2.81 static int key_get (struct socket * *so*, struct mbuf * *m*, const struct sadb_msghdr * *mhp*) [static]

Definition at line 5307 of file key.c.

References CMP_HEAD, sadb_msghdr::ext, sadb_msghdr::extlen, IPSEC_ASSERT, IPSEC_MODE_ANY, ipseclog, key_cmpsaidx(), key_getsavbyspi(), key_proto2satype(), key_satype2proto(), key_senderror(), key_sendup_mbuf(), KEY_SENDUP_ONE, key_setdumpsa(), KEY_SETSECASIDX, sadb_msghdr::msg, secasindex::proto, SAHTREE_LOCK, SAHTREE_UNLOCK, secashead::saidx, and secashead::state.

Here is the call graph for this function:



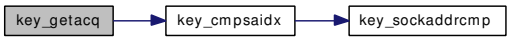
7.16.2.82 static struct secacq* key_getacq (const struct secasindex * *saidx*) [static]

Definition at line 5868 of file key.c.

References ACQ_LOCK, ACQ_UNLOCK, CMP_EXACTLY, key_cmpsaidx(), and secacq::saidx.

Referenced by key_acquire().

Here is the call graph for this function:



7.16.2.83 static struct secacq* key_getacqbyseq (u_int32_t *seq*) [static]

Definition at line 5883 of file key.c.

References ACQ_LOCK, ACQ_UNLOCK, and secacq::seq.

Referenced by key_acquire2(), and key_getspi().

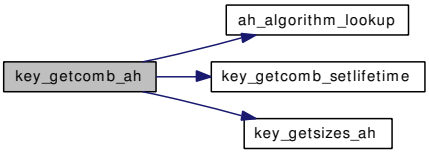
7.16.2.84 static struct mbuf* key_getcomb_ah () [static]

Definition at line 5517 of file key.c.

References _BITS, ah_algorithm_lookup(), IPSEC_ASSERT, key_getcomb_setlifetime(), and key_getsizes_ah().

Referenced by key_getcomb_esp(), and key_getprop().

Here is the call graph for this function:



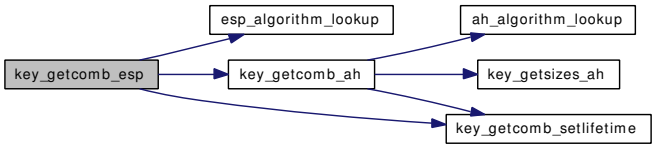
7.16.2.85 static struct mbuf* key_getcomb_esp () [static]

Definition at line 5415 of file key.c.

References _BITS, esp_algorithm_lookup(), IPSEC_ASSERT, key_getcomb_ah(), and key_getcomb_setlifetime().

Referenced by key_getprop().

Here is the call graph for this function:



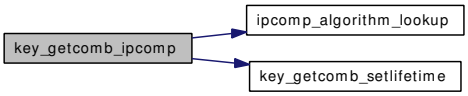
7.16.2.86 static struct mbuf* key_getcomb_ipcomp () [static]

Definition at line 5571 of file key.c.

References ipcomp_algorithm_lookup(), IPSEC_ASSERT, and key_getcomb_setlifetime().

Referenced by key_getprop().

Here is the call graph for this function:



7.16.2.87 static void key_getcomb_setlifetime (struct sadb_comb * *comb*) [static]

Definition at line 5396 of file key.c.

Referenced by key_getcomb_ah(), key_getcomb_esp(), and key_getcomb_ipcomp().

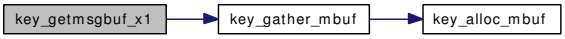
7.16.2.88 static struct mbuf* key_getmsgbuf_x1 (struct mbuf * *m*, const struct sadb_msghdr * *mhp*) [static]

Definition at line 5065 of file key.c.

References IPSEC_ASSERT, key_gather_mbuf(), and sadb_msghdr::msg.

Referenced by key_add(), and key_update().

Here is the call graph for this function:



7.16.2.89 static u_int32_t key_getnewspid () [static]

Definition at line 1974 of file key.c.

References ipseclog, KEY_FREESP, key_getspbyid(), key_spi_trycnt, and policy_id.

Referenced by key_spdadd().

Here is the call graph for this function:

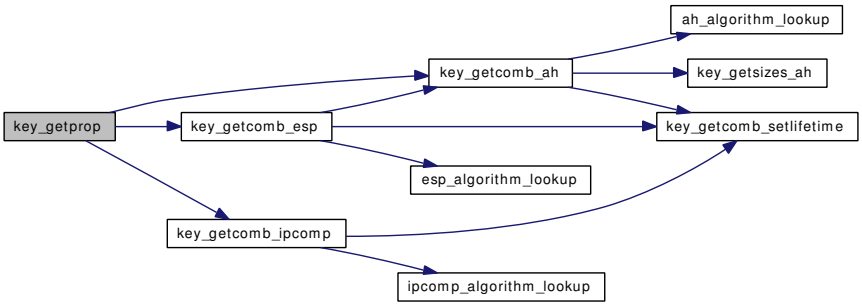
7.16.2.90 static struct mbuf* key_getprop (struct secasindex * *saidx*) const [static]

Definition at line 5615 of file key.c.

References key_getcomb_ah(), key_getcomb_esp(), and key_getcomb_ipcomp().

Referenced by key_acquire().

Here is the call graph for this function:



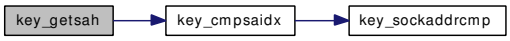
7.16.2.91 static struct secashead* key_getsah (struct secasindex * *saidx*) [static]

Definition at line 2874 of file key.c.

References CMP_REQID, key_cmpsaidx(), SAHTREE_LOCK, SAHTREE_UNLOCK, secashead::saidx, and secashead::state.

Referenced by key_add(), key_getspi(), and key_update().

Here is the call graph for this function:



7.16.2.92 static struct secasvar* key_getsavbyspi (struct secashead * *sah*, u_int32_t *spi*) [static]

Definition at line 2935 of file key.c.

References _ARRAYLEN, ipseclog, SAHTREE_LOCK_ASSERT, and secasvar::state.

Referenced by key_add(), key_checkspidup(), key_delete(), key_get(), and key_update().

7.16.2.93 static void key_getsizes_ah (const struct auth_hash * *ah*, int *alg*, u_int16_t * *min*, u_int16_t * *max*) [static]

Definition at line 5489 of file key.c.

References DPRINTF.

Referenced by key_getcomb_ah(), and key_register().

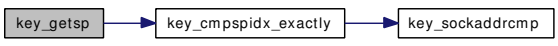
7.16.2.94 static struct secpolicy* key_getsp (struct secpolicyindex * *spidx*) [static]

Definition at line 1266 of file key.c.

References secpolicyindex::dir, IPSEC_ASSERT, IPSEC_SPSTATE_DEAD, key_cmpspidx_exactly(), SP_ADDREF, secpolicy::spidx, SPTREE_LOCK, SPTREE_UNLOCK, and secpolicy::state.

Referenced by key_spdadd(), and key_spddelete().

Here is the call graph for this function:



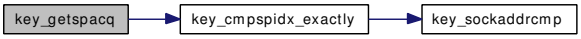
7.16.2.95 static struct secspacq* key_getspacq (struct secpolicyindex * *spidx*) [static]

Definition at line 5925 of file key.c.

References key_cmpspidx_exactly(), SPACQ_LOCK, SPACQ_UNLOCK, and secspacq::spidx.

Referenced by key_spdacquire(), and key_spdadd().

Here is the call graph for this function:



7.16.2.96 static struct secpolicy* key_getspbyid (u_int32_t *id*) [static]

Definition at line 1292 of file key.c.

References secpolicy::id, IPSEC_DIR_INBOUND, IPSEC_DIR_OUTBOUND, IPSEC_SPSTATE_DEAD, SP_ADDREF, SPTREE_LOCK, SPTREE_UNLOCK, and secpolicy::state.

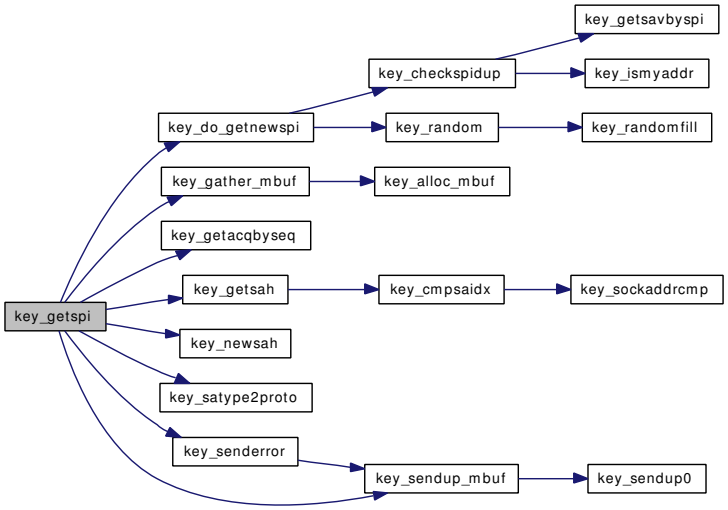
Referenced by key_getnewspid(), key_spddelete2(), and key_spdget().

7.16.2.97 static int key_getspi (struct socket * *so*, struct mbuf * *m*, const struct sadb_msghdr * *mhp*) [static]

Definition at line 4391 of file key.c.

References secacq::count, secacq::created, sadb_msghdr::ext, sadb_msghdr::extlen, IPSEC_ASSERT, IPSEC_MODE_ANY, ipseclog, key_do_getnewspi(), key_gather_mbuf(), key_getacqbyseq(), key_getsah(), key_newsah(), KEY_NEWSAV, key_satype2proto(), key_senderror(), key_sendup_mbuf(), KEY_SENDUP_ONE, KEY_SETSECASIDX, sadb_msghdr::msg, secasvar::seq, and secasvar::spi.

Here is the call graph for this function:



7.16.2.98 static u_int key_getspreqmsglen (struct secpolicy * *sp*) [static]

Definition at line 2477 of file key.c.

References IPSEC_POLICY_IPSEC, and ipsecrequest::next.

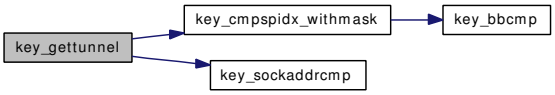
Referenced by key_sp2msg().

7.16.2.99 struct secpolicy* key_gettunnel (const struct sockaddr * *osrc*, const struct sockaddr * *odst*, const struct sockaddr * *isrc*, const struct sockaddr * *idst*, const char * *where*, int *tag*)

Definition at line 681 of file key.c.

References secpolicyindex::dst, secasindex::dst, secpolicy::id, IPSEC_DIR_INBOUND, IPSEC_MODE_-TUNNEL, IPSEC_SPSTATE_DEAD, ipseclog, key_cmpspidx_withmask(), key_sockaddrcmp(), KEY-DEBUG, KEYDEBUG_IPSEC_STAMP, secpolicy::lastused, secasindex::mode, ipsecrequest::next, secpolicy::refcnt, secpolicy::req, sockaddr_union::sa, ipsecrequest::saidx, SP_ADDREF, secpolicy::spidx, SPTREE_LOCK, SPTREE_UNLOCK, secpolicyindex::src, secasindex::src, and secpolicy::state.

Here is the call graph for this function:



7.16.2.100 int key_havesp (u_int *dir*)

Definition at line 554 of file key.c.

References IPSEC_DIR_INBOUND, and IPSEC_DIR_OUTBOUND.

Referenced by ipsec_getpolicybyaddr().

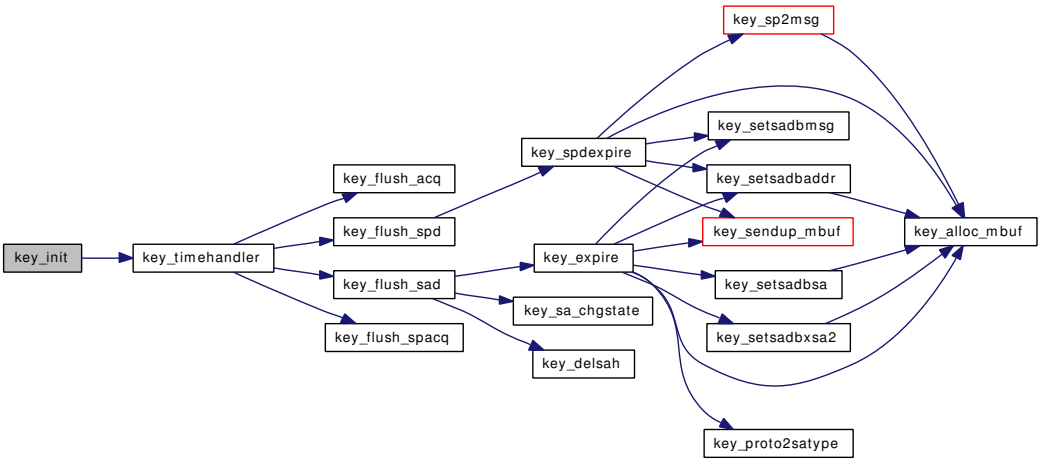
7.16.2.101 void key_init ()

Definition at line 7094 of file key.c.

References ACQ_LOCK_INIT, _keystat::getspi_count, ip4_def_policy, IPSEC_POLICY_NONE, key_-timehandler(), keystat, secpolicy::policy, secpolicy::refcnt, REGTREE_LOCK_INIT, SAHTREE_-LOCK_INIT, SPACQ_LOCK_INIT, and SPTREE_LOCK_INIT.

Referenced by key_init0().

Here is the call graph for this function:



7.16.2.102 int key_ismyaddr (struct sockaddr * *sa*)

Definition at line 3693 of file key.c.

References IPSEC_ASSERT.

Referenced by key_checkspidup().

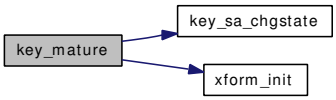
7.16.2.103 static int key_mature (struct secasvar * *sav*) [static]

Definition at line 3212 of file key.c.

References secasvar::alg_auth, secasvar::alg_enc, secasvar::flags, ipseclog, key_sa_chgstate(), secasindex::proto, secasvar::sah, SAHTREE_LOCK, SAHTREE_UNLOCK, secashead::saidx, secasvar::spi, XF_AH, XF_ESP, XF_IPCOMP, XF_TCPSIGNATURE, and xform_init().

Referenced by key_add(), and key_update().

Here is the call graph for this function:



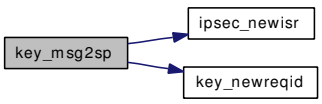
7.16.2.104 struct secpolicy* key_msg2sp (struct sadb_x_policy * *xpl0*, size_t *len*, int * *error*)

Definition at line 1352 of file key.c.

References secpolicyindex::dir, IPSEC_ASSERT, IPSEC_LEVEL_DEFAULT, IPSEC_LEVEL_REQUIRE, IPSEC_LEVEL_UNIQUE, IPSEC_LEVEL_USE, IPSEC_MANUAL_REQID_MAX, IPSEC_MODE_ANY, IPSEC_MODE_TRANSPORT, IPSEC_MODE_TUNNEL, ipsec_newisr(), IPSEC_POLICY_BYPASS, IPSEC_POLICY_DISCARD, IPSEC_POLICY_ENTRUST, IPSEC_POLICY_IPSEC, IPSEC_POLICY_NONE, ipseclog, KEY_FREESP, key_newreqid(), KEY_NEWSP, ipsecrequest::next, secpolicy::policy, secpolicy::req, and secpolicy::spidx.

Referenced by ipsec_set_policy(), and key_spdadd().

Here is the call graph for this function:



7.16.2.105 static struct secacq* key_newacq (const struct secasindex * *saidx*) [static]

Definition at line 5842 of file key.c.

References ACQ_LOCK, acq_seq, ACQ_UNLOCK, secacq::count, secacq::created, ipseclog, secacq::saidx, and secacq::seq.

Referenced by key_acquire().

7.16.2.106 static u_int32_t key_newreqid () [static]

Definition at line 1571 of file key.c.

References IPSEC_MANUAL_REQID_MAX.

Referenced by key_msg2sp().

7.16.2.107 static struct secashead* key_newsah (struct secasindex * *saidx*) [static]

Definition at line 2624 of file key.c.

References IPSEC_ASSERT, SAHTREE_LOCK, SAHTREE_UNLOCK, secashead::saidx, and secashead::state.

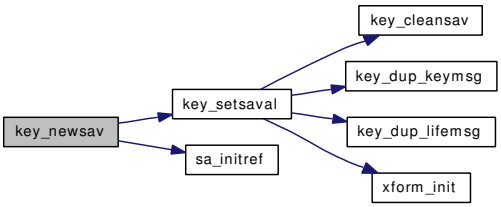
Referenced by key_add(), and key_getspi().

7.16.2.108 static struct secasvar* key_newsav (struct mbuf * *m*, const struct sadb_msghdr * *mhp*, struct secashead * *sah*, int * *errp*, const char* *where*, int *tag*) [static]

Definition at line 2703 of file key.c.

References acq_seq, secasvar::created, sadb_msghdr::ext, IPSEC_ASSERT, ipseclog, key_setsaval(), KEYDEBUG, KEYDEBUG_IPSEC_STAMP, LIST_INSERT_TAIL, sadb_msghdr::msg, secasvar::pid, sa_initref(), secasvar::sah, SECASVAR_LOCK_INIT, secasvar::seq, secasvar::spi, and secasvar::state.

Here is the call graph for this function:



7.16.2.109 struct secpolicy* key_newsp (const char * *where*, int *tag*)

Definition at line 1321 of file key.c.

References KEYDEBUG, KEYDEBUG_IPSEC_STAMP, and SECPOLICY_LOCK_INIT.

7.16.2.110 static struct secspacq* key_newspacq (struct secpolicyindex * *spidx*) [static]

Definition at line 5899 of file key.c.

References secspacq::count, secspacq::created, ipseclog, SPACQ_LOCK, SPACQ_UNLOCK, and secspacq::spidx.

Referenced by key_spdacquire().

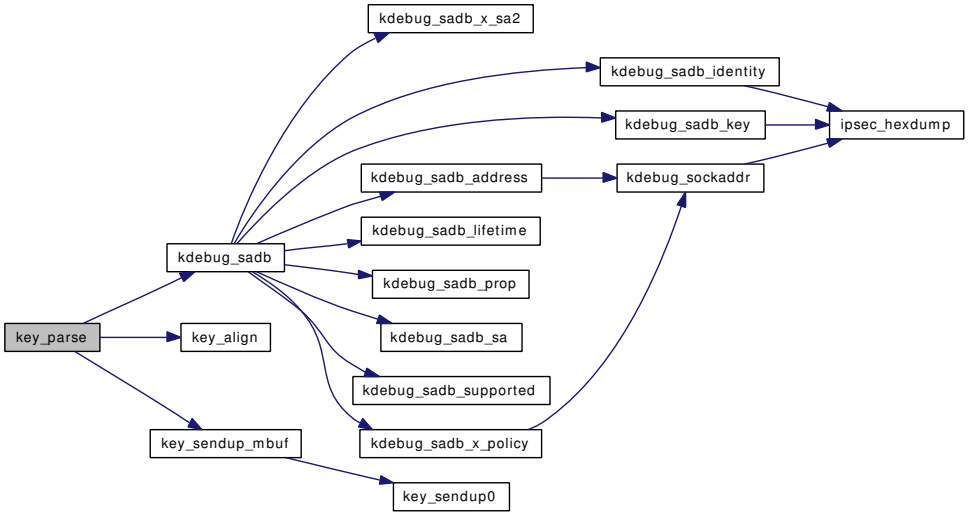
7.16.2.111 int key_parse (struct mbuf * *m*, struct socket * *so*)

Definition at line 6662 of file key.c.

References sadb_msghdr::ext, IPSEC_ASSERT, ipseclog, kdebug_sadb(), key_align(), key_sendup_-mbuf(), KEY_SENDUP_ONE, KEYDEBUG, KEYDEBUG_KEY_DUMP, and sadb_msghdr::msg.

Referenced by key_output().

Here is the call graph for this function:

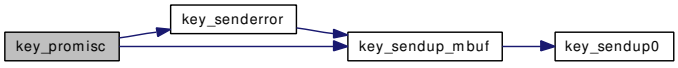


7.16.2.112 static int key_promisc (struct socket * *so*, struct mbuf * *m*, const struct sadb_msghdr * *mhp*) [static]

Definition at line 6573 of file key.c.

References IPSEC_ASSERT, key_senderror(), KEY_SENDUP_ALL, key_sendup_mbuf(), keycb::kp_-promisc, and sadb_msghdr::msg.

Here is the call graph for this function:



7.16.2.113 static u_int8_t key_proto2satype (u_int16_t *proto*) [static]

Definition at line 4359 of file key.c.

Referenced by key_acquire(), key_do_allocsa_policy(), key_dump(), key_expire(), and key_get().

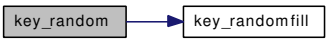
7.16.2.114 u_long key_random ()

Definition at line 4292 of file key.c.

References key_randomfill().

Referenced by key_do_getnewspi().

Here is the call graph for this function:



7.16.2.115 void key_randomfill (void * *p*, size_t *l*)

Definition at line 4301 of file key.c.

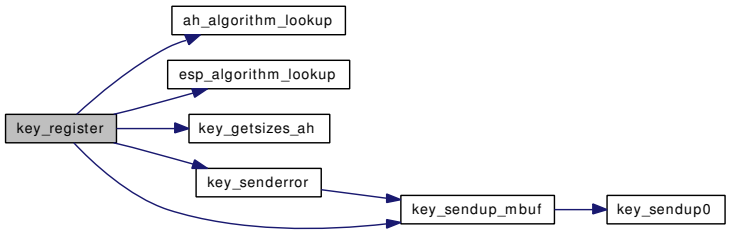
Referenced by esp_init(), key_random(), and key_sa_stir_iv().

7.16.2.116 static int key_register (struct socket * *so*, struct mbuf * *m*, const struct sadb_msghdr * *mhp*) [static]

Definition at line 6078 of file key.c.

References _BITS, ah_algorithm_lookup(), esp_algorithm_lookup(), IPSEC_ASSERT, ipseclog, key_getsizes_ah(), key_senderror(), key_sendup_mbuf(), KEY_SENDUP_REGISTERED, keycb::kp_registered, sadb_msghdr::msg, REGTREE_LOCK, REGTREE_UNLOCK, and secreg::so.

Here is the call graph for this function:



7.16.2.117 static void key_sa_chgstate (struct secasvar * *sav*, u_int8_t *state*) [static]

Definition at line 7217 of file key.c.

References __LIST_CHAINED, IPSEC_ASSERT, and SAHTREE_LOCK_ASSERT.

Referenced by key_delete(), key_delete_all(), key_do_allocsa_policy(), key_flush(), key_flush_sad(), and key_mature().

7.16.2.118 void key_sa_recordxfer (struct secasvar * *sav*, struct mbuf * *m*)

Definition at line 7155 of file key.c.

References IPSEC_ASSERT.

Referenced by ipsec_process_done().

7.16.2.119 void key_sa_routechange (struct sockaddr * *dst*)

Definition at line 7198 of file key.c.

References secashead::sa_route, SAHTREE_LOCK, and SAHTREE_UNLOCK.

7.16.2.120 void key_sa_stir_iv (struct secasvar * *sav*)

Definition at line 7233 of file key.c.

References IPSEC_ASSERT, and key_randomfill().

Here is the call graph for this function:

7.16.2.121 static u_int16_t key_satype2proto (u_int8_t *satype*) [static]

Definition at line 4333 of file key.c.

References IPSEC_PROTO_ANY.

Referenced by key_acquire2(), key_add(), key_delete(), key_dump(), key_flush(), key_get(), key_getspi(), and key_update().

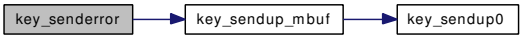
7.16.2.122 static int key_senderror (struct socket * *so*, struct mbuf * *m*, int *code*) [static]

Definition at line 6912 of file key.c.

References IPSEC_ASSERT, key_sendup_mbuf(), and KEY_SENDUP_ONE.

Referenced by key_acquire2(), key_add(), key_delete(), key_delete_all(), key_dump(), key_flush(), key_-get(), key_getspi(), key_promisc(), key_register(), key_spdadd(), key_spddelete(), key_spddelete2(), key_-spddump(), key_spdflush(), key_spdget(), and key_update().

Here is the call graph for this function:



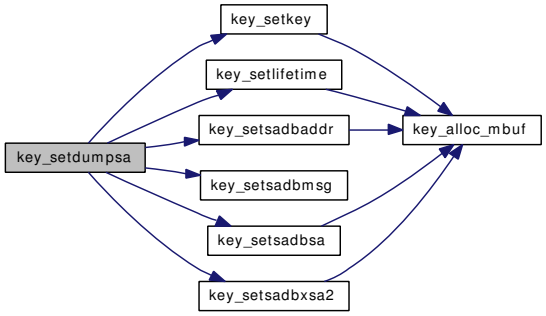
7.16.2.123 static struct mbuf* key_setdumpsa (struct secasvar * *sav*, u_int8_t *type*, u_int8_t *satype*, u_int32_t *seq*, u_int32_t *pid*) [static]

Definition at line 3293 of file key.c.

References FULLMASK, IPSEC_ULPROTO_ANY, key_setkey(), key_setlifetime(), key_setsadbaddr(), key_setsadbmsg(), key_setsadbsa(), and key_setsadbxsa2().

Referenced by key_dump(), and key_get().

Here is the call graph for this function:



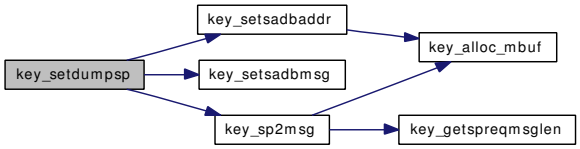
7.16.2.124 static struct mbuf* key_setdumpsp (struct secpolicy * *sp*, u_int8_t *type*, u_int32_t *seq*, u_int32_t *pid*) [static]

Definition at line 2419 of file key.c.

References key_setsadbaddr(), key_setsadbmsg(), and key_sp2msg().

Referenced by key_spddump(), and key_spdget().

Here is the call graph for this function:



7.16.2.125 static int key_setident (struct secashead * *sah*, struct mbuf * *m*, const struct sadb_msghdr * *mhp*) [static]

Definition at line 4989 of file key.c.

References sadb_msghdr::ext, sadb_msghdr::extlen, IPSEC_ASSERT, ipseclog, and sadb_msghdr::msg.

Referenced by key_add(), and key_update().

7.16.2.126 static struct mbuf * key_setkey (struct seckey * *src*, u_int16_t *exttype*) [static]

Definition at line 7294 of file key.c.

References _KEYBUF, _KEYLEN, seckey::bits, key_alloc_mbuf(), and seckey::key_data.

Referenced by key_setdumpsa().

Here is the call graph for this function:

7.16.2.127 static struct mbuf * key_setlifetime (struct seclifetime * *src*, u_int16_t *exttype*) [static]

Definition at line 7331 of file key.c.

References seclifetime::addtime, seclifetime::allocations, seclifetime::bytes, key_alloc_mbuf(), and seclifetime::usetime.

Referenced by key_setdumpsa().

Here is the call graph for this function:

7.16.2.128 static struct mbuf* key_setsadbaddr (u_int16_t *exttype*, const struct sockaddr * *saddr*, u_int8_t *prefixlen*, u_int16_t *ul_proto*) [static]

Definition at line 3514 of file key.c.

References FULLMASK, and key_alloc_mbuf().

Referenced by key_acquire(), key_do_allocsa_policy(), key_expire(), key_setdumpsa(), key_setdumpsp(), and key_spdexpire().

Here is the call graph for this function:

7.16.2.129 static struct mbuf* key_setsadbmsg (u_int8_t *type*, u_int16_t *tlen*, u_int8_t *satype*, u_int32_t *seq*, pid_t *pid*, u_int16_t *reserved*) [static]

Definition at line 3434 of file key.c.

Referenced by key_acquire(), key_do_allocsa_policy(), key_expire(), key_setdumpsa(), key_setdumpsp(), key_spdacquire(), and key_spdexpire().

7.16.2.130 static struct mbuf* key_setsadbsa (struct secasvar * *sav*) [static]

Definition at line 3480 of file key.c.

References key_alloc_mbuf().

Referenced by key_do_allocsa_policy(), key_expire(), and key_setdumpsa().

Here is the call graph for this function:

7.16.2.131 static struct mbuf* key_setsadbxpolicy (u_int16_t *type*, u_int8_t *dir*, u_int32_t *id*) [static]

Definition at line 3599 of file key.c.

References key_alloc_mbuf().

Referenced by key_acquire().

Here is the call graph for this function:

7.16.2.132 static struct mbuf* key_setsadbxsa2 (u_int8_t *mode*, u_int32_t *seq*, u_int32_t *reqid*) [static]

Definition at line 3565 of file key.c.

References key_alloc_mbuf().

Referenced by key_expire(), and key_setdumpsa().

Here is the call graph for this function:

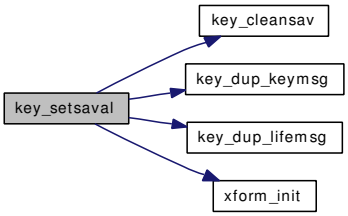
7.16.2.133 static int key_setsaval (struct secasvar * *sav*, struct mbuf * *m*, const struct sadb_msghdr * *mhp*) [static]

Definition at line 2977 of file key.c.

References sadb_msghdr::ext, sadb_msghdr::extlen, IPSEC_ASSERT, ipseclog, key_cleansav(), key_dup_keymsg(), key_dup_lifemsg(), sadb_msghdr::msg, XF_AH, XF_ESP, XF_IPCOMP, XF_-TCPSIGNATURE, and xform_init().

Referenced by key_newsav(), and key_update().

Here is the call graph for this function:



7.16.2.134 static int key_sockaddrcmp (const struct sockaddr * *sa1*, const struct sockaddr * *sa2*, int *port*) [static]

Definition at line 3968 of file key.c.

References satosin, and satosin6.

Referenced by key_allocsa(), key_allocsp2(), key_cmpsaidx(), key_cmpspidx_exactly(), and key_-gettunnel().

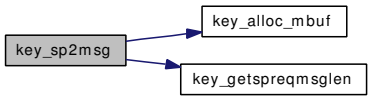
7.16.2.135 struct mbuf* key_sp2msg (struct secpolicy * *sp*)

Definition at line 1587 of file key.c.

References secasindex::dst, IPSEC_ASSERT, IPSEC_POLICY_IPSEC, key_alloc_mbuf(), key_-getspreqmsglen(), ipsecrequest::level, secasindex::mode, ipsecrequest::next, secasindex::proto, secasin-dex::reqid, sockaddr_union::sa, ipsecrequest::saidx, and secasindex::src.

Referenced by ipsec_get_policy(), key_setdumpsp(), and key_spdexpire().

Here is the call graph for this function:



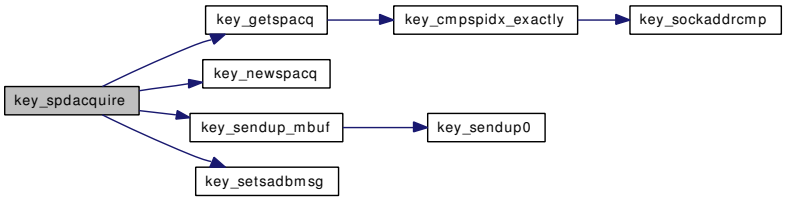
7.16.2.136 int key_spdacquire (struct secpolicy * *sp*)

Definition at line 2258 of file key.c.

References secspacq::count, IPSEC_ASSERT, IPSEC_POLICY_IPSEC, key_getspacq(), key_newspacq(), key_sendup_mbuf(), KEY_SENDUP_REGISTERED, key_setsadbmsg(), and SPACQ_UNLOCK.

Referenced by ipsec4_checkpolicy().

Here is the call graph for this function:



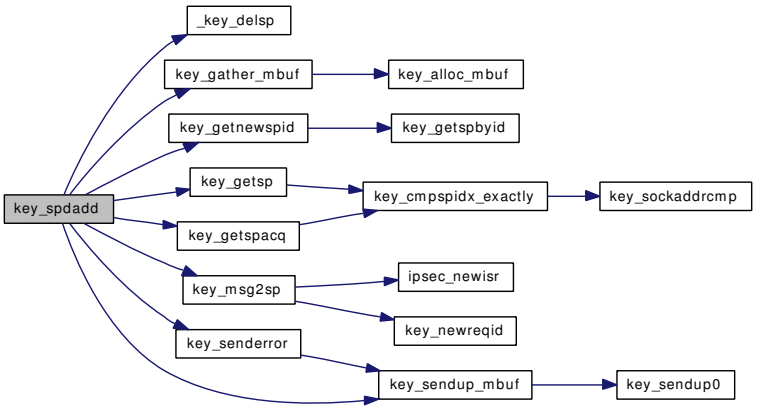
7.16.2.137 static int key_spdadd (struct socket * *so*, struct mbuf * *m*, const struct sadb_msghdr * *mhp*) [static]

Definition at line 1749 of file key.c.

References _key_delsp(), secspacq::count, secspacq::created, secpolicy::created, secpolicyindex::dir, secasindex::dst, sadb_msghdr::ext, sadb_msghdr::extlen, secpolicy::id, IPSEC_ASSERT, IPSEC_DIR_-INBOUND, IPSEC_DIR_OUTBOUND, IPSEC_POLICY_BYPASS, IPSEC_POLICY_ENTRUST,

IPSEC_POLICY_IPSEC, IPSEC_SPSTATE_ALIVE, IPSEC_SPSTATE_DEAD, ipseclog, KEY_-FREESP, key_gather_mbuf(), key_getnewspid(), key_getsp(), key_getspacq(), key_msg2sp(), key_-senderror(), KEY_SENDUP_ALL, key_sendup_mbuf(), KEY_SETSECSPIDX, secpolicy::lastused, secpolicy::lifetime, LIST_INSERT_TAIL, sadb_msghdr::msg, secpolicy::refcnt, secpolicy::req, sockaddr_union::sa, ipsecrequest::saidx, SPACQ_UNLOCK, secspacq::spidx, secpolicy::spidx, se-casindex::src, secpolicy::state, and secpolicy::validtime.

Here is the call graph for this function:

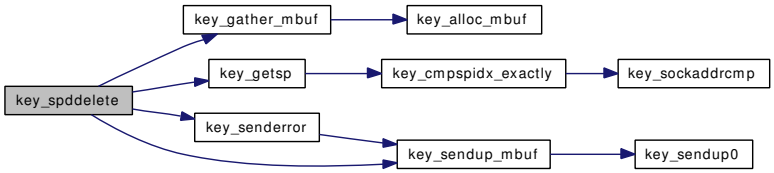


7.16.2.138 static int key_spddelete (struct socket * *so*, struct mbuf * *m*, const struct sadb_msghdr * *mhp*) [static]

Definition at line 2012 of file key.c.

References sadb_msghdr::ext, sadb_msghdr::extlen, secpolicy::id, IPSEC_ASSERT, IPSEC_DIR_-INBOUND, IPSEC_DIR_OUTBOUND, IPSEC_SPSTATE_DEAD, ipseclog, KEY_FREESP, key_-gather_mbuf(), key_getsp(), key_senderror(), KEY_SENDUP_ALL, key_sendup_mbuf(), KEY_-SETSECSPIDX, sadb_msghdr::msg, and secpolicy::state.

Here is the call graph for this function:

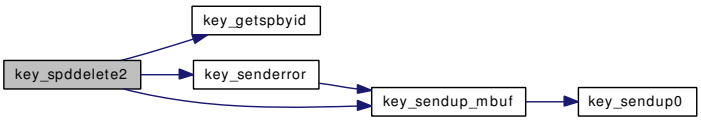


7.16.2.139 static int key_spddelete2 (struct socket * *so*, struct mbuf * *m*, const struct sadb_msghdr * *mhp*) [static]

Definition at line 2110 of file key.c.

References sadb_msghdr::ext, sadb_msghdr::extlen, sadb_msghdr::extoff, IPSEC_ASSERT, IPSEC_-SPSTATE_DEAD, ipseclog, KEY_FREESP, key_getspbyid(), key_senderror(), KEY_SENDUP_ALL, key_sendup_mbuf(), sadb_msghdr::msg, and secpolicy::state.

Here is the call graph for this function:

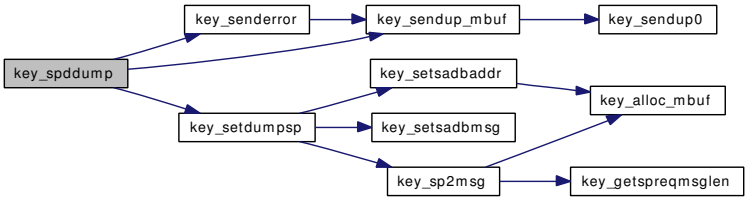


7.16.2.140 static int key_spddump (struct socket * *so*, struct mbuf * *m*, const struct sadb_msghdr * *mhp*) [static]

Definition at line 2377 of file key.c.

References IPSEC_ASSERT, key_senderror(), key_sendup_mbuf(), KEY_SENDUP_ONE, key_setdumpsp(), and sadb_msghdr::msg.

Here is the call graph for this function:



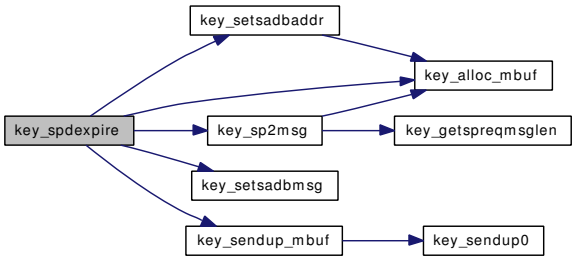
7.16.2.141 static int key_spdexpire (struct secpolicy * *sp*) [static]

Definition at line 2515 of file key.c.

References IPSEC_ASSERT, key_alloc_mbuf(), key_sendup_mbuf(), KEY_SENDUP_REGISTERED, key_setsadbaddr(), key_setsadbmsg(), and key_sp2msg().

Referenced by key_flush_spd().

Here is the call graph for this function:

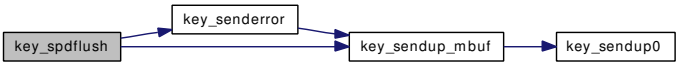


7.16.2.142 static int key_spdflush (struct socket * *so*, struct mbuf * *m*, const struct sadb_msghdr * *mhp*) [static]

Definition at line 2325 of file key.c.

References IPSEC_ASSERT, IPSEC_SPSTATE_DEAD, ipseclog, key_senderror(), KEY_SENDUP_-ALL, key_sendup_mbuf(), sadb_msghdr::msg, SPTREE_LOCK, SPTREE_UNLOCK, and secpolicy::state.

Here is the call graph for this function:

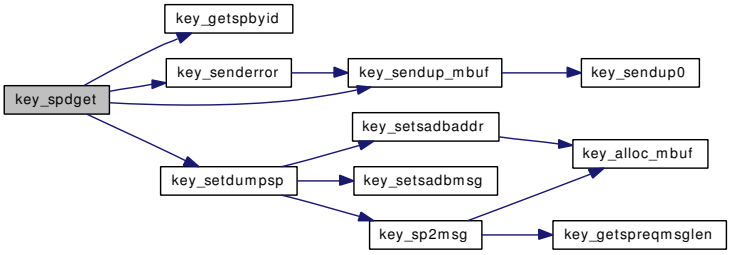


7.16.2.143 static int key_spdget (struct socket * *so*, struct mbuf * *m*, const struct sadb_msghdr * *mhp*) [static]

Definition at line 2205 of file key.c.

References sadb_msghdr::ext, sadb_msghdr::extlen, IPSEC_ASSERT, ipseclog, key_getspbyid(), key_-senderror(), key_sendup_mbuf(), KEY_SENDUP_ONE, key_setdumpsp(), and sadb_msghdr::msg.

Here is the call graph for this function:



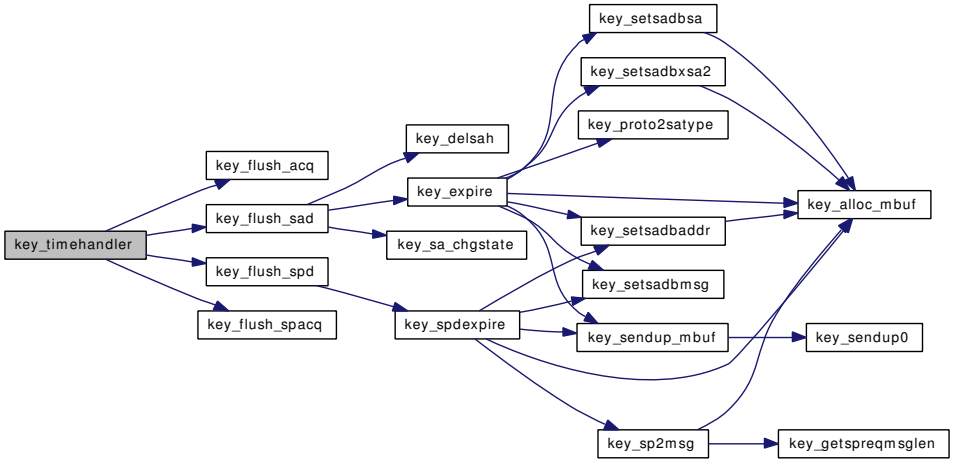
7.16.2.144 void key_timehandler (void)

Definition at line 4276 of file key.c.

References key_flush_acq(), key_flush_sad(), key_flush_spacq(), and key_flush_spd().

Referenced by key_init().

Here is the call graph for this function:

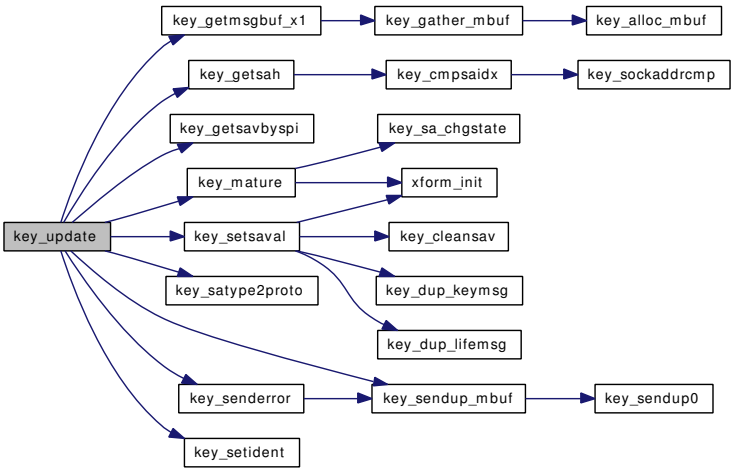


7.16.2.145 static int key_update (struct socket * *so*, struct mbuf * *m*, const struct sadb_msghdr * *mhp*) [static]

Definition at line 4668 of file key.c.

References sadb_msghdr::ext, sadb_msghdr::extlen, IPSEC_ASSERT, IPSEC_MODE_ANY, ipseclog, KEY_FREESAV, key_getmsgbuf_x1(), key_getsah(), key_getsavbyspi(), key_mature(), key_satype2proto(), key_senderror(), KEY_SENDUP_ALL, key_sendup_mbuf(), key_setident(), key_setsaval(), KEY_SETSECASIDX, sadb_msghdr::msg, secasvar::pid, secasindex::proto, secasvar::sah, SAHTREE_LOCK, SAHTREE_UNLOCK, secashead::saidx, and secasvar::spi.

Here is the call graph for this function:



7.16.2.146 static int key_validate_ext (struct sadb_ext * *ext*, int *len*) const [static]

Definition at line 7035 of file key.c.

7.16.2.147 static LIST_HEAD (_sptree, secpolicy) [static]

Definition at line 127 of file key.c.

7.16.2.148 MALLOC_DEFINE (M_IPSEC_SAR, "ipsec-reg", "ipsec sa acquire")

7.16.2.149 MALLOC_DEFINE (M_IPSEC_SAQ, "ipsec-saq", "ipsec sa acquire")

7.16.2.150 MALLOC_DEFINE (M_IPSEC_MISC, "ipsec-misc", "ipsec miscellaneous")

7.16.2.151 MALLOC_DEFINE (M_IPSEC_SR, "ipsecrequest", "ipsec security request")

7.16.2.152 MALLOC_DEFINE (M_IPSEC_SP, "ipsecpolicy", "ipsec security policy")

7.16.2.153 MALLOC_DEFINE (M_IPSEC_SAH, "sahead", "ipsec sa head")

7.16.2.154 MALLOC_DEFINE (M_IPSEC_SA, "secasvar", "ipsec security association")

7.16.2.155 static __inline void sa_addref (struct secasvar * *sav*) [static]

Definition at line 513 of file key.c.

References IPSEC_ASSERT, and secasvar::refcnt.

Referenced by key_allocsa(), and key_do_allocsa_policy().

7.16.2.156 static __inline int sa_delref (struct secasvar * *sav*) [static]

Definition at line 520 of file key.c.

References IPSEC_ASSERT, and secasvar::refcnt.

Referenced by key_freesav().

7.16.2.157 static __inline void sa_initref (struct secasvar * *sav*) [static]

Definition at line 507 of file key.c.

References secasvar::refcnt.

Referenced by key_newsav().

7.16.2.158 SYSCTL_INT (_net_key, KEYCTL_PREFERED_OLDSA, preferred_oldsa, CTLFLAG_RW, & *key_preferred_oldsa*, 0, "")

7.16.2.159 SYSCTL_INT (_net_key, KEYCTL_AH_KEYMIN, ah_keymin, CTLFLAG_RW, & *ipsec_ah_keymin*, 0, "")

7.16.2.160 SYSCTL_INT (_net_key, KEYCTL_ESP_KEYMIN, esp_keymin, CTLFLAG_RW, & *ipsec_esp_keymin*, 0, "")

7.16.2.161 SYSCTL_INT (_net_key, KEYCTL_ESP_AUTH, esp_auth, CTLFLAG_RW, & *ipsec_esp_auth*, 0, "")

7.16.2.162 SYSCTL_INT (_net_key, KEYCTL_BLOCKACQ_LIFETIME, blockacq_lifetime, CTLFLAG_RW, & *key_blockacq_lifetime*, 0, "")

7.16.2.163 SYSCTL_INT (_net_key, KEYCTL_BLOCKACQ_COUNT, blockacq_count, CTLFLAG_RW, & *key_blockacq_count*, 0, "")

7.16.2.164 SYSCTL_INT (_net_key, KEYCTL_LARVAL_LIFETIME, larval_lifetime, CTLFLAG_RW, & *key_larval_lifetime*, 0, "")

7.16.2.165 SYSCTL_INT (_net_key, KEYCTL_RANDOM_INT, int_random, CTLFLAG_RW, & *key_int_random*, 0, "")

7.16.2.166 SYSCTL_INT (_net_key, KEYCTL_SPI_MAX_VALUE, spi_maxval, CTLFLAG_RW, & *key_spi_maxval*, 0, "")

7.16.2.167 SYSCTL_INT (_net_key, KEYCTL_SPI_MIN_VALUE, spi_minval, CTLFLAG_RW, & *key_spi_minval*, 0, "")

7.16.2.168 SYSCTL_INT (_net_key, KEYCTL_SPI_TRY, spi_trycnt, CTLFLAG_RW, & *key_spi_trycnt*, 0, "")

7.16.2.169 SYSCTL_INT (_net_key, KEYCTL_DEBUG_LEVEL, debug, CTLFLAG_RW, & *key_debug_level*, 0, "")

7.16.3 Variable Documentation

7.16.3.1 u_int32_t acq_seq = 0 [static]

Definition at line 125 of file key.c.

Referenced by key_newacq(), and key_newsav().

7.16.3.2 int ipsec_ah_keymin = 128 [static]

Definition at line 239 of file key.c.

7.16.3.3 int ipsec_esp_auth = 0 [static]

Definition at line 238 of file key.c.

7.16.3.4 int ipsec_esp_keymin = 256 [static]

Definition at line 237 of file key.c.

7.16.3.5 int key_blockacq_count = 10 [static]

Definition at line 121 of file key.c.

7.16.3.6 int key_blockacq_lifetime = 20 [static]

Definition at line 122 of file key.c.

Referenced by key_flush_acq(), and key_flush_spacq().

7.16.3.7 u_int32_t key_debug_level = 0

Definition at line 114 of file key.c.

7.16.3.8 u_int key_int_random = 60 [static]

Definition at line 119 of file key.c.

7.16.3.9 u_int key_larval_lifetime = 30 [static]

Definition at line 120 of file key.c.

Referenced by key_flush_sad().

7.16.3.10 int key_preferred_oldsa = 1 [static]

Definition at line 123 of file key.c.

Referenced by key_allocsa(), key_allocsa_policy(), and key_do_allocsa_policy().

7.16.3.11 u_int32_t key_spi_maxval = 0x0fffffff [static]

Definition at line 117 of file key.c.

Referenced by key_do_getnewspi().

7.16.3.12 u_int32_t key_spi_minval = 0x100 [static]

Definition at line 116 of file key.c.

Referenced by key_do_getnewspi().

7.16.3.13 u_int key_spi_trycnt = 1000 [static]

Definition at line 115 of file key.c.

Referenced by key_do_getnewspi(), and key_getnewspid().

7.16.3.14 struct _keystat keystat

Referenced by key_do_getnewspi(), and key_init().

7.16.3.15 const int maxsize[ ] [static]

Initial value:

```
{
        sizeof(struct sadb_msg),
        sizeof(struct sadb_sa),
        sizeof(struct sadb_lifetime),
        sizeof(struct sadb_lifetime),
        sizeof(struct sadb_lifetime),
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        sizeof(struct sadb_spirange),
        0,
        0,
        sizeof(struct sadb_x_sa2),
}
```

Definition at line 214 of file key.c.

7.16.3.16 const int minsize[ ] [static]

Initial value:

```
{
        sizeof(struct sadb_msg),
        sizeof(struct sadb_sa),
        sizeof(struct sadb_lifetime),
        sizeof(struct sadb_lifetime),
        sizeof(struct sadb_lifetime),
        sizeof(struct sadb_address),
        sizeof(struct sadb_address),
        sizeof(struct sadb_address),
        sizeof(struct sadb_key),
        sizeof(struct sadb_key),
        sizeof(struct sadb_ident),
        sizeof(struct sadb_ident),
        sizeof(struct sadb_sens),
        sizeof(struct sadb_prop),
        sizeof(struct sadb_supported),
        sizeof(struct sadb_supported),
        sizeof(struct sadb_spirange),
        0,
        sizeof(struct sadb_x_policy),
        sizeof(struct sadb_x_sa2),
}
```

Definition at line 192 of file key.c.

7.16.3.17 u_int32_t policy_id = 0 [static]

Definition at line 118 of file key.c.

Referenced by key_getnewspid().

7.16.3.18 u_int saorder_state_alive[ ] [static]

Initial value:

```
 {

        SADB_SASTATE_MATURE, SADB_SASTATE_DYING, SADB_SASTATE_LARVAL
}
```

Definition at line 183 of file key.c.

7.16.3.19 u_int saorder_state_any[ ] [static]

Initial value:

```
 {
        SADB_SASTATE_MATURE, SADB_SASTATE_DYING,
        SADB_SASTATE_LARVAL, SADB_SASTATE_DEAD
}
```

Definition at line 187 of file key.c.

7.16.3.20 const u_int saorder_state_valid_prefer_new[ ] [static]

Initial value:

```
 {
        SADB_SASTATE_MATURE, SADB_SASTATE_DYING,
}
```

Definition at line 180 of file key.c.

7.17 /usr/src/sys/netipsec/key.h File Reference

This graph shows which files directly or indirectly include this file:

Defines

- #define KEY_ALLOCSP(spidx, dir) key_allocsp(spidx, dir, __FILE__, __LINE__)
- #define KEY_ALLOCSP2(spi, dst, proto, dir) key_allocsp2(spi, dst, proto, dir, __FILE__, __LINE__)
- #define KEY_NEWSP() key_newsp(__FILE__, __LINE__)
- #define KEY_GETTUNNEL(osrc, odst, isrc, idst) key_gettunnel(osrc, odst, isrc, idst, __FILE__, __LINE__)
- #define KEY_FREESP(spp) _key_freesp(spp, __FILE__, __LINE__)
- #define KEY_ALLOCSA(dst, proto, spi) key_allocsa(dst, proto, spi, __FILE__, __LINE__)
- #define KEY_FREESAV(psav) key_freesav(psav, __FILE__, __LINE__)

Functions

- void key_addref (struct secpolicy *sp)
- int key_havesp (u_int dir)
- secpolicy * key_allocsp (struct secpolicyindex *, u_int, const char *, int)
- secpolicy * key_allocsp2 (u_int32_t spi, union sockaddr_union *dst, u_int8_t proto, u_int dir, const char *, int)
- secpolicy * key_newsp (const char *, int)
- secpolicy * key_gettunnel (const struct sockaddr *, const struct sockaddr *, const struct sockaddr *, const struct sockaddr *, const char *, int)
- void _key_freesp (struct secpolicy **, const char *, int)
- secasvar * key_allocsa (union sockaddr_union *, u_int, u_int32_t, const char *, int)
- void key_freesav (struct secasvar **, const char *, int)
- void key_freeso __P ((struct socket *))
- int key_checktunnelsanity __P ((struct secasvar *, u_int, caddr_t, caddr_t))
- int key_checkrequest __P ((struct ipsecrequest *isr, const struct secasindex *))

- secpolicy *key_msg2sp __P ((struct sadb_x_policy *, size_t, int *))
- mbuf *key_sp2msg __P ((struct secpolicy *))
- int key_ismyaddr __P ((struct sockaddr *))
- void key_timehandler __P ((void))
- void key_randomfill __P ((void *, size_t))
- int key_parse __P ((struct mbuf *, struct socket *))
- void key_sa_recordxfer __P ((struct secasvar *, struct mbuf *))
- void key_sa_stir_iv __P ((struct secasvar *))

7.17.1 Define Documentation

7.17.1.1 #define KEY_ALLOCSA(dst, proto, spi) key_allocsa(dst, proto, spi, __FILE__, __LINE__)

Definition at line 77 of file key.h.

Referenced by ah_input_cb(), ah_output_cb(), esp_input_cb(), esp_output_cb(), ipcomp_input_cb(), ipcomp_output_cb(), and ipsec_common_input().

7.17.1.2 #define KEY_ALLOCSP(spidx, dir) key_allocsp(spidx, dir, __FILE__, __LINE__)

Definition at line 62 of file key.h.

Referenced by ipsec_getpolicybyaddr(), and ipsec_getpolicybysock().

7.17.1.3 #define KEY_ALLOCSP2(spi, dst, proto, dir) key_allocsp2(spi, dst, proto, dir, __FILE__, __LINE__)

Definition at line 64 of file key.h.

Referenced by ipsec_getpolicy().

7.17.1.4 #define KEY_FREESAV(psav) key_freesav(psav, __FILE__, __LINE__)

Definition at line 79 of file key.h.

Referenced by ah_input_cb(), ah_output_cb(), esp_input_cb(), esp_output_cb(), ipcomp_input_cb(), ipcomp_output_cb(), ipsec_common_input(), ipsec_process_done(), key_add(), key_checkrequest(), key_delete(), key_delete_all(), key_delsah(), key_delsp(), key_do_allocsa_policy(), key_flush(), key_flush_sad(), and key_update().

7.17.1.5 #define KEY_FREESP(spp) _key_freesp(spp, __FILE__, __LINE__)

Definition at line 70 of file key.h.

Referenced by ipsec4_checkpolicy(), ipsec4_delete_pcbpolicy(), ipsec4_hdrsiz(), ipsec4_in_reject(), ipsec_copy_policy(), ipsec_init_policy(), ipsec_set_policy(), key_flush_spd(), key_freesp_so(), key_getnewspid(), key_msg2sp(), key_spdadd(), key_spddelete(), and key_spddelete2().

7.17.1.6 #define KEY_GETTUNNEL(osrc, odst, isrc, idst) key_gettunnel(osrc, odst, isrc, idst, __FILE__, __LINE__)

Definition at line 68 of file key.h.

7.17.1.7 #define KEY_NEWSP() key_newsp(__FILE__, __LINE__)

Definition at line 66 of file key.h.

Referenced by ipsec_deepcopy_policy(), ipsec_init_policy(), and key_msg2sp().

7.17.2 Function Documentation

7.17.2.1 void key_sa_stir_iv __P ((struct secasvar *))

7.17.2.2 void key_sa_recordxfer __P ((struct secasvar *, struct mbuf *))

7.17.2.3 int key_parse __P ((struct mbuf *, struct socket *))

7.17.2.4 void key_randomfill __P ((void *, size_t))

7.17.2.5 struct secreg *keydb_newsecreg __P ((void))

7.17.2.6 void key_sa_routechange __P ((struct sockaddr *))

7.17.2.7 int key_spdacquire __P ((struct secpolicy *))

7.17.2.8 struct secpolicy* key_msg2sp __P ((struct sadb_x_policy *, size_t, int *))

7.17.2.9 int key_checkrequest __P ((struct ipsecrequest *isr, const struct secasindex *))

7.17.2.10 int key_checktunnelsanity __P ((struct secasvar *, u_int, caddr_t, caddr_t))

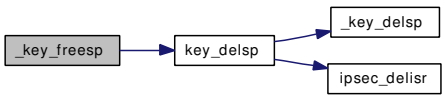
7.17.2.11 void key_freereg __P ((struct socket *))

7.17.2.12 void _key_freesp (struct secpolicy **, const char *, int)

Definition at line 1115 of file key.c.

References secpolicy::id, IPSEC_ASSERT, key_delsp(), KEYDEBUG, KEYDEBUG_IPSEC_STAMP, secpolicy::refcnt, SP_DELREF, SPTREE_LOCK, and SPTREE_UNLOCK.

Here is the call graph for this function:



7.17.2.13 void key_addref (struct secpolicy * *sp*)

Definition at line 541 of file key.c.

References SP_ADDREF, SPTREE_LOCK, and SPTREE_UNLOCK.

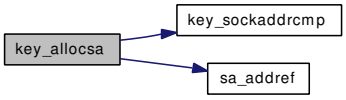
Referenced by ipsec_getpolicybysock(), and key_allocsp_default().

7.17.2.14 struct secasvar* key_allocsa (union sockaddr_union *, u_int, u_int32_t, const char *, int)

Definition at line 1042 of file key.c.

References _ARRAYLEN, IPSEC_ASSERT, KEY_CHKSASTATE, key_preferred_oldsa, key_sockaddrcmp(), KEYDEBUG, KEYDEBUG_IPSEC_STAMP, secasvar::refcnt, sockaddr_union::sa, sa_addref(), secasvar::sah, SAHTREE_LOCK, SAHTREE_UNLOCK, and secasvar::state.

Here is the call graph for this function:

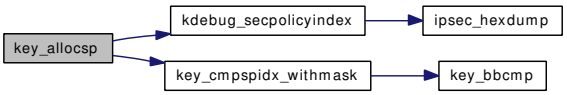


7.17.2.15 struct secpolicy* key_allocsp (struct secpolicyindex *, u_int, const char *, int)

Definition at line 568 of file key.c.

References secpolicyindex::dir, secpolicy::id, IPSEC_ASSERT, IPSEC_DIR_INBOUND, IPSEC_DIR_OUTBOUND, IPSEC_SPSTATE_DEAD, kdebug_secpolicyindex(), KEY_CHKSPDIR, key_cmpspidx_withmask(), KEYDEBUG, KEYDEBUG_IPSEC_DATA, KEYDEBUG_IPSEC_STAMP, secpolicy::lastused, secpolicy::refcnt, SP_ADDREF, secpolicy::spidx, SPTREE_LOCK, SPTREE_UNLOCK, and secpolicy::state.

Here is the call graph for this function:

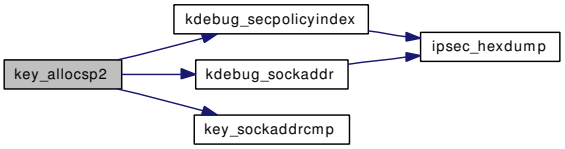


7.17.2.16 struct secpolicy* key_allocsp2 (u_int32_t *spi*, union sockaddr_union * *dst*, u_int8_t *proto*, u_int *dir*, const char *, int)

Definition at line 620 of file key.c.

References secpolicyindex::dir, secpolicyindex::dst, secpolicy::id, IPSEC_ASSERT, IPSEC_DIR_INBOUND, IPSEC_DIR_OUTBOUND, IPSEC_SPSTATE_DEAD, kdebug_secpolicyindex(), kdebug_sockaddr(), KEY_CHKSPDIR, key_sockaddrcmp(), KEYDEBUG, KEYDEBUG_IPSEC_DATA, KEYDEBUG_IPSEC_STAMP, secpolicy::lastused, secpolicy::refcnt, secpolicy::req, sockaddr_union::sa, ipsecrequest::sav, SP_ADDREF, secasvar::spi, secpolicy::spidx, SPTREE_LOCK, SPTREE_UNLOCK, secpolicy::state, and secpolicyindex::ul_proto.

Here is the call graph for this function:

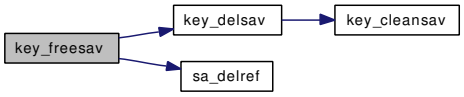


7.17.2.17 void key_freesav (struct secasvar **, const char *, int)

Definition at line 1208 of file key.c.

References IPSEC_ASSERT, key_delsav(), KEYDEBUG, KEYDEBUG_IPSEC_STAMP, secasvar::refcnt, sa_delref(), and secasvar::spi.

Here is the call graph for this function:

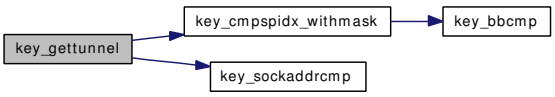


7.17.2.18 struct secpolicy* key_gettunnel (const struct sockaddr *, const struct sockaddr *, const struct sockaddr *, const struct sockaddr *, const char *, int)

Definition at line 681 of file key.c.

References secasindex::dst, secpolicyindex::dst, secpolicy::id, IPSEC_DIR_INBOUND, IPSEC_MODE_TUNNEL, IPSEC_SPSTATE_DEAD, ipseclog, key_cmpspidx_withmask(), key_sockaddrcmp(), KEYDEBUG, KEYDEBUG_IPSEC_STAMP, secpolicy::lastused, secasindex::mode, ipsecrequest::next, secpolicy::refcnt, secpolicy::req, sockaddr_union::sa, ipsecrequest::saidx, SP_ADDREF, secpolicy::spidx, SPTREE_LOCK, SPTREE_UNLOCK, secasindex::src, secpolicyindex::src, and secpolicy::state.

Here is the call graph for this function:



7.17.2.19 int key_havesp (u_int *dir*)

Definition at line 554 of file key.c.

References IPSEC_DIR_INBOUND, and IPSEC_DIR_OUTBOUND.

Referenced by ipsec_getpolicybyaddr().

7.17.2.20 struct secpolicy* key_newsp (const char *, int)

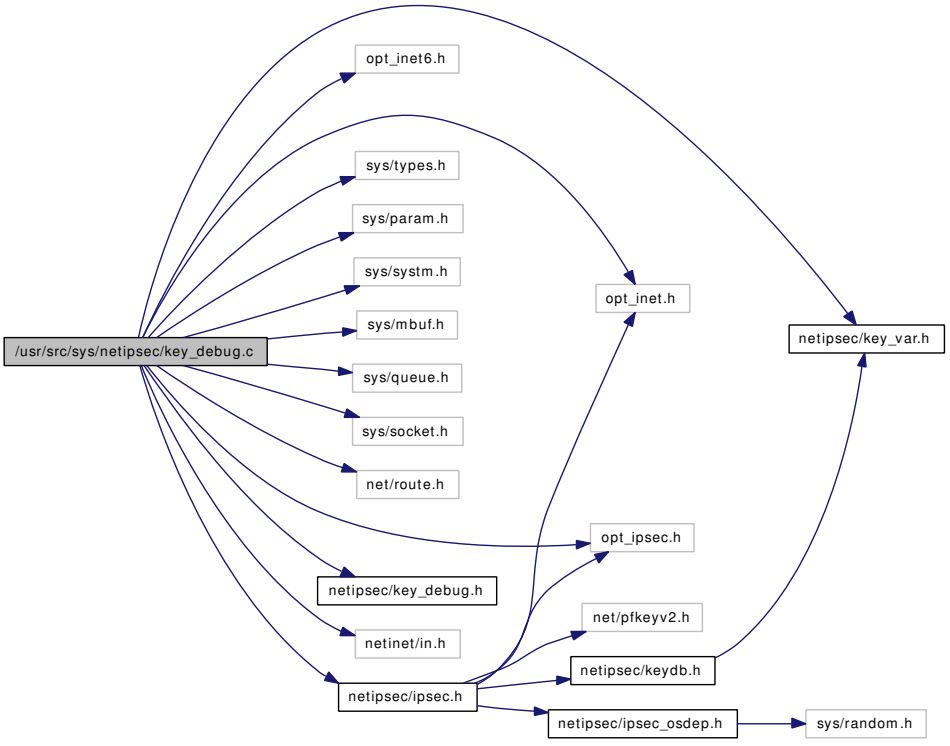
Definition at line 1321 of file key.c.

References KEYDEBUG, KEYDEBUG_IPSEC_STAMP, and SECPOLICY_LOCK_INIT.

7.18 /usr/src/sys/netipsec/key_debug.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include "opt_ipsec.h"
#include <sys/types.h>
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/mbuf.h>
#include <sys/queue.h>
#include <sys/socket.h>
#include <net/route.h>
#include <netipsec/key_var.h>
#include <netipsec/key_debug.h>
#include <netinet/in.h>
#include <netipsec/ipsec.h>
```

Include dependency graph for key_debug.c:



Functions

- static void kdebug_sadb_prop __P ((struct sadb_ext *))
- static void kdebug_secreplay __P ((struct secreplay *))
- void kdebug_sadb (struct sadb_msg *base)
- static void kdebug_sadb_prop (struct sadb_ext *ext)
- static void kdebug_sadb_identity (struct sadb_ext *ext)
- static void kdebug_sadb_supported (struct sadb_ext *ext)
- static void kdebug_sadb_lifetime (struct sadb_ext *ext)
- static void kdebug_sadb_sa (struct sadb_ext *ext)
- static void kdebug_sadb_address (struct sadb_ext *ext)
- static void kdebug_sadb_key (struct sadb_ext *ext)
- static void kdebug_sadb_x_sa2 (struct sadb_ext *ext)
- void kdebug_sadb_x_policy (struct sadb_ext *ext)
- void kdebug_secpolicy (struct secpolicy *sp)
- void kdebug_secpolicyindex (struct secpolicyindex *spidx)
- void kdebug_secasindex (struct secasindex *saidx)
- void kdebug_secasv (struct secasvar *sav)
- static void kdebug_secreplay (struct secreplay *rpl)
- void kdebug_mbufhdr (struct mbuf *m)
- void kdebug_mbuf (struct mbuf *m0)
- void kdebug_sockaddr (struct sockaddr *addr)
- void ipsec_bindump (caddr_t buf, int len)
- void ipsec_hexdump (caddr_t buf, int len)

7.18.1 Function Documentation

7.18.1.1 static void kdebug_secreplay __P ((struct secreplay *)) [static]

7.18.1.2 static void kdebug_sadb_x_sa2 __P ((struct sadb_ext *)) [static]

7.18.1.3 void ipsec_bindump (caddr_t *buf*, int *len*)

Definition at line 720 of file key_debug.c.

7.18.1.4 void ipsec_hexdump (caddr_t *buf*, int *len*)

Definition at line 734 of file key_debug.c.

Referenced by kdebug_sadb_identity(), kdebug_sadb_key(), kdebug_secasindex(), kdebug_secasv(), kdebug_secpolicyindex(), and kdebug_sockaddr().

7.18.1.5 void kdebug_mbuf (struct mbuf * *m0*)

Definition at line 656 of file key_debug.c.

References kdebug_mbufhdr().

Referenced by ipsec_get_policy(), and key_output().

Here is the call graph for this function:

7.18.1.6 void kdebug_mbufhdr (struct mbuf * *m*)

Definition at line 628 of file key_debug.c.

Referenced by kdebug_mbuf().

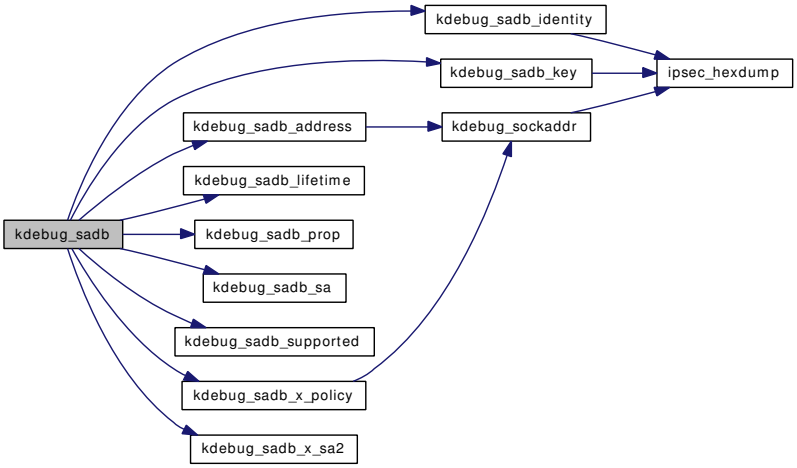
7.18.1.7 void kdebug_sadb (struct sadb_msg * *base*)

Definition at line 83 of file key_debug.c.

References kdebug_sadb_address(), kdebug_sadb_identity(), kdebug_sadb_key(), kdebug_sadb_-lifetime(), kdebug_sadb_prop(), kdebug_sadb_sa(), kdebug_sadb_supported(), kdebug_sadb_x_policy(), and kdebug_sadb_x_sa2().

Referenced by key_parse(), and key_sendup().

Here is the call graph for this function:



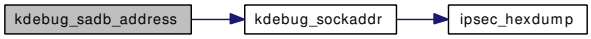
7.18.1.8 static void kdebug_sadb_address (struct sadb_ext * *ext*) [static]

Definition at line 331 of file key_debug.c.

References kdebug_sockaddr().

Referenced by kdebug_sadb().

Here is the call graph for this function:



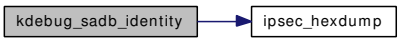
7.18.1.9 static void kdebug_sadb_identity (struct sadb_ext * *ext*) [static]

Definition at line 222 of file key_debug.c.

References ipsec_hexdump().

Referenced by kdebug_sadb().

Here is the call graph for this function:



7.18.1.10 static void kdebug_sadb_key (struct sadb_ext * *ext*) [static]

Definition at line 351 of file key_debug.c.

References ipsec_hexdump().

Referenced by kdebug_sadb(), and kdebug_secasv().

Here is the call graph for this function:

7.18.1.11 static void kdebug_sadb_lifetime (struct sadb_ext * *ext*) [static]

Definition at line 292 of file key_debug.c.

Referenced by kdebug_sadb(), and kdebug_secasv().

7.18.1.12 static void kdebug_sadb_prop (struct sadb_ext * *ext*) [static]

Definition at line 172 of file key_debug.c.

Referenced by kdebug_sadb().

7.18.1.13 static void kdebug_sadb_sa (struct sadb_ext * *ext*) [static]

Definition at line 312 of file key_debug.c.

Referenced by kdebug_sadb().

7.18.1.14 static void kdebug_sadb_supported (struct sadb_ext * *ext*) [static]

Definition at line 265 of file key_debug.c.

Referenced by kdebug_sadb().

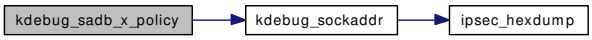
7.18.1.15 void kdebug_sadb_x_policy (struct sadb_ext * *ext*)

Definition at line 399 of file key_debug.c.

References IPSEC_POLICY_IPSEC, and kdebug_sockaddr().

Referenced by ipsec_set_policy(), and kdebug_sadb().

Here is the call graph for this function:



7.18.1.16 static void kdebug_sadb_x_sa2 (struct sadb_ext * *ext*) [static]

Definition at line 380 of file key_debug.c.

Referenced by kdebug_sadb().

7.18.1.17 void kdebug_secasindex (struct secasindex * *saidx*)

Definition at line 536 of file key_debug.c.

References ipsec_hexdump().

Referenced by kdebug_secasv(), and kdebug_secpolicy().

Here is the call graph for this function:

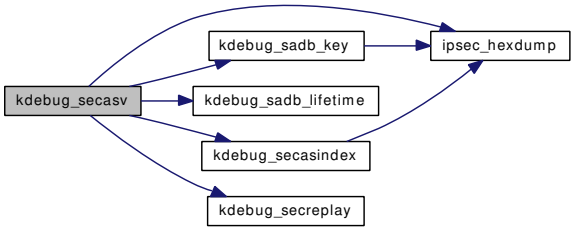
7.18.1.18 void kdebug_secasv (struct secasvar * *sav*)

Definition at line 557 of file key_debug.c.

References ipsec_hexdump(), kdebug_sadb_key(), kdebug_sadb_lifetime(), kdebug_secasindex(), and kdebug_secreplay().

Referenced by kdebug_secpolicy().

Here is the call graph for this function:



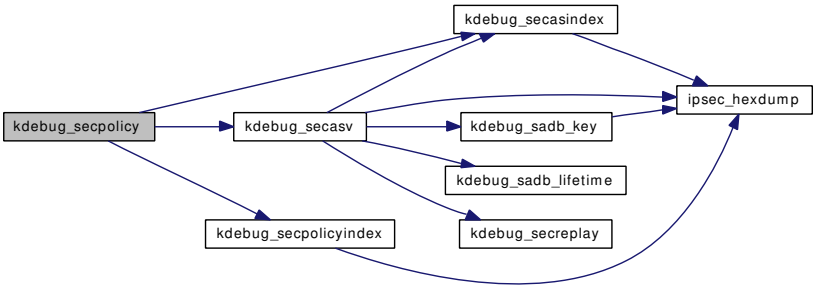
7.18.1.19 void kdebug_secpolicy (struct secpolicy * *sp*)

Definition at line 467 of file key_debug.c.

References IPSEC_POLICY_BYPASS, IPSEC_POLICY_DISCARD, IPSEC_POLICY_ENTRUST, IPSEC_POLICY_IPSEC, IPSEC_POLICY_NONE, kdebug_secasindex(), kdebug_secasv(), kdebug_-secpolicyindex(), ipsecrequest::level, ipsecrequest::next, ipsecrequest::saidx, and ipsecrequest::sav.

Referenced by ipsec_hdrsiz(), ipsec_in_reject(), and ipsec_set_policy().

Here is the call graph for this function:



7.18.1.20 void kdebug_secpolicyindex (struct secpolicyindex * *spidx*)

Definition at line 515 of file key_debug.c.

References ipsec_hexdump().

Referenced by kdebug_secpolicy(), key_allocsp(), and key_allocsp2().

Here is the call graph for this function:

7.18.1.21 static void kdebug_secreplay (struct secreplay * *rpl*) [static]

Definition at line 599 of file key_debug.c.

Referenced by kdebug_secasv().

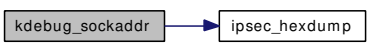
7.18.1.22 void kdebug_sockaddr (struct sockaddr * *addr*)

Definition at line 681 of file key_debug.c.

References ipsec_hexdump().

Referenced by kdebug_sadb_address(), kdebug_sadb_x_policy(), and key_allocsp2().

Here is the call graph for this function:



7.19 /usr/src/sys/netipsec/key_debug.h File Reference

This graph shows which files directly or indirectly include this file:

Defines

- #define KEYDEBUG_STAMP 0x00000001
- #define KEYDEBUG_DATA 0x00000002
- #define KEYDEBUG_DUMP 0x00000004
- #define KEYDEBUG_KEY 0x00000010
- #define KEYDEBUG_ALG 0x00000020
- #define KEYDEBUG_IPSEC 0x00000040
- #define KEYDEBUG_KEY_STAMP (KEYDEBUG_KEY | KEYDEBUG_STAMP)
- #define KEYDEBUG_KEY_DATA (KEYDEBUG_KEY | KEYDEBUG_DATA)
- #define KEYDEBUG_KEY_DUMP (KEYDEBUG_KEY | KEYDEBUG_DUMP)
- #define KEYDEBUG_ALG_STAMP (KEYDEBUG_ALG | KEYDEBUG_STAMP)
- #define KEYDEBUG_ALG_DATA (KEYDEBUG_ALG | KEYDEBUG_DATA)
- #define KEYDEBUG_ALG_DUMP (KEYDEBUG_ALG | KEYDEBUG_DUMP)
- #define KEYDEBUG_IPSEC_STAMP (KEYDEBUG_IPSEC | KEYDEBUG_STAMP)
- #define KEYDEBUG_IPSEC_DATA (KEYDEBUG_IPSEC | KEYDEBUG_DATA)
- #define KEYDEBUG_IPSEC_DUMP (KEYDEBUG_IPSEC | KEYDEBUG_DUMP)
- #define KEYDEBUG(lev, arg) do { if ((key_debug_level & (lev)) == (lev)) { arg; } } while (0)

Functions

- void kdebug_sadb __P ((struct sadb_msg *))
- void kdebug_sadb_x_policy __P ((struct sadb_ext *))
- void kdebug_secpolicy __P ((struct secpolicy *))
- void kdebug_secpolicyindex __P ((struct secpolicyindex *))
- void kdebug_secasindex __P ((struct secasindex *))
- void kdebug_secasv __P ((struct secasvar *))
- void kdebug_mbufhdr __P ((struct mbuf *))

- void kdebug_sockaddr __P ((struct sockaddr *))
- void ipsec_hexdump __P ((caddr_t, int))

Variables

- u_int32_t key_debug_level

7.19.1 Define Documentation

7.19.1.1 #define KEYDEBUG(lev, arg) do { if ((key_debug_level & (lev)) == (lev)) { arg; } } while (0)

Definition at line 56 of file key_debug.h.

Referenced by _key_freesp(), ipsec4_hdrsiz(), ipsec_get_policy(), ipsec_getpolicybysock(), ipsec_hdrsiz(), ipsec_in_reject(), ipsec_set_policy(), ipsec_setspidx(), key_allocsa(), key_allocsp(), key_allocsp2(), key_allocsp_default(), key_do_allocsa_policy(), key_freesav(), key_gettunnel(), key_newsav(), key_newsp(), key_output(), key_parse(), and key_sendup().

7.19.1.2 #define KEYDEBUG_ALG 0x00000020

Definition at line 43 of file key_debug.h.

7.19.1.3 #define KEYDEBUG_ALG_DATA (KEYDEBUG_ALG | KEYDEBUG_DATA)

Definition at line 50 of file key_debug.h.

7.19.1.4 #define KEYDEBUG_ALG_DUMP (KEYDEBUG_ALG | KEYDEBUG_DUMP)

Definition at line 51 of file key_debug.h.

7.19.1.5 #define KEYDEBUG_ALG_STAMP (KEYDEBUG_ALG | KEYDEBUG_STAMP)

Definition at line 49 of file key_debug.h.

7.19.1.6 #define KEYDEBUG_DATA 0x00000002

Definition at line 39 of file key_debug.h.

7.19.1.7 #define KEYDEBUG_DUMP 0x00000004

Definition at line 40 of file key_debug.h.

7.19.1.8 #define KEYDEBUG_IPSEC 0x00000040

Definition at line 44 of file key_debug.h.

7.19.1.9 #define KEYDEBUG_IPSEC_DATA (KEYDEBUG_IPSEC | KEYDEBUG_DATA)

Definition at line 53 of file key_debug.h.

Referenced by ipsec4_hdrsiz(), ipsec_hdrsiz(), ipsec_in_reject(), key_allocsp(), and key_allocsp2().

7.19.1.10 #define KEYDEBUG_IPSEC_DUMP (KEYDEBUG_IPSEC | KEYDEBUG_DUMP)

Definition at line 54 of file key_debug.h.

Referenced by ipsec_setspidx().

7.19.1.11 #define KEYDEBUG_IPSEC_STAMP (KEYDEBUG_IPSEC | KEYDEBUG_STAMP)

Definition at line 52 of file key_debug.h.

Referenced by _key_freesp(), ipsec_getpolicybysock(), key_allocsa(), key_allocsp(), key_allocsp2(), key_allocsp_default(), key_do_allocsa_policy(), key_freesav(), key_gettunnel(), key_newsav(), and key_newsp().

7.19.1.12 #define KEYDEBUG_KEY 0x00000010

Definition at line 42 of file key_debug.h.

7.19.1.13 #define KEYDEBUG_KEY_DATA (KEYDEBUG_KEY | KEYDEBUG_DATA)

Definition at line 47 of file key_debug.h.

7.19.1.14 #define KEYDEBUG_KEY_DUMP (KEYDEBUG_KEY | KEYDEBUG_DUMP)

Definition at line 48 of file key_debug.h.

Referenced by key_output(), key_parse(), and key_sendup().

7.19.1.15 #define KEYDEBUG_KEY_STAMP (KEYDEBUG_KEY | KEYDEBUG_STAMP)

Definition at line 46 of file key_debug.h.

7.19.1.16 #define KEYDEBUG_STAMP 0x00000001

Definition at line 38 of file key_debug.h.

7.19.2 Function Documentation

7.19.2.1 void ipsec_bindump __P ((caddr_t, int))

7.19.2.2 void kdebug_sockaddr __P ((struct sockaddr *))

7.19.2.3 void kdebug_mbufhdr __P ((struct mbuf *))

7.19.2.4 void kdebug_secasv __P ((struct secasvar *))

7.19.2.5 void kdebug_secasindex __P ((struct secasindex *))

7.19.2.6 void kdebug_secpolicyindex __P ((struct secpolicyindex *))

7.19.2.7 void kdebug_secpolicy __P ((struct secpolicy *))

7.19.2.8 void kdebug_sadb_x_policy __P ((struct sadb_ext *))

7.19.2.9 void kdebug_sadb __P ((struct sadb_msg *))

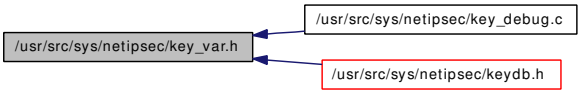
7.19.3 Variable Documentation

7.19.3.1 u_int32_t key_debug_level

Definition at line 114 of file key.c.

7.20 /usr/src/sys/netipsec/key_var.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define KEYCTL_DEBUG_LEVEL 1
- #define KEYCTL_SPI_TRY 2
- #define KEYCTL_SPI_MIN_VALUE 3
- #define KEYCTL_SPI_MAX_VALUE 4
- #define KEYCTL_RANDOM_INT 5
- #define KEYCTL_LARVAL_LIFETIME 6
- #define KEYCTL_BLOCKACQ_COUNT 7
- #define KEYCTL_BLOCKACQ_LIFETIME 8
- #define KEYCTL_ESP_KEYMIN 9
- #define KEYCTL_ESP_AUTH 10
- #define KEYCTL_AH_KEYMIN 11
- #define KEYCTL_PREFERED_OLDSA 12
- #define KEYCTL_MAXID 13
- #define KEYCTL_NAMES
- #define _ARRAYLEN(p) (sizeof(p)/sizeof(p[0]))
- #define _KEYLEN(key) ((u_int)((key) → bits >> 3))
- #define _KEYBITS(key) ((u_int)((key) → bits))
- #define _KEYBUF(key) ((caddr_t)((caddr_t)(key) + sizeof(struct sadb_key)))

7.20.1 Define Documentation

7.20.1.1 #define _ARRAYLEN(p) (sizeof(p)/sizeof(p[0]))

Definition at line 68 of file key_var.h.

Referenced by key_allocsa(), key_delete_all(), key_delsah(), key_dump(), key_flush(), and key_-getsavbyspi().

7.20.1.2 #define _KEYBITS(key) ((u_int)((key) → bits))

Definition at line 70 of file key_var.h.

Referenced by ah_init0(), ah_input(), ah_output(), esp_init(), esp_input(), and esp_output().

7.20.1.3 #define _KEYBUF(key) ((caddr_t)((caddr_t)(key) + sizeof(struct sadb_key)))

Definition at line 71 of file key_var.h.

Referenced by key_setkey().

7.20.1.4 #define _KEYLEN(key) ((u_int)((key) → bits >> 3))

Definition at line 69 of file key_var.h.

Referenced by ah_init0(), ah_zeroize(), esp_init(), esp_zeroize(), key_cleansav(), key_setkey(), tcpsignature_init(), and tcpsignature_zeroize().

7.20.1.5 #define KEYCTL_AH_KEYMIN 11

Definition at line 47 of file key_var.h.

7.20.1.6 #define KEYCTL_BLOCKACQ_COUNT 7

Definition at line 43 of file key_var.h.

7.20.1.7 #define KEYCTL_BLOCKACQ_LIFETIME 8

Definition at line 44 of file key_var.h.

7.20.1.8 #define KEYCTL_DEBUG_LEVEL 1

Definition at line 37 of file key_var.h.

7.20.1.9 #define KEYCTL_ESP_AUTH 10

Definition at line 46 of file key_var.h.

7.20.1.10 #define KEYCTL_ESP_KEYMIN 9

Definition at line 45 of file key_var.h.

7.20.1.11 #define KEYCTL_LARVAL_LIFETIME 6

Definition at line 42 of file key_var.h.

7.20.1.12 #define KEYCTL_MAXID 13

Definition at line 49 of file key_var.h.

7.20.1.13 #define KEYCTL_NAMES

Value:

```
{ \
        { 0, 0 }, \
        { "debug", CTLTYPE_INT }, \
        { "spi_try", CTLTYPE_INT }, \
        { "spi_min_value", CTLTYPE_INT }, \
        { "spi_max_value", CTLTYPE_INT }, \
```

```
        { "random_int", CTLTYPE_INT }, \
        { "larval_lifetime", CTLTYPE_INT }, \
        { "blockacq_count", CTLTYPE_INT }, \
        { "blockacq_lifetime", CTLTYPE_INT }, \
        { "esp_keymin", CTLTYPE_INT }, \
        { "esp_auth", CTLTYPE_INT }, \
        { "ah_keymin", CTLTYPE_INT }, \
        { "prefered_oldsa", CTLTYPE_INT }, \
}
```

Definition at line 51 of file key_var.h.

7.20.1.14 #define KEYCTL_PREFERED_OLDSA 12

Definition at line 48 of file key_var.h.

7.20.1.15 #define KEYCTL_RANDOM_INT 5

Definition at line 41 of file key_var.h.

7.20.1.16 #define KEYCTL_SPI_MAX_VALUE 4

Definition at line 40 of file key_var.h.

7.20.1.17 #define KEYCTL_SPI_MIN_VALUE 3

Definition at line 39 of file key_var.h.

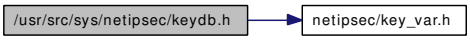
7.20.1.18 #define KEYCTL_SPI_TRY 2

Definition at line 38 of file key_var.h.

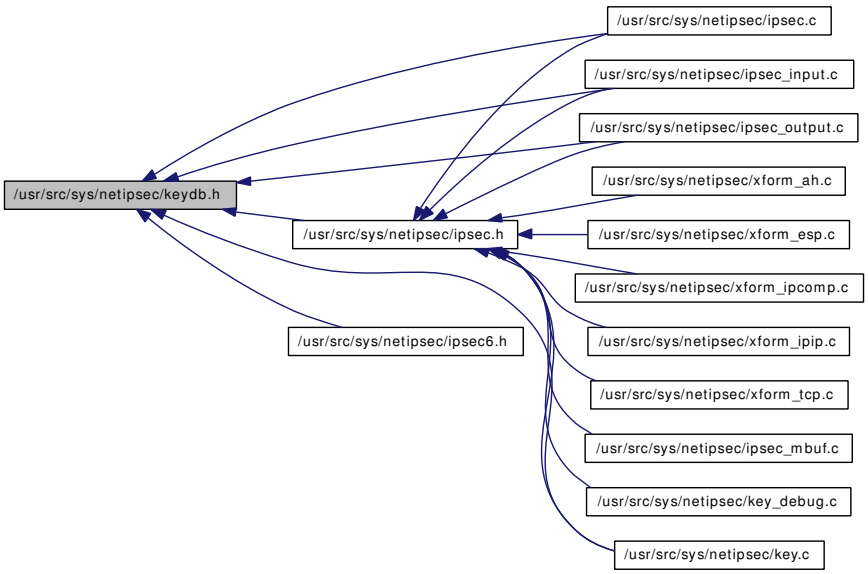
7.21 /usr/src/sys/netipsec/keydb.h File Reference

```
#include <netipsec/key_var.h>
```

Include dependency graph for keydb.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- union sockaddr_union
- struct secasindex
- struct secident
- struct seckey
- struct seclifetime
- struct secashead
- struct secasvar
- struct secreplay
- struct secreg
- struct secacq

Defines

- #define SECASVAR_LOCK_INIT(_sav) mtx_init(&(_sav) → lock, "ipsec association", NULL, MTX_DEF)
- #define SECASVAR_LOCK(_sav) mtx_lock(&(_sav) → lock)
- #define SECASVAR_UNLOCK(_sav) mtx_unlock(&(_sav) → lock)
- #define SECASVAR_LOCK_DESTROY(_sav) mtx_destroy(&(_sav) → lock)

- #define SECASVAR_LOCK_ASSERT(_sav) mtx_assert(&(_sav) → lock, MA_OWNED)
- #define SADB_KILL_INTERVAL 600

Functions

- secpolicy *keydb_newsecpolicy __P ((void))
- void keydb_delsecpolicy __P ((struct secpolicy *))
- void keydb_delsecashead __P ((struct secashead *))
- void keydb_refsecasvar __P ((struct secasvar *))
- secreplay *keydb_newsecreplay __P ((size_t))
- void keydb_delsecreplay __P ((struct secreplay *))
- void keydb_delsecreg __P ((struct secreg *))

7.21.1 Define Documentation

7.21.1.1 #define SADB_KILL_INTERVAL 600

Definition at line 194 of file keydb.h.

7.21.1.2 #define SECASVAR_LOCK(_sav) mtx_lock(&(_sav) → lock)

Definition at line 158 of file keydb.h.

7.21.1.3 #define SECASVAR_LOCK_ASSERT(_sav) mtx_assert(&(_sav) → lock, MA_OWNED)

Definition at line 161 of file keydb.h.

7.21.1.4 #define SECASVAR_LOCK_DESTROY(_sav) mtx_destroy(&(_sav) → lock)

Definition at line 160 of file keydb.h.

Referenced by key_delsav().

7.21.1.5 #define SECASVAR_LOCK_INIT(_sav) mtx_init(&(_sav) → lock, "ipsec association", NULL, MTX_DEF)

Definition at line 156 of file keydb.h.

Referenced by key_newsav().

7.21.1.6 #define SECASVAR_UNLOCK(_sav) mtx_unlock(&(_sav) → lock)

Definition at line 159 of file keydb.h.

7.21.2 Function Documentation

7.21.2.1 void keydb_delsecreg __P ((struct secreg *))

7.21.2.2 void keydb_delsecreplay __P ((struct secreplay *))

7.21.2.3 struct secreplay* keydb_newsecreplay __P ((size_t))

7.21.2.4 void keydb_refsecasvar __P ((struct secasvar *))

7.21.2.5 void keydb_delsecashead __P ((struct secashead *))

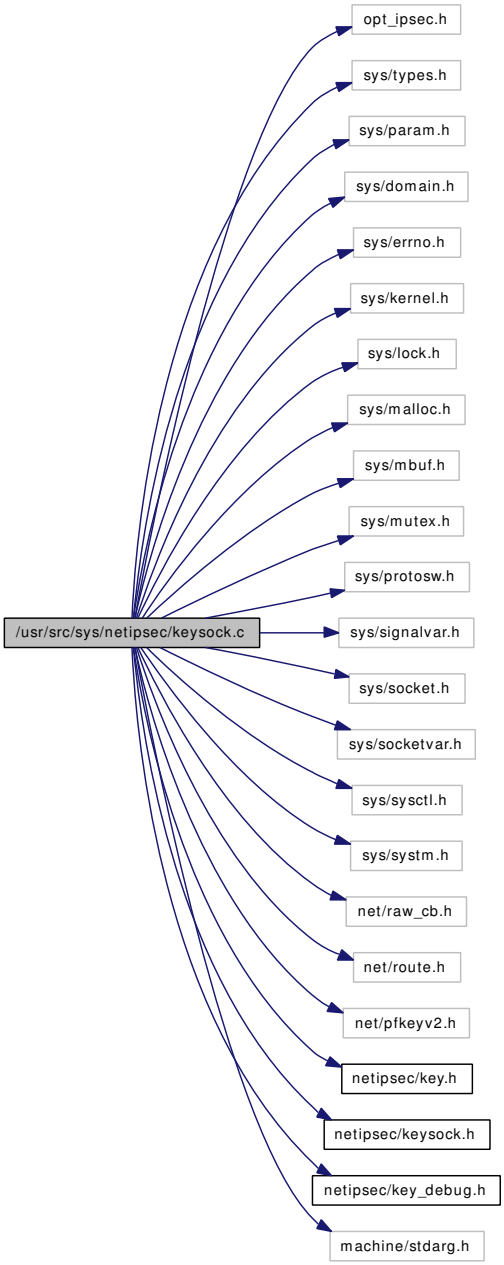
7.21.2.6 void keydb_delsecpolicy __P ((struct secpolicy *))

7.21.2.7 struct secpolicy* keydb_newsecpolicy __P ((void))

7.22 /usr/src/sys/netipsec/keysock.c File Reference

```
#include "opt_ipsec.h"
#include <sys/types.h>
#include <sys/param.h>
#include <sys/domain.h>
#include <sys/errno.h>
#include <sys/kernel.h>
#include <sys/lock.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/mutex.h>
#include <sys/protosw.h>
#include <sys/signalvar.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <sys/sysctl.h>
#include <sys/systm.h>
#include <net/raw_cb.h>
#include <net/route.h>
#include <net/pfkeyv2.h>
#include <netipsec/key.h>
#include <netipsec/keysock.h>
#include <netipsec/key_debug.h>
#include <machine/stdarg.h>
```

Include dependency graph for keysock.c:



Data Structures

- struct key_cb

Functions

- static int key_sendup0 __P ((struct rawcb *, struct mbuf *, int))
- int key_output (struct mbuf *m, struct socket *so)
- static int key_sendup0 (struct rawcb *rp, struct mbuf *m, int promisc)
- int key_sendup (struct socket *so, struct sadb_msg *msg, u_int len, int target)

- int key_sendup_mbuf (struct socket *so, struct mbuf *m, int target)
- static void key_abort (struct socket *so)
- static int key_attach (struct socket *so, int proto, struct thread *td)
- static int key_bind (struct socket *so, struct sockaddr *nam, struct thread *td)
- static void key_close (struct socket *so)
- static int key_connect (struct socket *so, struct sockaddr *nam, struct thread *td)
- static void key_detach (struct socket *so)
- static int key_disconnect (struct socket *so)
- static int key_peeraddr (struct socket *so, struct sockaddr **nam)
- static int key_send (struct socket *so, int flags, struct mbuf *m, struct sockaddr *nam, struct mbuf *control, struct thread *td)
- static int key_shutdown (struct socket *so)
- static int key_sockaddr (struct socket *so, struct sockaddr **nam)
- SYSCTL_NODE (_net, PF_KEY, key, CTLFLAG_RW, 0,"Key Family")
- static void key_init0 (void)
- DOMAIN_SET (key)

Variables

- static struct key_cb key_cb
- static struct sockaddr key_dst = { 2, PF_KEY, }
- static struct sockaddr key_src = { 2, PF_KEY, }
- pfkeystat pfkeystat
- pr_usrreqs key_usrreqs
- domain keydomain
- protosw keysw [ ]
- domain keydomain

7.22.1 Function Documentation

7.22.1.1 static int key_sendup0 __P ((struct rawcb *, struct mbuf *, int)) [static]

7.22.1.2 DOMAIN_SET (key)

7.22.1.3 static void key_abort (struct socket * *so*) [static]

Definition at line 373 of file keysock.c.

7.22.1.4 static int key_attach (struct socket * *so*, int *proto*, struct thread * *td*) [static]

Definition at line 384 of file keysock.c.

References key_cb::any_count, key_cb, key_cb::key_count, key_dst, and key_src.

7.22.1.5 static int key_bind (struct socket * *so*, struct sockaddr * *nam*, struct thread * *td*) [static]

Definition at line 432 of file keysock.c.

7.22.1.6 static void key_close (struct socket * *so*) [static]

Definition at line 446 of file keysock.c.

7.22.1.7 static int key_connect (struct socket * *so*, struct sockaddr * *nam*, struct thread * *td*) [static]

Definition at line 457 of file keysock.c.

7.22.1.8 static void key_detach (struct socket * *so*) [static]

Definition at line 471 of file keysock.c.

References key_cb::any_count, key_cb, key_cb::key_count, key_freereg(), and keycb::kp_raw.

Here is the call graph for this function:

7.22.1.9 static int key_disconnect (struct socket * *so*) [static]

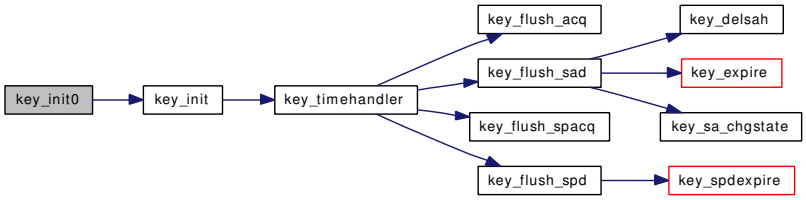
Definition at line 490 of file keysock.c.

7.22.1.10 static void key_init0 (void) [static]

Definition at line 593 of file keysock.c.

References key_cb, and key_init().

Here is the call graph for this function:

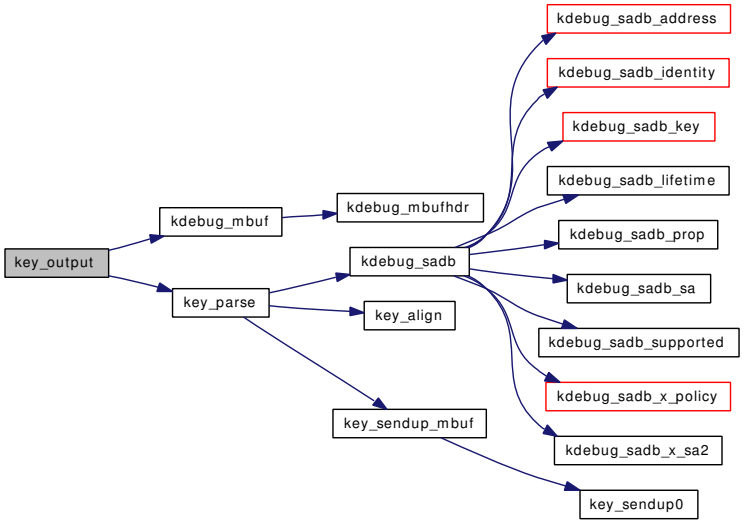


7.22.1.11 int key_output (struct mbuf * *m*, struct socket * *so*)

Definition at line 80 of file keysock.c.

References kdebug_mbuf(), key_parse(), KEYDEBUG, KEYDEBUG_KEY_DUMP, pfkeystat::out_bytes, pfkeystat::out_invlen, pfkeystat::out_msgtype, pfkeystat::out_nomem, pfkeystat::out_tooshort, pfkeystat::out_total, and pfkeystat.

Here is the call graph for this function:



7.22.1.12 static int key_peeraddr (struct socket * *so*, struct sockaddr ** *nam*) [static]

Definition at line 504 of file keysock.c.

7.22.1.13 static int key_send (struct socket * *so*, int *flags*, struct mbuf * *m*, struct sockaddr * *nam*, struct mbuf * *control*, struct thread * *td*) [static]

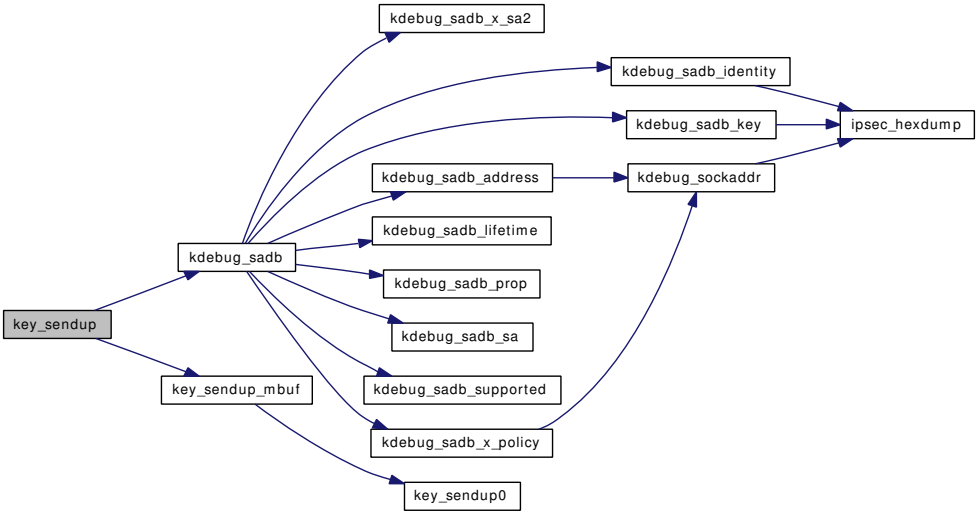
Definition at line 518 of file keysock.c.

7.22.1.14 int key_sendup (struct socket * *so*, struct sadb_msg * *msg*, u_int *len*, int *target*)

Definition at line 177 of file keysock.c.

References pfkeystat::in_bytes, pfkeystat::in_msgtype, pfkeystat::in_nomem, pfkeystat::in_total, kdebug_-sadb(), key_sendup_mbuf(), KEYDEBUG, KEYDEBUG_KEY_DUMP, and pfkeystat.

Here is the call graph for this function:



7.22.1.15 static int key_sendup0 (struct rawcb * *rp*, struct mbuf * *m*, int *promisc*) [static]

Definition at line 134 of file keysock.c.

References pfkeystat::in_msgtype, pfkeystat::in_nomem, key_src, and pfkeystat.

Referenced by key_sendup_mbuf().

7.22.1.16 int key_sendup_mbuf (struct socket * *so*, struct mbuf * *m*, int *target*)

Definition at line 262 of file keysock.c.

References pfkeystat::in_bytes, pfkeystat::in_msgtarget, pfkeystat::in_msgtype, pfkeystat::in_nomem, pfkeystat::in_total, key_sendup0(), KEY_SENDUP_ALL, KEY_SENDUP_ONE, KEY_SENDUP_REGISTERED, keycb::kp_promisc, keycb::kp_registered, and pfkeystat.

Referenced by key_acquire(), key_acquire2(), key_add(), key_delete(), key_delete_all(), key_do_allocsa_policy(), key_dump(), key_expire(), key_flush(), key_get(), key_getspi(), key_parse(), key_promisc(), key_register(), key_senderror(), key_sendup(), key_spdacquire(), key_spdadd(), key_spddelete(), key_spddelete2(), key_spddump(), key_spdexpire(), key_spdflush(), key_spdget(), and key_update().

Here is the call graph for this function:

7.22.1.17 static int key_shutdown (struct socket * *so*) [static]

Definition at line 533 of file keysock.c.

7.22.1.18 static int key_sockaddr (struct socket * *so*, struct sockaddr ** *nam*) [static]

Definition at line 547 of file keysock.c.

7.22.1.19 SYSCTL_NODE (_net, PF_KEY, key, CTLFLAG_RW, 0, "Key Family")

7.22.2 Variable Documentation

7.22.2.1 struct key_cb key_cb [static]

Definition at line 67 of file keysock.c.

Referenced by key_attach(), key_detach(), and key_init0().

7.22.2.2 struct sockaddr key_dst = { 2, PF_KEY, } [static]

Definition at line 69 of file keysock.c.

Referenced by key_attach().

7.22.2.3 struct sockaddr key_src = { 2, PF_KEY, } [static]

Definition at line 70 of file keysock.c.

Referenced by key_attach(), and key_sendup0().

7.22.2.4 struct pr_usrreqs key_usrreqs

Initial value:

```
{
        .pru_abort =            key_abort,
        .pru_attach =           key_attach,
        .pru_bind =             key_bind,
        .pru_connect =          key_connect,
        .pru_detach =           key_detach,
        .pru_disconnect =       key_disconnect,
        .pru_peeraddr =         key_peeraddr,
        .pru_send =             key_send,
        .pru_shutdown =         key_shutdown,
        .pru_sockaddr =         key_sockaddr,
        .pru_close =            key_close,
}
```

Definition at line 556 of file keysock.c.

7.22.2.5 struct domain keydomain

Initial value:

```
{
        .dom_family =           PF_KEY,
        .dom_name =             "key",
        .dom_init =             key_init0,
        .dom_protosw =          keysw,
        .dom_protoswNPROTOSW =  &keysw[sizeof(keysw)/sizeof(keysw[0])]
}
```

Definition at line 599 of file keysock.c.

7.22.2.6 struct domain keydomain

Definition at line 599 of file keysock.c.

7.22.2.7 struct protosw keysw[ ]

Initial value:

```
 {
{
        .pr_type =              SOCK_RAW,
        .pr_domain =            &keydomain,
        .pr_protocol =          PF_KEY_V2,
        .pr_flags =             PR_ATOMIC|PR_ADDR,
        .pr_output =            key_output,
        .pr_ctlinput =          raw_ctlinput,
        .pr_init =              raw_init,
        .pr_usrreqs =           &key_usrreqs
}
}
```

Definition at line 579 of file keysock.c.

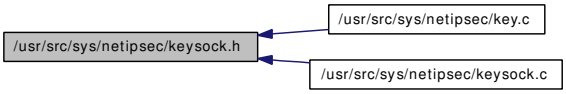
7.22.2.8 struct pfkeystat pfkeystat

Definition at line 74 of file keysock.c.

Referenced by key_output(), key_sendup(), key_sendup0(), and key_sendup_mbuf().

7.23 /usr/src/sys/netipsec/keysock.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct pfkeystat
- struct keycb

Defines

- #define KEY_SENDUP_ONE 0
- #define KEY_SENDUP_ALL 1
- #define KEY_SENDUP_REGISTERED 2

Functions

- int key_output (struct mbuf *m, struct socket *so)
- int key_usrreq __P ((struct socket *, int, struct mbuf *, struct mbuf *, struct mbuf *))
- int key_sendup __P ((struct socket *, struct sadb_msg *, u_int, int))
- int key_sendup_mbuf __P ((struct socket *, struct mbuf *, int))

Variables

- pfkeystat pfkeystat

7.23.1 Define Documentation

7.23.1.1 #define KEY_SENDUP_ALL 1

Definition at line 62 of file keysock.h.

Referenced by key_add(), key_delete(), key_delete_all(), key_flush(), key_promisc(), key_sendup_mbuf(), key_spdadd(), key_spddelete(), key_spddelete2(), key_spdflush(), and key_update().

7.23.1.2 #define KEY_SENDUP_ONE 0

Definition at line 61 of file keysock.h.

Referenced by key_dump(), key_get(), key_getspi(), key_parse(), key_senderror(), key_sendup_mbuf(), key_spddump(), and key_spdget().

7.23.1.3 #define KEY_SENDUP_REGISTERED 2

Definition at line 63 of file keysock.h.

Referenced by key_acquire(), key_acquire2(), key_do_allocsa_policy(), key_expire(), key_register(), key_sendup_mbuf(), key_spdacquire(), and key_spdexpire().

7.23.2 Function Documentation

7.23.2.1 int key_sendup_mbuf __P ((struct socket *, struct mbuf *, int))

7.23.2.2 int key_sendup __P ((struct socket *, struct sadb_msg *, u_int, int))

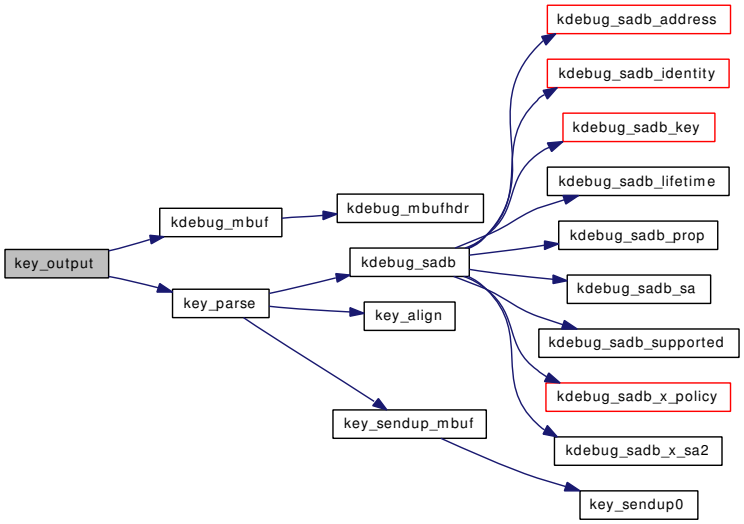
7.23.2.3 int key_usrreq __P ((struct socket *, int, struct mbuf *, struct mbuf *, struct mbuf *))

7.23.2.4 int key_output (struct mbuf * *m*, struct socket * *so*)

Definition at line 80 of file keysock.c.

References kdebug_mbuf(), key_parse(), KEYDEBUG, KEYDEBUG_KEY_DUMP, pfkeystat::out_bytes, pfkeystat::out_invlen, pfkeystat::out_msgtype, pfkeystat::out_nomem, pfkeystat::out_tooshort, pfkeystat::out_total, and pfkeystat.

Here is the call graph for this function:



7.23.3 Variable Documentation

7.23.3.1 struct pfkeystat pfkeystat

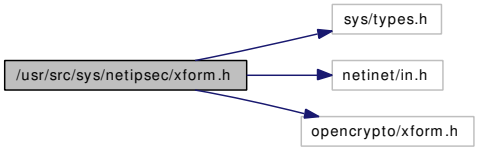
Definition at line 74 of file keysock.c.

Referenced by key_output(), key_sendup(), key_sendup0(), and key_sendup_mbuf().

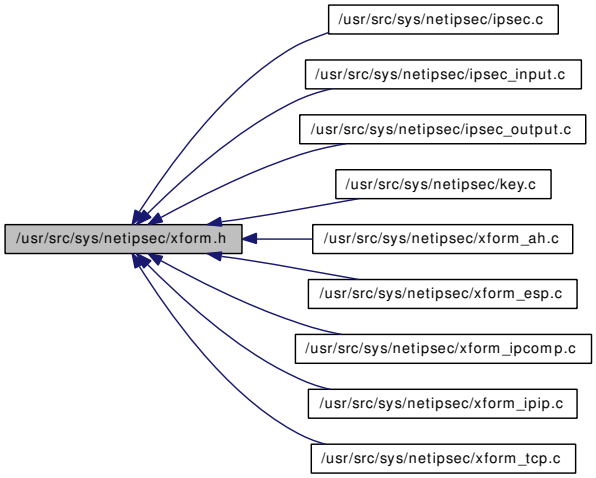
7.24 /usr/src/sys/netipsec/xform.h File Reference

```
#include <sys/types.h>
#include <netinet/in.h>
#include <opencrypto/xform.h>
```

Include dependency graph for xform.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct tdb_ident
- struct tdb_crypto
- struct xformsw

Defines

- #define AH_HMAC_HASHLEN 12
- #define AH_HMAC_INITIAL_RPL 1
- #define XF_IP4 1
- #define XF_AH 2
- #define XF_ESP 3
- #define XF_TCPSIGNATURE 5
- #define XF_IPCOMP 6
- #define XFT_AUTH 0x0001

- #define XFT_CONF 0x0100
- #define XFT_COMP 0x1000

Functions

- void xform_register (struct xformsw *)
- int xform_init (struct secasvar *sav, int xftype)
- int ip4_input6 (struct mbuf **m, int *offp, int proto)
- void ip4_input (struct mbuf *m, int)
- int ipip_output (struct mbuf *, struct ipsecrequest *, struct mbuf **, int, int)
- int ah_init0 (struct secasvar *, struct xformsw *, struct cryptoini *)
- int ah_zeroize (struct secasvar *sav)
- auth_hash * ah_algorithm_lookup (int alg)
- size_t ah_hdrsiz (struct secasvar *)
- enc_xform * esp_algorithm_lookup (int alg)
- size_t esp_hdrsiz (struct secasvar *sav)
- comp_algo * ipcomp_algorithm_lookup (int alg)

7.24.1 Define Documentation

7.24.1.1 #define AH_HMAC_HASHLEN 12

Definition at line 48 of file xform.h.

Referenced by esp_input(), esp_input_cb(), esp_output(), and esp_output_cb().

7.24.1.2 #define AH_HMAC_INITIAL_RPL 1

Definition at line 49 of file xform.h.

7.24.1.3 #define XF_AH 2

Definition at line 82 of file xform.h.

Referenced by key_mature(), and key_setsaval().

7.24.1.4 #define XF_ESP 3

Definition at line 83 of file xform.h.

Referenced by key_mature(), and key_setsaval().

7.24.1.5 #define XF_IP4 1

Definition at line 81 of file xform.h.

Referenced by ipip_output().

7.24.1.6 #define XF_IPCOMP 6

Definition at line 85 of file xform.h.

Referenced by key_mature(), and key_setsaval().

7.24.1.7 #define XF_TCPSIGNATURE 5

Definition at line 84 of file xform.h.

Referenced by key_mature(), and key_setsaval().

7.24.1.8 #define XFT_AUTH 0x0001

Definition at line 87 of file xform.h.

7.24.1.9 #define XFT_COMP 0x1000

Definition at line 89 of file xform.h.

7.24.1.10 #define XFT_CONF 0x0100

Definition at line 88 of file xform.h.

7.24.2 Function Documentation

7.24.2.1 struct auth_hash* ah_algorithm_lookup (int *alg*)

Definition at line 111 of file xform_ah.c.

References AH_ALG_MAX.

Referenced by ah_init0(), key_getcomb_ah(), and key_register().

7.24.2.2 size_t ah_hdrsiz (struct secasvar *)

Definition at line 139 of file xform_ah.c.

References AUTHSIZE, HDRSIZE, IPSEC_ASSERT, and secasvar::tdb_authalgxform.

Referenced by esp_hdrsiz(), and ipsec_hdrsiz().

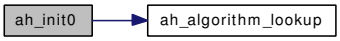
7.24.2.3 int ah_init0 (struct secasvar *, struct xformsw *, struct cryptoini *)

Definition at line 160 of file xform_ah.c.

References _KEYBITS, _KEYLEN, ah_algorithm_lookup(), secasvar::alg_auth, AUTHSIZE, DPRINTF, secasvar::flags, secasvar::key_auth, seckey::key_data, secasvar::replay, secasvar::tdb_authalgxform, and secasvar::tdb_xform.

Referenced by ah_init(), and esp_init().

Here is the call graph for this function:



7.24.2.4 int ah_zeroize (struct secasvar * *sav*)

Definition at line 230 of file xform_ah.c.

References _KEYLEN, secasvar::key_auth, seckey::key_data, secasvar::tdb_authalgxform, secasvar::tdb_-cryptoid, and secasvar::tdb_xform.

Referenced by esp_zeroize().

7.24.2.5 struct enc_xform* esp_algorithm_lookup (int *alg*)

Definition at line 97 of file xform_esp.c.

References ESP_ALG_MAX.

Referenced by esp_init(), key_getcomb_esp(), and key_register().

7.24.2.6 size_t esp_hdrsiz (struct secasvar * *sav*)

Definition at line 121 of file xform_esp.c.

References ah_hdrsiz(), esp_max_ivlen, secasvar::flags, IPSEC_ASSERT, secasvar::replay, secasvar::tdb_authalgxform, and secasvar::tdb_encalgxform.

Referenced by ipsec_hdrsiz().

Here is the call graph for this function:

7.24.2.7 void ip4_input (struct mbuf * *m*, int)

7.24.2.8 int ip4_input6 (struct mbuf ** *m*, int * *offp*, int *proto*)

7.24.2.9 struct comp_algo* ipcomp_algorithm_lookup (int *alg*)

Definition at line 82 of file xform_ipcomp.c.

References IPCOMP_ALG_MAX.

Referenced by ipcomp_init(), and key_getcomb_ipcomp().

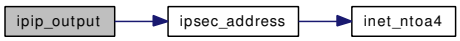
7.24.2.10 int ipip_output (struct mbuf *, struct ipsecrequest *, struct mbuf **, int, int)

Definition at line 386 of file xform_ipip.c.

References DPRINTF, secasindex::dst, ipipstat::ipips_family, ipipstat::ipips_hdrops, ipipstat::ipips_-obytes, ipipstat::ipips_opackets, ipipstat::ipips_unspec, ipipstat, ipsec_address(), IPSEC_ASSERT,

IPSEC_SPLASSERT_SOFTNET, sockaddr_union::sa, secasvar::sah, secashead::saidx, ipsecrequest::sav, sockaddr_union::sin, sockaddr_union::sin6, secasvar::spi, secasindex::src, secasvar::tdb_xform, XF_IP4, and xformsw::xf_type.

Here is the call graph for this function:



7.24.2.11 int xform_init (struct secasvar * *sav*, int *xftype*)

Definition at line 1943 of file ipsec.c.

References secasvar::tdb_xform, xformsw::xf_init, xformsw::xf_next, xformsw::xf_type, and xforms.

Referenced by key_mature(), and key_setsaval().

7.24.2.12 void xform_register (struct xformsw *)

Definition at line 1933 of file ipsec.c.

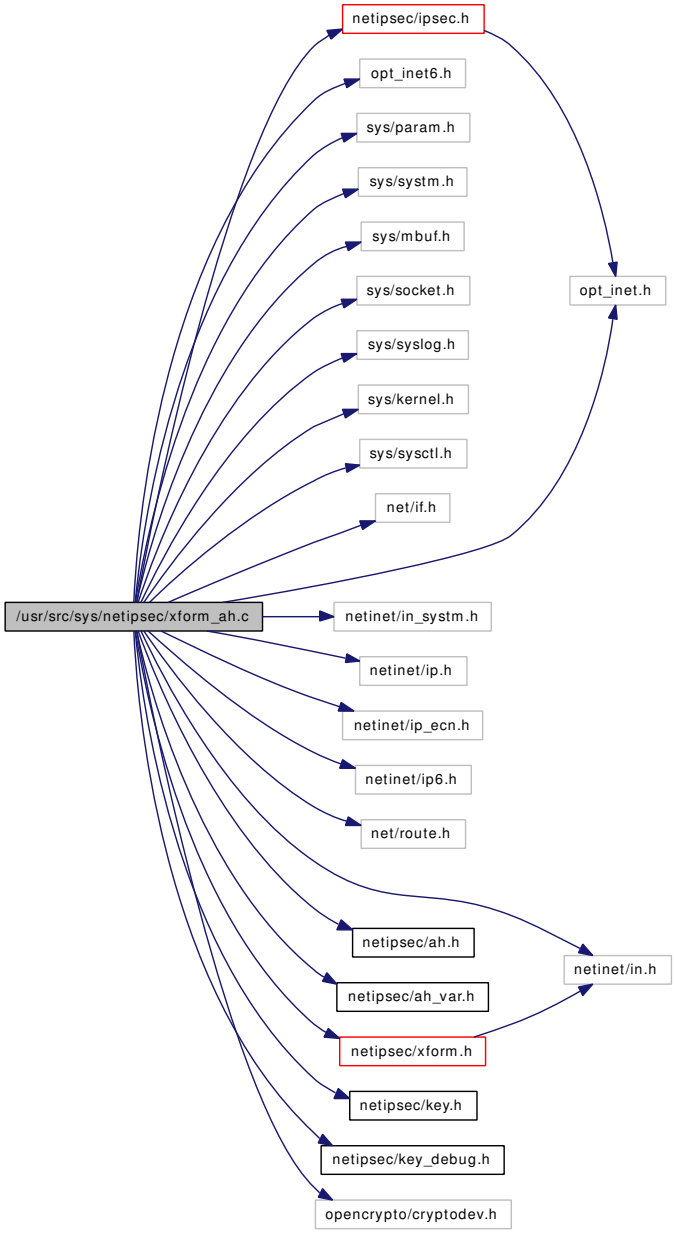
References xforms.

Referenced by ah_attach(), esp_attach(), ipcomp_attach(), and tcpsignature_attach().

7.25 /usr/src/sys/netipsec/xform_ah.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/mbuf.h>
#include <sys/socket.h>
#include <sys/syslog.h>
#include <sys/kernel.h>
#include <sys/sysctl.h>
#include <net/if.h>
#include <netinet/in.h>
#include <netinet/in_systm.h>
#include <netinet/ip.h>
#include <netinet/ip_ecn.h>
#include <netinet/ip6.h>
#include <net/route.h>
#include <netipsec/ipsec.h>
#include <netipsec/ah.h>
#include <netipsec/ah_var.h>
#include <netipsec/xform.h>
#include <netipsec/key.h>
#include <netipsec/key_debug.h>
#include <opencrypto/cryptodev.h>
```

Include dependency graph for xform_ah.c:



Defines

- #define HDRSIZE(sav)
- #define AUTHSIZE(sav) ((sav → flags & SADB_X_EXT_OLD) ? 16 : AH_HMAC_HASHLEN)
- #define IPSEC_COMMON_INPUT_CB(m, sav, skip, protoff, mtag) (error = ipsec4_common_-input_cb(m, sav, skip, protoff, mtag))

Functions

- SYSCTL_DECL (_net_inet_ah)
- SYSCTL_INT (_net_inet_ah, OID_AUTO, ah_enable, CTLFLAG_RW,&ah_enable, 0,"")

- SYSCTL_INT (_net_inet_ah, OID_AUTO, ah_cleartos, CTLFLAG_RW,&ah_cleartos, 0,"")
- SYSCTL_STRUCT (_net_inet_ah, IPSECCTL_STATS, stats, CTLFLAG_RD,&ahstat, ahstat,"")
- static int ah_input_cb (struct cryptop *)
- static int ah_output_cb (struct cryptop *)
- auth_hash * ah_algorithm_lookup (int alg)
- size_t ah_hdrsiz (struct secasvar *sav)
- int ah_init0 (struct secasvar *sav, struct xformsw *xsp, struct cryptoini *cria)
- static int ah_init (struct secasvar *sav, struct xformsw *xsp)
- int ah_zeroize (struct secasvar *sav)
- static int ah_massage_headers (struct mbuf **m0, int proto, int skip, int alg, int out)
- static int ah_input (struct mbuf *m, struct secasvar *sav, int skip, int protoff)
- static int ah_output (struct mbuf *m, struct ipsecrequest *isr, struct mbuf **mp, int skip, int protoff)
- static void ah_attach (void)
- SYSINIT (ah_xform_init, SI_SUB_PROTO_DOMAIN, SI_ORDER_MIDDLE, ah_attach, NULL)

Variables

- int ah_enable = 1
- int ah_cleartos = 1
- ahstat ahstat
- static unsigned char ipseczeroes [256]
- static struct xformsw ah_xformsw

7.25.1 Define Documentation

7.25.1.1 #define AUTHSIZE(sav) ((sav → flags & SADB_X_EXT_OLD) ? 16 : AH_HMAC_HASHLEN)

Definition at line 87 of file xform_ah.c.

Referenced by ah_hdrsiz(), ah_init0(), ah_input(), ah_input_cb(), ah_output(), and ah_output_cb().

7.25.1.2 #define HDRSIZE(sav)

Value:

```
(((sav)->flags & SADB_X_EXT_OLD) ? \
                sizeof (struct ah) : sizeof (struct ah) + sizeof (u_int32_t))
```

Definition at line 79 of file xform_ah.c.

Referenced by ah_hdrsiz(), ah_input(), ah_input_cb(), and ah_output().

7.25.1.3 #define IPSEC_COMMON_INPUT_CB(m, sav, skip, protoff, mtag) (error = ipsec4_common_input_cb(m, sav, skip, protoff, mtag))

Definition at line 715 of file xform_ah.c.

Referenced by ah_input_cb(), esp_input_cb(), and ipcomp_input_cb().

7.25.2 Function Documentation

7.25.2.1 struct auth_hash* ah_algorithm_lookup (int *alg*)

Definition at line 111 of file xform_ah.c.

References AH_ALG_MAX.

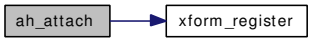
Referenced by ah_init0(), key_getcomb_ah(), and key_register().

7.25.2.2 static void ah_attach (void) [static]

Definition at line 1224 of file xform_ah.c.

References ah_xformsw, and xform_register().

Here is the call graph for this function:



7.25.2.3 size_t ah_hdrsiz (struct secasvar * *sav*)

Definition at line 139 of file xform_ah.c.

References AUTHSIZE, HDRSIZE, IPSEC_ASSERT, and secasvar::tdb_authalgxform.

Referenced by esp_hdrsiz(), and ipsec_hdrsiz().

7.25.2.4 static int ah_init (struct secasvar * *sav*, struct xformsw * *xsp*) [static]

Definition at line 214 of file xform_ah.c.

References ah_init0(), crypto_support, and secasvar::tdb_cryptoid.

Here is the call graph for this function:

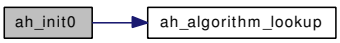
7.25.2.5 int ah_init0 (struct secasvar * *sav*, struct xformsw * *xsp*, struct cryptoini * *cria*)

Definition at line 160 of file xform_ah.c.

References _KEYBITS, _KEYLEN, ah_algorithm_lookup(), secasvar::alg_auth, AUTHSIZE, DPRINTF, secasvar::flags, secasvar::key_auth, seckey::key_data, secasvar::replay, secasvar::tdb_authalgxform, and secasvar::tdb_xform.

Referenced by ah_init(), and esp_init().

Here is the call graph for this function:

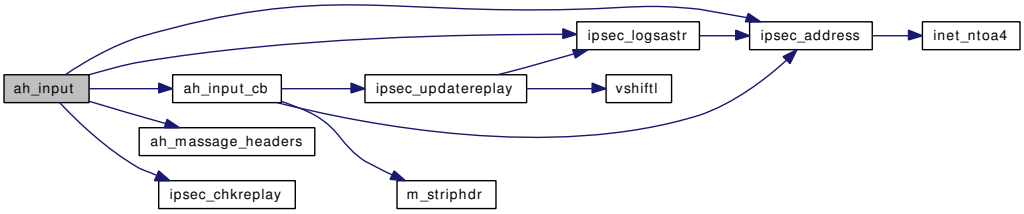


7.25.2.6 static int ah_input (struct mbuf * *m*, struct secasvar * *sav*, int *skip*, int *protoff*) [static]

Definition at line 552 of file xform_ah.c.

References _KEYBITS, ah_input_cb(), newah::ah_len, ah_massage_headers(), newah::ah_seq, ahstat::ahs_badauthl, ahstat::ahs_crypto, ahstat::ahs_hdrops, ahstat::ahs_ibytes, ahstat::ahs_replay, ahstat, AUTHSIZE, DPRINTF, tdb_ident::dst, secasindex::dst, HDRSIZE, ipsec_address(), IPSEC_ASSERT, ipsec_chkreplay(), ipsec_logsastr(), IPSEC_SPLASSERT_SOFTNET, ipseczeroes, secasvar::key_auth, seckey::key_data, secasindex::proto, tdb_ident::proto, secasvar::replay, sockaddr_union::sa, secasvar::sah, secashead::saidx, tdb_ident::spi, secasvar::spi, tdb_crypto::tc_dst, tdb_crypto::tc_nxt, tdb_crypto::tc_proto, tdb_crypto::tc_protoff, tdb_crypto::tc_ptr, tdb_crypto::tc_skip, tdb_crypto::tc_spi, secasvar::tdb_authalgxform, and secasvar::tdb_cryptoid.

Here is the call graph for this function:



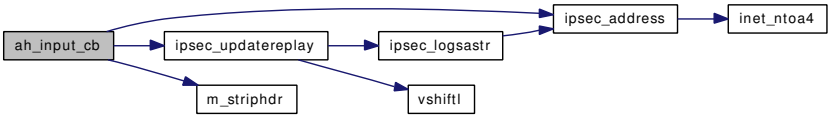
7.25.2.7 static int ah_input_cb (struct cryptop *) [static]

Definition at line 723 of file xform_ah.c.

References ahstat::ahs_badauth, ahstat::ahs_crypto, ahstat::ahs_hdrops, ahstat::ahs_hist, ahstat::ahs_notdb, ahstat::ahs_noxform, ahstat::ahs_replay, ahstat, secasvar::alg_auth, AUTHSIZE, DPRINTF, secasindex::dst, HDRSIZE, ipsec_address(), IPSEC_ASSERT, IPSEC_COMMON_INPUT_CB, ipsec_updatereplay(), KEY_ALLOCSA, KEY_FREESAV, m_striphdr(), secasvar::replay, sockaddr_union::sa, secasvar::sah, secashead::saidx, secasvar::spi, tdb_crypto::tc_dst, tdb_crypto::tc_nxt, tdb_crypto::tc_proto, tdb_crypto::tc_protoff, tdb_crypto::tc_ptr, tdb_crypto::tc_skip, tdb_crypto::tc_spi, secasvar::tdb_authalgxform, and secasvar::tdb_cryptoid.

Referenced by ah_input().

Here is the call graph for this function:



7.25.2.8 static int ah_massage_headers (struct mbuf ** *m0*, int *proto*, int *skip*, int *alg*, int *out*) [static]

Definition at line 248 of file xform_ah.c.

References ah_cleartos, DPRINTF, and ipseczeroes.

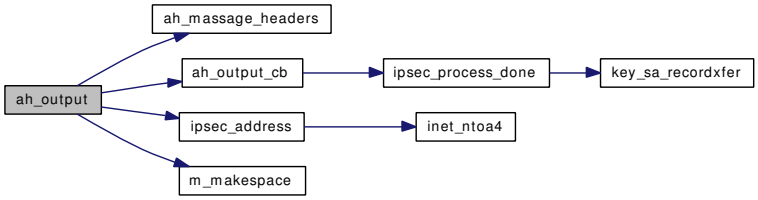
Referenced by ah_input(), and ah_output().

7.25.2.9 static int ah_output (struct mbuf * *m*, struct ipsecrequest * *isr*, struct mbuf ** *mp*, int *skip*, int *protoff*) [static]

Definition at line 885 of file xform_ah.c.

References _KEYBITS, newah::ah_len, ah_massage_headers(), newah::ah_nxt, ah_output_cb(), newah::ah_reserve, newah::ah_seq, newah::ah_spi, ahstat::ahs_crypto, ahstat::ahs_hdrops, ahstat::ahs_nopf, ahstat::ahs_obytes, ahstat::ahs_output, ahstat::ahs_toobig, ahstat::ahs_wrap, ahstat, AUTHSIZE, sec replay::count, DPRINTF, secasindex::dst, secasvar::flags, HDRSIZE, ipsec_address(), IPSEC_ASSERT, IPSEC_SPLASSERT_SOFTNET, ipseczeroes, secasvar::key_auth, seckey::key_data, m_makespace(), secasindex::proto, secasvar::replay, sockaddr_union::sa, secasvar::sah, secashead::saidx, ipsecrequest::sav, secasvar::spi, tdb_crypto::tc_dst, tdb_crypto::tc_isr, tdb_crypto::tc_proto, tdb_crypto::tc_protoff, tdb_crypto::tc_skip, tdb_crypto::tc_spi, secasvar::tdb_authalgxform, and secasvar::tdb_cryptoid.

Here is the call graph for this function:



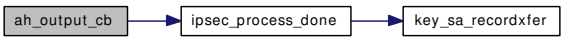
7.25.2.10 static int ah_output_cb (struct cryptop *) [static]

Definition at line 1118 of file xform_ah.c.

References ahstat::ahs_crypto, ahstat::ahs_hist, ahstat::ahs_notdb, ahstat::ahs_noxform, ahstat, secasvar::alg_auth, AUTHSIZE, DPRINTF, IPSEC_ASSERT, ipsec_process_done(), IPSECREQUEST_LOCK, IPSECREQUEST_UNLOCK, ipseczeroes, KEY_ALLOCSA, KEY_FREESAV, ipsecrequest::sav, tdb_crypto::tc_dst, tdb_crypto::tc_isr, tdb_crypto::tc_proto, tdb_crypto::tc_protoff, tdb_crypto::tc_skip, tdb_crypto::tc_spi, and secasvar::tdb_cryptoid.

Referenced by ah_output().

Here is the call graph for this function:



7.25.2.11 int ah_zeroize (struct secasvar * *sav*)

Definition at line 230 of file xform_ah.c.

References _KEYLEN, secasvar::key_auth, seckey::key_data, secasvar::tdb_authalgxform, secasvar::tdb_cryptoid, and secasvar::tdb_xform.

Referenced by esp_zeroize().

7.25.2.12 SYSCTL_DECL (_net_inet_ah)

7.25.2.13 SYSCTL_INT (_net_inet_ah, OID_AUTO, ah_cleartos, CTLFLAG_RW, & *ah_cleartos*, 0, "")

7.25.2.14 SYSCTL_INT (_net_inet_ah, OID_AUTO, ah_enable, CTLFLAG_RW, & *ah_enable*, 0, "")

7.25.2.15 SYSCTL_STRUCT (_net_inet_ah, IPSECCTL_STATS, stats, CTLFLAG_RD, & *ahstat*, ahstat, "")

7.25.2.16 SYSINIT (ah_xform_init, SI_SUB_PROTO_DOMAIN, SI_ORDER_MIDDLE, ah_attach, NULL)

7.25.3 Variable Documentation

7.25.3.1 int ah_cleartos = 1

Definition at line 91 of file xform_ah.c.

Referenced by ah_massage_headers().

7.25.3.2 int ah_enable = 1

Definition at line 90 of file xform_ah.c.

Referenced by ipsec_common_input(), and ipsec_nextisr().

7.25.3.3 struct xformsw ah_xformsw [static]

Initial value:

```
{
        XF_AH,          XFT_AUTH,       "IPsec AH",
        ah_init,        ah_zeroize,     ah_input,       ah_output,
}
```

Definition at line 1218 of file xform_ah.c.

Referenced by ah_attach().

7.25.3.4 struct ahstat ahstat

Definition at line 92 of file xform_ah.c.

Referenced by ah_input(), ah_input_cb(), ah_output(), and ah_output_cb().

7.25.3.5 unsigned char ipseczeroes[256] [static]

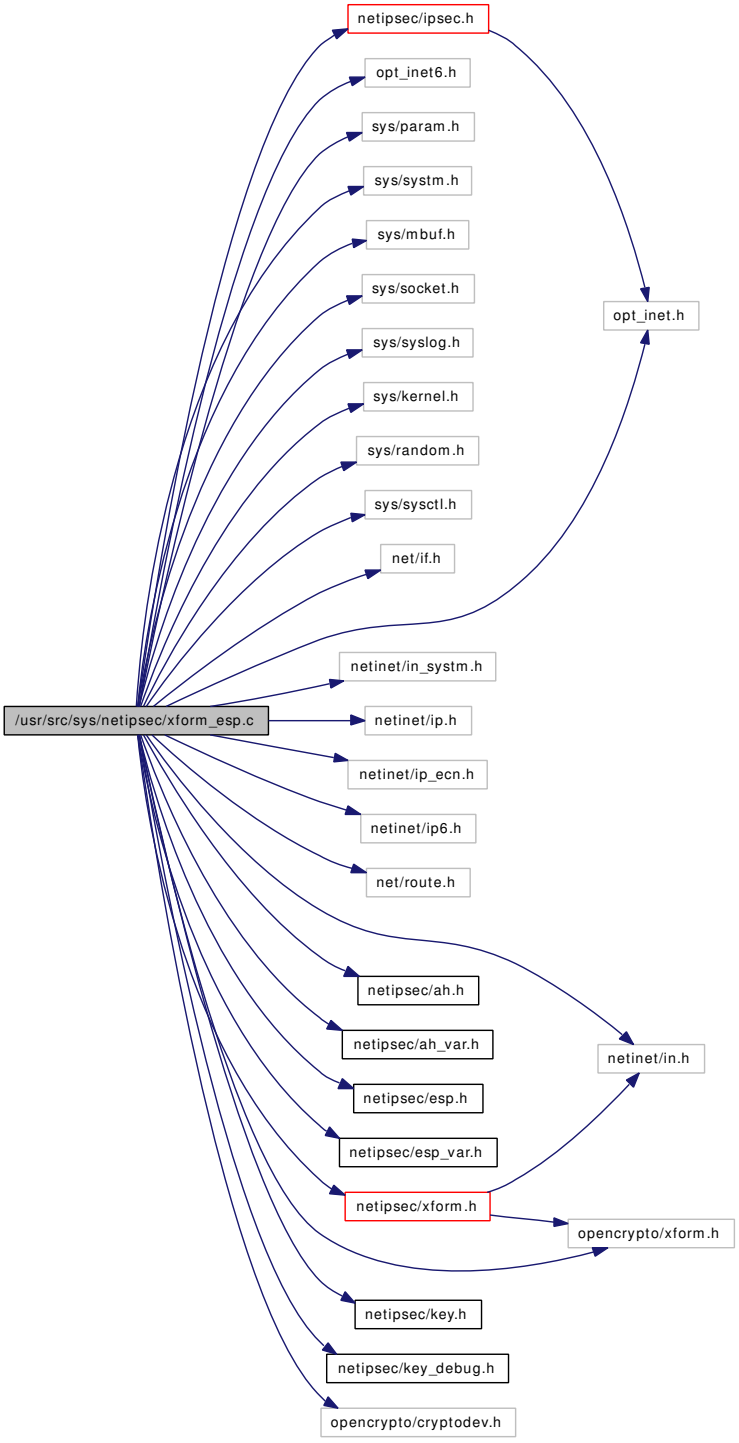
Definition at line 102 of file xform_ah.c.

Referenced by ah_input(), ah_massage_headers(), ah_output(), ah_output_cb(), and esp_output_cb().

7.26 /usr/src/sys/netipsec/xform_esp.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/mbuf.h>
#include <sys/socket.h>
#include <sys/syslog.h>
#include <sys/kernel.h>
#include <sys/random.h>
#include <sys/sysctl.h>
#include <net/if.h>
#include <netinet/in.h>
#include <netinet/in_systm.h>
#include <netinet/ip.h>
#include <netinet/ip_ecn.h>
#include <netinet/ip6.h>
#include <net/route.h>
#include <netipsec/ipsec.h>
#include <netipsec/ah.h>
#include <netipsec/ah_var.h>
#include <netipsec/esp.h>
#include <netipsec/esp_var.h>
#include <netipsec/xform.h>
#include <netipsec/key.h>
#include <netipsec/key_debug.h>
#include <opencrypto/cryptodev.h>
#include <opencrypto/xform.h>
```

Include dependency graph for xform_esp.c:



Defines

- #define IPSEC_COMMON_INPUT_CB(m, sav, skip, protoff, mtag) (error = ipsec4_common_-input_cb(m, sav, skip, protoff, mtag))
- #define MAXIV(xform)

Functions

- SYSCTL_DECL (_net_inet_esp)
- SYSCTL_INT (_net_inet_esp, OID_AUTO, esp_enable, CTLFLAG_RW,&esp_enable, 0,"")
- SYSCTL_STRUCT (_net_inet_esp, IPSECCTL_STATS, stats, CTLFLAG_RD,&espstat, espstat,"")
- static int esp_input_cb (struct cryptop *op)
- static int esp_output_cb (struct cryptop *crp)
- enc_xform * esp_algorithm_lookup (int alg)
- size_t esp_hdrsiz (struct secasvar *sav)
- static int esp_init (struct secasvar *sav, struct xformsw *xsp)
- static int esp_zeroize (struct secasvar *sav)
- static int esp_input (struct mbuf *m, struct secasvar *sav, int skip, int protoff)
- static int esp_output (struct mbuf *m, struct ipsecrequest *isr, struct mbuf **mp, int skip, int protoff)
- static void esp_attach (void)
- SYSINIT (esp_xform_init, SI_SUB_PROTO_DOMAIN, SI_ORDER_MIDDLE, esp_attach, NULL)

Variables

- int esp_enable = 1
- espstat espstat
- static int esp_max_ivlen
- static struct xformsw esp_xformsw

7.26.1 Define Documentation

7.26.1.1 #define IPSEC_COMMON_INPUT_CB(m, sav, skip, protoff, mtag) (error = ipsec4_common_input_cb(m, sav, skip, protoff, mtag))

Definition at line 441 of file xform_esp.c.

7.26.1.2 #define MAXIV(xform)

Value:

```
if (xform.blocksize > esp_max_ivlen)            \
                esp_max_ivlen = xform.blocksize         \
```

Referenced by esp_attach().

7.26.2 Function Documentation

7.26.2.1 struct enc_xform* esp_algorithm_lookup (int *alg*)

Definition at line 97 of file xform_esp.c.

References ESP_ALG_MAX.

Referenced by esp_init(), key_getcomb_esp(), and key_register().

7.26.2.2 static void esp_attach (void) [static]

Definition at line 995 of file xform_esp.c.

References esp_max_ivlen, esp_xformsw, MAXIV, and xform_register().

Here is the call graph for this function:

7.26.2.3 size_t esp_hdrsiz (struct secasvar * *sav*)

Definition at line 121 of file xform_esp.c.

References ah_hdrsiz(), esp_max_ivlen, secasvar::flags, IPSEC_ASSERT, secasvar::replay, secasvar::tdb_authalgxform, and secasvar::tdb_encalgxform.

Referenced by ipsec_hdrsiz().

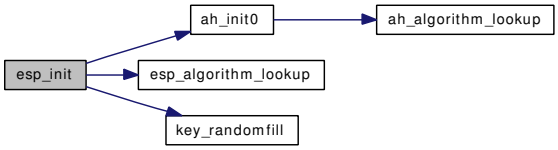
Here is the call graph for this function:

7.26.2.4 static int esp_init (struct secasvar * *sav*, struct xformsw * *xsp*) [static]

Definition at line 155 of file xform_esp.c.

References _KEYBITS, _KEYLEN, ah_init0(), secasvar::alg_auth, secasvar::alg_enc, crypto_support, DPRINTF, esp_algorithm_lookup(), secasvar::flags, secasvar::iv, secasvar::ivlen, seckey::key_data, secasvar::key_enc, key_randomfill(), secasvar::tdb_authalgxform, secasvar::tdb_cryptoid, secasvar::tdb_encalgxform, and secasvar::tdb_xform.

Here is the call graph for this function:



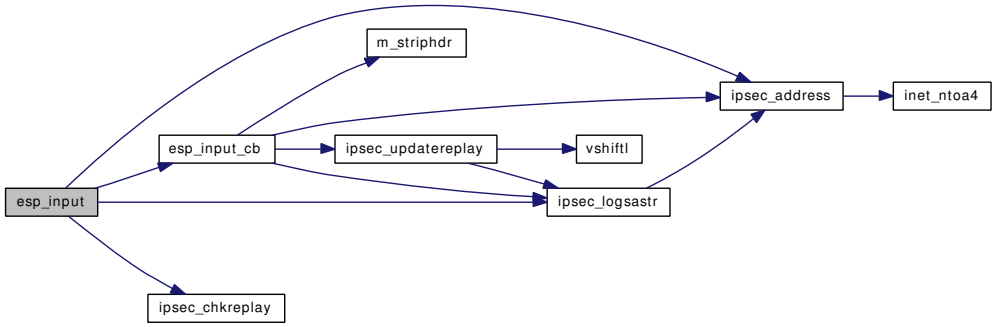
7.26.2.5 static int esp_input (struct mbuf * *m*, struct secasvar * *sav*, int *skip*, int *protoff*) [static]

Definition at line 265 of file xform_esp.c.

References _KEYBITS, AH_HMAC_HASHLEN, DPRINTF, tdb_ident::dst, esp_input_cb(), newesp::esp_seq, espstat::esps_badilen, espstat::esps_crypto, espstat::esps_ibytes, espstat::esps_replay, espstat, secasvar::flags, ipsec_address(), IPSEC_ASSERT, ipsec_chkreplay(), ipsec_logsastr(), IPSEC_SPLASSERT_SOFTNET, tdb_ident::proto, tdb_ident::spi, tdb_crypto::tc_dst, tdb_crypto::tc_proto,

tdb_crypto::tc_protoff, tdb_crypto::tc_ptr, tdb_crypto::tc_skip, tdb_crypto::tc_spi, secasvar::tdb_-authalgxform, and secasvar::tdb_encalgxform.

Here is the call graph for this function:



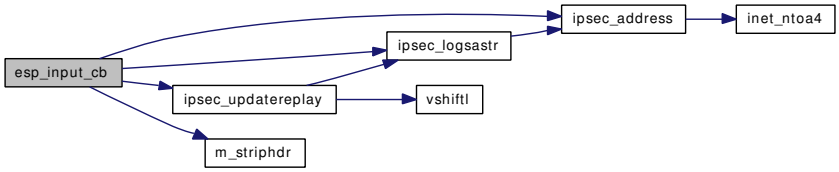
7.26.2.6 static int esp_input_cb (struct cryptop * *op*) [static]

Definition at line 449 of file xform_esp.c.

References AH_HMAC_HASHLEN, secasvar::alg_auth, secasvar::alg_enc, DPRINTF, secasindex::dst, espstat::esps_badauth, espstat::esps_badenc, espstat::esps_badilen, espstat::esps_crypto, espstat::esps_-hdrops, espstat::esps_hist, espstat::esps_notdb, espstat::esps_noxform, espstat::esps_replay, esp-stat, secasvar::flags, ipsec_address(), IPSEC_ASSERT, IPSEC_COMMON_INPUT_CB, ipsec_-logsastr(), ipsec_updatereplay(), KEY_ALLOCSA, KEY_FREESAV, m_striphdr(), secasvar::replay, sockaddr_union::sa, secasvar::sah, secashead::saidx, secasvar::spi, tdb_crypto::tc_dst, tdb_crypto::tc_-proto, tdb_crypto::tc_protoff, tdb_crypto::tc_ptr, tdb_crypto::tc_skip, tdb_crypto::tc_spi, secasvar::tdb_-authalgxform, secasvar::tdb_cryptoid, and secasvar::tdb_encalgxform.

Referenced by esp_input().

Here is the call graph for this function:



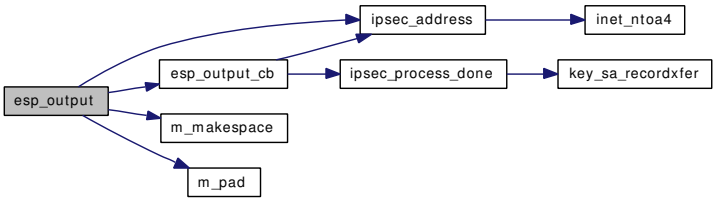
7.26.2.7 static int esp_output (struct mbuf * *m*, struct ipsecrequest * *isr*, struct mbuf ** *mp*, int *skip*, int *protoff*) [static]

Definition at line 651 of file xform_esp.c.

References _KEYBITS, AH_HMAC_HASHLEN, DPRINTF, secasindex::dst, esp_output_cb(), espstat::esps_crypto, espstat::esps_hdrops, espstat::esps_nopf, espstat::esps_obytes, espstat::esps_output, espstat::esps_toobig, espstat, secasvar::flags, ipsec_address(), IPSEC_ASSERT, IPSEC_SPLASSERT_-SOFTNET, m_makespace(), m_pad(), secasindex::proto, sockaddr_union::sa, ipsecrequest::sav, tdb_-

crypto::tc_dst, tdb_crypto::tc_isr, tdb_crypto::tc_proto, tdb_crypto::tc_spi, secasvar::tdb_authalgxform, and secasvar::tdb_encalgxform.

Here is the call graph for this function:



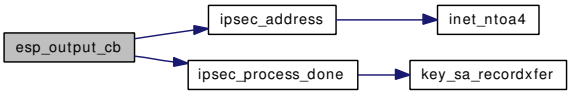
7.26.2.8 static int esp_output_cb (struct cryptop * *crp*) [static]

Definition at line 889 of file xform_esp.c.

References AH_HMAC_HASHLEN, secasvar::alg_auth, secasvar::alg_enc, DPRINTF, espstat::esps_-crypto, espstat::esps_hist, espstat::esps_notdb, espstat::esps_noxform, espstat, ipsec_address(), IPSEC_-ASSERT, ipsec_process_done(), IPSECREQUEST_LOCK, IPSECREQUEST_UNLOCK, ipseczeroes, KEY_ALLOCSA, KEY_FREESAV, ipsecrequest::sav, tdb_crypto::tc_dst, tdb_crypto::tc_isr, tdb_-crypto::tc_proto, tdb_crypto::tc_spi, secasvar::tdb_authalgxform, and secasvar::tdb_cryptoid.

Referenced by esp_output().

Here is the call graph for this function:



7.26.2.9 static int esp_zeroize (struct secasvar * *sav*) [static]

Definition at line 245 of file xform_esp.c.

References _KEYLEN, ah_zeroize(), secasvar::iv, seckey::key_data, secasvar::key_enc, secasvar::tdb_-encalgxform, and secasvar::tdb_xform.

Here is the call graph for this function:

7.26.2.10 SYSCTL_DECL (_net_inet_esp)

7.26.2.11 SYSCTL_INT (_net_inet_esp, OID_AUTO, esp_enable, CTLFLAG_RW, & *esp_enable*, 0, "")

7.26.2.12 SYSCTL_STRUCT (_net_inet_esp, IPSECCTL_STATS, stats, CTLFLAG_RD, & *espstat*, espstat, "")

7.26.2.13 SYSINIT (esp_xform_init, SI_SUB_PROTO_DOMAIN, SI_ORDER_MIDDLE, esp_attach, NULL)

7.26.3 Variable Documentation

7.26.3.1 int esp_enable = 1

Definition at line 78 of file xform_esp.c.

Referenced by ipsec_common_input(), and ipsec_nextisr().

7.26.3.2 int esp_max_ivlen [static]

Definition at line 87 of file xform_esp.c.

Referenced by esp_attach(), and esp_hdrsiz().

7.26.3.3 struct xformsw esp_xformsw [static]

Initial value:

```
{
        XF_ESP,         XFT_CONF|XFT_AUTH,      "IPsec ESP",
        esp_init,       esp_zeroize,            esp_input,
        esp_output
}
```

Definition at line 988 of file xform_esp.c.

Referenced by esp_attach().

7.26.3.4 struct espstat espstat

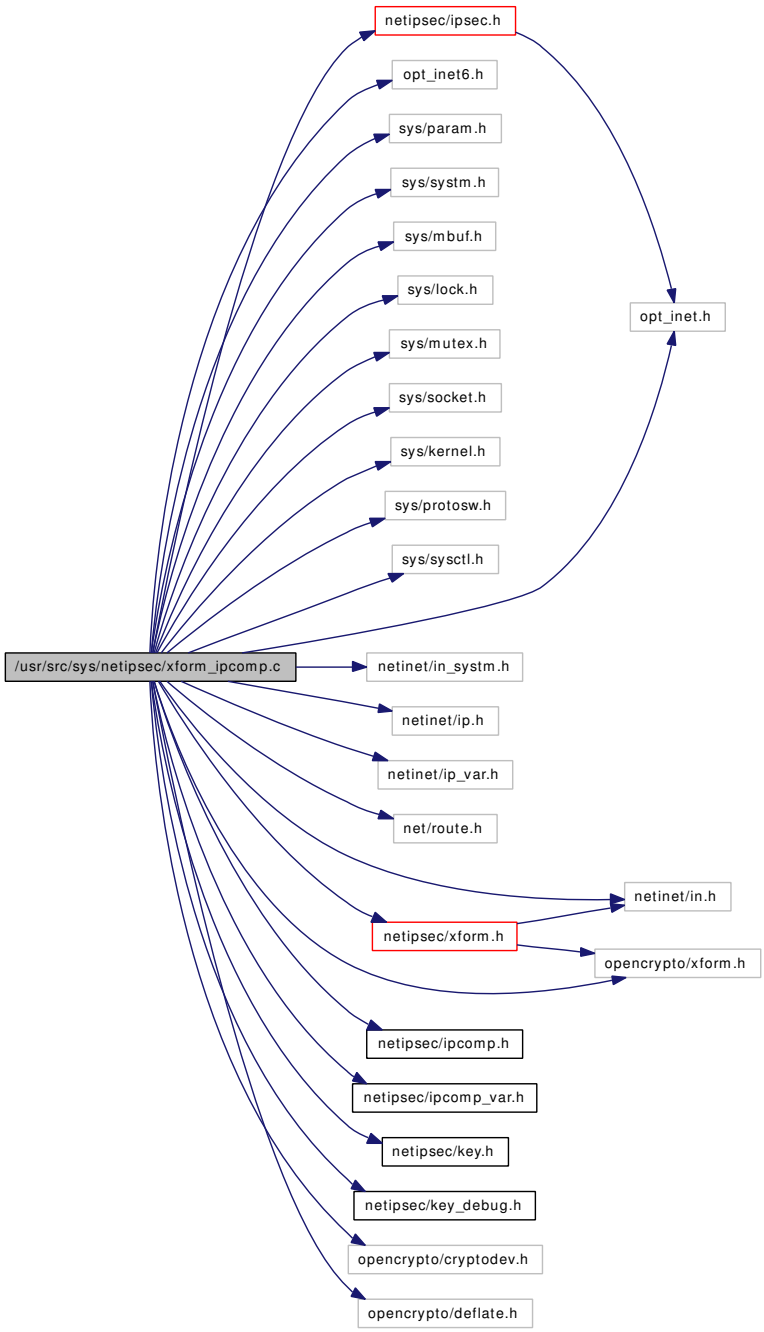
Definition at line 79 of file xform_esp.c.

Referenced by esp_input(), esp_input_cb(), esp_output(), and esp_output_cb().

7.27 /usr/src/sys/netipsec/xform_ipcomp.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/mbuf.h>
#include <sys/lock.h>
#include <sys/mutex.h>
#include <sys/socket.h>
#include <sys/kernel.h>
#include <sys/protosw.h>
#include <sys/sysctl.h>
#include <netinet/in.h>
#include <netinet/in_systm.h>
#include <netinet/ip.h>
#include <netinet/ip_var.h>
#include <net/route.h>
#include <netipsec/ipsec.h>
#include <netipsec/xform.h>
#include <netipsec/ipcomp.h>
#include <netipsec/ipcomp_var.h>
#include <netipsec/key.h>
#include <netipsec/key_debug.h>
#include <opencrypto/cryptodev.h>
#include <opencrypto/deflate.h>
#include <opencrypto/xform.h>
```

Include dependency graph for xform_ipcomp.c:



Defines

- #define IPSEC_COMMON_INPUT_CB(m, sav, skip, protoff, mtag) (error = ipsec4_common_-input_cb(m, sav, skip, protoff, mtag))

Functions

- SYSCTL_DECL (_net_inet_ipcomp)

- SYSCTL_INT (_net_inet_ipcomp, OID_AUTO, ipcomp_enable, CTLFLAG_RW,&ipcomp_enable, 0,"")
- SYSCTL_STRUCT (_net_inet_ipcomp, IPSECCTL_STATS, stats, CTLFLAG_RD,&ipcompstat, ipcompstat,"")
- static int ipcomp_input_cb (struct cryptop *crp)
- static int ipcomp_output_cb (struct cryptop *crp)
- comp_algo * ipcomp_algorithm_lookup (int alg)
- static int ipcomp_init (struct secasvar *sav, struct xformsw *xsp)
- static int ipcomp_zeroize (struct secasvar *sav)
- static int ipcomp_input (struct mbuf *m, struct secasvar *sav, int skip, int protoff)
- static int ipcomp_output (struct mbuf *m, struct ipsecrequest *isr, struct mbuf **mp, int skip, int protoff)
- static void ipcomp_attach (void)
- SYSINIT (ipcomp_xform_init, SI_SUB_PROTO_DOMAIN, SI_ORDER_MIDDLE, ipcomp_attach, NULL)

Variables

- int ipcomp_enable = 0
- ipcompstat ipcompstat
- static struct xformsw ipcomp_xformsw

7.27.1 Define Documentation

7.27.1.1 #define IPSEC_COMMON_INPUT_CB(m, sav, skip, protoff, mtag) (error = ipsec4_common_input_cb(m, sav, skip, protoff, mtag))

Definition at line 201 of file xform_ipcomp.c.

7.27.2 Function Documentation

7.27.2.1 struct comp_algo* ipcomp_algorithm_lookup (int *alg*)

Definition at line 82 of file xform_ipcomp.c.

References IPCOMP_ALG_MAX.

Referenced by ipcomp_init(), and key_getcomb_ipcomp().

7.27.2.2 static void ipcomp_attach (void) [static]

Definition at line 606 of file xform_ipcomp.c.

References ipcomp_xformsw, and xform_register().

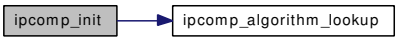
Here is the call graph for this function:

7.27.2.3 static int ipcomp_init (struct secasvar * *sav*, struct xformsw * *xsp*) [static]

Definition at line 97 of file xform_ipcomp.c.

References secasvar::alg_comp, secasvar::alg_enc, crypto_support, DPRINTF, ipcomp_algorithm_-lookup(), secasvar::tdb_compalgxform, secasvar::tdb_cryptoid, and secasvar::tdb_xform.

Here is the call graph for this function:

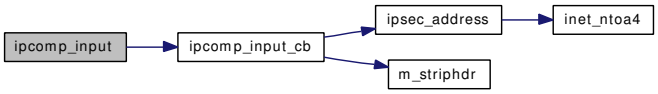


7.27.2.4 static int ipcomp_input (struct mbuf * *m*, struct secasvar * *sav*, int *skip*, int *protoff*) [static]

Definition at line 137 of file xform_ipcomp.c.

References DPRINTF, secasindex::dst, IPCOMP_HLENGTH, ipcomp_input_cb(), ipcompstat::ipcomps_-crypto, ipcompstat, IPSEC_SPLASSERT_SOFTNET, secasindex::proto, secasvar::sah, secashead::saidx, secasvar::spi, secasvar::tdb_compalgxform, and secasvar::tdb_cryptoid.

Here is the call graph for this function:



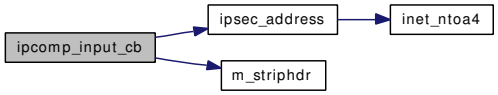
7.27.2.5 static int ipcomp_input_cb (struct cryptop * *crp*) [static]

Definition at line 209 of file xform_ipcomp.c.

References secasvar::alg_comp, DPRINTF, secasindex::dst, IPCOMP_HLENGTH, ipcompstat::ipcomps_crypto, ipcompstat::ipcomps_hdrops, ipcompstat::ipcomps_hist, ipcompstat::ipcomps_notdb, ipcompstat::ipcomps_noxform, ipcompstat, ipsec_address(), IPSEC_-ASSERT, IPSEC_COMMON_INPUT_CB, KEY_ALLOCSA, KEY_FREESAV, m_striphdr(), sockaddr_-union::sa, secasvar::sah, secashead::saidx, secasvar::spi, tdb_crypto::tc_dst, tdb_crypto::tc_proto, tdb_crypto::tc_protoff, tdb_crypto::tc_ptr, tdb_crypto::tc_skip, tdb_crypto::tc_spi, and secasvar::tdb_-cryptoid.

Referenced by ipcomp_input().

Here is the call graph for this function:

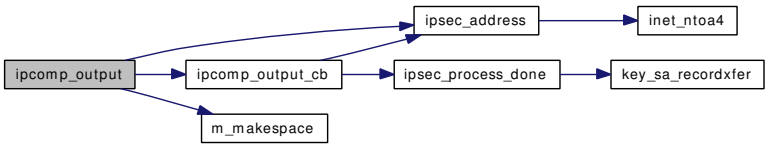


7.27.2.6 static int ipcomp_output (struct mbuf * *m*, struct ipsecrequest * *isr*, struct mbuf ** *mp*, int *skip*, int *protoff*) [static]

Definition at line 328 of file xform_ipcomp.c.

References ipcomp::comp_cpi, ipcomp::comp_flags, ipcomp::comp_nxt, DPRINTF, secasindex::dst, IPCOMP_HLENGTH, ipcomp_output_cb(), ipcompstat::ipcomps_crypto, ipcompstat::ipcomps_hdrops, ipcompstat::ipcomps_nopf, ipcompstat::ipcomps_obytes, ipcompstat::ipcomps_output, ipcompstat::ipcomps_toobig, ipcompstat::ipcomps_wrap, ipcompstat, ipsec_address(), IPSEC_ASSERT, IPSEC_SPLASSERT_SOFTNET, m_makespace(), secasindex::proto, sockaddr_union::sa, secasvar::sah, secashead::saidx, ipsecrequest::sav, secasvar::spi, tdb_crypto::tc_dst, tdb_crypto::tc_isr, tdb_crypto::tc_proto, tdb_crypto::tc_skip, tdb_crypto::tc_spi, secasvar::tdb_compalgxform, and secasvar::tdb_cryptoid.

Here is the call graph for this function:



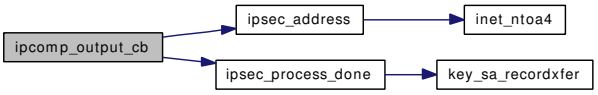
7.27.2.7 static int ipcomp_output_cb (struct cryptop * *crp*) [static]

Definition at line 494 of file xform_ipcomp.c.

References secasvar::alg_comp, DPRINTF, secasindex::dst, ipcompstat::ipcomps_crypto, ipcompstat::ipcomps_hist, ipcompstat::ipcomps_nopf, ipcompstat::ipcomps_notdb, ipcompstat::ipcomps_noxform, ipcompstat, ipsec_address(), IPSEC_ASSERT, ipsec_process_done(), IPSECREQUEST_LOCK, IPSECREQUEST_UNLOCK, KEY_ALLOCSA, KEY_FREESAV, sockaddr_union::sa, secasvar::sah, secashead::saidx, ipsecrequest::sav, secasvar::spi, tdb_crypto::tc_dst, tdb_crypto::tc_isr, tdb_crypto::tc_proto, tdb_crypto::tc_skip, tdb_crypto::tc_spi, and secasvar::tdb_cryptoid.

Referenced by ipcomp_output().

Here is the call graph for this function:



7.27.2.8 static int ipcomp_zeroize (struct secasvar * *sav*) [static]

Definition at line 124 of file xform_ipcomp.c.

References secasvar::tdb_cryptoid.

7.27.2.9 SYSCTL_DECL (_net_inet_ipcomp)

7.27.2.10 SYSCTL_INT (_net_inet_ipcomp, OID_AUTO, ipcomp_enable, CTLFLAG_RW, & *ipcomp_enable*, 0, "")

7.27.2.11 SYSCTL_STRUCT (_net_inet_ipcomp, IPSECCTL_STATS, stats, CTLFLAG_RD, & *ipcompstat*, ipcompstat, "")

7.27.2.12 SYSINIT (ipcomp_xform_init, SI_SUB_PROTO_DOMAIN, SI_ORDER_MIDDLE, ipcomp_attach, NULL)

7.27.3 Variable Documentation

7.27.3.1 int ipcomp_enable = 0

Definition at line 69 of file xform_ipcomp.c.

Referenced by ipsec_common_input(), and ipsec_nextisr().

7.27.3.2 struct xformsw ipcomp_xformsw [static]

Initial value:

```
{
        XF_IPCOMP,              XFT_COMP,               "IPcomp",
        ipcomp_init,            ipcomp_zeroize,         ipcomp_input,
        ipcomp_output
}
```

Definition at line 599 of file xform_ipcomp.c.

Referenced by ipcomp_attach().

7.27.3.3 struct ipcompstat ipcompstat

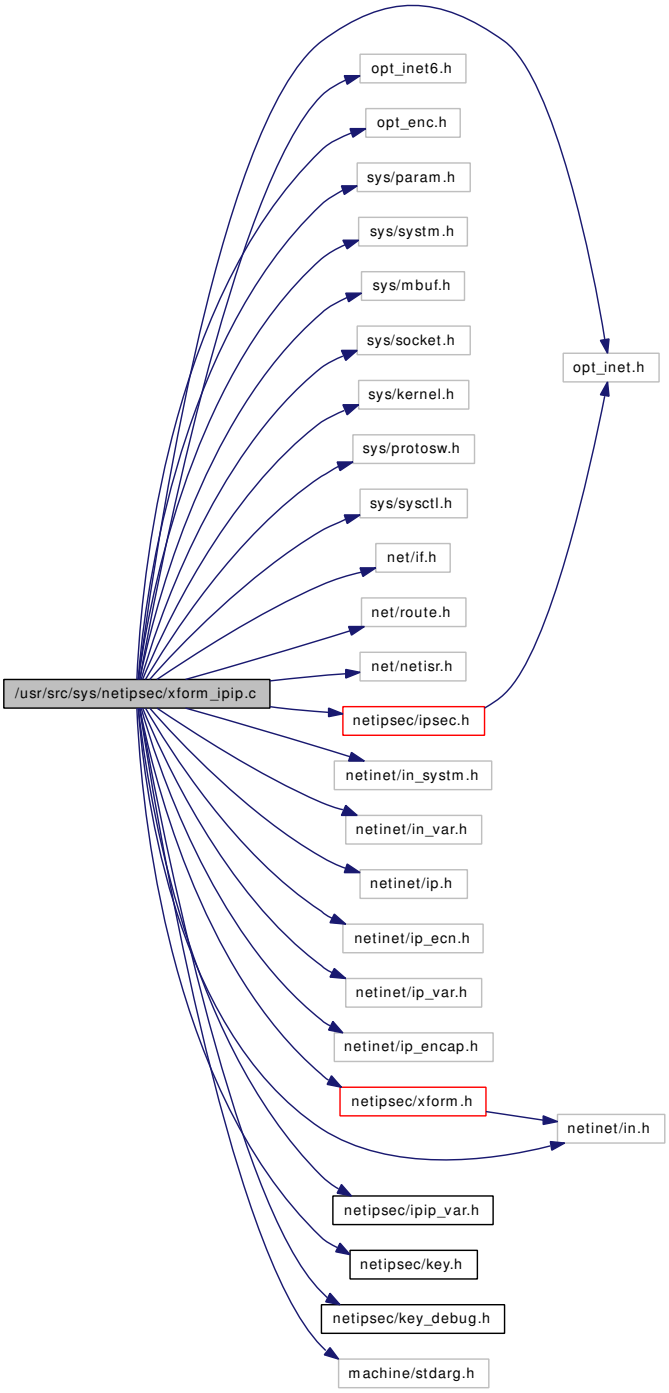
Definition at line 70 of file xform_ipcomp.c.

Referenced by ipcomp_input(), ipcomp_input_cb(), ipcomp_output(), and ipcomp_output_cb().

7.28 /usr/src/sys/netipsec/xform_ipip.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include "opt_enc.h"
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/mbuf.h>
#include <sys/socket.h>
#include <sys/kernel.h>
#include <sys/protosw.h>
#include <sys/sysctl.h>
#include <net/if.h>
#include <net/route.h>
#include <net/netisr.h>
#include <netinet/in.h>
#include <netinet/in_systm.h>
#include <netinet/in_var.h>
#include <netinet/ip.h>
#include <netinet/ip_ecn.h>
#include <netinet/ip_var.h>
#include <netinet/ip_encap.h>
#include <netipsec/ipsec.h>
#include <netipsec/xform.h>
#include <netipsec/ipip_var.h>
#include <netipsec/key.h>
#include <netipsec/key_debug.h>
#include <machine/stdarg.h>
```

Include dependency graph for xform_ipip.c:



Defines

- #define M_IPSEC (M_AUTHIPHDR|M_AUTHIPDGM|M_DECRYPTED)

Functions

- SYSCTL_DECL (_net_inet_ipip)
- SYSCTL_INT (_net_inet_ipip, OID_AUTO, ipip_allow, CTLFLAG_RW,&ipip_allow, 0,"")
- SYSCTL_STRUCT (_net_inet_ipip, IPSECCTL_STATS, stats, CTLFLAG_RD,&ipipstat, ipipstat,"")
- static void _ipip_input (struct mbuf *m, int iphlen, struct ifnet *gifp)
- int ipip_output (struct mbuf *m, struct ipsecrequest *isr, struct mbuf **mp, int skip, int protoff)

Variables

- int ipip_allow = 0
- ipipstat ipipstat

7.28.1 Define Documentation

7.28.1.1 #define M_IPSEC (M_AUTHIPHDR|M_AUTHIPDGM|M_DECRYPTED)

Definition at line 102 of file xform_ipip.c.

7.28.2 Function Documentation

7.28.2.1 static void _ipip_input (struct mbuf * *m*, int *iphlen*, struct ifnet * *gifp*) [static]

Definition at line 155 of file xform_ipip.c.

References DPRINTF, ip4_ipsec_ecn, ip6_ipsec_ecn, ipip_allow, ipipstat::ipips_family, ipipstat::ipips_hdrops, ipipstat::ipips_ibytes, ipipstat::ipips_ipackets, ipipstat::ipips_qfull, ipipstat::ipips_spoof, ipipstat, and ipsec_filter().

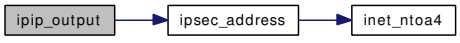
Here is the call graph for this function:

7.28.2.2 int ipip_output (struct mbuf * *m*, struct ipsecrequest * *isr*, struct mbuf ** *mp*, int *skip*, int *protoff*)

Definition at line 386 of file xform_ipip.c.

References DPRINTF, secasindex::dst, ipipstat::ipips_family, ipipstat::ipips_hdrops, ipipstat::ipips_obytes, ipipstat::ipips_opackets, ipipstat::ipips_unspec, ipipstat, ipsec_address(), IPSEC_ASSERT, IPSEC_SPLASSERT_SOFTNET, sockaddr_union::sa, secasvar::sah, secashead::saidx, ipsecrequest::sav, sockaddr_union::sin, sockaddr_union::sin6, secasvar::spi, secasindex::src, secasvar::tdb_xform, XF_IP4, and xformsw::xf_type.

Here is the call graph for this function:



7.28.2.3 SYSCTL_DECL (_net_inet_ipip)

7.28.2.4 SYSCTL_INT (_net_inet_ipip, OID_AUTO, ipip_allow, CTLFLAG_RW, & *ipip_allow*, 0, "")

7.28.2.5 SYSCTL_STRUCT (_net_inet_ipip, IPSECCTL_STATS, stats, CTLFLAG_RD, & *ipipstat*, ipipstat, "")

7.28.3 Variable Documentation

7.28.3.1 int ipip_allow = 0

Definition at line 92 of file xform_ipip.c.

Referenced by _ipip_input().

7.28.3.2 struct ipipstat ipipstat

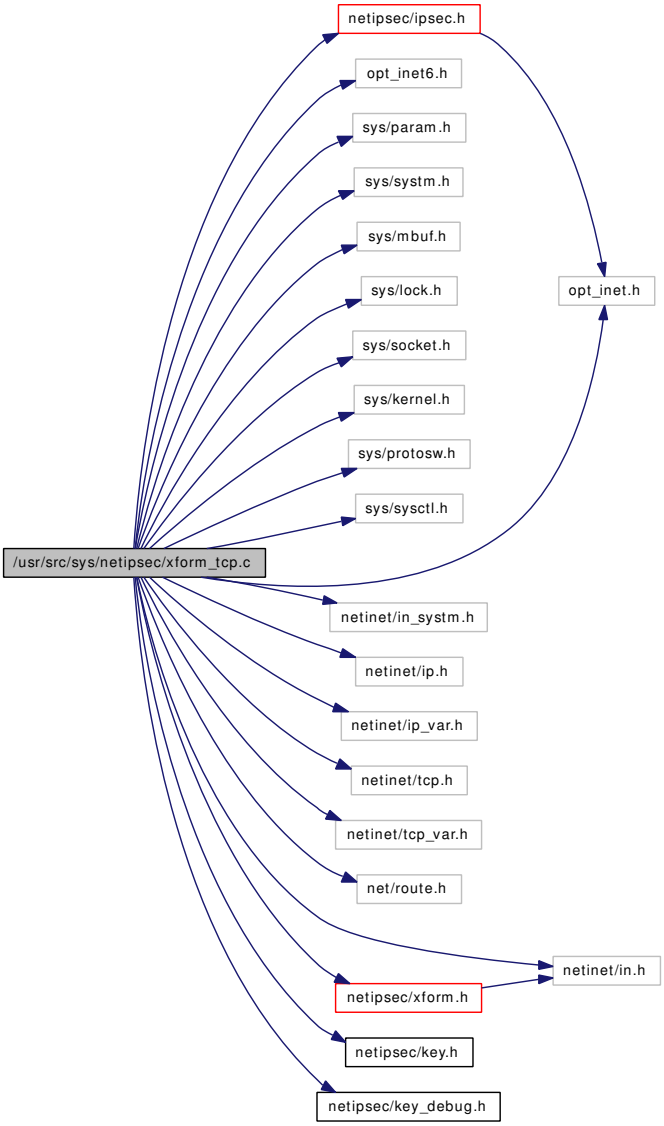
Definition at line 93 of file xform_ipip.c.

Referenced by _ipip_input(), and ipip_output().

7.29 /usr/src/sys/netipsec/xform_tcp.c File Reference

```
#include "opt_inet.h"
#include "opt_inet6.h"
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/mbuf.h>
#include <sys/lock.h>
#include <sys/socket.h>
#include <sys/kernel.h>
#include <sys/protosw.h>
#include <sys/sysctl.h>
#include <netinet/in.h>
#include <netinet/in_systm.h>
#include <netinet/ip.h>
#include <netinet/ip_var.h>
#include <netinet/tcp.h>
#include <netinet/tcp_var.h>
#include <net/route.h>
#include <netipsec/ipsec.h>
#include <netipsec/xform.h>
#include <netipsec/key.h>
#include <netipsec/key_debug.h>
```

Include dependency graph for xform_tcp.c:



Functions

- static int tcpsignature_init (struct secasvar *sav, struct xformsw *xsp)
- static int tcpsignature_zeroize (struct secasvar *sav)
- static int tcpsignature_input (struct mbuf *m, struct secasvar *sav, int skip, int protoff)
- static int tcpsignature_output (struct mbuf *m, struct ipsecrequest *isr, struct mbuf **mp, int skip, int protoff)
- static void tcpsignature_attach (void)

Variables

- static struct xformsw tcpsignature_xformsw

7.29.1 Function Documentation

7.29.1.1 static void tcpsignature_attach (void) [static]

Definition at line 162 of file xform_tcp.c.

References tcpsignature_xformsw, and xform_register().

Here is the call graph for this function:

7.29.1.2 static int tcpsignature_init (struct secasvar * *sav*, struct xformsw * *xsp*) [static]

Definition at line 83 of file xform_tcp.c.

References _KEYLEN, secasvar::alg_auth, DPRINTF, secasvar::key_auth, and secasvar::spi.

7.29.1.3 static int tcpsignature_input (struct mbuf * *m*, struct secasvar * *sav*, int *skip*, int *protoff*) [static]

Definition at line 135 of file xform_tcp.c.

7.29.1.4 static int tcpsignature_output (struct mbuf * *m*, struct ipsecrequest * *isr*, struct mbuf ** *mp*, int *skip*, int *protoff*) [static]

Definition at line 148 of file xform_tcp.c.

7.29.1.5 static int tcpsignature_zeroize (struct secasvar * *sav*) [static]

Definition at line 116 of file xform_tcp.c.

References _KEYLEN, secasvar::key_auth, seckey::key_data, secasvar::tdb_authalgxform, secasvar::tdb_-cryptoid, and secasvar::tdb_xform.

7.29.2 Variable Documentation

7.29.2.1 struct xformsw tcpsignature_xformsw [static]

Initial value:

```
{
        XF_TCPSIGNATURE,        XFT_AUTH,               "TCPMD5",
        tcpsignature_init,      tcpsignature_zeroize,
        tcpsignature_input,     tcpsignature_output
}
```

Definition at line 155 of file xform_tcp.c.

Referenced by tcpsignature_attach().

Index