TLS handshake

Generated by Doxygen 1.8.12

Contents

Chapter 1

Progetto di Advanced Programming of Cryptographic methods

ToDo list

- SSL/TLS
- In particolare, lo schema di comunicazione client<->server
- In particolare, l'handshaking

hello client

- Tutta la comunicazione client<->server (senza socket)
- Il mezzo di comunicazione da usare sarà un file condiviso
- Ogni entità legge il file e lo ripulisce
- Nella comunicazione ci sarà un token
- 2 file: uno di comunicazione, l'altro con il token di chi deve parlare
- La comunicazione prosegue finché non si genera un master secret (la chiave condivisa)
- Per la generazione del master secret usare hash, md5, generatori random
- Si possono usare librerie già fatte
- Devono essere due programmi: Client e Server
- Il protocollo è diviso in 4 fasi, arrivare fino alla 4ª fase
- CI FERMIAMO AL SEGRETO COMUNE

PARAMETRI DI VALUTAZIONE:

- Funzionante
- Porcherie di programmazione
- La relazione che sarà una spiegazione di cosa abbiamo fatto
- Come lo abbiamo pensato ed implementato

Libro: William Stallings - Network Security Applications and Standards

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Data Structure Documentation

4.1 certificate_message Struct Reference

```
#include <Certificate.h>
```

Data Fields

- uint32_t **cert_length**
- X509 ∗ **X509_certificate**

4.1.1 Detailed Description

Definition at line 30 of file Certificate.h.

4.1.2 Field Documentation

4.1.2.1 uint32_t cert_length

Definition at line 31 of file Certificate.h.

4.1.2.2 X509∗ X509_certificate

Definition at line 32 of file Certificate.h.

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

- /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/HandshakeMessages/**Certificate.h**

4.2 channel Struct Reference

```
#include <ServerClientBasic.h>
```

Data Fields

- **mode mod**
- char ∗ **channel_source**
- char ∗ **channel_destination**
- char ∗ **fileName**
- int **fd**
- void(∗ **onPacketReceive**)(struct **channel** ∗ch, **packet_basic** ∗p)
- int **isEnabled**
- pthread_t **thread**

4.2.1 Detailed Description

Definition at line 62 of file ServerClientBasic.h.

4.2.2 Field Documentation

4.2.2.1 char∗ channel_destination

Definition at line 68 of file ServerClientBasic.h.

4.2.2.2 char∗ channel_source

Definition at line 66 of file ServerClientBasic.h.

4.2.2.3 int fd

Definition at line 72 of file ServerClientBasic.h.

4.2.2.4 char∗ fileName

Definition at line 70 of file ServerClientBasic.h.

4.2.2.5 int isEnabled

Definition at line 76 of file ServerClientBasic.h.

4.2.2.6 mode mod

Definition at line 64 of file ServerClientBasic.h.

4.2.2.7 void(∗ onPacketReceive) (struct channel ∗ch, packet_basic ∗p)

Definition at line 74 of file ServerClientBasic.h.

4.2.2.8 pthread_t thread

Definition at line 78 of file ServerClientBasic.h.

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

- /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/**ServerClientBasic.h**

4.3 cipher_suite_t Struct Reference

```
#include <handshakeConstants.h>
```

Data Fields

- uint16_t **cipher_id**
- char ∗ **name**
- **key_exchange_algorithm kx**
- **authentication_algorithm au**
- uint16_t **key_size**
- **hash_algorithm hash**

4.3.1 Detailed Description

Definition at line 45 of file handshakeConstants.h.

4.3.2 Field Documentation

4.3.2.1 **authentication_algorithm** au

Definition at line 49 of file handshakeConstants.h.

4.3.2.2 uint16_t cipher_id

Definition at line 46 of file handshakeConstants.h.

4.3.2.3 **hash_algorithm** hash

Definition at line 51 of file handshakeConstants.h.

4.3.2.4 uint16_t key_size

Definition at line 50 of file handshakeConstants.h.

4.3.2.5 key_exchange_algorithm kx

Definition at line 48 of file handshakeConstants.h.

4.3.2.6 char∗ name

Definition at line 47 of file handshakeConstants.h.

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

- /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/**handshakeConstants.h**

4.4 client_key_exchange Struct Reference

```
#include <ServerClientKeyExchange.h>
```

Data Fields

- uint16_t **key_length**
- unsigned char ∗ **key**

4.4.1 Detailed Description

Definition at line 62 of file ServerClientKeyExchange.h.

4.4.2 Field Documentation

4.4.2.1 unsigned char∗ key

Definition at line 66 of file ServerClientKeyExchange.h.

4.4.2.2 uint16_t key_length

Definition at line 64 of file ServerClientKeyExchange.h.

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

- /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/HandshakeMessages/**ServerClient←KeyExchange.h**

4.5 compression_methods_t Struct Reference

```
#include <ServerClientHello.h>
```

Data Fields

- uint16_t **length**
- uint8_t **compression_id**

4.5.1 Detailed Description

Definition at line 37 of file ServerClientHello.h.

4.5.2 Field Documentation

4.5.2.1 uint8_t compression_id

Definition at line 39 of file ServerClientHello.h.

4.5.2.2 uint16_t length

Definition at line 38 of file ServerClientHello.h.

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

- /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/HandshakeMessages/**ServerClient↩ Hello.h**

4.6 DHE_server_key_exchange Struct Reference

```
#include <ServerClientKeyExchange.h>
```

Data Fields

- BIGNUM ∗ **p**
- BIGNUM ∗ **g**
- BIGNUM ∗ **pubKey**
- uint16_t **sign_hash_alg**
- unsigned int **signature_length**
- unsigned char ∗ **signature**

4.6.1 Detailed Description

Definition at line 45 of file ServerClientKeyExchange.h.

4.6.2 Field Documentation

4.6.2.1 BIGNUM∗ g

Definition at line 49 of file ServerClientKeyExchange.h.

4.6.2.2 BIGNUM∗ p

Definition at line 47 of file ServerClientKeyExchange.h.

4.6.2.3 BIGNUM∗ pubKey

Definition at line 51 of file ServerClientKeyExchange.h.

4.6.2.4 uint16_t sign_hash_alg

Definition at line 54 of file ServerClientKeyExchange.h.

4.6.2.5 unsigned char∗ signature

Definition at line 58 of file ServerClientKeyExchange.h.

4.6.2.6 unsigned int signature_length

Definition at line 56 of file ServerClientKeyExchange.h.

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

- /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/HandshakeMessages/**ServerClientKeyExchange.h**

4.7 ECDHE_server_key_exchange Struct Reference

```
#include <ServerClientKeyExchange.h>
```

Data Fields

- uint16_t **named_curve**
- BIGNUM ∗ **pub_key**
- uint16_t **sign_hash_alg**
- unsigned int **signature_length**
- unsigned char ∗ **signature**

4.7.1 Detailed Description

Definition at line 32 of file ServerClientKeyExchange.h.

4.7.2 Field Documentation

4.7.2.1 uint16_t named_curve

Definition at line 33 of file ServerClientKeyExchange.h.

4.7.2.2 BIGNUM* pub_key

Definition at line 35 of file ServerClientKeyExchange.h.

4.7.2.3 uint16_t sign_hash_alg

Definition at line 37 of file ServerClientKeyExchange.h.

4.7.2.4 unsigned char* signature

Definition at line 41 of file ServerClientKeyExchange.h.

4.7.2.5 unsigned int signature_length

Definition at line 39 of file ServerClientKeyExchange.h.

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

- /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/HandshakeMessages/**ServerClient↩ KeyExchange.h**

4.8 handshake Struct Reference

```
#include <ServerClientHandshakeProtocol.h>
```

Data Fields

- uint8_t **type**
- uint32_t **length**
- unsigned char * **message**

4.8.1 Detailed Description

Definition at line 36 of file ServerClientHandshakeProtocol.h.

4.8.2 Field Documentation

4.8.2.1 uint32_t length

Definition at line 38 of file ServerClientHandshakeProtocol.h.

4.8.2.2 unsigned char∗ message

Definition at line 39 of file ServerClientHandshakeProtocol.h.

4.8.2.3 uint8_t type

Definition at line 37 of file ServerClientHandshakeProtocol.h.

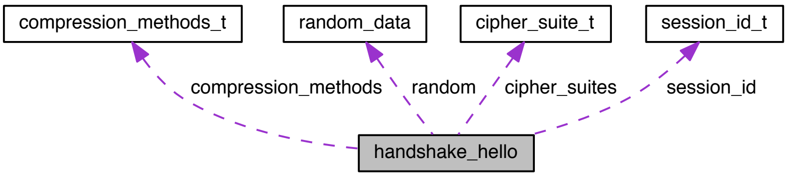
The documentation for this struct was generated from the following file:

- /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/**ServerClientHandshakeProtocol.h**

4.9 handshake_hello Struct Reference

```
#include <ServerClientHello.h>
```

Collaboration diagram for handshake_hello:



Data Fields

- uint16_t **TLS_version**
- **random_data random**
- **session_id session_id**
- uint16_t **cipher_suite_len**
- **cipher_suite_t** ∗ **cipher_suites**
- **compression_methods compression_methods**

4.9.1 Detailed Description

Definition at line 43 of file ServerClientHello.h.

4.9.2 Field Documentation

4.9.2.1 uint16_t cipher_suite_len

Definition at line 47 of file ServerClientHello.h.

4.9.2.2 cipher_suite_t∗ cipher_suites

Definition at line 48 of file ServerClientHello.h.

4.9.2.3 compression_methods compression_methods

Definition at line 49 of file ServerClientHello.h.

4.9.2.4 random_data random

Definition at line 45 of file ServerClientHello.h.

4.9.2.5 session_id session_id

Definition at line 46 of file ServerClientHello.h.

4.9.2.6 uint16_t TLS_version

Definition at line 44 of file ServerClientHello.h.

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

- /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/HandshakeMessages/**ServerClient↩ Hello.h**

4.10 packet_basic Struct Reference

```
#include <ServerClientBasic.h>
```

Data Fields

- char ∗ **source**
- char ∗ **destination**
- uint32_t **messageLen**
- unsigned char ∗ **message**

4.10.1 Detailed Description

Definition at line 51 of file ServerClientBasic.h.

4.10.2 Field Documentation

4.10.2.1 char∗ destination

Definition at line 53 of file ServerClientBasic.h.

4.10.2.2 unsigned char∗ message

Definition at line 55 of file ServerClientBasic.h.

4.10.2.3 uint32_t messageLen

Definition at line 54 of file ServerClientBasic.h.

4.10.2.4 char∗ source

Definition at line 52 of file ServerClientBasic.h.

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

- /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/**ServerClientBasic.h**

4.11 random_data Struct Reference

```
#include <ServerClientHello.h>
```

Data Fields

- uint32_t **UNIX_time**
- uint8_t **random_bytes** [28]

4.11.1 Detailed Description

Definition at line 25 of file ServerClientHello.h.

4.11.2 Field Documentation

4.11.2.1 uint8_t random_bytes[28]

Definition at line 27 of file ServerClientHello.h.

4.11.2.2 uint32_t UNIX_time

Definition at line 26 of file ServerClientHello.h.

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

- /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/HandshakeMessages/**ServerClient←Hello.h**

4.12 record Struct Reference

```
#include <ServerClientRecordProtocol.h>
```

Data Fields

- uint8_t **type**
- uint16_t **version**
- uint16_t **lenght**
- unsigned char ∗ **message**

4.12.1 Detailed Description

Definition at line 36 of file ServerClientRecordProtocol.h.

4.12.2 Field Documentation

4.12.2.1 uint16_t lenght

Definition at line 39 of file ServerClientRecordProtocol.h.

4.12.2.2 unsigned char∗ message

Definition at line 40 of file ServerClientRecordProtocol.h.

4.12.2.3 uint8_t type

Definition at line 37 of file ServerClientRecordProtocol.h.

4.12.2.4 uint16_t version

Definition at line 38 of file ServerClientRecordProtocol.h.

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

- /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/**ServerClientRecordProtocol.h**

4.13 session_id_t Struct Reference

```
#include <ServerClientHello.h>
```

Data Fields

- uint8_t **session_lenght**
- uint8_t ∗ **session_id**

4.13.1 Detailed Description

Definition at line 31 of file ServerClientHello.h.

4.13.2 Field Documentation

4.13.2.1 uint8_t∗ session_id

Definition at line 33 of file ServerClientHello.h.

4.13.2.2 uint8_t session_lenght

Definition at line 32 of file ServerClientHello.h.

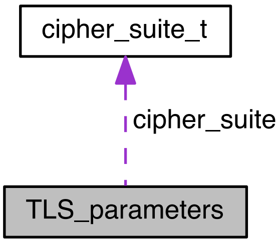
The documentation for this struct was generated from the following file:

- /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/HandshakeMessages/**ServerClient↩ Hello.h**

4.14 TLS_parameters Struct Reference

```
#include <TLS.h>
```

Collaboration diagram for TLS_parameters:



Data Fields

- uint16_t **tls_version**
- uint16_t **previous_state**
- **cipher_suite_t cipher_suite**
- unsigned char **client_random** [32]
- unsigned char **server_random** [32]
- void ∗ **server_key_ex**
- unsigned char ∗ **master_secret**
- int **master_secret_len**
- int **handshake_messages_len**
- unsigned char ∗ **handshake_messages**
- X509 ∗ **server_certificate**
- BIGNUM ∗ **private_key**

4.14.1 Detailed Description

Definition at line 21 of file TLS.h.

4.14.2 Field Documentation

4.14.2.1 cipher_suite_t cipher_suite

Definition at line 24 of file TLS.h.

4.14.2.2 unsigned char client_random[32]

Definition at line 26 of file TLS.h.

4.14.2.3 unsigned char∗ handshake_messages

Definition at line 35 of file TLS.h.

4.14.2.4 int handshake_messages_len

Definition at line 34 of file TLS.h.

4.14.2.5 unsigned char∗ master_secret

Definition at line 31 of file TLS.h.

4.14.2.6 int master_secret_len

Definition at line 32 of file TLS.h.

4.14.2.7 uint16_t previous_state

Definition at line 23 of file TLS.h.

4.14.2.8 BIGNUM∗ private_key

Definition at line 39 of file TLS.h.

4.14.2.9 X509∗ server_certificate

Definition at line 37 of file TLS.h.

4.14.2.10 void∗ server_key_ex

Definition at line 29 of file TLS.h.

4.14.2.11 unsigned char server_random[32]

Definition at line 27 of file TLS.h.

4.14.2.12 uint16_t tls_version

Definition at line 22 of file TLS.h.

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

- /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/**TLS.h**

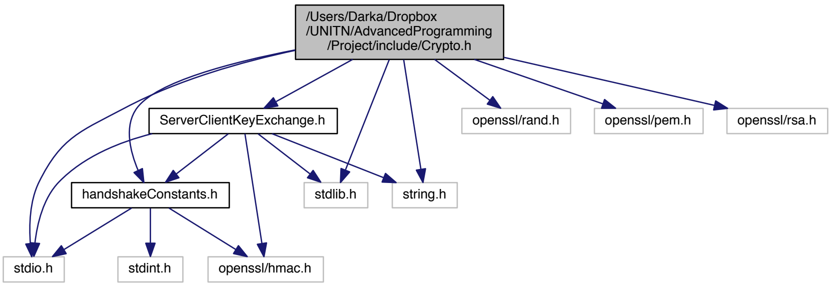
Chapter 5

File Documentation

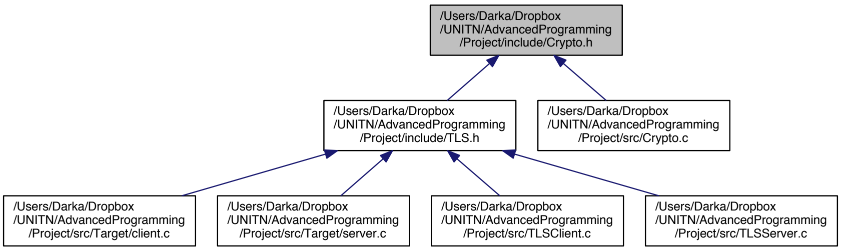
5.1 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/Crypto.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <openssl/rand.h>
#include <openssl/pem.h>
#include <openssl/rsa.h>
#include "handshakeConstants.h"
#include "ServerClientKeyExchange.h"
```

Include dependency graph for Crypto.h:



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


/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/Crypto.h
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/TLS.h
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/Crypto.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/Target/client.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/Target/server.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/TLSClient.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/TLSServer.c

Functions

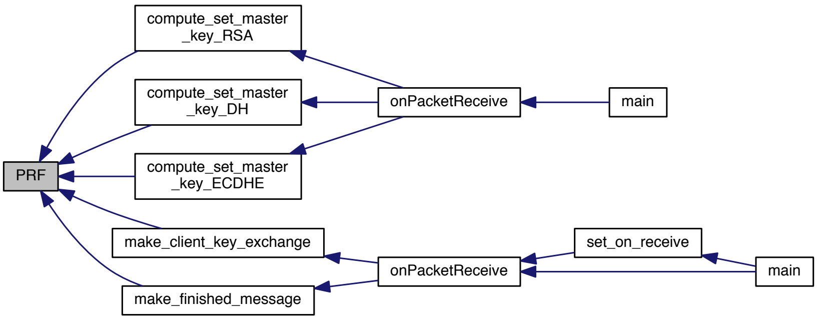
- void **PRF** (const EVP_MD ∗hash, unsigned char ∗secret, int secret_len, char ∗label, unsigned char ∗seed, int seed_len, int result_len, unsigned char ∗∗result)
- int **sign_DHE_server_key_ex** (unsigned char ∗client_random, unsigned char ∗server_random, **DHE_↩ server_key_exchange** ∗server_key_ex, **authentication_algorithm** au)
- int **verify_DHE_server_key_ex_sign** (X509 ∗certificate, unsigned char ∗client_random, unsigned char ∗server_random, **DHE_server_key_exchange** ∗server_key_ex, **authentication_algorithm** au)
- int **sign_ECDHE_server_key_ex** (unsigned char ∗client_random, unsigned char ∗server_random, **ECDH↩ E_server_key_exchange** ∗server_key_ex, **authentication_algorithm** au)
- int **verify_ECDHE_server_key_ex_sign** (X509 ∗certificate, unsigned char ∗client_random, unsigned char ∗server_random, **ECDHE_server_key_exchange** ∗server_key_ex, **authentication_algorithm** au)

5.1.1 Function Documentation

5.1.1.1 void PRF (const EVP_MD ∗ *hash,* unsigned char ∗ *secret,* int *secret_len,* char ∗ *label,* unsigned char ∗ *seed,* int *seed_len,* int *result_len,* unsigned char ∗∗ *result*)

Definition at line 15 of file Crypto.c.

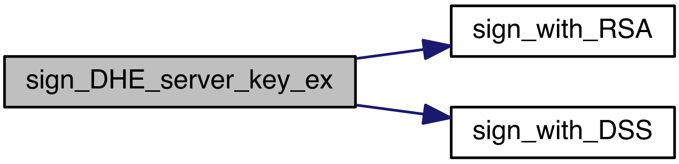
Here is the caller graph for this function:



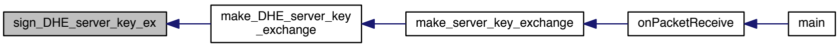
5.1.1.2 int sign_DHE_server_key_ex (unsigned char ∗ *client_random,* unsigned char ∗ *server_random,* **DHE_server_key_exchange** ∗ *server_key_ex,* **authentication_algorithm** *au*)

Definition at line 127 of file Crypto.c.

Here is the call graph for this function:



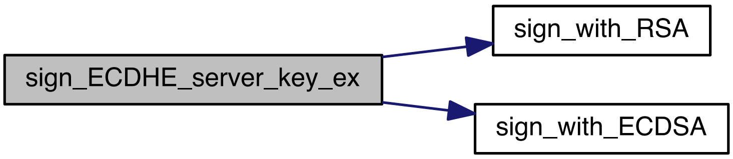
Here is the caller graph for this function:



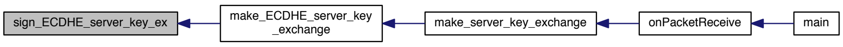
5.1.1.3 int sign_ECDHE_server_key_ex (unsigned char ∗ *client_random,* unsigned char ∗ *server_random,* ECDHE_server_key_exchange ∗ *server_key_ex,* authentication_algorithm *au*)

Definition at line 199 of file Crypto.c.

Here is the call graph for this function:



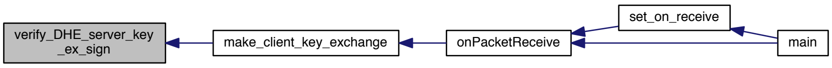
Here is the caller graph for this function:



5.1.1.4 int verify_DHE_server_key_ex_sign (X509 ∗ *certificate,* unsigned char ∗ *client_random,* unsigned char ∗ *server_random,* DHE_server_key_exchange ∗ *server_key_ex,* authentication_algorithm *au*)

Definition at line 42 of file Crypto.c.

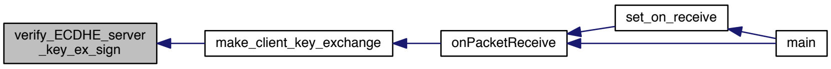
Here is the caller graph for this function:



5.1.1.5 int verify_ECDHE_server_key_ex_sign (X509 ∗ *certificate,* unsigned char ∗ *client_random,* unsigned char ∗ *server_random,* ECDHE_server_key_exchange ∗ *server_key_ex,* authentication_algorithm *au*)

Definition at line 259 of file Crypto.c.

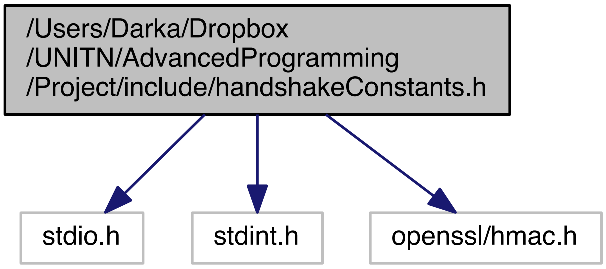
Here is the caller graph for this function:



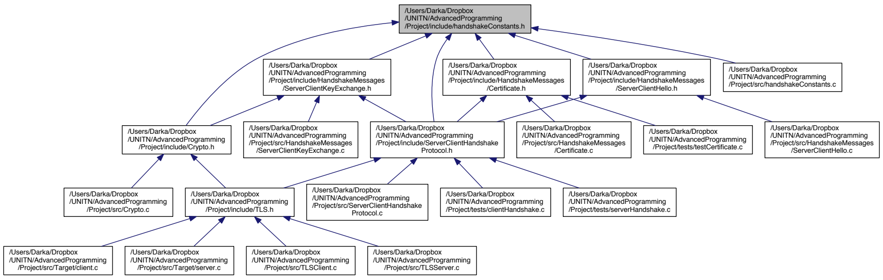
5.2 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/handshake←Constants.h File Reference

```
#include <stdio.h>
#include <stdint.h>
#include <openssl/hmac.h>
```

Include dependency graph for handshakeConstants.h:


/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/handshakeConstants.h
stdio.h
stdint.h
openssl/hmac.h

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


/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/handshakeConstants.h
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/HandshakeMessages
/ServerClientKeyExchange.h
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/HandshakeMessages
/Certificate.h
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/HandshakeMessages
/ServerClientHello.h
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/handshakeConstants.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/Crypto.h
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/HandshakeMessages
/ServerClientKeyExchange.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/ServerClientHandshake
Protocol.h
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/HandshakeMessages
/Certificate.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/tests/testCertificate.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/HandshakeMessages
/ServerClientHello.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/Crypto.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/TLS.h
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/ServerClientHandshake
Protocol.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/tests/clientHandshake.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/tests/serverHandshake.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/Target/client.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/Target/server.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/TLSClient.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/TLSServer.c

Data Structures

- struct **cipher_suite_t**

Macros

- #define **REV16**(value) ({(value & 0x00FFU) $<<$ 8 | (value & 0xFF00U) $>>$ 8;})
- #define **REV32**(value) ({(value & 0x000000FFU) $<<$ 24 | (value & 0x0000FF00U) $<<$ 8 |(value & 0x00FF0000U) $>>$ 8 | (value & 0xFF000000U) $>>$ 24;})
- #define **channel_mode_enum**
- #define **enum_record_version**
- #define **enum_handshake_type**

Enumerations

- enum **key_exchange_algorithm** { **RSA_KX** = 1, **DHE_KX** = 2, **ECDHE_KX** = 3 }
- enum **authentication_algorithm** { **RSA_AU** = 1, **DSS_AU** = 2, **ECDSA_AU** = 3 }
- enum **hash_algorithm** {
  none = 0, **md5** = 1, **sha1** = 2, **sha224** = 3,
  sha256 = 4, **sha384** = 5, **sha512** = 6 }
- enum **channel_mode** { **SERVER_MODE**, **CLIENT_MODE** }
- enum **TLS_version** { **SSL3_0** = 0x0300, **TLS1_0** = 0x0301, **TLS1_1** = 0x0302, **TLS1_2** = 0x0303 }
- enum {
  HELLO_REQUEST = 0x00, **CLIENT_HELLO** = 0x01, **SERVER_HELLO** = 0x02, **CERTIFICATE** = 0x0B,
  SERVER_KEY_EXCHANGE = 0x0C, **CERTIFICATE_REQUEST** = 0x0D, **SERVER_DONE** = 0x0E, **CERTIFICATE_VERIFY** = 0x0F,
  CLIENT_KEY_EXCHANGE = 0x10, **FINISHED** = 0x14 }

Functions

- **cipher_suite_t get_cipher_suite** (uint16_t id)
- const EVP_MD $*$ **get_hash_function** (**hash_algorithm** h)

5.2.1 Macro Definition Documentation

5.2.1.1 #define channel_mode_enum

Definition at line 58 of file handshakeConstants.h.

5.2.1.2 #define enum_handshake_type

Definition at line 90 of file handshakeConstants.h.

5.2.1.3 #define enum_record_version

Definition at line 77 of file handshakeConstants.h.

5.2.1.4 #define REV16(*value*) ({(value & 0x00FFU) $<<$ 8 $|$ (value & 0xFF00U) $>>$ 8;})

Definition at line 18 of file handshakeConstants.h.

5.2.1.5 #define REV32(*value*) ({(value & 0x000000FFU) $<<$ 24 $|$ (value & 0x0000FF00U) $<<$ 8 $|$(value & 0x00FF0000U) $>>$ 8 $|$ (value & 0xFF000000U) $>>$ 24;})

Definition at line 19 of file handshakeConstants.h.

5.2.2 Enumeration Type Documentation

5.2.2.1 anonymous enum

Enumerator

HELLO_REQUEST

CLIENT_HELLO

SERVER_HELLO

CERTIFICATE

SERVER_KEY_EXCHANGE

CERTIFICATE_REQUEST

SERVER_DONE

CERTIFICATE_VERIFY

CLIENT_KEY_EXCHANGE

FINISHED

Definition at line 91 of file handshakeConstants.h.

5.2.2.2 enum authentication_algorithm

Enumerator

RSA_AU

DSS_AU

ECDSA_AU

Definition at line 28 of file handshakeConstants.h.

5.2.2.3 enum channel_mode

Enumerator

SERVER_MODE

CLIENT_MODE

Definition at line 59 of file handshakeConstants.h.

5.2.2.4 enum hash_algorithm

Enumerator

none

md5

sha1

sha224

sha256

sha384

sha512

Definition at line 34 of file handshakeConstants.h.

5.2.2.5 enum key_exchange_algorithm

Enumerator

RSA_KX

DHE_KX

ECDHE_KX

Definition at line 22 of file handshakeConstants.h.

5.2.2.6 enum TLS_version

Enumerator

SSL3_0

TLS1_0

TLS1_1

TLS1_2

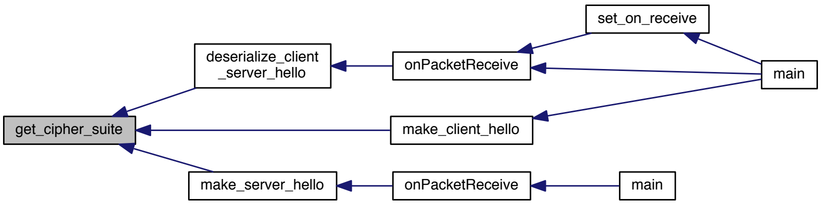
Definition at line 78 of file handshakeConstants.h.

5.2.3 Function Documentation

5.2.3.1 cipher_suite_t get_cipher_suite (uint16_t *id*)

Definition at line 103 of file handshakeConstants.c.

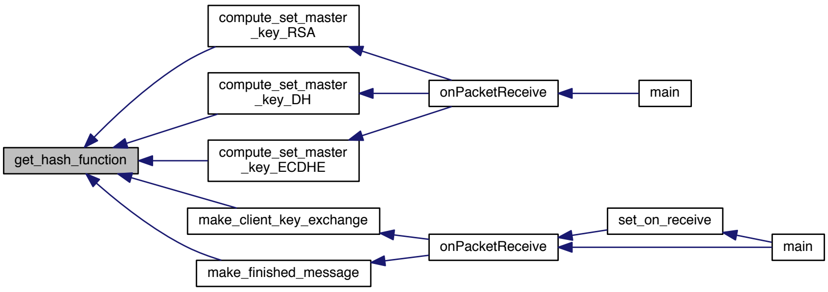
Here is the caller graph for this function:



5.2.3.2 const EVP_MD∗ get_hash_function (hash_algorithm *h*)

Definition at line 83 of file handshakeConstants.c.

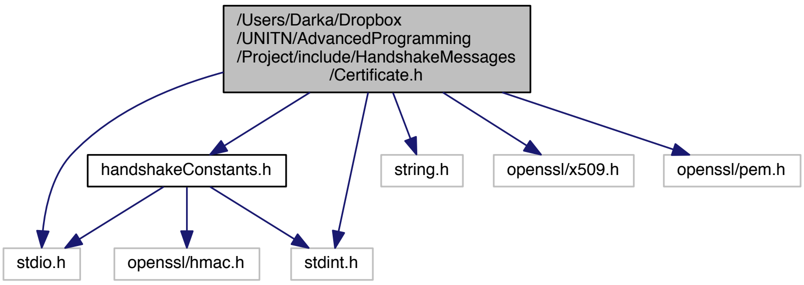
Here is the caller graph for this function:



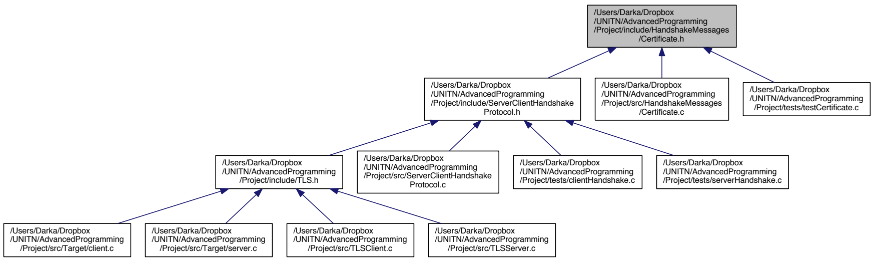
5.3 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/Handshake↩ Messages/Certificate.h File Reference

```
#include <stdio.h>
#include <stdint.h>
#include <string.h>
#include <openssl/x509.h>
#include <openssl/pem.h>
#include "handshakeConstants.h"
```

Include dependency graph for Certificate.h:


/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/include/HandshakeMessages /Certificate.h
handshakeConstants.h
string.h
openssl/x509.h
openssl/pem.h
stdio.h
openssl/hmac.h
stdint.h

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


/Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/HandshakeMessages/Certificate.h
/Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/ServerClientHandshakeProtocol.h
/Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/src/HandshakeMessages/Certificate.c
/Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/tests/testCertificate.c
/Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/TLS.h
/Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/src/ServerClientHandshakeProtocol.c
/Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/tests/clientHandshake.c
/Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/tests/serverHandshake.c
/Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/src/Target/client.c
/Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/src/Target/server.c
/Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/src/TLSClient.c
/Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/src/TLSServer.c

Data Structures

- struct **certificate_message**

Functions

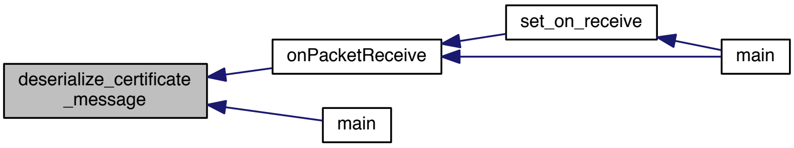
- **certificate_message** ∗ **make_certificate_message** (char ∗cert_file_name)
- void **serialize_certificate_message** (**certificate_message** ∗cert, unsigned char ∗∗stream, uint32_t ∗len)
- **certificate_message** ∗ **deserialize_certificate_message** (unsigned char ∗stream, uint32_t len)
- void **free_certificate_message** (**certificate_message** ∗cert)

5.3.1 Function Documentation

5.3.1.1 certificate_message∗ deserialize_certificate_message (unsigned char ∗ *stream,* uint32_t *len*)

Definition at line 83 of file Certificate.c.

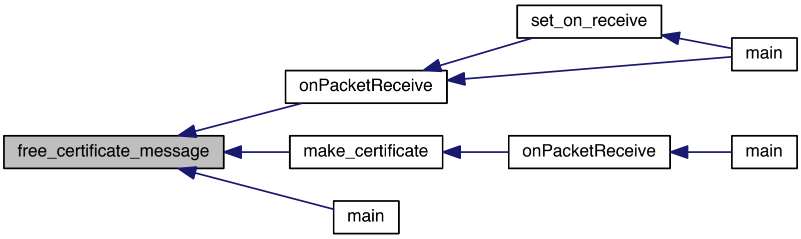
Here is the caller graph for this function:



5.3.1.2 void free_certificate_message (**certificate_message** ∗ *cert*)

Definition at line 105 of file Certificate.c.

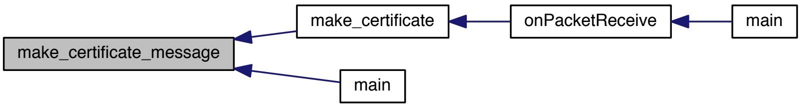
Here is the caller graph for this function:



5.3.1.3 **certificate_message**∗ make_certificate_message (char ∗ *cert_file_name*)

Definition at line 16 of file Certificate.c.

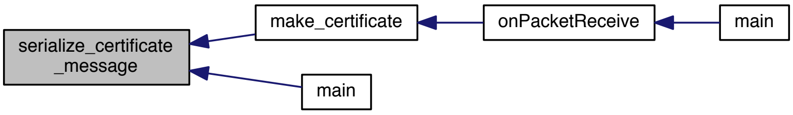
Here is the caller graph for this function:



5.3.1.4 void serialize_certificate_message (**certificate_message** ∗ *cert,* unsigned char ∗∗ *stream,* uint32_t ∗ *len*)

Definition at line 44 of file Certificate.c.

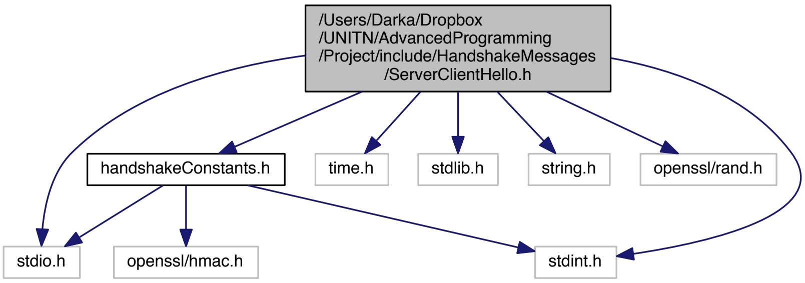
Here is the caller graph for this function:



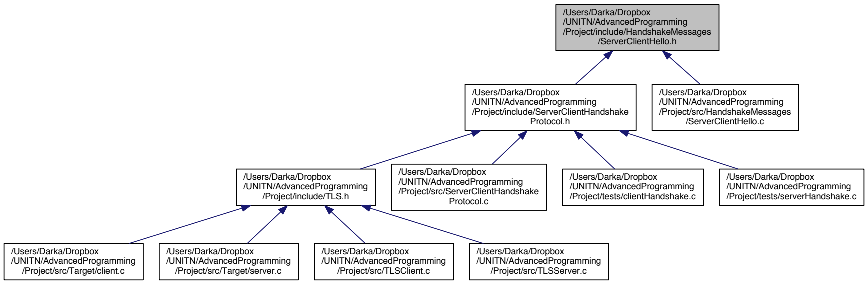
5.4 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/Handshake↩ Messages/ServerClientHello.h File Reference

```
#include <stdio.h>
#include <stdint.h>
#include <time.h>
#include <stdlib.h>
#include <string.h>
#include <openssl/rand.h>
#include "handshakeConstants.h"
```

Include dependency graph for ServerClientHello.h:



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


/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/HandshakeMessages
/ServerClientHello.h
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/ServerClientHandshake
Protocol.h
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/HandshakeMessages
/ServerClientHello.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/TLS.h
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/ServerClientHandshake
Protocol.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/tests/clientHandshake.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/tests/serverHandshake.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/Target/client.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/Target/server.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/TLSClient.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/TLSServer.c

Data Structures

- struct **random_data**
- struct **session_id_t**
- struct **compression_methods_t**
- struct **handshake_hello**

Typedefs

- typedef struct **session_id_t** **session_id**
- typedef struct **compression_methods_t** **compression_methods**

Functions

- **handshake_hello** ∗ **make_hello** (**session_id** session)
- void **serialize_client_server_hello** (**handshake_hello** ∗hello, unsigned char ∗∗stream, uint32_t ∗stream←Len, **channel_mode mode**)
- **handshake_hello** ∗ **deserialize_client_server_hello** (unsigned char ∗stream, uint32_t streamLen, **channel_mode mode**)
- void **free_hello** (**handshake_hello** ∗h)
- void **print_hello** (**handshake_hello** ∗h)

5.4.1 Typedef Documentation

5.4.1.1 typedef struct compression_methods_t compression_methods

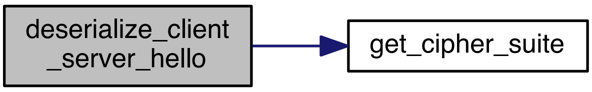
5.4.1.2 typedef struct session_id_t session_id

5.4.2 Function Documentation

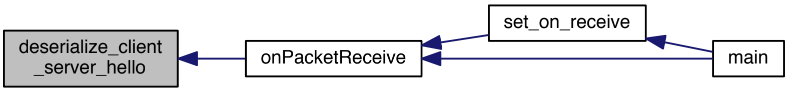
5.4.2.1 handshake_hello∗ deserialize_client_server_hello (unsigned char ∗ *stream,* uint32_t *streamLen,* channel_mode *mode*)

Definition at line 106 of file ServerClientHello.c.

Here is the call graph for this function:



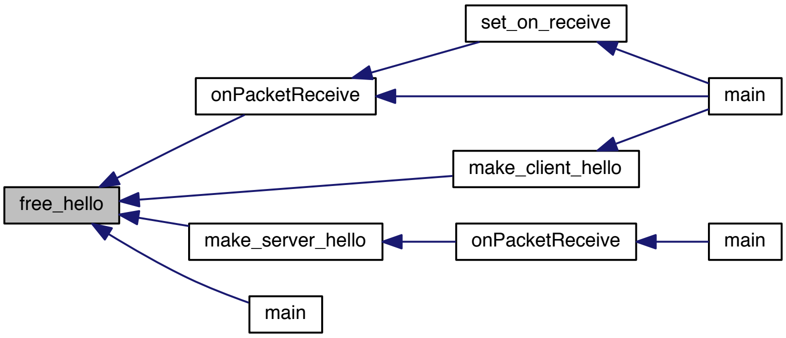
Here is the caller graph for this function:



5.4.2.2 void free_hello (handshake_hello ∗ *h*)

Definition at line 209 of file ServerClientHello.c.

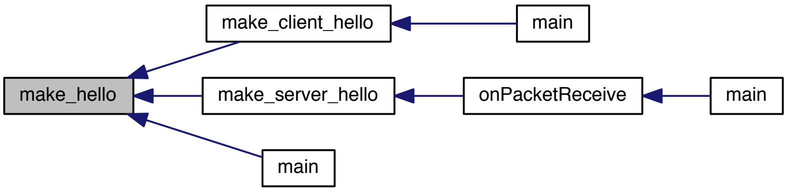
Here is the caller graph for this function:



5.4.2.3 handshake_hello∗ make_hello (session_id *session*)

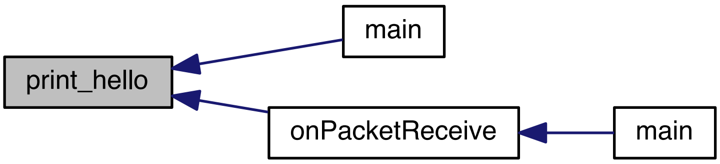
Definition at line 15 of file ServerClientHello.c.

Here is the caller graph for this function:



5.4.2.4 void print_hello (**handshake_hello** ∗ *h*)

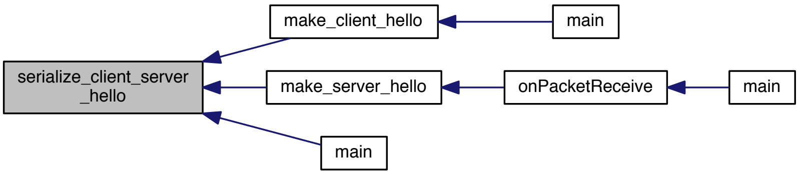
Here is the caller graph for this function:



5.4.2.5 void serialize_client_server_hello (**handshake_hello** ∗ *hello,* unsigned char ∗∗ *stream,* uint32_t ∗ *streamLen,* **channel_mode** *mode*)

Definition at line 37 of file ServerClientHello.c.

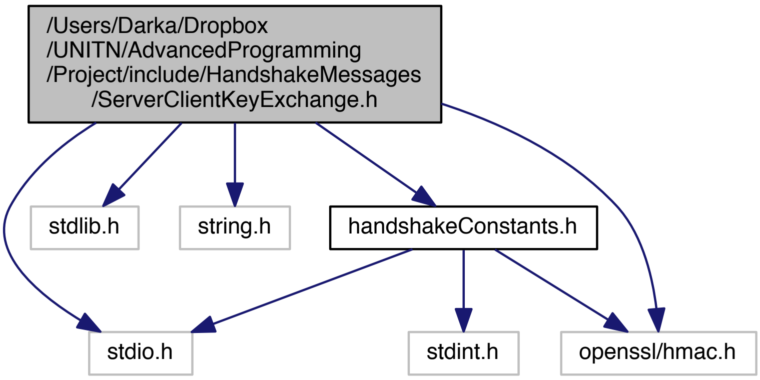
Here is the caller graph for this function:



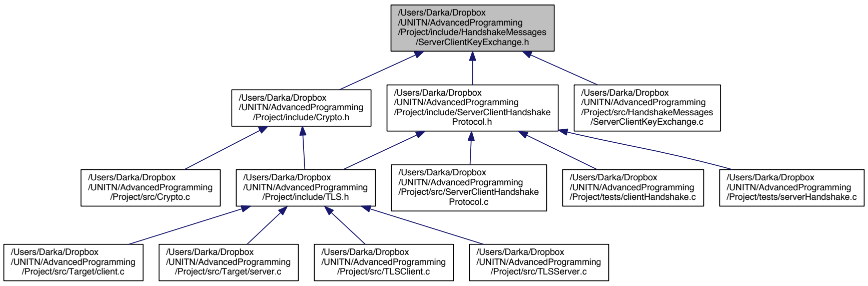
5.5 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/Handshake↩ Messages/ServerClientKeyExchange.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <openssl/hmac.h>
#include "handshakeConstants.h"
```

Include dependency graph for ServerClientKeyExchange.h:


/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/HandshakeMessages
/ServerClientKeyExchange.h
stdlib.h
string.h
handshakeConstants.h
stdio.h
stdint.h
openssl/hmac.h

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


/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/HandshakeMessages
/ServerClientKeyExchange.h
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/Crypto.h
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/ServerClientHandshake
Protocol.h
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/HandshakeMessages
/ServerClientKeyExchange.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/Crypto.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/TLS.h
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/ServerClientHandshake
Protocol.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/tests/clientHandshake.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/tests/serverHandshake.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/Target/client.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/Target/server.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/TLSClient.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/TLSServer.c

Data Structures

- struct **ECDHE_server_key_exchange**
- struct **DHE_server_key_exchange**
- struct **client_key_exchange**

Macros

- #define **server_key_exchange_structs**

Functions

- void **serialize_server_key_exchange** (void ∗server_key_exchange, unsigned char ∗∗stream, uint32_t ∗streamLen, **key_exchange_algorithm** kx)
- void ∗ **deserialize_server_key_exchange** (uint32_t message_len, unsigned char ∗message, **key_exchange_algorithm** kx)
- void **serialize_client_key_exchange** (**client_key_exchange** ∗**client_key_exchange**, unsigned char ∗∗stream, uint32_t ∗streamLen)
- void ∗ **deserialize_client_key_exchange** (uint32_t message_len, unsigned char ∗message)
- void **free_server_key_exchange** (void ∗server_key_ex, **cipher_suite_t** cipher_suite)

5.5.1 Macro Definition Documentation

5.5.1.1 #define server_key_exchange_structs

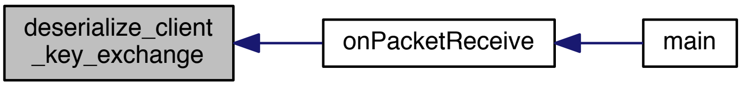
Definition at line 24 of file ServerClientKeyExchange.h.

5.5.2 Function Documentation

5.5.2.1 void∗ deserialize_client_key_exchange (uint32_t *message_len,* unsigned char ∗ *message*)

Definition at line 185 of file ServerClientKeyExchange.c.

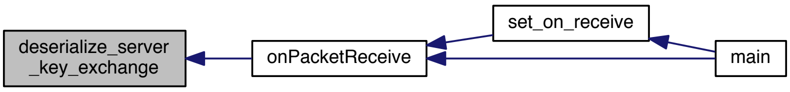
Here is the caller graph for this function:



5.5.2.2 void∗ deserialize_server_key_exchange (uint32_t *message_len,* unsigned char ∗ *message,* key_exchange_algorithm *kx*)

Definition at line 104 of file ServerClientKeyExchange.c.

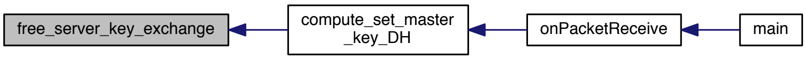
Here is the caller graph for this function:



5.5.2.3 void free_server_key_exchange (void ∗ *server_key_ex,* cipher_suite_t *cipher_suite*)

Definition at line 199 of file ServerClientKeyExchange.c.

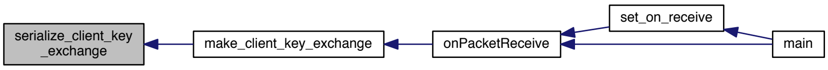
Here is the caller graph for this function:



5.5.2.4 void serialize_client_key_exchange (client_key_exchange ∗ *client_key_exchange,* unsigned char ∗∗ *stream,* uint32_t ∗ *streamLen*)

Definition at line 171 of file ServerClientKeyExchange.c.

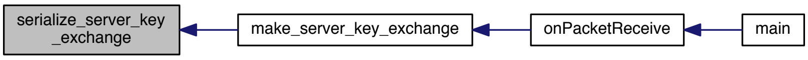
Here is the caller graph for this function:



5.5.2.5 void serialize_server_key_exchange (void ∗ *server_key_exchange,* unsigned char ∗∗ *stream,* uint32_t ∗ *streamLen,* key_exchange_algorithm *kx*)

Definition at line 15 of file ServerClientKeyExchange.c.

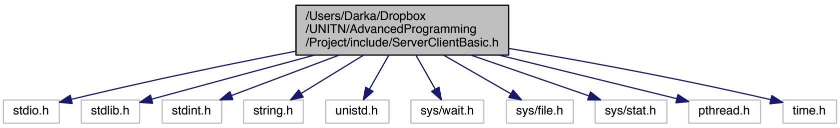
Here is the caller graph for this function:



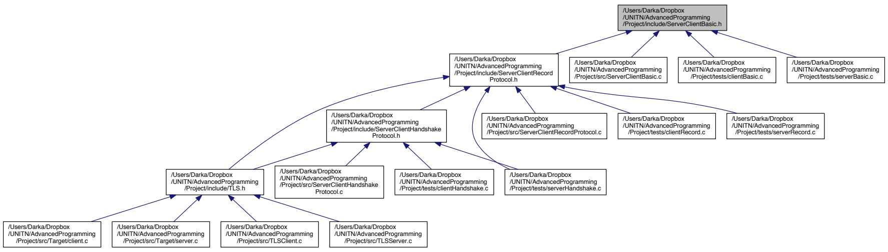
5.6 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/ServerClient↩ Basic.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <string.h>
#include <unistd.h>
#include <sys/wait.h>
#include <sys/file.h>
#include <sys/stat.h>
#include <pthread.h>
#include <time.h>
```

Include dependency graph for ServerClientBasic.h:


/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/include/ServerClientBasic.h
stdio.h
stdlib.h
stdint.h
string.h
unistd.h
sys/wait.h
sys/file.h
sys/stat.h
pthread.h
time.h

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


/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/include/ServerClientBasic.h
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/include/ServerClientRecord Protocol.h
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/src/ServerClientBasic.c
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/tests/clientBasic.c
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/tests/serverBasic.c
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/include/ServerClientHandshake Protocol.h
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/src/ServerClientRecordProtocol.c
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/tests/clientRecord.c
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/tests/serverRecord.c
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/include/TLS.h
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/src/ServerClientHandshake Protocol.c
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/tests/clientHandshake.c
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/tests/serverHandshake.c
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/src/Target/client.c
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/src/Target/server.c
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/src/TLSClient.c
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/src/TLSServer.c

Data Structures

- struct **packet_basic**
- struct **channel**

Macros

- #define **DELAY_TIME** 50
- #define **enum_mode**
- #define **struct_packet**
- #define **struct_channel**

Typedefs

- typedef struct **channel channel**

Enumerations

- enum **mode** { **SERVER**, **CLIENT** }

Functions

- **channel** ∗ **create_channel** (char ∗fileName, char ∗channelFrom, char ∗channelTo, **mode** channelMode)
- int **set_on_receive** (**channel** ∗ch, void(∗**onPacketReceive**)(**channel** ∗ch, **packet_basic** ∗p))
- int **send_packet** (**channel** ∗ch, **packet_basic** ∗p)
- int **start_listener** (**channel** ∗ch)
- void **wait_channel** (**channel** ∗ch)
- void **stop_channel** (**channel** ∗ch)
- **packet_basic** ∗ **create_packet** (char ∗from, char ∗to, unsigned char ∗message, uint32_t messageLen)
- void **free_packet** (**packet_basic** ∗p)

5.6.1 Macro Definition Documentation

5.6.1.1 #define DELAY_TIME 50

Definition at line 38 of file ServerClientBasic.h.

5.6.1.2 #define enum_mode

Definition at line 42 of file ServerClientBasic.h.

5.6.1.3 #define struct_channel

Definition at line 61 of file ServerClientBasic.h.

5.6.1.4 #define struct_packet

Definition at line 49 of file ServerClientBasic.h.

5.6.2 Typedef Documentation

5.6.2.1 typedef struct channel channel

5.6.3 Enumeration Type Documentation

5.6.3.1 enum mode

Enumerator

SERVER

CLIENT

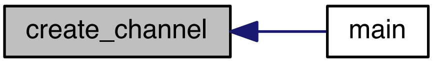
Definition at line 43 of file ServerClientBasic.h.

5.6.4 Function Documentation

5.6.4.1 channel∗ create_channel (char ∗ *fileName,* char ∗ *channelFrom,* char ∗ *channelTo,* mode *channelMode*)

Definition at line 19 of file ServerClientBasic.c.

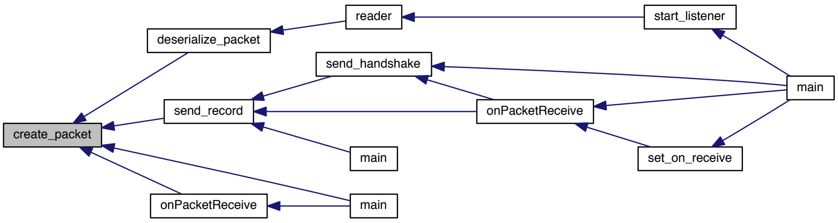
Here is the caller graph for this function:



5.6.4.2 packet_basic∗ create_packet (char ∗ *from,* char ∗ *to,* unsigned char ∗ *message,* uint32_t *messageLen*)

Definition at line 131 of file ServerClientBasic.c.

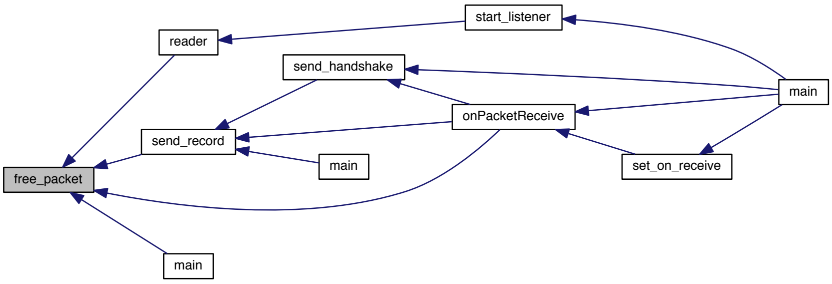
Here is the caller graph for this function:



5.6.4.3 void free_packet (packet_basic ∗ *p*)

Definition at line 152 of file ServerClientBasic.c.

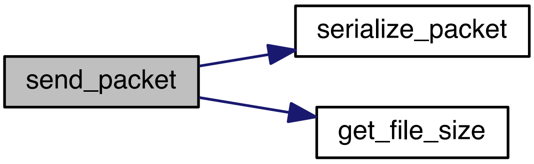
Here is the caller graph for this function:



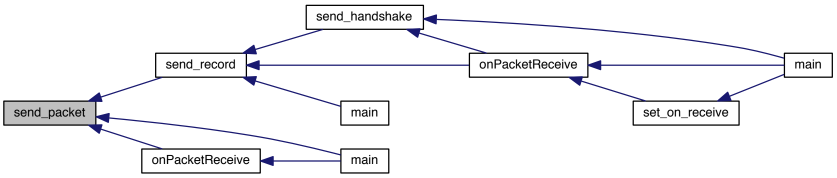
5.6.4.4 int send_packet (channel ∗ *ch,* packet_basic ∗ *p*)

Definition at line 41 of file ServerClientBasic.c.

Here is the call graph for this function:



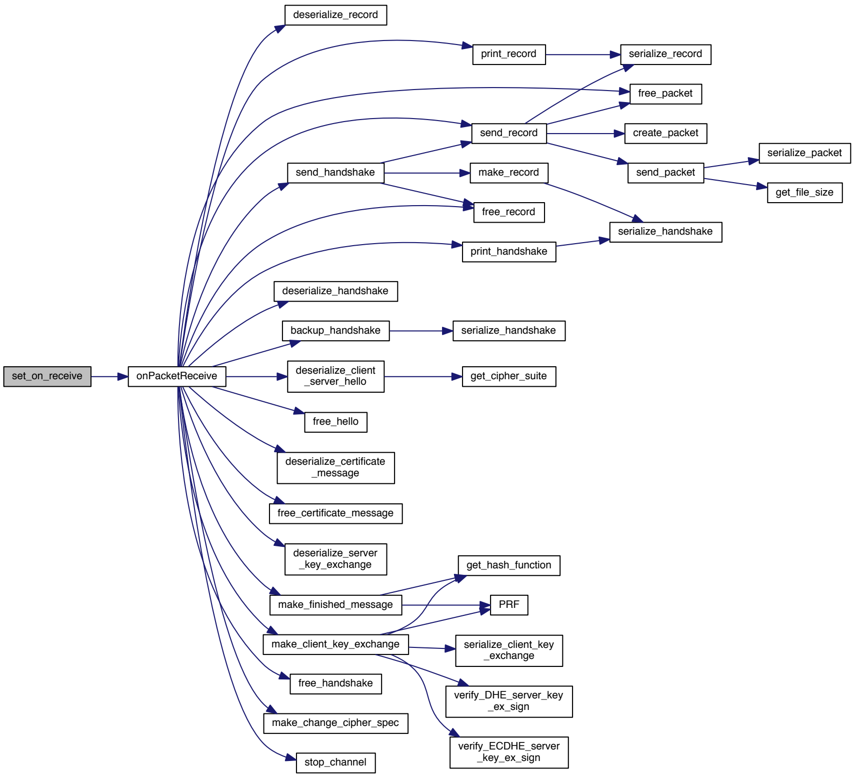
Here is the caller graph for this function:



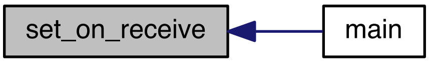
5.6.4.5 int set_on_receive (channel ∗ *ch,* void(∗)(channel ∗ch, packet_basic ∗p) *onPacketReceive*)

Definition at line 32 of file ServerClientBasic.c.

Here is the call graph for this function:



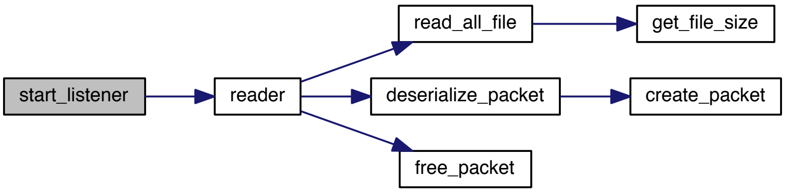
Here is the caller graph for this function:



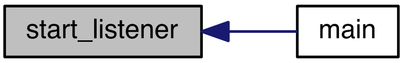
5.6.4.6 int start_listener (channel ∗ *ch*)

Definition at line 103 of file ServerClientBasic.c.

Here is the call graph for this function:



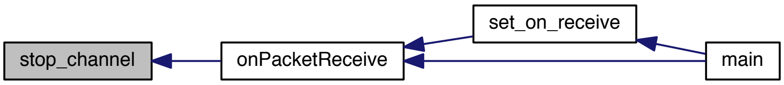
Here is the caller graph for this function:



5.6.4.7 void stop_channel (channel ∗ *ch*)

Definition at line 121 of file ServerClientBasic.c.

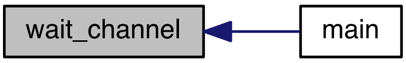
Here is the caller graph for this function:



5.6.4.8 void wait_channel (**channel** ∗ *ch*)

Definition at line 127 of file ServerClientBasic.c.

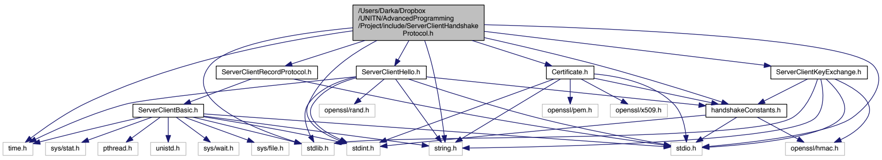
Here is the caller graph for this function:



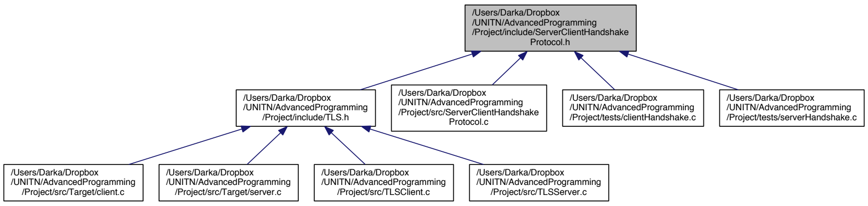
5.7 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/ServerClient↩ HandshakeProtocol.h File Reference

```
#include <stdio.h>
#include <stdint.h>
#include <time.h>
#include <stdlib.h>
#include <string.h>
#include "ServerClientRecordProtocol.h"
#include "handshakeConstants.h"
#include "ServerClientHello.h"
#include "Certificate.h"
#include "ServerClientKeyExchange.h"
```

Include dependency graph for ServerClientHandshakeProtocol.h:



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


/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/include/ServerClientHandshake Protocol.h
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/include/TLS.h
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/src/ServerClientHandshake Protocol.c
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/tests/clientHandshake.c
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/tests/serverHandshake.c
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/src/Target/client.c
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/src/Target/server.c
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/src/TLSClient.c
/Users/Darka/Dropbox /UNITN/AdvancedProgramming /Project/src/TLSServer.c

Data Structures

- struct **handshake**

Functions

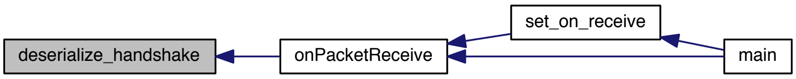
- int **send_handshake** (**channel** ∗ch, **handshake** ∗h)
- void **serialize_handshake** (**handshake** ∗h, unsigned char ∗∗stream, uint32_t ∗streamLen)
- **handshake** ∗ **deserialize_handshake** (unsigned char ∗message, uint32_t messageLen)
- void **print_handshake** (**handshake** ∗h)
- void **free_handshake** (**handshake** ∗h)

5.7.1 Function Documentation

5.7.1.1 handshake∗ deserialize_handshake (unsigned char ∗ *message,* uint32_t *messageLen*)

Definition at line 50 of file ServerClientHandshakeProtocol.c.

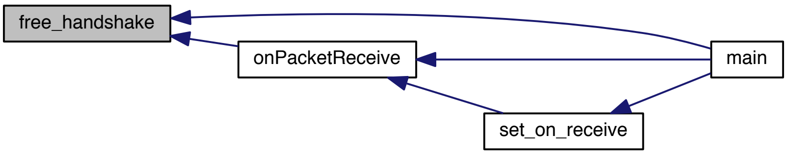
Here is the caller graph for this function:



5.7.1.2 void free_handshake (handshake ∗ *h*)

Definition at line 66 of file ServerClientHandshakeProtocol.c.

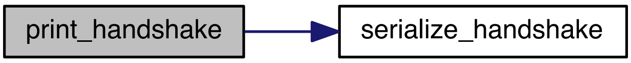
Here is the caller graph for this function:



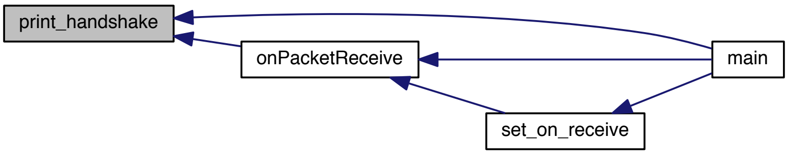
5.7.1.3 void print_handshake (handshake ∗ *h*)

Definition at line 73 of file ServerClientHandshakeProtocol.c.

Here is the call graph for this function:



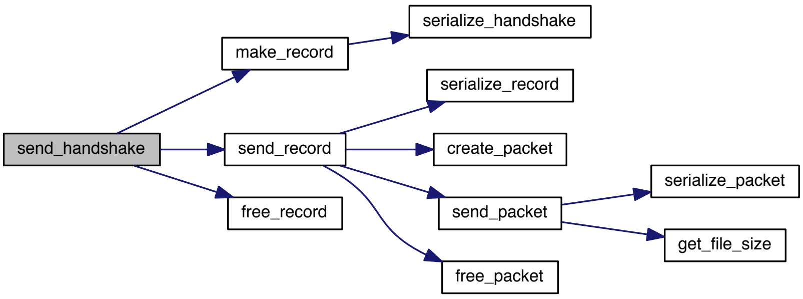
Here is the caller graph for this function:



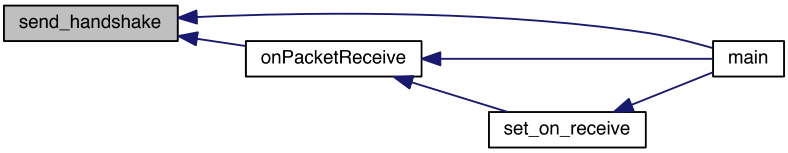
5.7.1.4 int send_handshake (channel * *ch*, handshake * *h*)

Definition at line 25 of file ServerClientHandshakeProtocol.c.

Here is the call graph for this function:



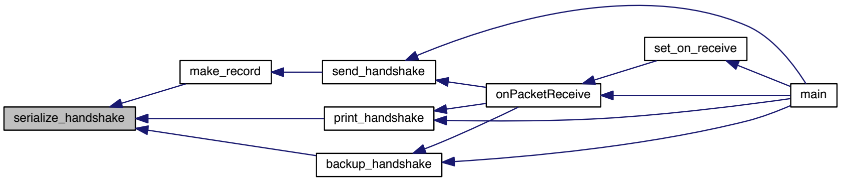
Here is the caller graph for this function:



5.7.1.5 void serialize_handshake (handshake * *h*, unsigned char ** *stream*, uint32_t * *streamLen*)

Definition at line 35 of file ServerClientHandshakeProtocol.c.

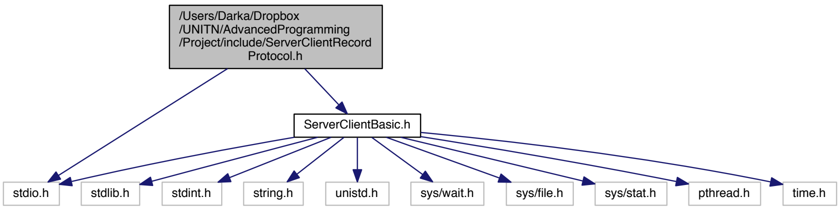
Here is the caller graph for this function:



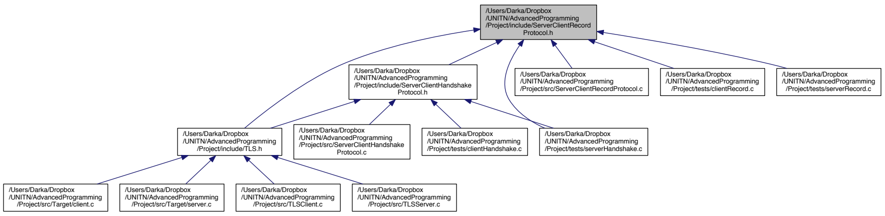
5.8 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/ServerClient↩ RecordProtocol.h File Reference

```
#include <stdio.h>
#include "ServerClientBasic.h"
```

Include dependency graph for ServerClientRecordProtocol.h:



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


/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/ServerClientRecord
Protocol.h
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/ServerClientHandshake
Protocol.h
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/ServerClientRecordProtocol.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/tests/clientRecord.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/tests/serverRecord.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/TLS.h
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/ServerClientHandshake
Protocol.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/tests/clientHandshake.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/tests/serverHandshake.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/Target/client.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/Target/server.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/TLSClient.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/TLSServer.c

Data Structures

- struct **record**

Macros

- #define **REV16**(value) ({(value & 0x00FFU) << 8 | (value & 0xFF00U) >> 8;})
- #define **enum_recordtype**
- #define **struct_record**

Typedefs

- typedef struct **record record**

Enumerations

- enum **recordType** { **HANDSHAKE** = 0x16, **CHANGE_CIPHER_SPEC** = 0x14, **ALERT** = 0x15, **APPLICA↩ TION_DATA** = 0x17 }

Functions

- int **send_record** (**channel** ∗ch, **record** ∗r)
- **record** ∗ **deserialize_record** (unsigned char ∗message, uint32_t messageLen)
- void **serialize_record** (**record** ∗r, unsigned char ∗∗message, uint16_t ∗messageLen)

  Serialize record in a byte stream message and message length are used for return.
- void **print_record** (**record** ∗r)
- void **free_record** (**record** ∗r)

5.8.1 Macro Definition Documentation

5.8.1.1 #define enum_recordtype

Definition at line 22 of file ServerClientRecordProtocol.h.

5.8.1.2 #define REV16(*value*) ({(value & 0x00FFU) << 8 | (value & 0xFF00U) >> 8;})

Definition at line 14 of file ServerClientRecordProtocol.h.

5.8.1.3 #define struct_record

Definition at line 35 of file ServerClientRecordProtocol.h.

5.8.2 Typedef Documentation

5.8.2.1 typedef struct record record

5.8.3 Enumeration Type Documentation

5.8.3.1 enum recordType

Enumerator

HANDSHAKE

CHANGE_CIPHER_SPEC

ALERT

APPLICATION_DATA

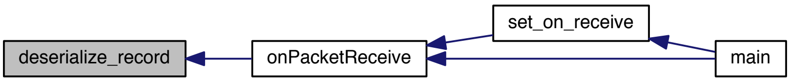
Definition at line 23 of file ServerClientRecordProtocol.h.

5.8.4 Function Documentation

5.8.4.1 record∗ deserialize_record (unsigned char ∗ *message*, uint32_t *messageLen*)

Definition at line 22 of file ServerClientRecordProtocol.c.

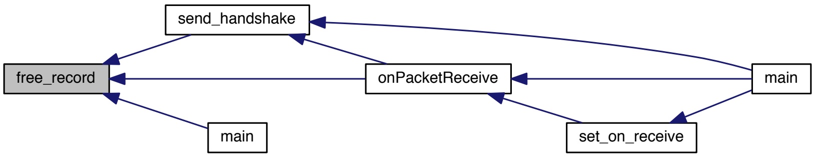
Here is the caller graph for this function:



5.8.4.2 void free_record (record ∗ *r*)

Definition at line 68 of file ServerClientRecordProtocol.c.

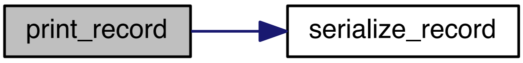
Here is the caller graph for this function:



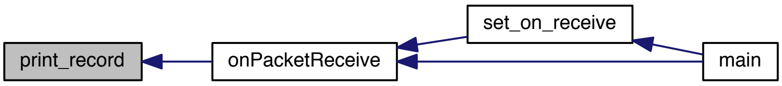
5.8.4.3 void print_record (record ∗ *r*)

Definition at line 48 of file ServerClientRecordProtocol.c.

Here is the call graph for this function:



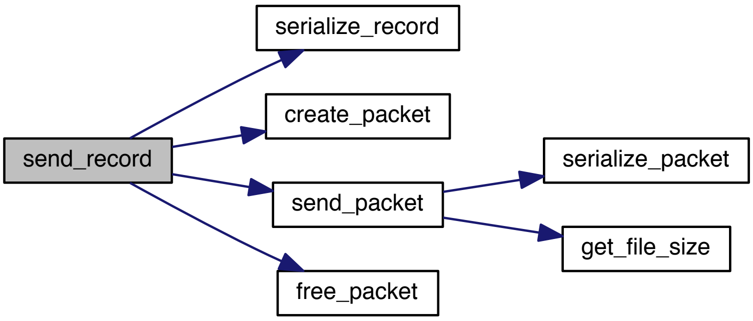
Here is the caller graph for this function:



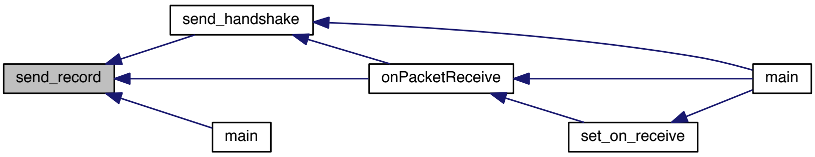
5.8.4.4 int send_record (**channel** ∗ *ch,* **record** ∗ *r*)

Definition at line 36 of file ServerClientRecordProtocol.c.

Here is the call graph for this function:



Here is the caller graph for this function:



5.8.4.5 void serialize_record (**record** ∗ *r,* unsigned char ∗∗ *message,* uint16_t ∗ *messageLen*)

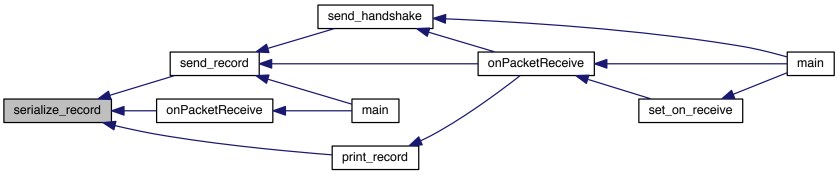
Serialize record in a byte stream message and message length are used for return.

Parameters

message	: pointer to null (the function allocate space for you)
messageLen	: pointer to integer (will contains the message length)

Definition at line 11 of file ServerClientRecordProtocol.c.

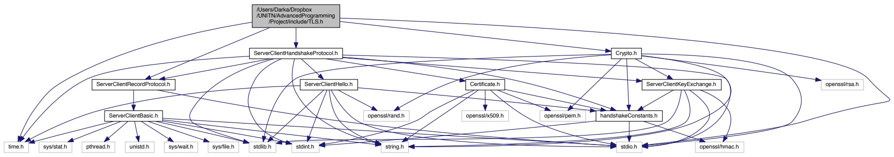
Here is the caller graph for this function:



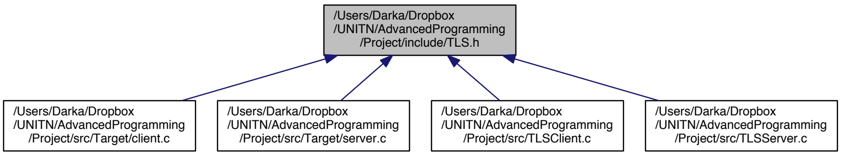
5.9 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/include/TLS.h File Reference

```
#include <stdio.h>
#include <time.h>
#include "ServerClientHandshakeProtocol.h"
#include "ServerClientRecordProtocol.h"
#include "Crypto.h"
```

Include dependency graph for TLS.h:



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


/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/include/TLS.h
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/Target/client.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/Target/server.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/TLSClient.c
/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/TLSServer.c

Data Structures

- struct **TLS_parameters**

Macros

- #define **TLS_parameter_enum**

Functions

- **handshake** ∗ **make_client_hello** (unsigned char ∗client_random)
- **handshake** ∗ **make_client_key_exchange** (**TLS_parameters** ∗**TLS_param**, uint16_t key_ex_alg)
- **record** ∗ **make_change_cipher_spec** ()
- **handshake** ∗ **make_finished_message** (**TLS_parameters** ∗**TLS_param**)
- **handshake** ∗ **make_server_hello** (**TLS_parameters** ∗**TLS_param**, **handshake_hello** ∗client_hello)
- **handshake** ∗ **make_certificate** (**TLS_parameters** ∗**TLS_param**)
- **handshake** ∗ **make_server_key_exchange** (**TLS_parameters** ∗**TLS_param**)
- **handshake** ∗ **make_server_hello_done** ()
- **DHE_server_key_exchange** ∗ **make_DHE_server_key_exchange** (**TLS_parameters** ∗**TLS_param**)
- **ECDHE_server_key_exchange** ∗ **make_ECDHE_server_key_exchange** (**TLS_parameters** ∗**TLS_↩ param**)
- void **backup_handshake** (**TLS_parameters** ∗**TLS_param**, **handshake** ∗h)

5.9.1 Macro Definition Documentation

5.9.1.1 #define TLS_parameter_enum

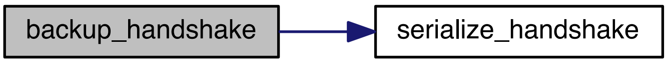
Definition at line 20 of file TLS.h.

5.9.2 Function Documentation

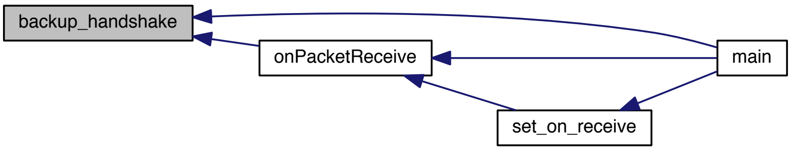
5.9.2.1 void backup_handshake (TLS_parameters ∗ *TLS_param*, handshake ∗ *h*)

Definition at line 13 of file TLSClient.c.

Here is the call graph for this function:



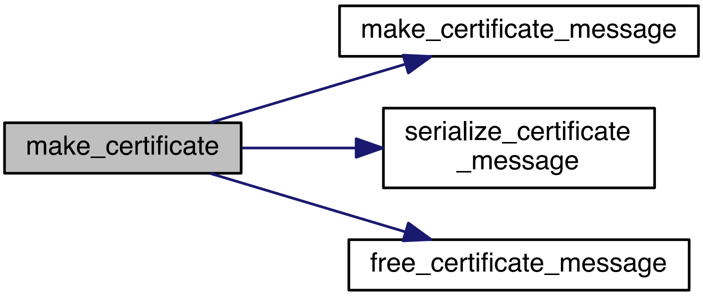
Here is the caller graph for this function:



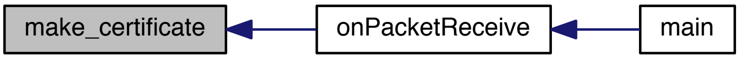
5.9.2.2 handshake∗ make_certificate (TLS_parameters ∗ *TLS_param*)

Definition at line 48 of file TLSServer.c.

Here is the call graph for this function:



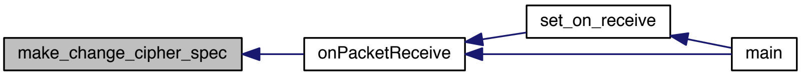
Here is the caller graph for this function:



5.9.2.3 record∗ make_change_cipher_spec ( )

Definition at line 250 of file TLSClient.c.

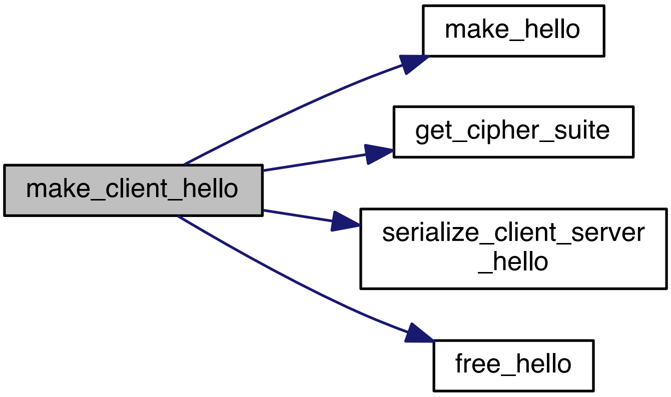
Here is the caller graph for this function:



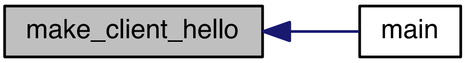
5.9.2.4 handshake∗ make_client_hello (unsigned char ∗ *client_random*)

Definition at line 31 of file TLSClient.c.

Here is the call graph for this function:



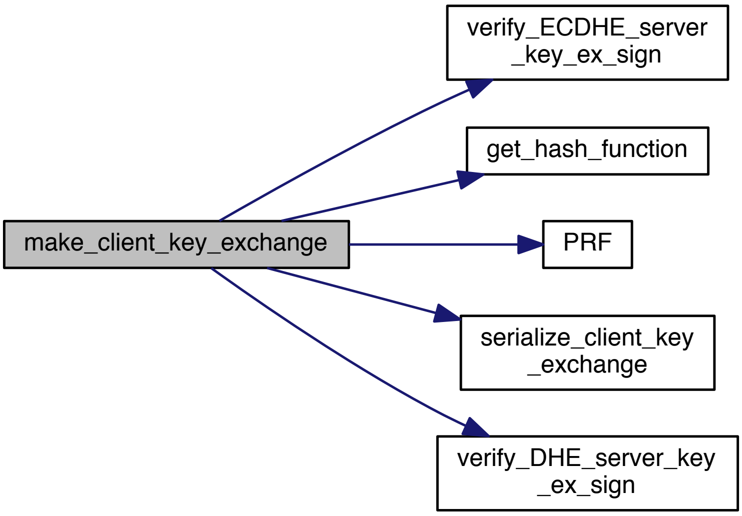
Here is the caller graph for this function:



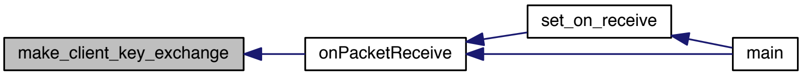
5.9.2.5 handshake∗ make_client_key_exchange (TLS_parameters ∗ *TLS_param,* uint16_t *key_ex_alg*)

Definition at line 65 of file TLSClient.c.

Here is the call graph for this function:


verify_ECDHE_server_key_ex_sign
get_hash_function
make_client_key_exchange
PRF
serialize_client_key_exchange
verify_DHE_server_key_ex_sign

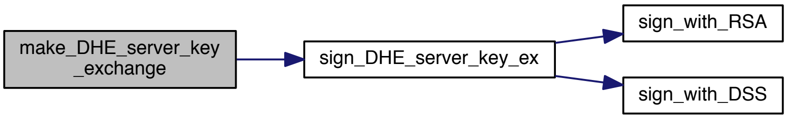
Here is the caller graph for this function:



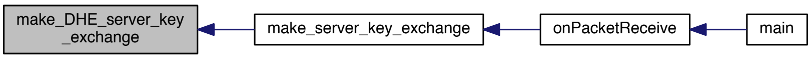
5.9.2.6 DHE_server_key_exchange∗ make_DHE_server_key_exchange (TLS_parameters ∗ *TLS_param*)

Definition at line 113 of file TLSServer.c.

Here is the call graph for this function:



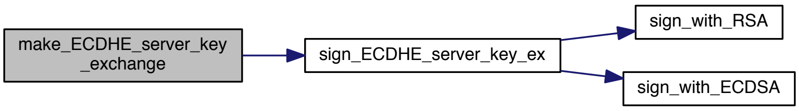
Here is the caller graph for this function:



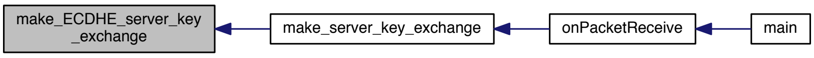
5.9.2.7 ECDHE_server_key_exchange∗ make_ECDHE_server_key_exchange (TLS_parameters ∗ *TLS_param*)

Definition at line 153 of file TLSServer.c.

Here is the call graph for this function:



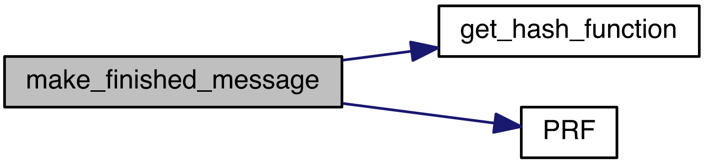
Here is the caller graph for this function:



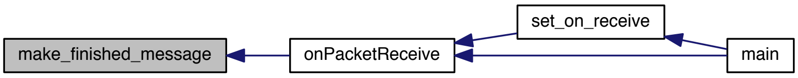
5.9.2.8 handshake∗ make_finished_message (TLS_parameters ∗ *TLS_param*)

Definition at line 262 of file TLSClient.c.

Here is the call graph for this function:



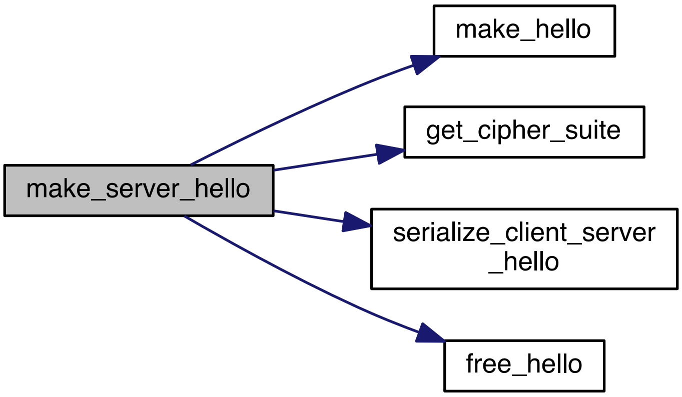
Here is the caller graph for this function:



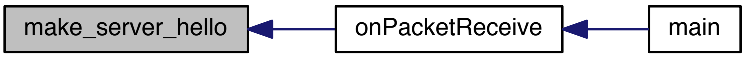
5.9.2.9 handshake∗ make_server_hello (TLS_parameters ∗ *TLS_param,* handshake_hello ∗ *client_hello*)

Definition at line 11 of file TLSServer.c.

Here is the call graph for this function:



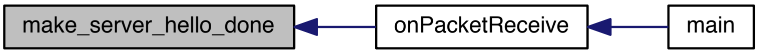
Here is the caller graph for this function:



5.9.2.10 handshake* make_server_hello_done ( )

Definition at line 188 of file TLSServer.c.

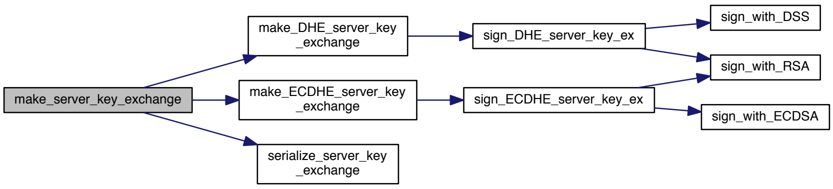
Here is the caller graph for this function:



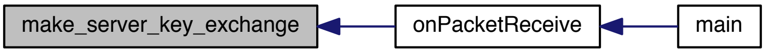
5.9.2.11 handshake* make_server_key_exchange (TLS_parameters * *TLS_param*)

Definition at line 84 of file TLSServer.c.

Here is the call graph for this function:



Here is the caller graph for this function:

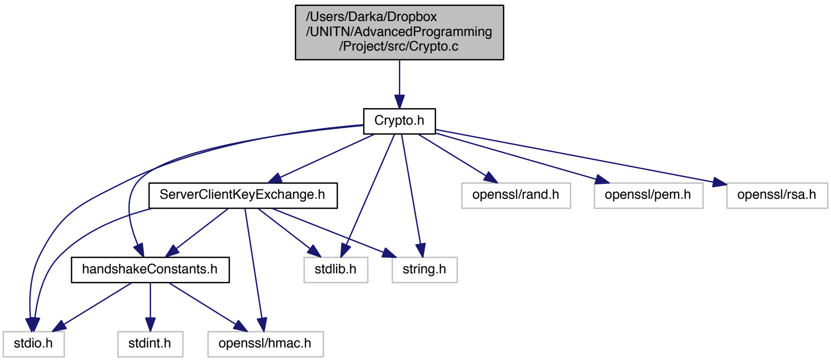


5.10 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/README.md File Reference

5.11 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/src/Crypto.c File Reference

```
#include "Crypto.h"
```

Include dependency graph for Crypto.c:



Functions

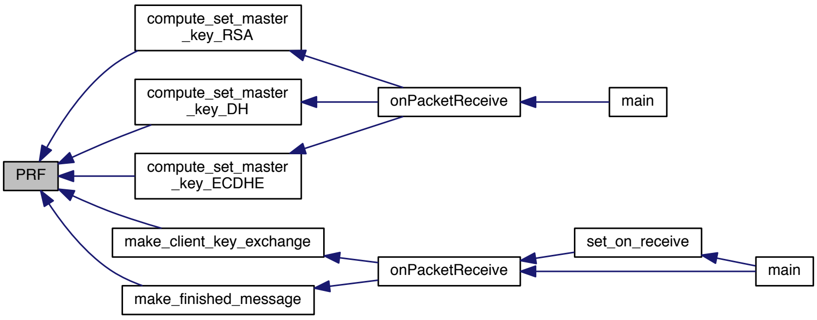
- int **sign_with_RSA** (unsigned char ∗∗signature, unsigned int ∗signature_length, unsigned int to_sign_len, unsigned char ∗to_sign)
- int **sign_with_DSS** (unsigned char ∗∗signature, unsigned int ∗signature_length, unsigned int to_sign_len, unsigned char ∗to_sign)
- int **sign_with_ECDSA** (unsigned char ∗∗signature, unsigned int ∗signature_length, unsigned int to_sign_len, unsigned char ∗to_sign)
- void **PRF** (const EVP_MD ∗hash, unsigned char ∗secret, int secret_len, char ∗label, unsigned char ∗seed, int seed_len, int result_len, unsigned char ∗∗result)
- int **verify_DHE_server_key_ex_sign** (X509 ∗certificate, unsigned char ∗client_random, unsigned char ∗server_random, **DHE_server_key_exchange** ∗server_key_ex, **authentication_algorithm** au)
- int **sign_DHE_server_key_ex** (unsigned char ∗client_random, unsigned char ∗server_random, **DHE_server_key_exchange** ∗server_key_ex, **authentication_algorithm** au)
- int **sign_ECDHE_server_key_ex** (unsigned char ∗client_random, unsigned char ∗server_random, **ECDHE_server_key_exchange** ∗server_key_ex, **authentication_algorithm** au)
- int **verify_ECDHE_server_key_ex_sign** (X509 ∗certificate, unsigned char ∗client_random, unsigned char ∗server_random, **ECDHE_server_key_exchange** ∗server_key_ex, **authentication_algorithm** au)

5.11.1 Function Documentation

5.11.1.1 void PRF (const EVP_MD ∗ *hash,* unsigned char ∗ *secret,* int *secret_len,* char ∗ *label,* unsigned char ∗ *seed,* int *seed_len,* int *result_len,* unsigned char ∗∗ *result*)

Definition at line 15 of file Crypto.c.

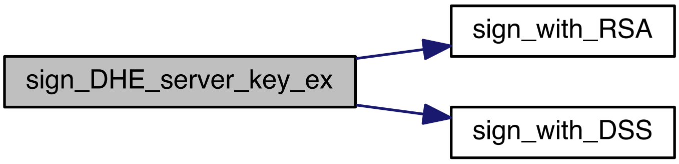
Here is the caller graph for this function:



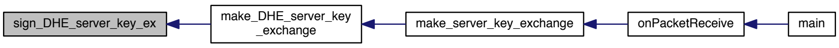
5.11.1.2 int sign_DHE_server_key_ex (unsigned char ∗ *client_random*, unsigned char ∗ *server_random*, DHE_server_key_exchange ∗ *server_key_ex*, authentication_algorithm *au*)

Definition at line 127 of file Crypto.c.

Here is the call graph for this function:



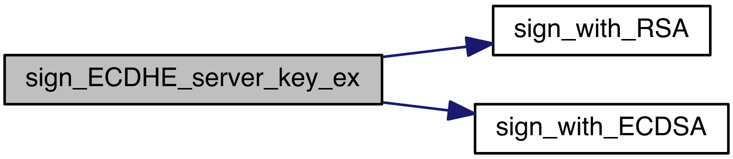
Here is the caller graph for this function:



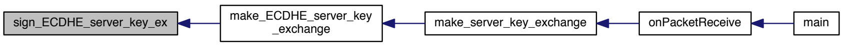
5.11.1.3 int sign_ECDHE_server_key_ex (unsigned char ∗ *client_random*, unsigned char ∗ *server_random*, ECDHE_server_key_exchange ∗ *server_key_ex*, authentication_algorithm *au*)

Definition at line 199 of file Crypto.c.

Here is the call graph for this function:



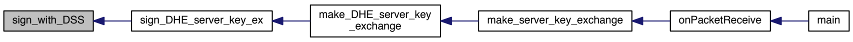
Here is the caller graph for this function:



5.11.1.4 int sign_with_DSS (unsigned char ∗∗ *signature,* unsigned int ∗ *signature_length,* unsigned int *to_sign_len,* unsigned char ∗ *to_sign*)

Definition at line 337 of file Crypto.c.

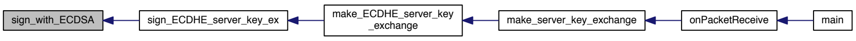
Here is the caller graph for this function:



5.11.1.5 int sign_with_ECDSA (unsigned char ∗∗ *signature,* unsigned int ∗ *signature_length,* unsigned int *to_sign_len,* unsigned char ∗ *to_sign*)

Definition at line 384 of file Crypto.c.

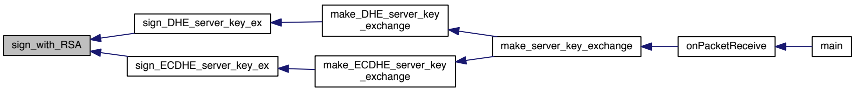
Here is the caller graph for this function:



5.11.1.6 int sign_with_RSA (unsigned char ∗∗ *signature,* unsigned int ∗ *signature_length,* unsigned int *to_sign_len,* unsigned char ∗ *to_sign*)

Definition at line 358 of file Crypto.c.

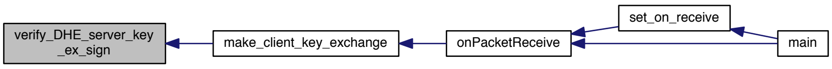
Here is the caller graph for this function:



5.11.1.7 int verify_DHE_server_key_ex_sign (X509 ∗ *certificate,* unsigned char ∗ *client_random,* unsigned char ∗ *server_random,* DHE_server_key_exchange ∗ *server_key_ex,* authentication_algorithm *au*)

Definition at line 42 of file Crypto.c.

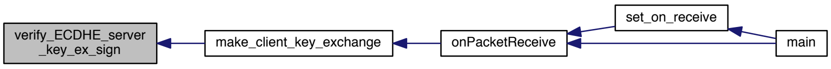
Here is the caller graph for this function:



5.11.1.8 int verify_ECDHE_server_key_ex_sign (X509 ∗ *certificate,* unsigned char ∗ *client_random,* unsigned char ∗ *server_random,* ECDHE_server_key_exchange ∗ *server_key_ex,* authentication_algorithm *au*)

Definition at line 259 of file Crypto.c.

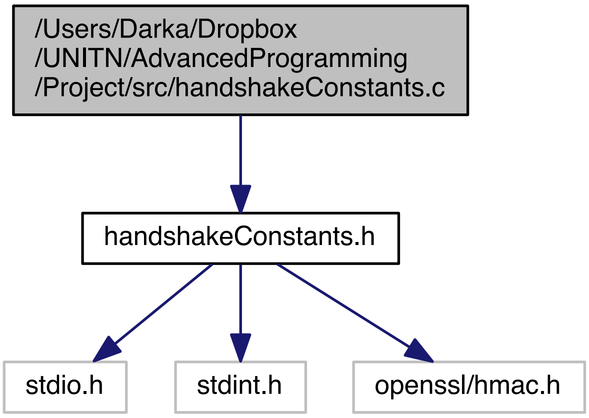
Here is the caller graph for this function:



5.12 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/src/handshake↩ Constants.c File Reference

`#include "handshakeConstants.h"`

Include dependency graph for handshakeConstants.c:



Functions

- const EVP_MD ∗ **get_hash_function** (**hash_algorithm** h)
- **cipher_suite_t get_cipher_suite** (uint16_t id)

Variables

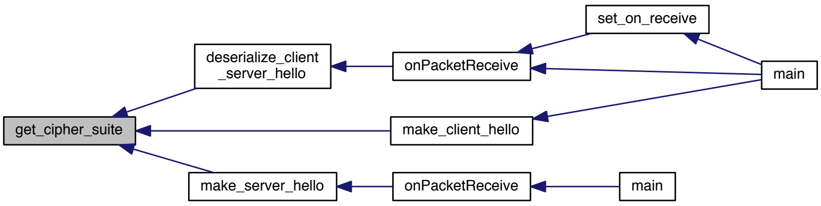
- int **cipher_suite_len** = 61
- **cipher_suite_t cipher_suite_list** [ ]

5.12.1 Function Documentation

5.12.1.1 cipher_suite_t get_cipher_suite (uint16_t *id*)

Definition at line 103 of file handshakeConstants.c.

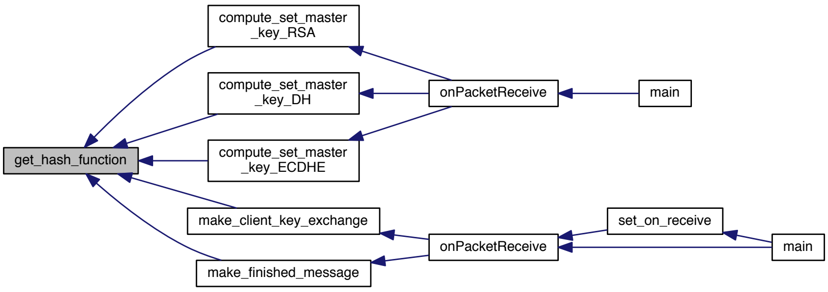
Here is the caller graph for this function:



5.12.1.2 const EVP_MD∗ get_hash_function (hash_algorithm *h*)

Definition at line 83 of file handshakeConstants.c.

Here is the caller graph for this function:



5.12.2 Variable Documentation

5.12.2.1 int cipher_suite_len = 61

Definition at line 11 of file handshakeConstants.c.

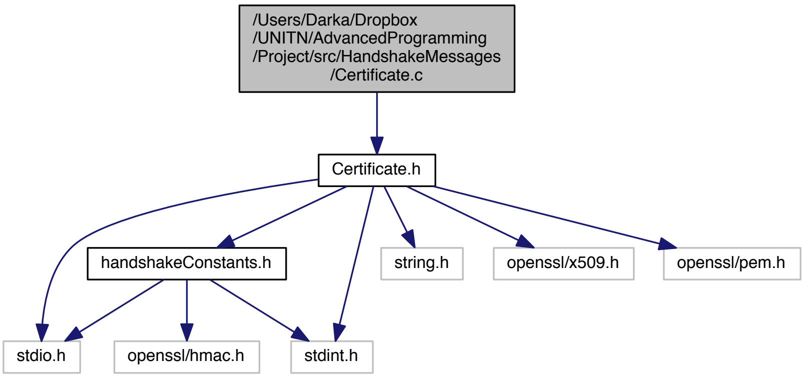
5.12.2.2 cipher_suite_t cipher_suite_list[ ]

Definition at line 13 of file handshakeConstants.c.

5.13 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/src/Handshake←Messages/Certificate.c File Reference

```
#include "Certificate.h"
```

Include dependency graph for Certificate.c:



Functions

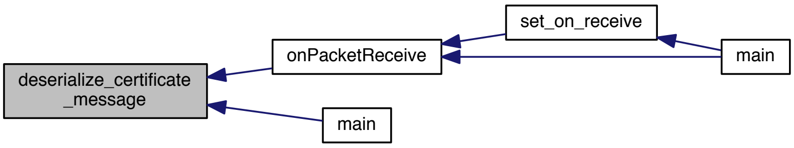
- **certificate_message** ∗ **make_certificate_message** (char ∗cert_file_name)
- void **serialize_certificate_message** (**certificate_message** ∗cert, unsigned char ∗∗stream, uint32_t ∗len)
- **certificate_message** ∗ **deserialize_certificate_message** (unsigned char ∗stream, uint32_t len)
- void **free_certificate_message** (**certificate_message** ∗cert)

5.13.1 Function Documentation

5.13.1.1 certificate_message∗ deserialize_certificate_message (unsigned char ∗ *stream,* uint32_t *len*)

Definition at line 83 of file Certificate.c.

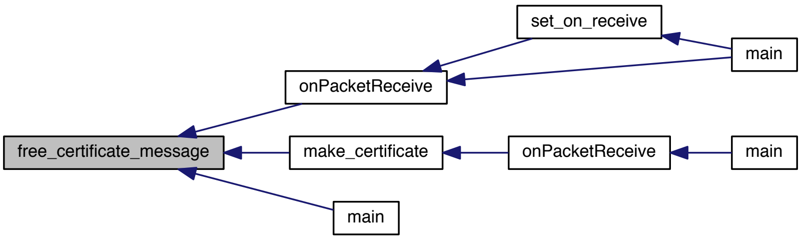
Here is the caller graph for this function:



5.13.1.2 void free_certificate_message (certificate_message ∗ *cert*)

Definition at line 105 of file Certificate.c.

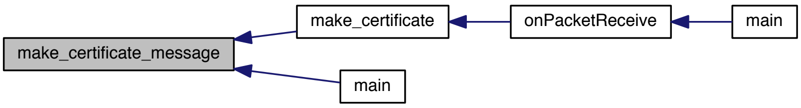
Here is the caller graph for this function:



5.13.1.3 certificate_message∗ make_certificate_message (char ∗ *cert_file_name*)

Definition at line 16 of file Certificate.c.

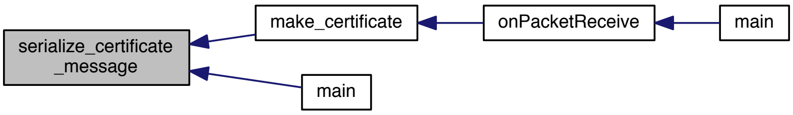
Here is the caller graph for this function:



5.13.1.4 void serialize_certificate_message (certificate_message ∗ *cert,* unsigned char ∗∗ *stream,* uint32_t ∗ *len*)

Definition at line 44 of file Certificate.c.

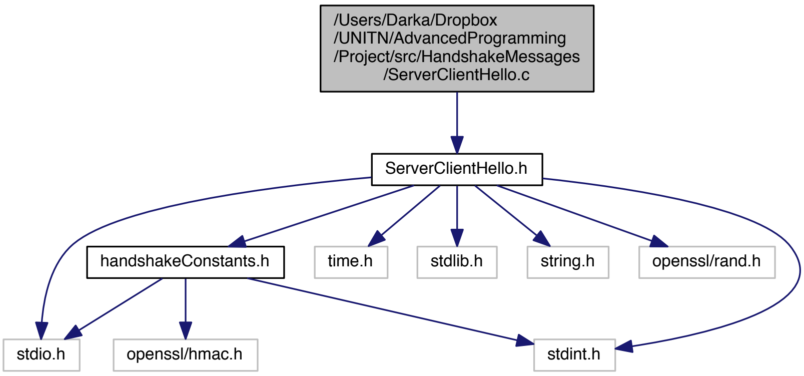
Here is the caller graph for this function:



5.14 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/src/Handshake↩ Messages/ServerClientHello.c File Reference

`#include "ServerClientHello.h"`

Include dependency graph for ServerClientHello.c:



Functions

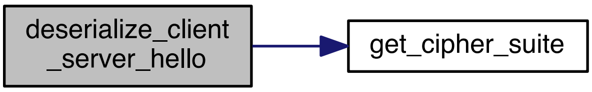
- **handshake_hello** ∗ **make_hello** (**session_id** session)
- void **serialize_client_server_hello** (**handshake_hello** ∗hello, unsigned char ∗∗stream, uint32_t ∗streamLen, **channel_mode mode**)
- **handshake_hello** ∗ **deserialize_client_server_hello** (unsigned char ∗stream, uint32_t streamLen, **channel_mode mode**)
- void **free_hello** (**handshake_hello** ∗h)

5.14.1 Function Documentation

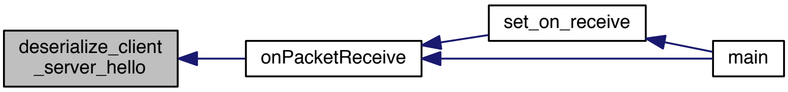
5.14.1.1 handshake_hello∗ deserialize_client_server_hello (unsigned char ∗ *stream*, uint32_t *streamLen*, channel_mode *mode*)

Definition at line 106 of file ServerClientHello.c.

Here is the call graph for this function:



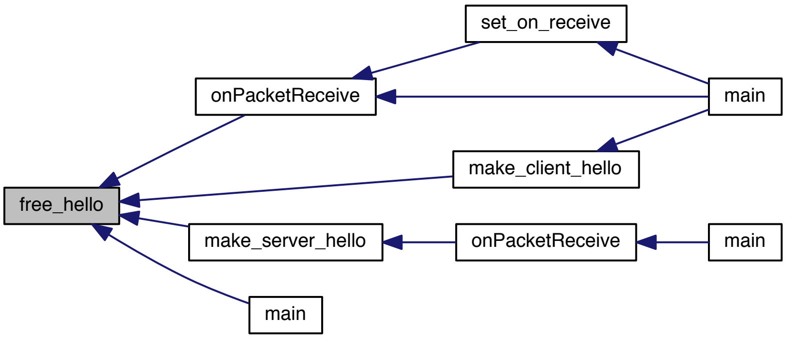
Here is the caller graph for this function:



5.14.1.2 void free_hello (handshake_hello ∗ *h*)

Definition at line 209 of file ServerClientHello.c.

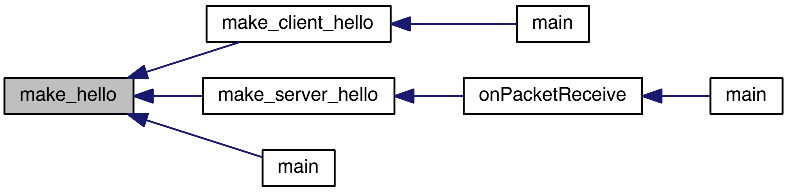
Here is the caller graph for this function:



5.14.1.3 handshake_hello∗ make_hello (session_id *session*)

Definition at line 15 of file ServerClientHello.c.

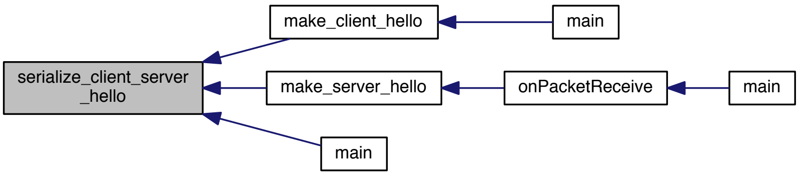
Here is the caller graph for this function:



5.14.1.4 void serialize_client_server_hello (handshake_hello ∗ *hello,* unsigned char ∗∗ *stream,* uint32_t ∗ *streamLen,* channel_mode *mode*)

Definition at line 37 of file ServerClientHello.c.

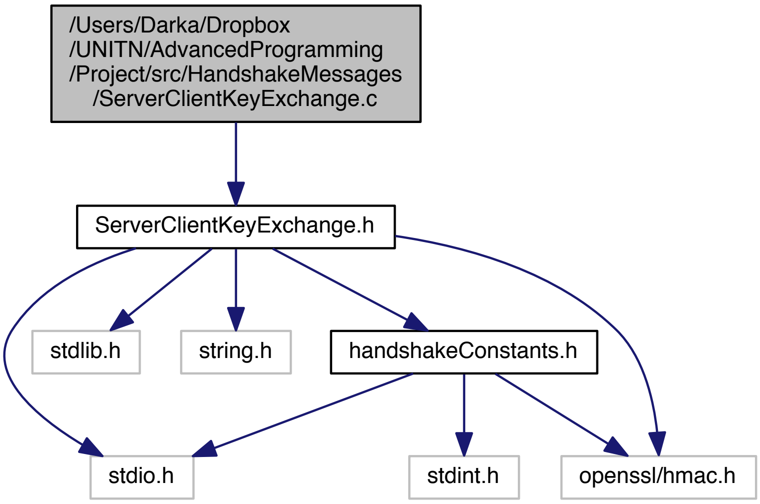
Here is the caller graph for this function:



5.15 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/src/Handshake↩ Messages/ServerClientKeyExchange.c File Reference

```
#include "ServerClientKeyExchange.h"
```

Include dependency graph for ServerClientKeyExchange.c:


/Users/Darka/Dropbox
/UNITN/AdvancedProgramming
/Project/src/HandshakeMessages
/ServerClientKeyExchange.c
ServerClientKeyExchange.h
stdlib.h
string.h
handshakeConstants.h
stdio.h
stdint.h
openssl/hmac.h

Functions

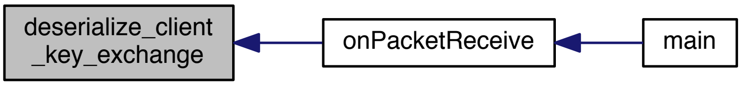
- void **serialize_server_key_exchange** (void ∗server_key_exchange, unsigned char ∗∗stream, uint32_↩ t ∗streamLen, **key_exchange_algorithm** kx)
- void ∗ **deserialize_server_key_exchange** (uint32_t message_len, unsigned char ∗message, **key_↩ exchange_algorithm** kx)
- void **serialize_client_key_exchange** (**client_key_exchange** ∗**client_key_exchange**, unsigned char ∗∗stream, uint32_t ∗streamLen)
- void ∗ **deserialize_client_key_exchange** (uint32_t message_len, unsigned char ∗message)
- void **free_server_key_exchange** (void ∗server_key_ex, **cipher_suite_t** cipher_suite)

5.15.1 Function Documentation

5.15.1.1 void∗ deserialize_client_key_exchange (uint32_t *message_len,* unsigned char ∗ *message*)

Definition at line 185 of file ServerClientKeyExchange.c.

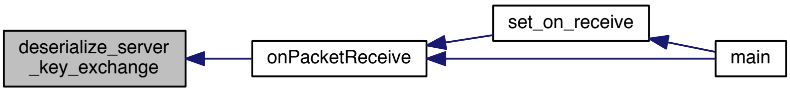
Here is the caller graph for this function:



5.15.1.2 void∗ deserialize_server_key_exchange (uint32_t *message_len,* unsigned char ∗ *message,* key_exchange_algorithm *kx*)

Definition at line 104 of file ServerClientKeyExchange.c.

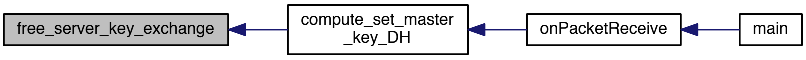
Here is the caller graph for this function:



5.15.1.3 void free_server_key_exchange (void ∗ *server_key_ex,* cipher_suite_t *cipher_suite*)

Definition at line 199 of file ServerClientKeyExchange.c.

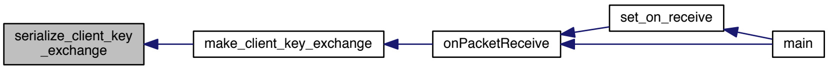
Here is the caller graph for this function:



5.15.1.4 void serialize_client_key_exchange (client_key_exchange ∗ *client_key_exchange,* unsigned char ∗∗ *stream,* uint32_t ∗ *streamLen*)

Definition at line 171 of file ServerClientKeyExchange.c.

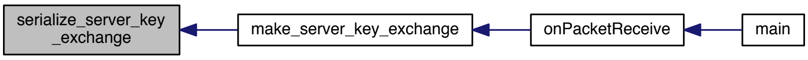
Here is the caller graph for this function:



5.15.1.5 void serialize_server_key_exchange (void ∗ *server_key_exchange,* unsigned char ∗∗ *stream,* uint32_t ∗ *streamLen,* key_exchange_algorithm *kx*)

Definition at line 15 of file ServerClientKeyExchange.c.

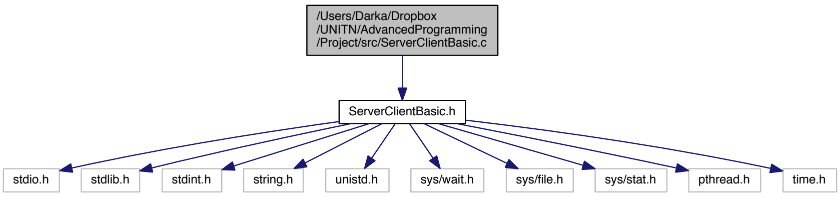
Here is the caller graph for this function:



5.16 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/src/ServerClient↩ Basic.c File Reference

```
#include "ServerClientBasic.h"
```

Include dependency graph for ServerClientBasic.c:



Functions

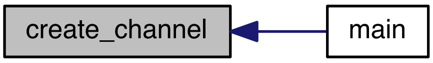
- void **free_packet** (**packet_basic** ∗p)
- long long **get_file_size** (int fd)
- uint32_t **read_all_file** (int fd, unsigned char ∗∗p)
- **packet_basic** ∗ **deserialize_packet** (unsigned char ∗str, uint32_t fileLen)
- void **serialize_packet** (**packet_basic** ∗p, unsigned char ∗∗str, uint32_t ∗strLen)
- **channel** ∗ **create_channel** (char ∗fileName, char ∗channelFrom, char ∗channelTo, **mode** channelMode)
- int **set_on_receive** (**channel** ∗ch, void(∗**onPacketReceive**)(**channel** ∗ch, **packet_basic** ∗p))
- int **send_packet** (**channel** ∗ch, **packet_basic** ∗p)
- void **reader** (void ∗data)
- int **start_listener** (**channel** ∗ch)
- void **stop_channel** (**channel** ∗ch)
- void **wait_channel** (**channel** ∗ch)
- **packet_basic** ∗ **create_packet** (char ∗from, char ∗to, unsigned char ∗message, uint32_t messageLen)

5.16.1 Function Documentation

5.16.1.1 channel∗ create_channel (char ∗ *fileName,* char ∗ *channelFrom,* char ∗ *channelTo,* mode *channelMode*)

Definition at line 19 of file ServerClientBasic.c.

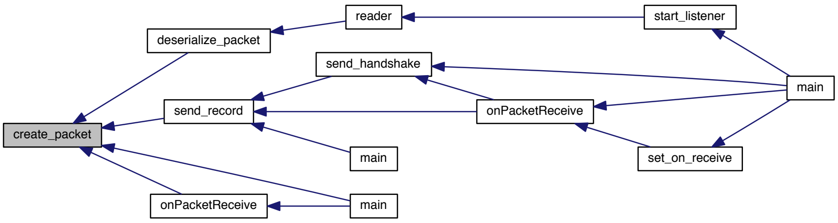
Here is the caller graph for this function:



5.16.1.2 packet_basic∗ create_packet (char ∗ *from,* char ∗ *to,* unsigned char ∗ *message,* uint32_t *messageLen*)

Definition at line 131 of file ServerClientBasic.c.

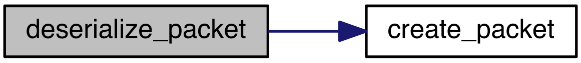
Here is the caller graph for this function:



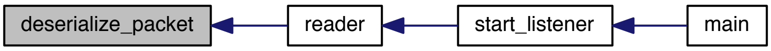
5.16.1.3 packet_basic ∗ deserialize_packet (unsigned char ∗ *str,* uint32_t *fileLen*)

Definition at line 205 of file ServerClientBasic.c.

Here is the call graph for this function:



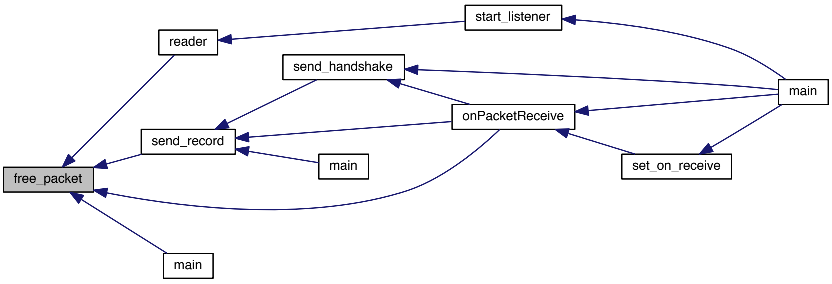
Here is the caller graph for this function:



5.16.1.4 void free_packet (packet_basic ∗ *p*)

Definition at line 152 of file ServerClientBasic.c.

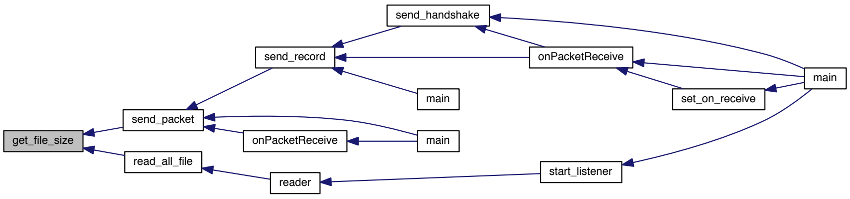
Here is the caller graph for this function:



5.16.1.5 long long get_file_size (int *fd*)

Definition at line 169 of file ServerClientBasic.c.

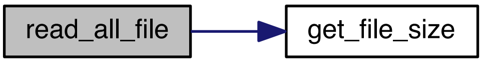
Here is the caller graph for this function:



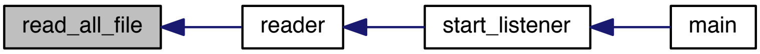
5.16.1.6 uint32_t read_all_file (int *fd*, unsigned char ∗∗ *p*)

Definition at line 184 of file ServerClientBasic.c.

Here is the call graph for this function:



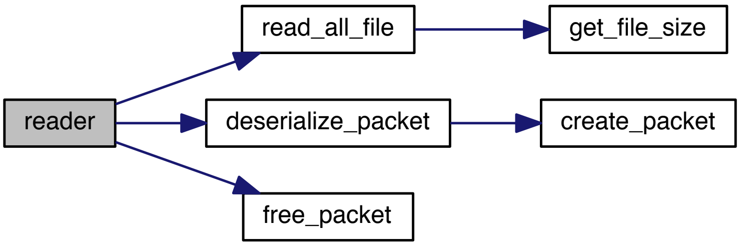
Here is the caller graph for this function:



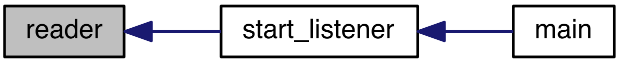
5.16.1.7 void reader (void ∗ *data*)

Definition at line 77 of file ServerClientBasic.c.

Here is the call graph for this function:



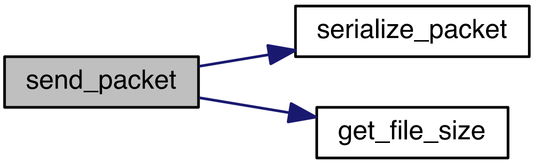
Here is the caller graph for this function:



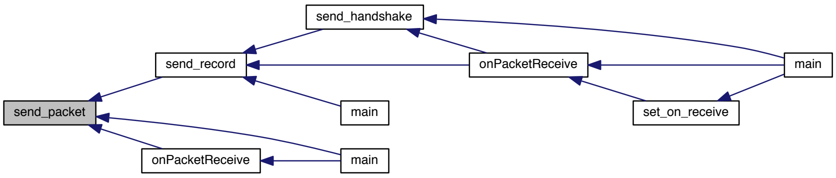
5.16.1.8 int send_packet (channel ∗ *ch,* packet_basic ∗ *p*)

Definition at line 41 of file ServerClientBasic.c.

Here is the call graph for this function:



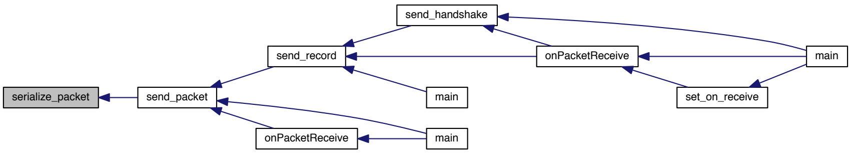
Here is the caller graph for this function:



5.16.1.9 void serialize_packet (packet_basic ∗ *p,* unsigned char ∗∗ *str,* uint32_t ∗ *strLen*)

Definition at line 241 of file ServerClientBasic.c.

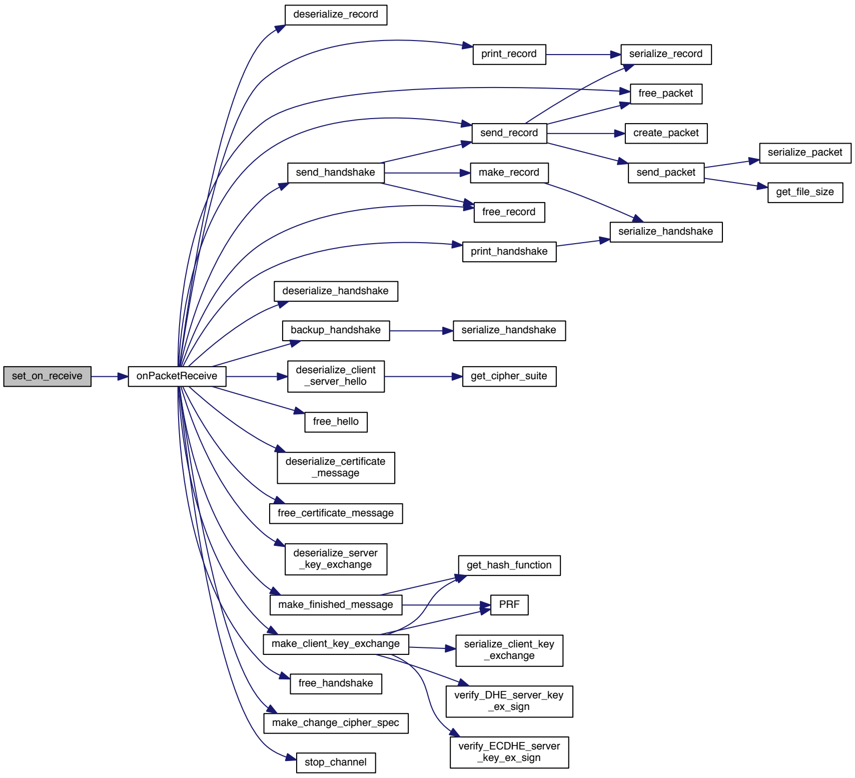
Here is the caller graph for this function:



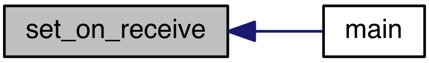
5.16.1.10 int set_on_receive (channel * *ch*, void(*)(channel *ch, packet_basic *p) *onPacketReceive*)

Definition at line 32 of file ServerClientBasic.c.

Here is the call graph for this function:



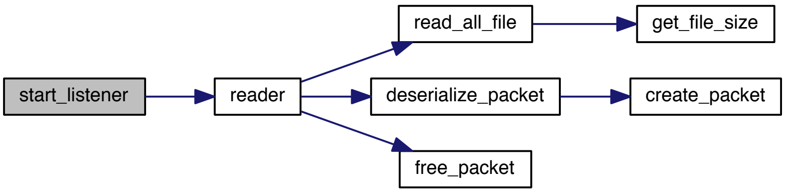
Here is the caller graph for this function:



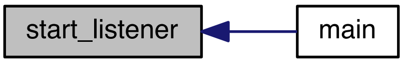
5.16.1.11 int start_listener (channel ∗ *ch*)

Definition at line 103 of file ServerClientBasic.c.

Here is the call graph for this function:



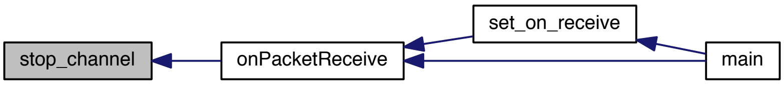
Here is the caller graph for this function:



5.16.1.12 void stop_channel (channel ∗ *ch*)

Definition at line 121 of file ServerClientBasic.c.

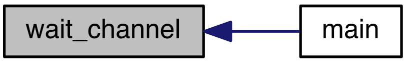
Here is the caller graph for this function:



5.16.1.13 void wait_channel (**channel** ∗ *ch*)

Definition at line 127 of file ServerClientBasic.c.

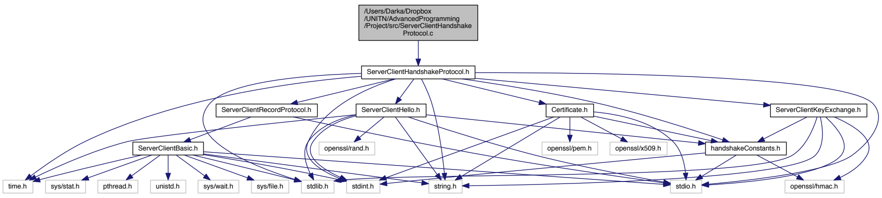
Here is the caller graph for this function:



5.17 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/src/ServerClient↩ HandshakeProtocol.c File Reference

```
#include "ServerClientHandshakeProtocol.h"
```

Include dependency graph for ServerClientHandshakeProtocol.c:



Functions

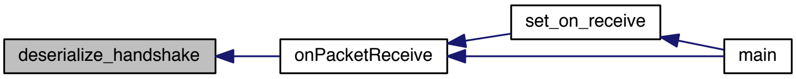
- **record** ∗ **make_record** (**handshake** ∗h)
- int **send_handshake** (**channel** ∗ch, **handshake** ∗h)
- void **serialize_handshake** (**handshake** ∗h, unsigned char ∗∗stream, uint32_t ∗streamLen)
- **handshake** ∗ **deserialize_handshake** (unsigned char ∗message, uint32_t messageLen)
- void **free_handshake** (**handshake** ∗h)
- void **print_handshake** (**handshake** ∗h)

5.17.1 Function Documentation

5.17.1.1 handshake∗ deserialize_handshake (unsigned char ∗ *message,* uint32_t *messageLen*)

Definition at line 50 of file ServerClientHandshakeProtocol.c.

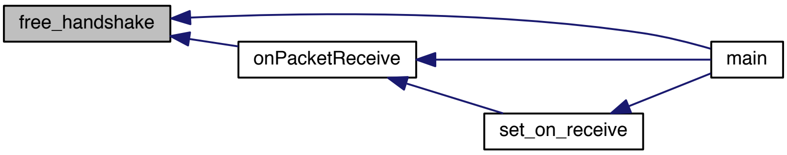
Here is the caller graph for this function:



5.17.1.2 void free_handshake (handshake ∗ *h*)

Definition at line 66 of file ServerClientHandshakeProtocol.c.

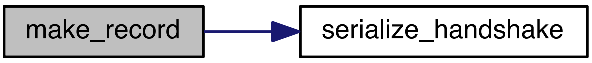
Here is the caller graph for this function:



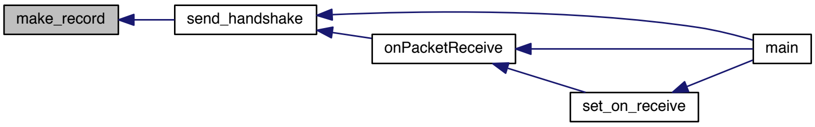
5.17.1.3 record∗ make_record (handshake ∗ *h*)

Definition at line 11 of file ServerClientHandshakeProtocol.c.

Here is the call graph for this function:



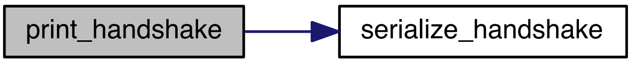
Here is the caller graph for this function:



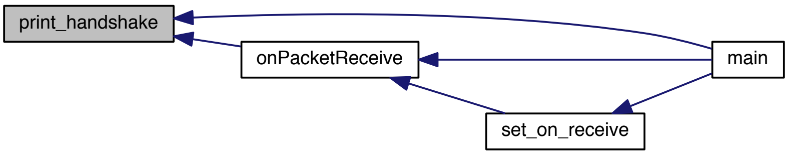
5.17.1.4 void print_handshake (handshake ∗ *h*)

Definition at line 73 of file ServerClientHandshakeProtocol.c.

Here is the call graph for this function:



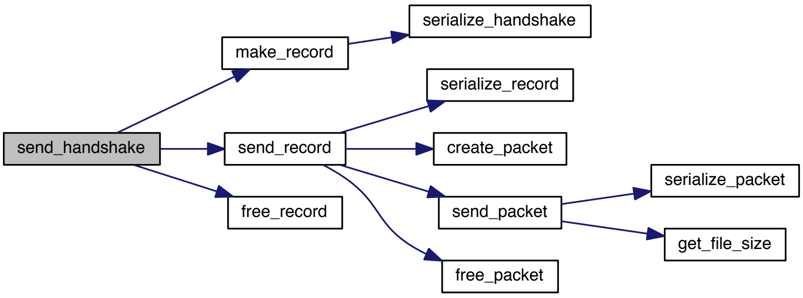
Here is the caller graph for this function:



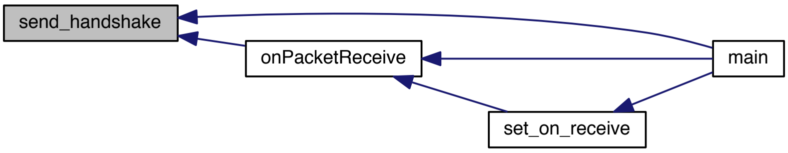
5.17.1.5 int send_handshake (channel ∗ *ch,* handshake ∗ *h*)

Definition at line 25 of file ServerClientHandshakeProtocol.c.

Here is the call graph for this function:



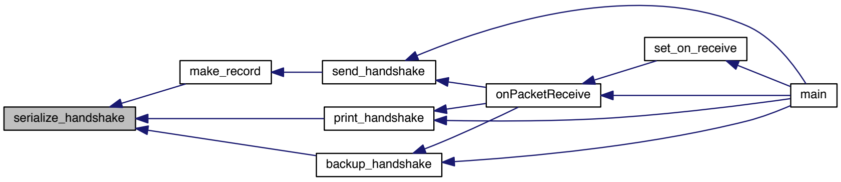
Here is the caller graph for this function:



5.17.1.6 void serialize_handshake (handshake ∗ *h,* unsigned char ∗∗ *stream,* uint32_t ∗ *streamLen*)

Definition at line 35 of file ServerClientHandshakeProtocol.c.

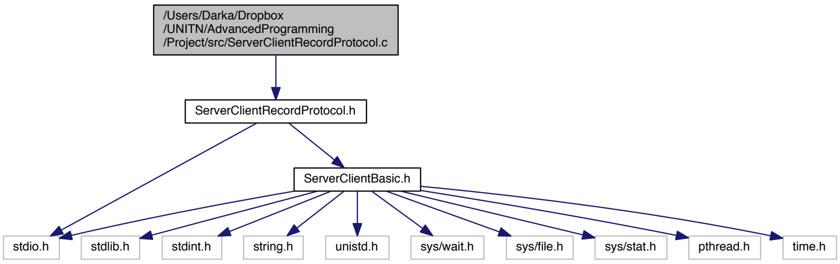
Here is the caller graph for this function:



5.18 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/src/ServerClient↩ RecordProtocol.c File Reference

```
#include "ServerClientRecordProtocol.h"
```

Include dependency graph for ServerClientRecordProtocol.c:



Functions

- void **serialize_record** (**record** ∗r, unsigned char ∗∗message, uint16_t ∗messageLen)

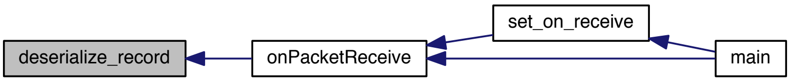
  Serialize record in a byte stream message and message length are used for return.
- **record** ∗ **deserialize_record** (unsigned char ∗message, uint32_t messageLen)
- int **send_record** (**channel** ∗ch, **record** ∗r)
- void **print_record** (**record** ∗r)
- void **free_record** (**record** ∗r)

5.18.1 Function Documentation

5.18.1.1 record∗ deserialize_record (unsigned char ∗ *message,* uint32_t *messageLen*)

Definition at line 22 of file ServerClientRecordProtocol.c.

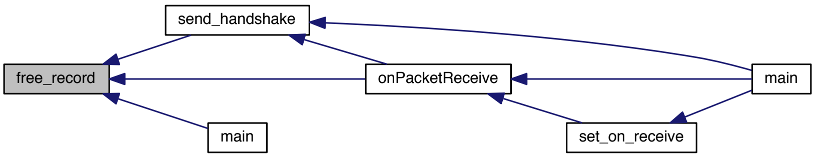
Here is the caller graph for this function:



5.18.1.2 void free_record (record ∗ *r*)

Definition at line 68 of file ServerClientRecordProtocol.c.

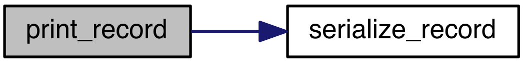
Here is the caller graph for this function:



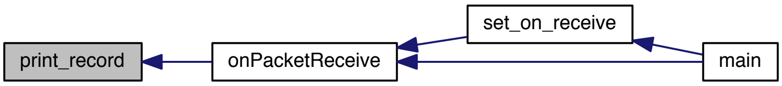
5.18.1.3 void print_record (record ∗ *r*)

Definition at line 48 of file ServerClientRecordProtocol.c.

Here is the call graph for this function:



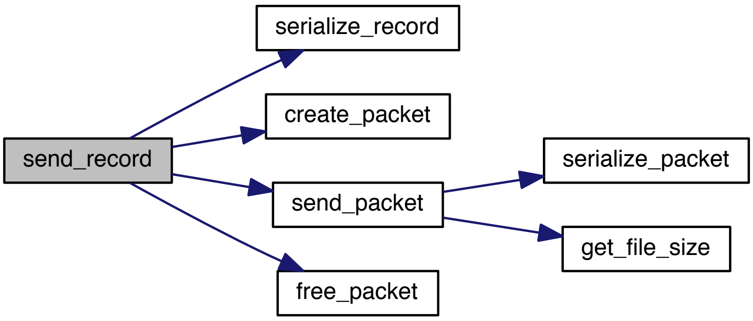
Here is the caller graph for this function:



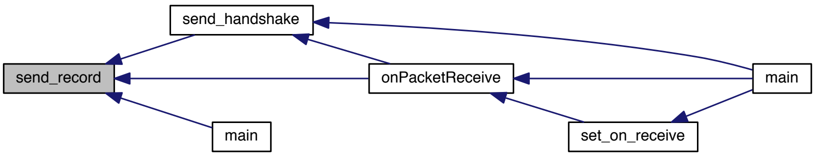
5.18.1.4 int send_record (channel ∗ *ch,* record ∗ *r*)

Definition at line 36 of file ServerClientRecordProtocol.c.

Here is the call graph for this function:



Here is the caller graph for this function:



5.18.1.5 void serialize_record (record ∗ *r,* unsigned char ∗∗ *message,* uint16_t ∗ *messageLen*)

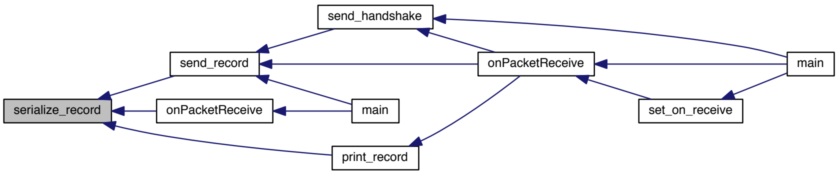
Serialize record in a byte stream message and message length are used for return.

Parameters

message	: pointer to null (the function allocate space for you)
messageLen	: pointer to integer (will contains the message length)

Definition at line 11 of file ServerClientRecordProtocol.c.

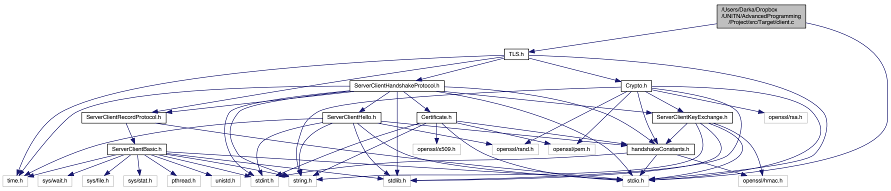
Here is the caller graph for this function:



5.19 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/src/Target/client.c File Reference

```
#include <stdio.h>
#include "TLS.h"
```

Include dependency graph for client.c:



Functions

- void **onPacketReceive** (**channel** ∗ch, **packet_basic** ∗p)
- void **print_master_secret** ()
- void **print_random** ()
- int **main** ()

Variables

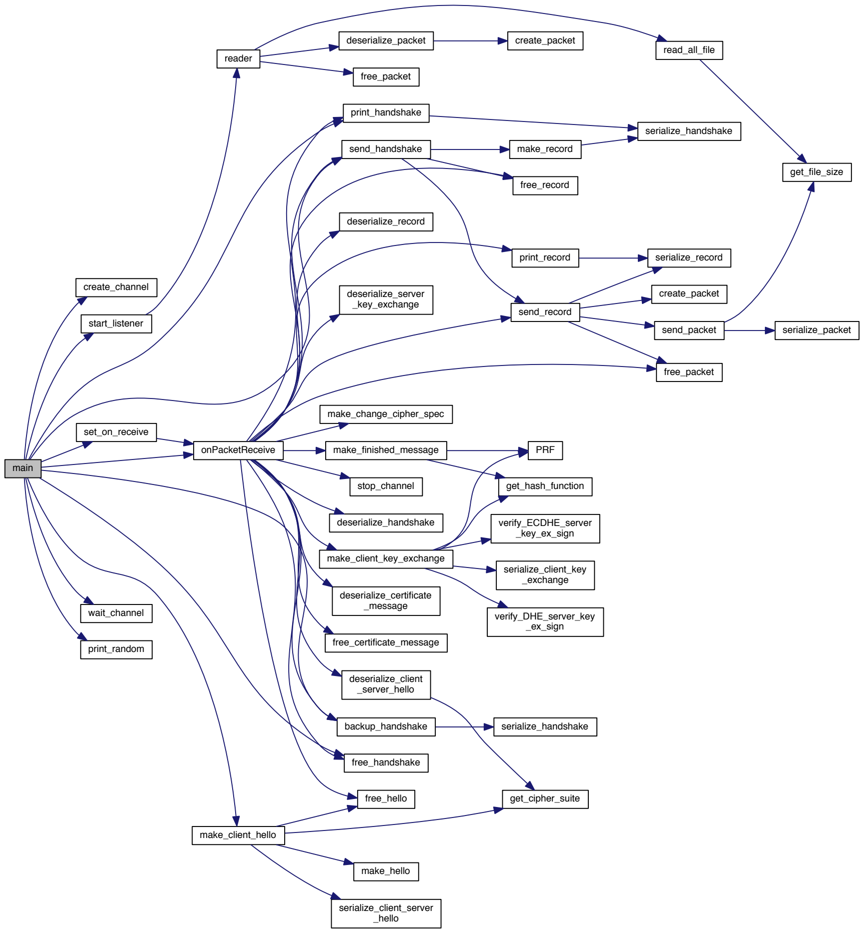
- **TLS_parameters TLS_param**

5.19.1 Function Documentation

5.19.1.1 int main ( )

Definition at line 19 of file client.c.

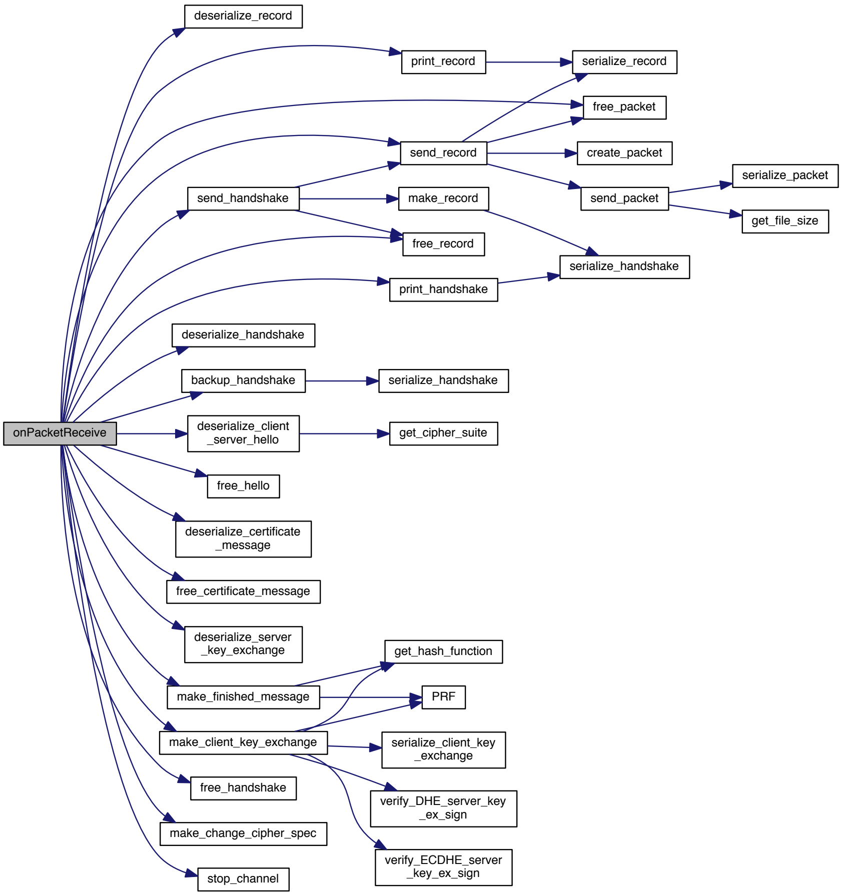
Here is the call graph for this function:



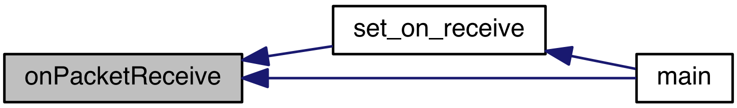
5.19.1.2 void onPacketReceive (channel ∗ *ch*, packet_basic ∗ *p*)

Definition at line 69 of file client.c.

Here is the call graph for this function:



Here is the caller graph for this function:



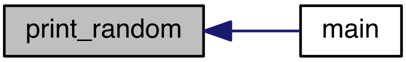
5.19.1.3 void print_master_secret ( )

Definition at line 184 of file client.c.

5.19.1.4 void print_random ( )

Definition at line 191 of file client.c.

Here is the caller graph for this function:



5.19.2 Variable Documentation

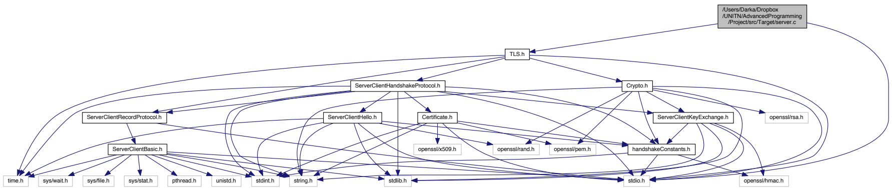
5.19.2.1 TLS_parameters TLS_param

Definition at line 17 of file client.c.

5.20 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/src/Target/server.c File Reference

```
#include <stdio.h>
#include "TLS.h"
```

Include dependency graph for server.c:



Functions

- void **print_random** ()
- void **print_master_secret** ()
- void **compute_set_master_key_RSA** (**client_key_exchange** ∗**client_key_exchange**)
- void **compute_set_master_key_DH** (**client_key_exchange** ∗cliet_public)
- void **compute_set_master_key_ECDHE** (**client_key_exchange** ∗cliet_public)
- void **onPacketReceive** (**channel** ∗server2client, **packet_basic** ∗p)
- int **main** ()

Variables

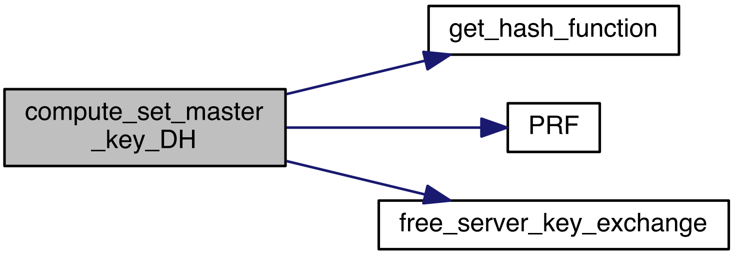
- **TLS_parameters TLS_param**

5.20.1 Function Documentation

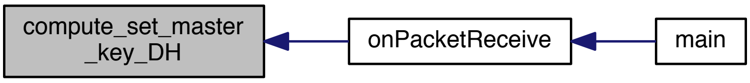
5.20.1.1 void compute_set_master_key_DH (client_key_exchange ∗ *cliet_public*)

Definition at line 250 of file server.c.

Here is the call graph for this function:



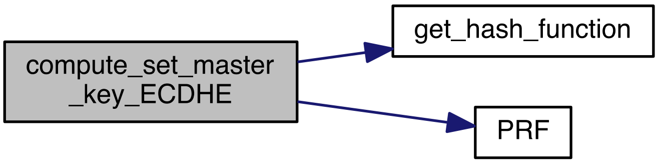
Here is the caller graph for this function:



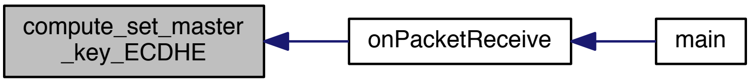
5.20.1.2 void compute_set_master_key_ECDHE (client_key_exchange ∗ *cliet_public*)

Definition at line 279 of file server.c.

Here is the call graph for this function:



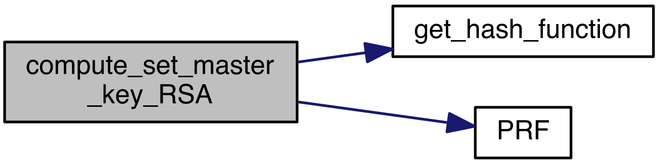
Here is the caller graph for this function:



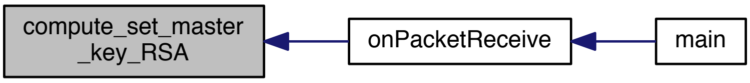
5.20.1.3 void compute_set_master_key_RSA (client_key_exchange ∗ *client_key_exchange*)

Definition at line 213 of file server.c.

Here is the call graph for this function:



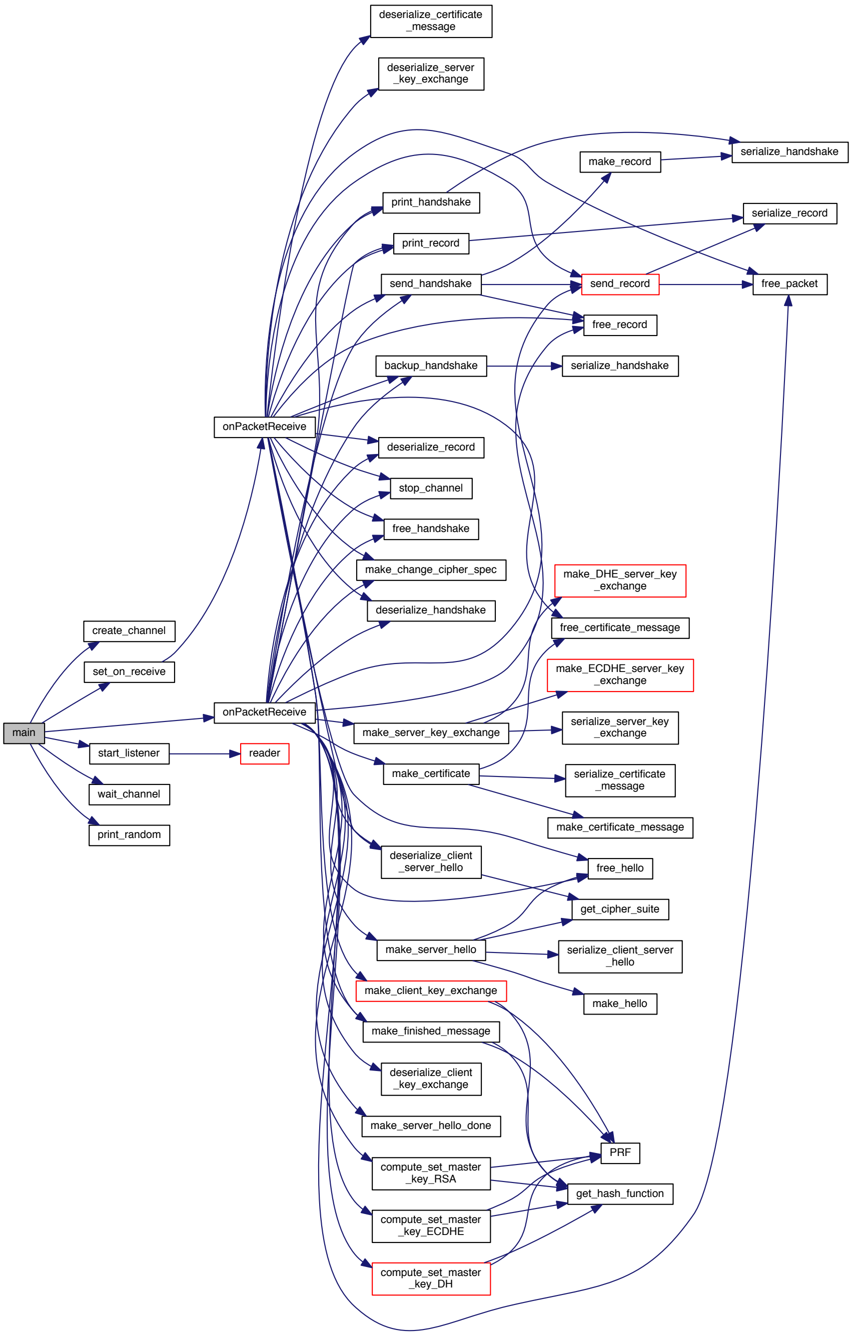
Here is the caller graph for this function:



5.20.1.4 int main ( )

Definition at line 21 of file server.c.

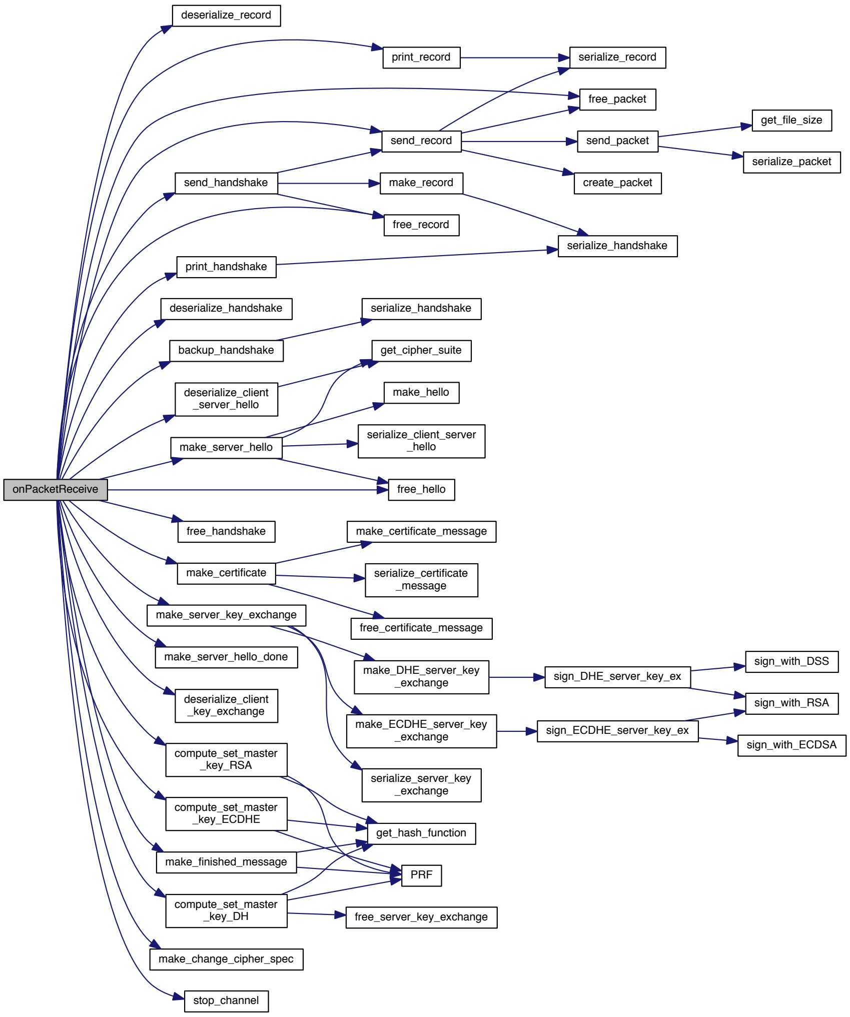
Here is the call graph for this function:



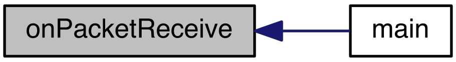
5.20.1.5 void onPacketReceive (channel ∗ *server2client,* packet_basic ∗ *p*)

Definition at line 58 of file server.c.

Here is the call graph for this function:



Here is the caller graph for this function:



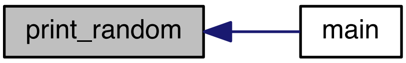
5.20.1.6 void print_master_secret ( )

Definition at line 205 of file server.c.

5.20.1.7 void print_random ( )

Definition at line 194 of file server.c.

Here is the caller graph for this function:



5.20.2 Variable Documentation

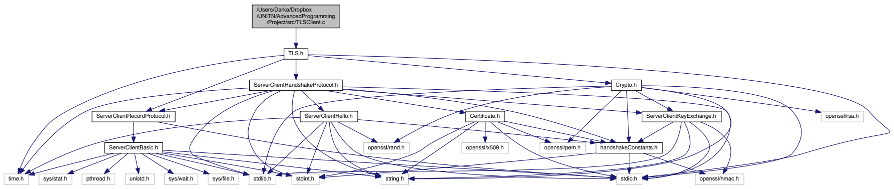
5.20.2.1 TLS_parameters TLS_param

Definition at line 19 of file server.c.

5.21 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/src/TLSClient.c File Reference

```
#include "TLS.h"
```

Include dependency graph for TLSClient.c:



Functions

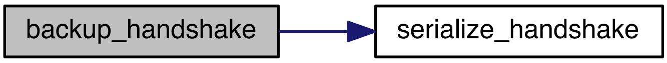
- void **backup_handshake** (**TLS_parameters** ∗**TLS_param**, **handshake** ∗h)
- **handshake** ∗ **make_client_hello** (unsigned char ∗client_random)
- **handshake** ∗ **make_client_key_exchange** (**TLS_parameters** ∗**TLS_param**, uint16_t key_ex_alg)
- **record** ∗ **make_change_cipher_spec** ()
- **handshake** ∗ **make_finished_message** (**TLS_parameters** ∗**TLS_param**)

5.21.1 Function Documentation

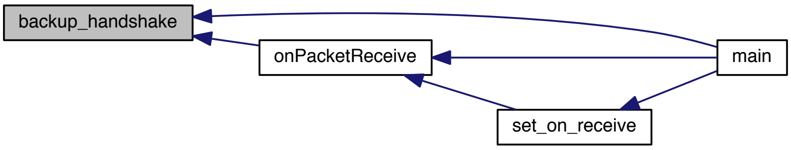
5.21.1.1 void backup_handshake (**TLS_parameters** ∗ *TLS_param,* **handshake** ∗ *h*)

Definition at line 13 of file TLSClient.c.

Here is the call graph for this function:



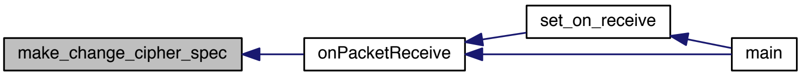
Here is the caller graph for this function:



5.21.1.2 **record**∗ make_change_cipher_spec ( )

Definition at line 250 of file TLSClient.c.

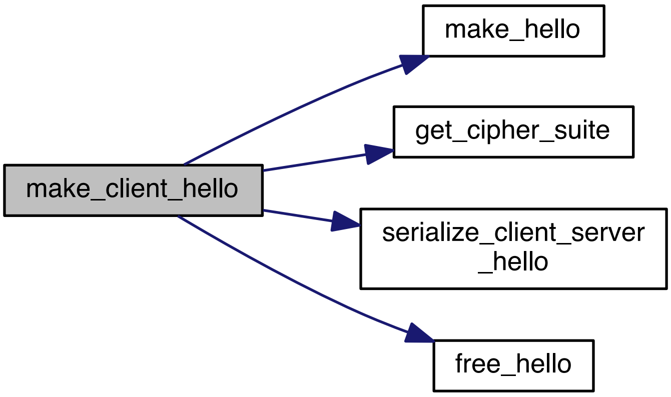
Here is the caller graph for this function:



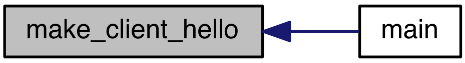
5.21.1.3 handshake∗ make_client_hello (unsigned char ∗ *client_random*)

Definition at line 31 of file TLSClient.c.

Here is the call graph for this function:



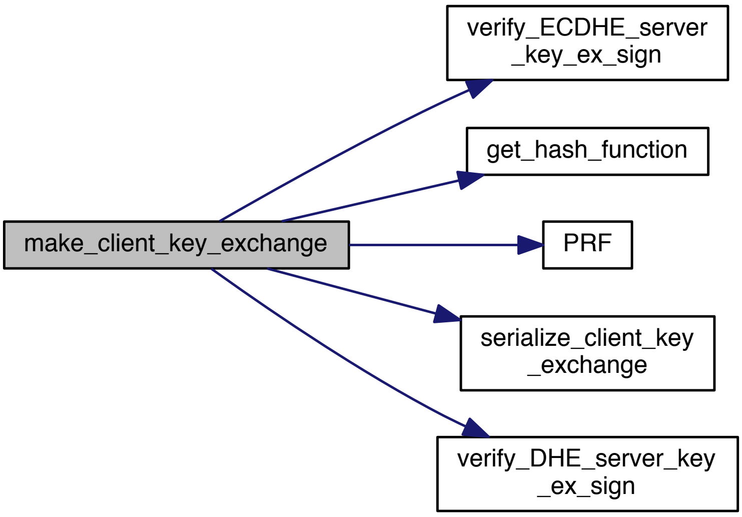
Here is the caller graph for this function:



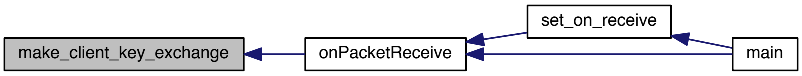
5.21.1.4 handshake∗ make_client_key_exchange (TLS_parameters ∗ *TLS_param,* uint16_t *key_ex_alg*)

Definition at line 65 of file TLSClient.c.

Here is the call graph for this function:


verify_ECDHE_server_key_ex_sign
get_hash_function
make_client_key_exchange
PRF
serialize_client_key_exchange
verify_DHE_server_key_ex_sign

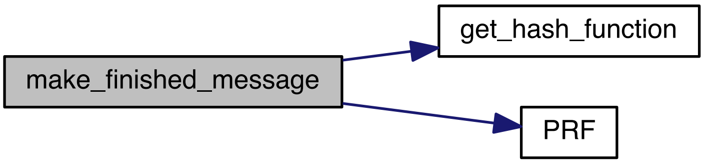
Here is the caller graph for this function:



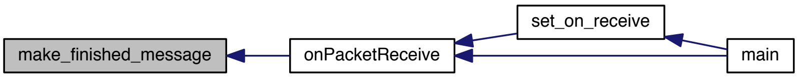
5.21.1.5 handshake∗ make_finished_message (TLS_parameters ∗ *TLS_param*)

Definition at line 262 of file TLSClient.c.

Here is the call graph for this function:



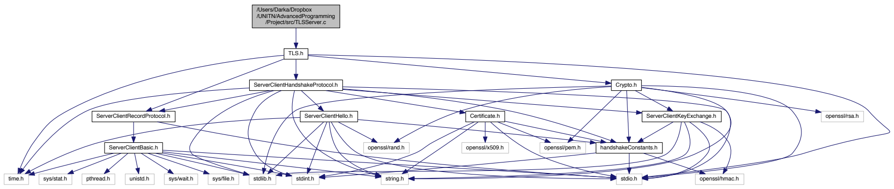
Here is the caller graph for this function:



5.22 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/src/TLSServer.c File Reference

```
#include "TLS.h"
```

Include dependency graph for TLSServer.c:



Functions

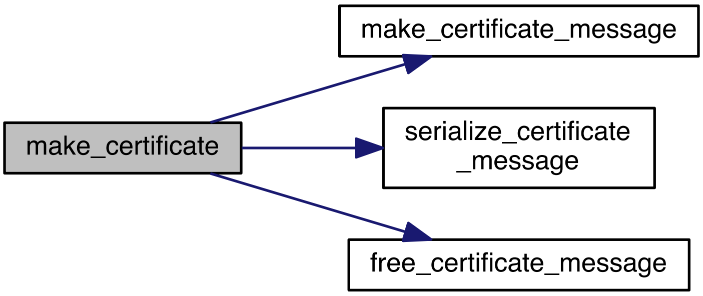
- **handshake** ∗ **make_server_hello** (**TLS_parameters** ∗**TLS_param**, **handshake_hello** ∗client_hello)
- **handshake** ∗ **make_certificate** (**TLS_parameters** ∗**TLS_param**)
- **handshake** ∗ **make_server_key_exchange** (**TLS_parameters** ∗**TLS_param**)
- **DHE_server_key_exchange** ∗ **make_DHE_server_key_exchange** (**TLS_parameters** ∗**TLS_param**)
- **ECDHE_server_key_exchange** ∗ **make_ECDHE_server_key_exchange** (**TLS_parameters** ∗**TLS_↩ param**)
- **handshake** ∗ **make_server_hello_done** ()

5.22.1 Function Documentation

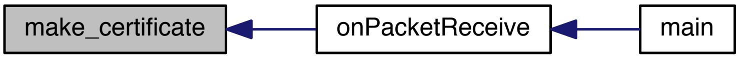
5.22.1.1 handshake∗ make_certificate (TLS_parameters ∗ *TLS_param*)

Definition at line 48 of file TLSServer.c.

Here is the call graph for this function:



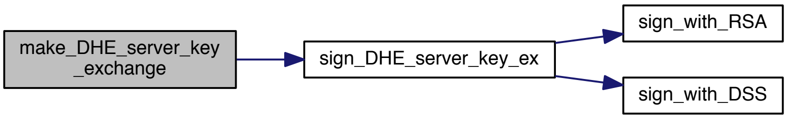
Here is the caller graph for this function:



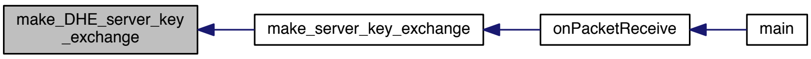
5.22.1.2 DHE_server_key_exchange∗ make_DHE_server_key_exchange (TLS_parameters ∗ *TLS_param*)

Definition at line 113 of file TLSServer.c.

Here is the call graph for this function:



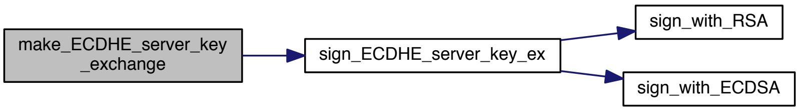
Here is the caller graph for this function:



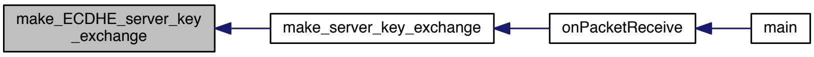
5.22.1.3 ECDHE_server_key_exchange∗ make_ECDHE_server_key_exchange (TLS_parameters ∗ *TLS_param*)

Definition at line 153 of file TLSServer.c.

Here is the call graph for this function:



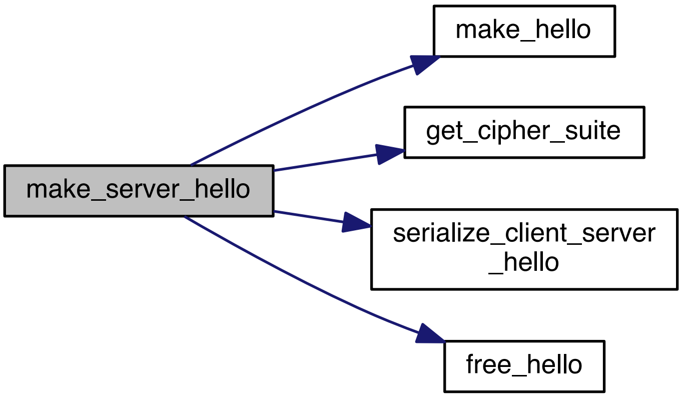
Here is the caller graph for this function:



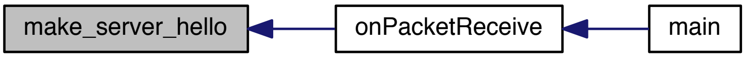
5.22.1.4 handshake∗ make_server_hello (TLS_parameters ∗ *TLS_param,* handshake_hello ∗ *client_hello*)

Definition at line 11 of file TLSServer.c.

Here is the call graph for this function:



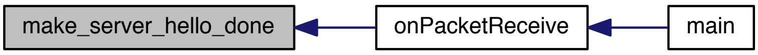
Here is the caller graph for this function:



5.22.1.5 handshake∗ make_server_hello_done ( )

Definition at line 188 of file TLSServer.c.

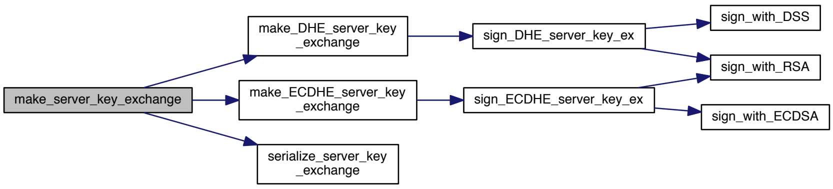
Here is the caller graph for this function:



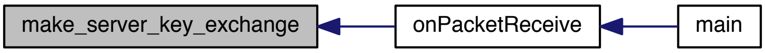
5.22.1.6 handshake∗ make_server_key_exchange (TLS_parameters ∗ *TLS_param*)

Definition at line 84 of file TLSServer.c.

Here is the call graph for this function:



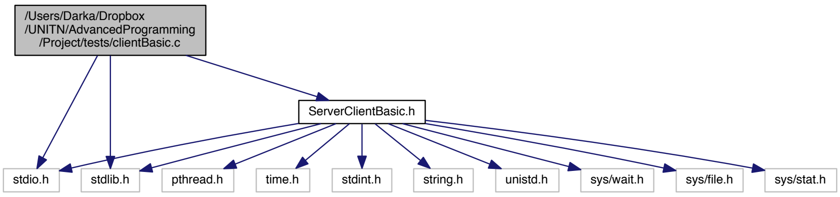
Here is the caller graph for this function:



5.23 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/tests/clientBasic.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "ServerClientBasic.h"
```

Include dependency graph for clientBasic.c:



Functions

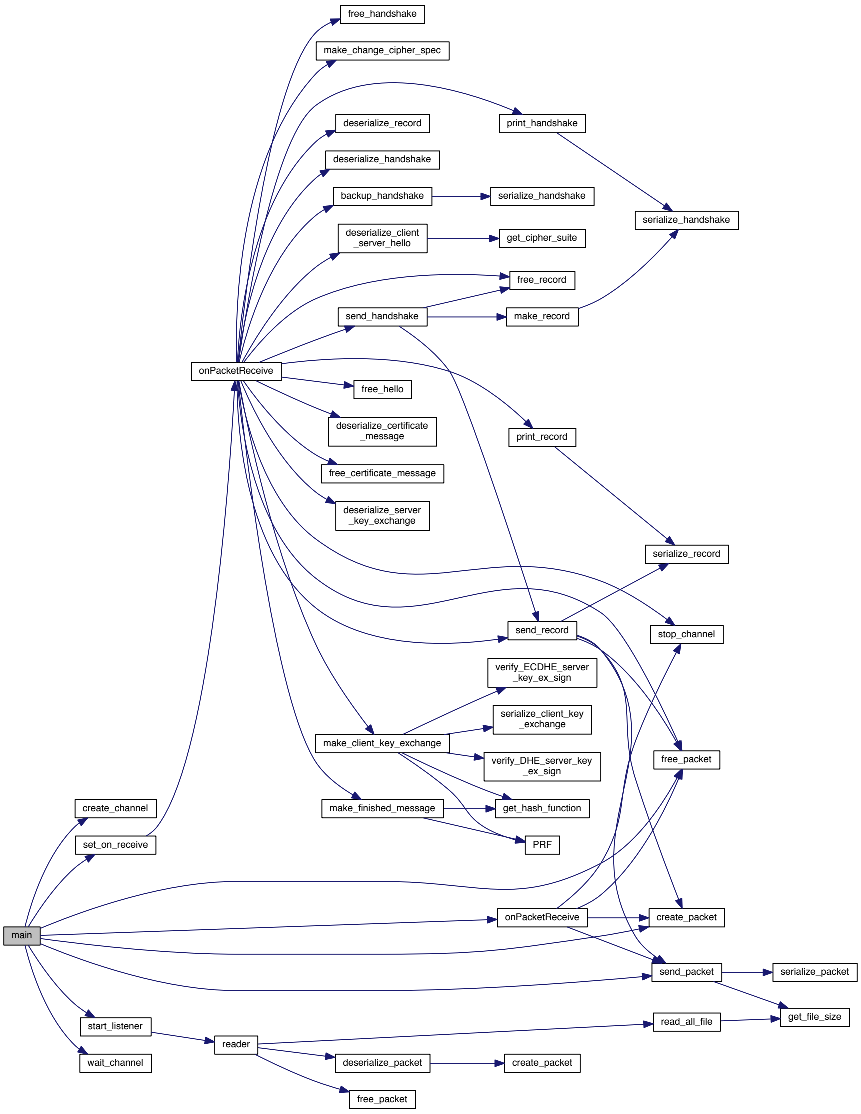
- void **onPacketReceive** (**channel** ∗ch, **packet_basic** ∗p)
- int **main** (int argc, char ∗∗argv)

5.23.1 Function Documentation

5.23.1.1 int main (int *argc,* char ∗∗ *argv*)

Definition at line 16 of file clientBasic.c.

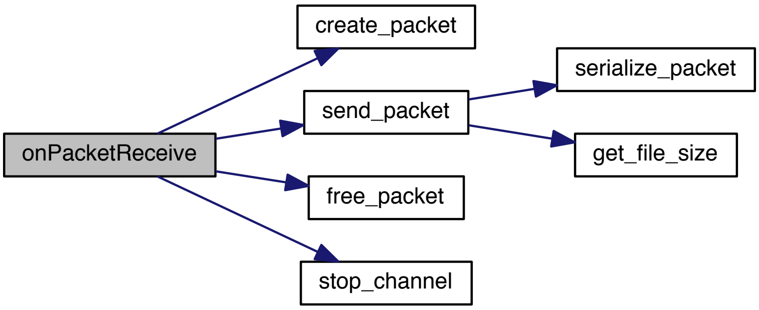
Here is the call graph for this function:



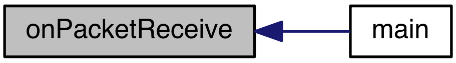
5.23.1.2 void onPacketReceive (channel ∗ *ch,* packet_basic ∗ *p*)

Definition at line 38 of file clientBasic.c.

Here is the call graph for this function:



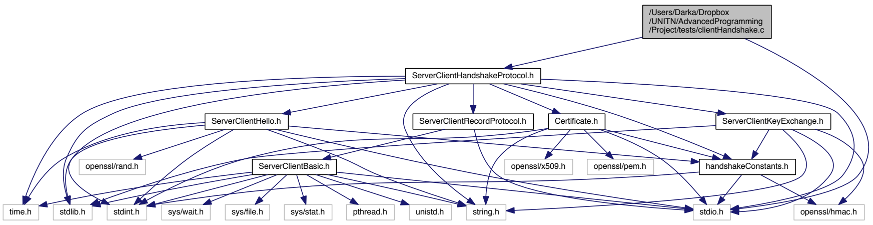
Here is the caller graph for this function:



5.24 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/tests/clientHandshake.c File Reference

```
#include <stdio.h>
#include "ServerClientHandshakeProtocol.h"
```

Include dependency graph for clientHandshake.c:



Functions

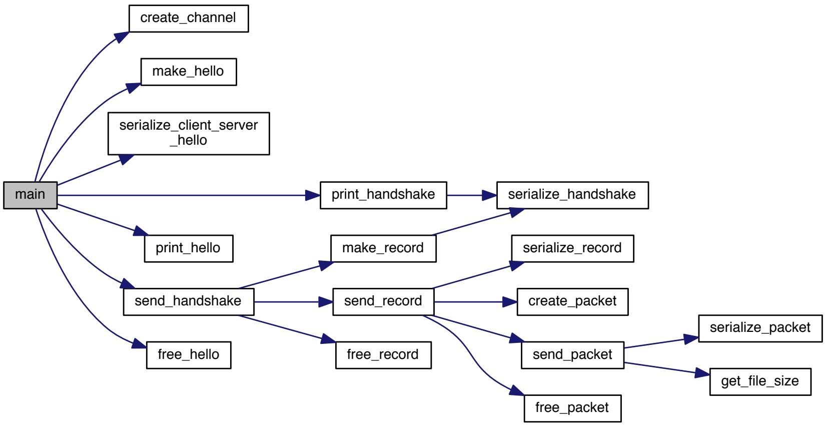
- int **main** ()

5.24.1 Function Documentation

5.24.1.1 int main ( )

Definition at line 12 of file clientHandshake.c.

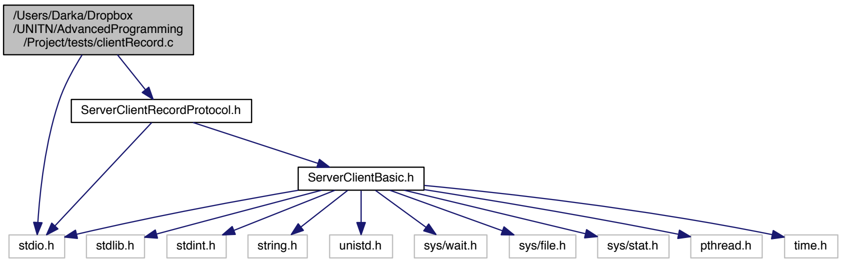
Here is the call graph for this function:



5.25 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/tests/clientRecord.c File Reference

```
#include <stdio.h>
#include "ServerClientRecordProtocol.h"
```

Include dependency graph for clientRecord.c:



Functions

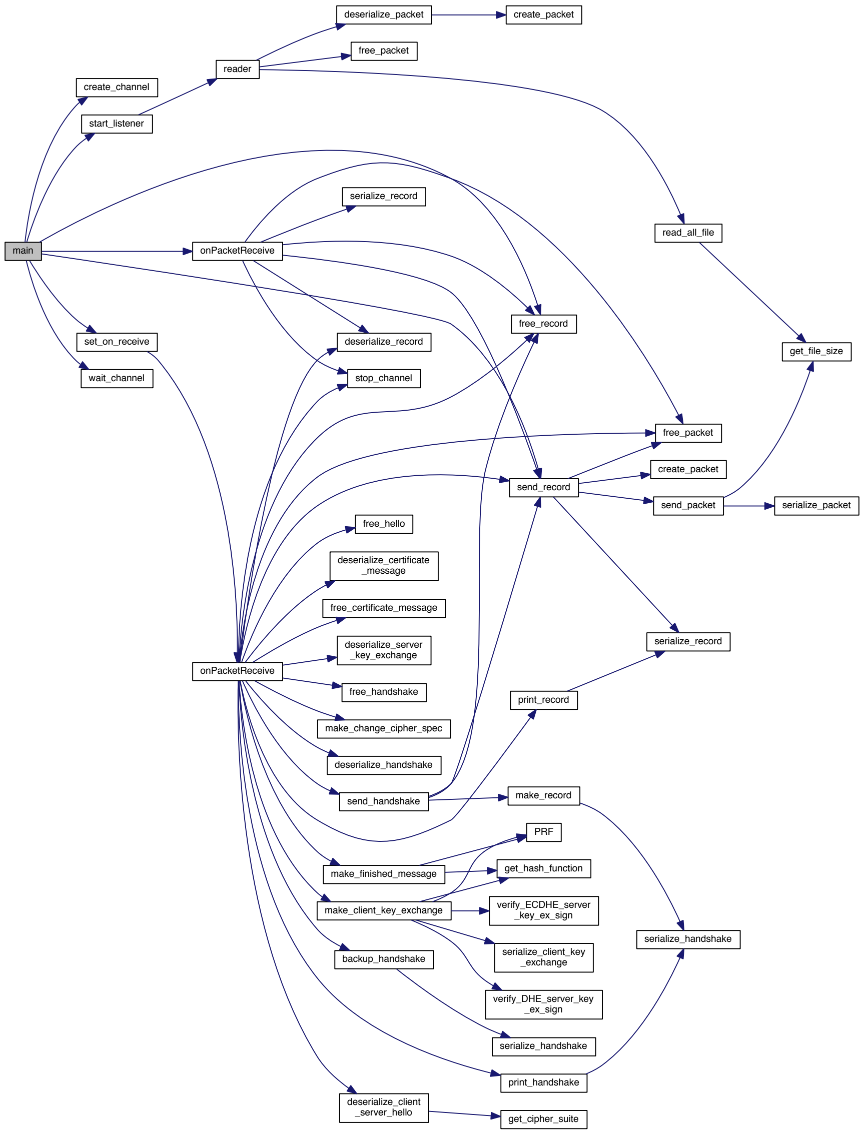
- void **onPacketReceive** (**channel** ∗ch, **packet_basic** ∗p)
- int **main** (int argc, const char ∗argv[ ])

5.25.1 Function Documentation

5.25.1.1 int main (int *argc*, const char ∗ *argv[ ]*)

Definition at line 15 of file clientRecord.c.

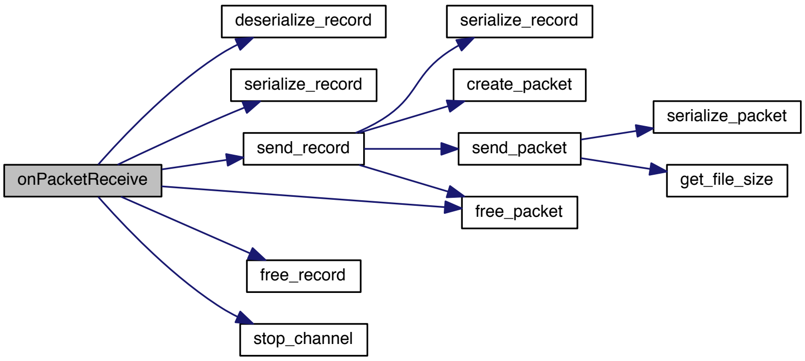
Here is the call graph for this function:



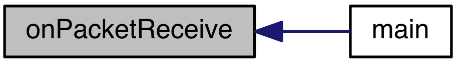
5.25.1.2 void onPacketReceive (channel ∗ *ch*, packet_basic ∗ *p*)

Definition at line 40 of file clientRecord.c.

Here is the call graph for this function:



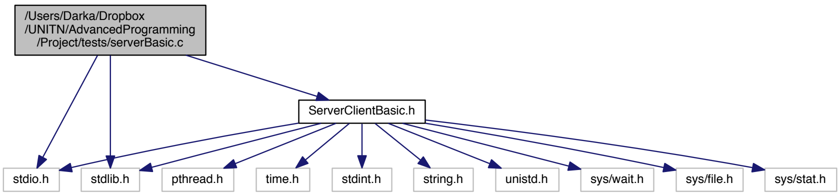
Here is the caller graph for this function:



5.26 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/tests/serverBasic.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "ServerClientBasic.h"
```

Include dependency graph for serverBasic.c:



Functions

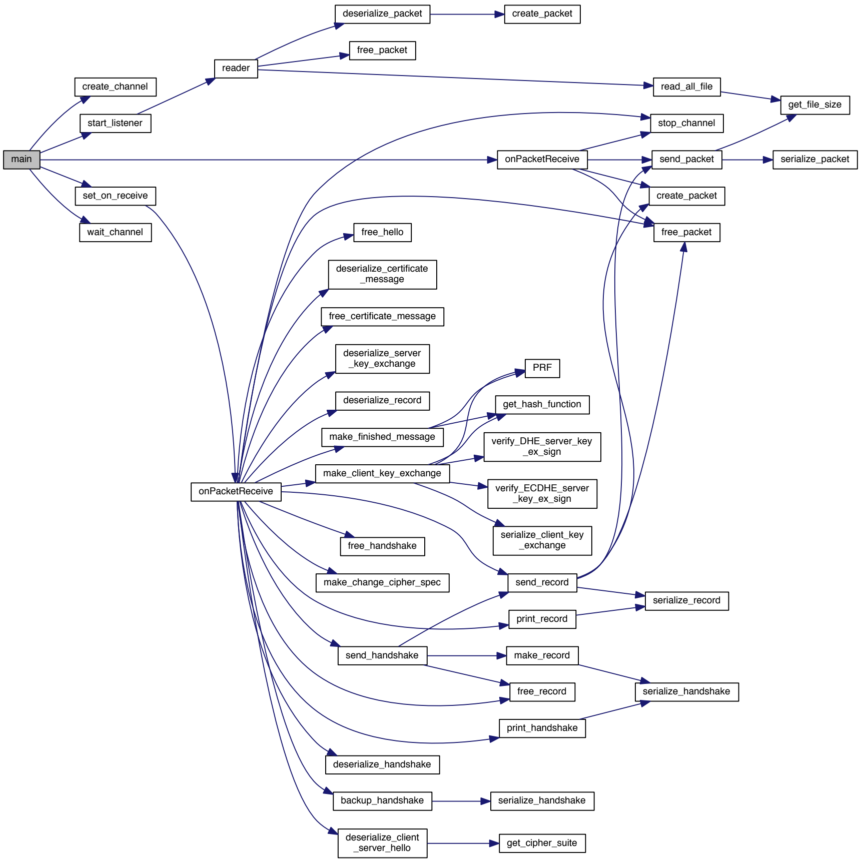
- void **onPacketReceive** (**channel** ∗ch, **packet_basic** ∗p)
- int **main** (int argc, char ∗∗argv)

5.26.1 Function Documentation

5.26.1.1 int main (int *argc,* char ∗∗ *argv*)

Definition at line 16 of file serverBasic.c.

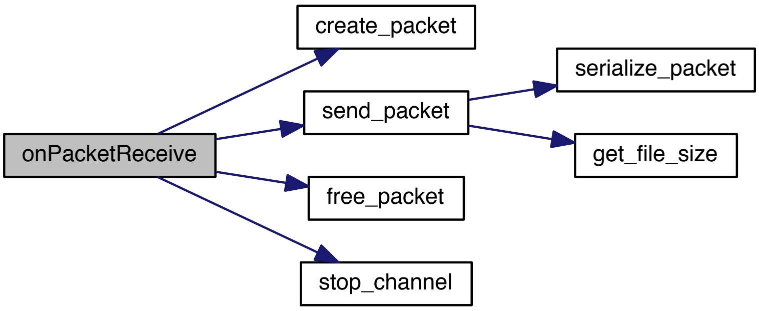
Here is the call graph for this function:



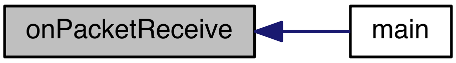
5.26.1.2 void onPacketReceive (**channel** ∗ *ch,* **packet_basic** ∗ *p*)

Definition at line 32 of file serverBasic.c.

Here is the call graph for this function:



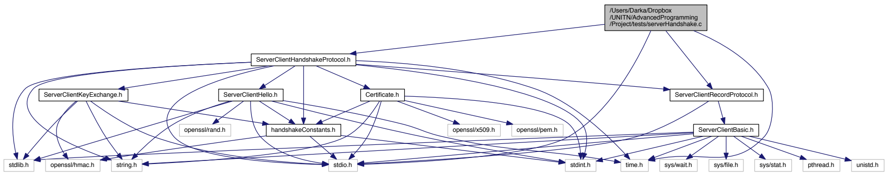
Here is the caller graph for this function:



5.27 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/tests/serverHandshake.c File Reference

```
#include <stdio.h>
#include <time.h>
#include "ServerClientHandshakeProtocol.h"
#include "ServerClientRecordProtocol.h"
```

Include dependency graph for serverHandshake.c:



Functions

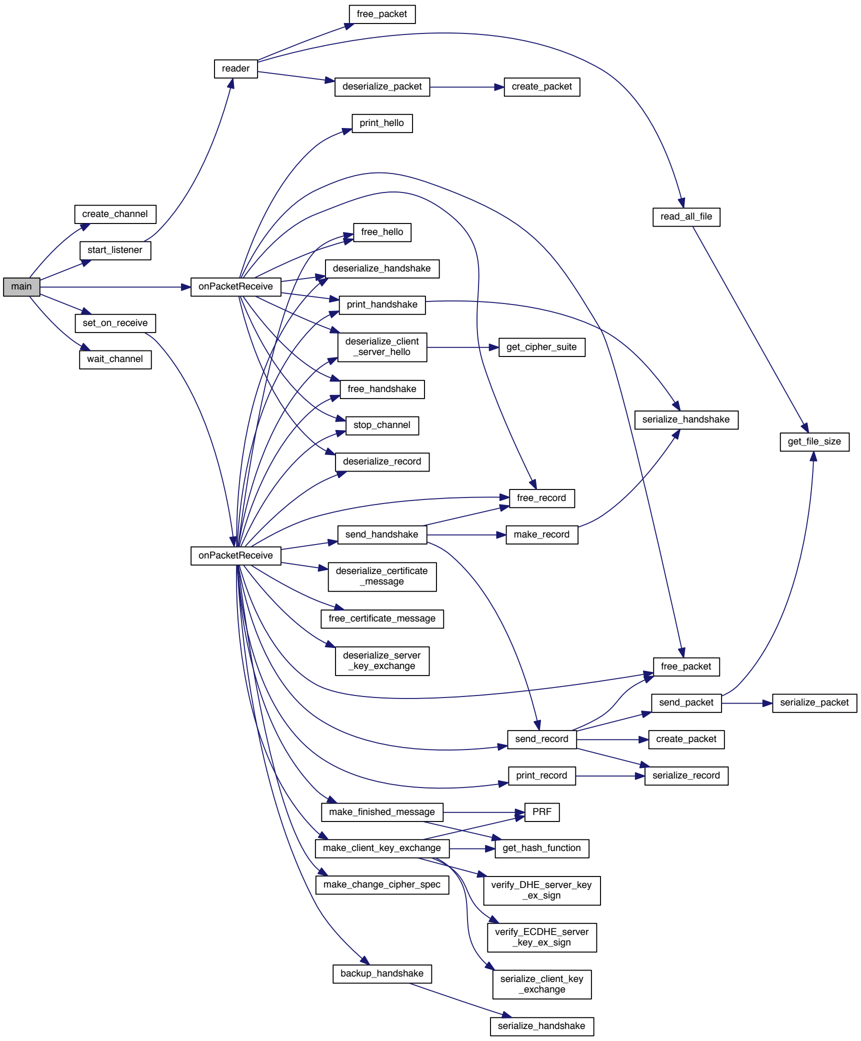
- void **onPacketReceive** (**channel** ∗ch, **packet_basic** ∗p)
- int **main** ()

5.27.1 Function Documentation

5.27.1.1 int main ( )

Definition at line 18 of file serverHandshake.c.

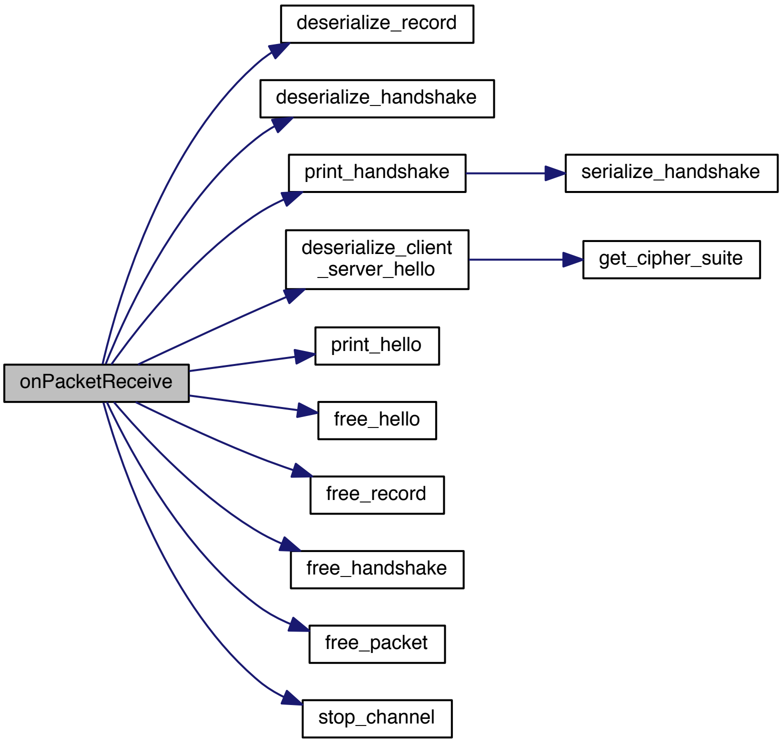
Here is the call graph for this function:



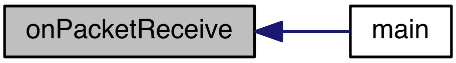
5.27.1.2 void onPacketReceive (channel ∗ *ch,* packet_basic ∗ *p*)

Definition at line 34 of file serverHandshake.c.

Here is the call graph for this function:



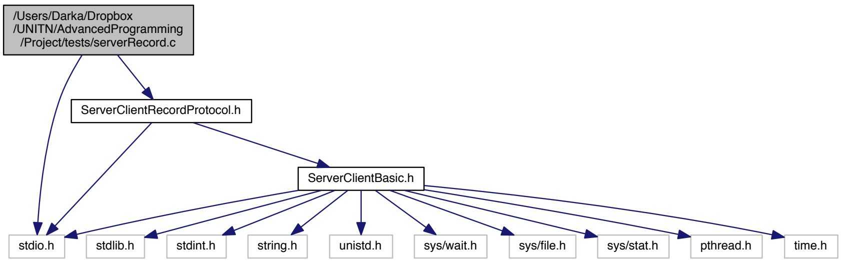
Here is the caller graph for this function:



5.28 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/tests/serverRecord.c File Reference

```
#include <stdio.h>
#include "ServerClientRecordProtocol.h"
```

Include dependency graph for serverRecord.c:



Functions

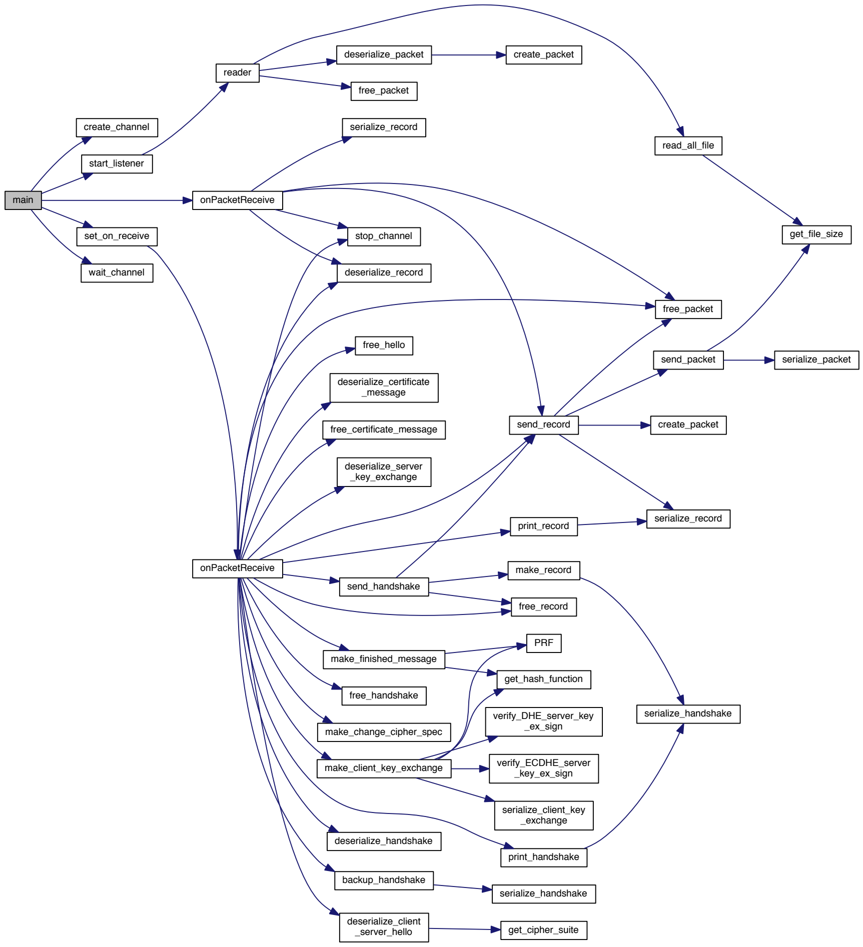
- void **onPacketReceive** (**channel** ∗ch, **packet_basic** ∗p)
- int **main** (int argc, const char ∗argv[ ])

5.28.1 Function Documentation

5.28.1.1 int main (int *argc,* const char ∗ *argv[ ]*)

Definition at line 15 of file serverRecord.c.

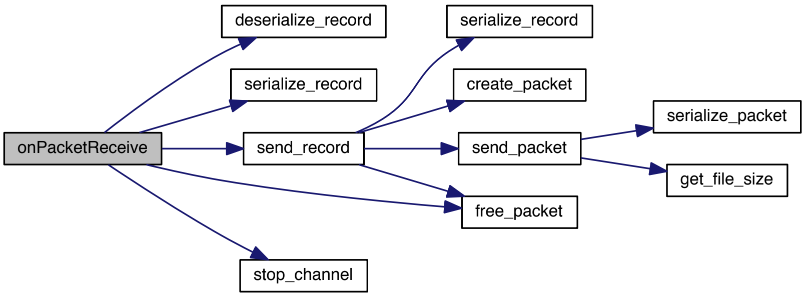
Here is the call graph for this function:



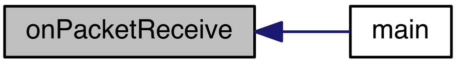
5.28.1.2 void onPacketReceive (channel ∗ *ch,* packet_basic ∗ *p*)

Definition at line 33 of file serverRecord.c.

Here is the call graph for this function:



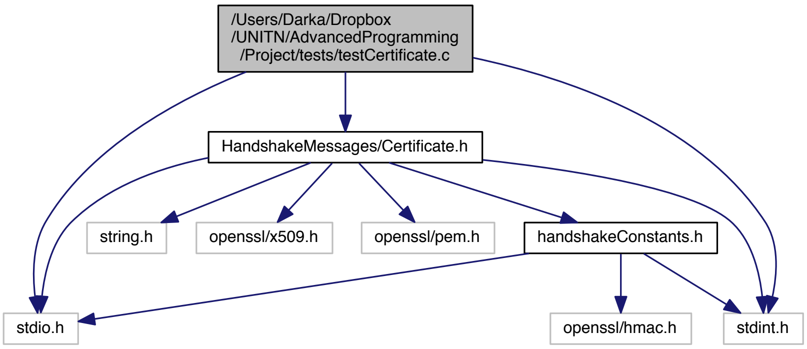
Here is the caller graph for this function:



5.29 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/tests/testCertificate.c File Reference

```
#include <stdio.h>
#include <stdint.h>
#include "HandshakeMessages/Certificate.h"
```

Include dependency graph for testCertificate.c:



Functions

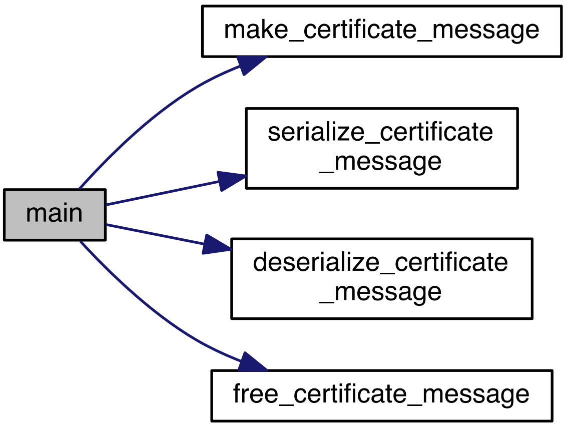
- int **main** ()

5.29.1 Function Documentation

5.29.1.1 int main ( )

Definition at line 5 of file testCertificate.c.

Here is the call graph for this function:


make_certificate_message
serialize_certificate _message
main
deserialize_certificate _message
free_certificate_message

5.30 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩ Intermediates/TLSProject.build/Debug/StructTest.build/Objects-normal/x86_64/main.d File Reference

5.31 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Debug/StructTest.build/Objects-normal/x86_64/main.d File Reference

5.32 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩ Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/↩ Certificate.d File Reference

5.33 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩ Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/↩ Certificate.d File Reference

5.34 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/↩ Certificate.d File Reference

5.35 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/↩ Certificate.d File Reference

5.36 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Release/TLSClient.build/Objects-normal/x86_64/↩ Certificate.d File Reference

5.37 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Release/TLSServer.build/Objects-normal/x86_64/↩ Certificate.d File Reference

5.38 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩ Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/client.d File Reference

5.39 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/client.d File Reference

5.40 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Release/TLSClient.build/Objects-normal/x86_↩ 64/client.d File Reference

5.41 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩ Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/↩ Crypto.d File Reference

5.42 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩ Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/↩ Crypto.d File Reference

5.43 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/↩ Crypto.d File Reference

5.44 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/↩Crypto.d File Reference

5.45 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Release/TLSClient.build/Objects-normal/x86_64/↩Crypto.d File Reference

5.46 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Release/TLSServer.build/Objects-normal/x86_64/↩Crypto.d File Reference

5.47 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/handshake↩Constants.d File Reference

5.48 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/handshake↩Constants.d File Reference

5.49 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/handshake↩Constants.d File Reference

5.50 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/handshake↩Constants.d File Reference

5.51 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Release/TLSClient.build/Objects-normal/x86_↩64/handshakeConstants.d File Reference

5.52 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Release/TLSServer.build/Objects-normal/x86_↩64/handshakeConstants.d File Reference

5.53 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/↩ServerClientBasic.d File Reference

5.54 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/↩ServerClientBasic.d File Reference

5.55 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/↩ServerClientBasic.d File Reference

5.56 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/↩ServerClientBasic.d File Reference

5.57 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Release/TLSClient.build/Objects-normal/x86_64/↩ServerClientBasic.d File Reference

5.58 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Release/TLSServer.build/Objects-normal/x86_64/↩ServerClientBasic.d File Reference

5.59 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/↩ServerClientHandshakeProtocol.d File Reference

5.60 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/↩ServerClientHandshakeProtocol.d File Reference

5.61 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/↩ServerClientHandshakeProtocol.d File Reference

5.62 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/↩ ServerClientHandshakeProtocol.d File Reference

5.63 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Release/TLSClient.build/Objects-normal/x86_64/↩ ServerClientHandshakeProtocol.d File Reference

5.64 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Release/TLSServer.build/Objects-normal/x86_64/↩ ServerClientHandshakeProtocol.d File Reference

5.65 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩ Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/↩ ServerClientHello.d File Reference

5.66 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩ Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/↩ ServerClientHello.d File Reference

5.67 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/↩ ServerClientHello.d File Reference

5.68 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/↩ ServerClientHello.d File Reference

5.69 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Release/TLSClient.build/Objects-normal/x86_64/↩ ServerClientHello.d File Reference

5.70 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Release/TLSServer.build/Objects-normal/x86_64/↩ ServerClientHello.d File Reference

5.71 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/↩ServerClientKeyExchange.d File Reference

5.72 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/↩ServerClientKeyExchange.d File Reference

5.73 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/↩ServerClientKeyExchange.d File Reference

5.74 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/↩ServerClientKeyExchange.d File Reference

5.75 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Release/TLSClient.build/Objects-normal/x86_64/↩ServerClientKeyExchange.d File Reference

5.76 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Release/TLSServer.build/Objects-normal/x86_64/↩ServerClientKeyExchange.d File Reference

5.77 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/↩ServerClientRecordProtocol.d File Reference

5.78 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/↩ServerClientRecordProtocol.d File Reference

5.79 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/↩ServerClientRecordProtocol.d File Reference

5.80 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/↩ ServerClientRecordProtocol.d File Reference

5.81 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Release/TLSClient.build/Objects-normal/x86_64/↩ ServerClientRecordProtocol.d File Reference

5.82 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Release/TLSServer.build/Objects-normal/x86_64/↩ ServerClientRecordProtocol.d File Reference

5.83 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩ Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/TL↩ SClient.d File Reference

5.84 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩ Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/TL↩ SClient.d File Reference

5.85 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/TL↩ SClient.d File Reference

5.86 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/TL↩ SClient.d File Reference

5.87 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Release/TLSClient.build/Objects-normal/x86_64/T↩ LSClient.d File Reference

5.88 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩ Intermediates/TLSProject.build/Release/TLSServer.build/Objects-normal/x86_64/T↩ LSClient.d File Reference

5.89 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/TL↩SServer.d File Reference

5.90 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/TL↩SServer.d File Reference

5.91 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/TL↩SServer.d File Reference

5.92 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/TL↩SServer.d File Reference

5.93 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Release/TLSClient.build/Objects-normal/x86_64/T↩LSServer.d File Reference

5.94 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Release/TLSServer.build/Objects-normal/x86_64/T↩LSServer.d File Reference

5.95 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/Build/↩Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/server.d File Reference

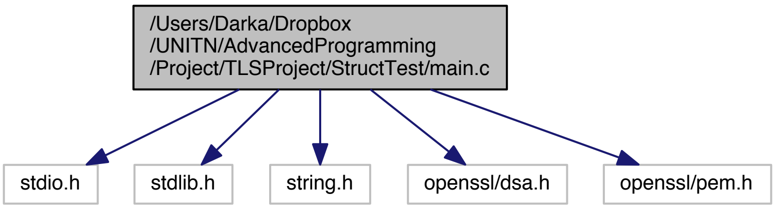
5.96 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Debug/TLSServer.build/Objects-normal/x86_64/server.d File Reference

5.97 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcode↩Intermediates/TLSProject.build/Release/TLSServer.build/Objects-normal/x86_↩64/server.d File Reference

5.98 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/StructTest/main.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <openssl/dsa.h>
#include <openssl/pem.h>
```

Include dependency graph for main.c:



Functions

- int **main** (int argc, const char ∗argv[ ])

5.98.1 Function Documentation

5.98.1.1 int main (int *argc,* const char ∗ *argv[ ]*)

Definition at line 15 of file main.c.

5.99 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcodeIntermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/TLSClient-3E1448B75D6FF18F.d File Reference

5.100 /Users/Darka/Dropbox/UNITN/AdvancedProgramming/Project/TLSProject/xcodeIntermediates/TLSProject.build/Debug/TLSClient.build/Objects-normal/x86_64/TLSClient-BF9D49D9B0317373.d File Reference

Index