Verteilte Systeme Labor

2.4

Generated by Doxygen 1.8.9.1

Tue Apr 26 2016 00:44:47

Contents

1 Module Index **1**

1.1 Modules 1

2 Data Structure Index **3**

2.1 Data Structures 3

3 File Index **5**

3.1 File List 5

4 Module Documentation **7**

4.1 Macros for internal use only 7

4.1.1 Detailed Description 8

4.1.2 Macro Definition Documentation 8

4.1.2.1 FNC_BROADCAST 8

4.1.2.2 FNC_DECRYPT 8

4.1.2.3 FNC_POLYNOME 8

4.1.2.4 FNC_STATUS 8

4.1.2.5 FNC_UNLOCK 8

4.1.2.6 HEADER_LENGTH 8

4.1.2.7 MAX_PACKET_LENGTH 8

4.1.2.8 MODE_CLIENT 8

4.1.2.9 MODE_SERVER 8

4.1.2.10 MODE_STATUS 9

4.1.2.11 MSG_ERROR 9

4.1.2.12 MSG_REQUEST 9

4.1.2.13 MSG_RESPONSE 9

4.1.2.14 NO_BLOCK_ID 9

4.1.2.15 PROTOCOL_VERSION 9

4.1.2.16 VALUE_RESERVED 9

4.2 General API functions 10

4.2.1 Detailed Description 10

4.2.2 Function Documentation 10

4.2.2.1 get_msg_type 10
4.2.2.2 recv_msg 10
4.3 Macros 12
4.3.1 Detailed Description 12
4.3.2 Macro Definition Documentation 13
4.3.2.1 ERR_ALLOC 13
4.3.2.2 ERR_DATA 13
4.3.2.3 ERR_DECRYPT 13
4.3.2.4 ERR_FUNCTIONEXEC 13
4.3.2.5 ERR_FUNCTIONTIMEOUT 13
4.3.2.6 ERR_HEADER_DATA 13
4.3.2.7 ERR_INVALIDMODE 13
4.3.2.8 ERR_INVALIDTYPE 13
4.3.2.9 ERR_INVALIDVERSION 13
4.3.2.10 ERR_NO_PACKET 14
4.3.2.11 ERR_NOSUCHFUNCTION 14
4.3.2.12 ERR_PACKETLENGTH 14
4.3.2.13 ERR_SERVERINUSE 14
4.3.2.14 ERR_UNKNOWN 14
4.3.2.15 ERROR 14
4.3.2.16 NO_ERROR 14
4.3.2.17 SUCCESS 14
4.3.3 Enumeration Type Documentation 14
4.3.3.1 FID 14
4.4 Structures 16
4.4.1 Detailed Description 16
4.4.2 Function Documentation 16
4.4.2.1 __attribute__ 16
4.5 Internal Functions 18
4.5.1 Detailed Description 18
4.5.2 Function Documentation 18
4.5.2.1 check_packet 18
4.5.2.2 send_msg 18
4.6 Client Functions 19
4.6.1 Detailed Description 19
4.6.2 Function Documentation 19
4.6.2.1 extract_brdcst_rsp 19
4.6.2.2 extract_dec_rsp 20
4.6.2.3 extract_error_rsp 20
4.6.2.4 extract_gp_rsp 20

4.6.2.5 extract_unlock_rsp . . . 21
4.6.2.6 init_client . . . 21
4.6.2.7 send_brdcst_req . . . 21
4.6.2.8 send_dec_req . . . 22
4.6.2.9 send_gp_req . . . 22
4.6.2.10 send_unlock_req . . . 23
4.7 Server Functions . . . 24
4.7.1 Detailed Description . . . 24
4.7.2 Function Documentation . . . 24
4.7.2.1 extract_brdcst_req . . . 24
4.7.2.2 extract_dec_req . . . 25
4.7.2.3 extract_gp_req . . . 25
4.7.2.4 extract_status_req . . . 26
4.7.2.5 extract_unlock_req . . . 26
4.7.2.6 init_server . . . 26
4.7.2.7 send_brdcst_rsp . . . 27
4.7.2.8 send_dec_rsp . . . 27
4.7.2.9 send_error_rsp . . . 27
4.7.2.10 send_gp_rsp . . . 28
4.7.2.11 send_status_rsp . . . 28
4.7.2.12 send_unlock_rsp . . . 29

5 Data Structure Documentation 31

5.1 dat_broadcast_response Struct Reference . . . 31
5.1.1 Detailed Description . . . 31
5.1.2 Field Documentation . . . 31
5.1.2.1 serverIP . . . 31
5.2 dat_decrypt_request Struct Reference . . . 31
5.2.1 Detailed Description . . . 32
5.2.2 Field Documentation . . . 32
5.2.2.1 blockID . . . 32
5.2.2.2 clientID . . . 32
5.2.2.3 firstElement . . . 32
5.3 dat_decrypt_response Struct Reference . . . 32
5.3.1 Detailed Description . . . 33
5.3.2 Field Documentation . . . 33
5.3.2.1 blockID . . . 33
5.3.2.2 clientID . . . 33
5.3.2.3 firstElement . . . 33
5.4 dat_polynom_request Struct Reference . . . 33

5.4.1 Detailed Description 33
5.4.2 Field Documentation 34
5.4.2.1 clientID 34
5.4.2.2 generator 34
5.5 dat_status_response Struct Reference 34
5.5.1 Detailed Description 34
5.5.2 Field Documentation 34
5.5.2.1 clientID 34
5.5.2.2 reserved 34
5.5.2.3 wordCount 35
5.6 dat_unlock_request Struct Reference 35
5.6.1 Detailed Description 35
5.6.2 Field Documentation 35
5.6.2.1 clientID 35
5.6.2.2 reserved 35
5.7 error Struct Reference 36
5.7.1 Detailed Description 36
5.7.2 Field Documentation 36
5.7.2.1 blockID 36
5.7.2.2 errCode 36
5.8 msg Struct Reference 37
5.8.1 Detailed Description 37
5.8.2 Field Documentation 37
5.8.2.1 data 37
5.8.2.2 header 38
5.9 msg_header Struct Reference 38
5.9.1 Detailed Description 38
5.9.2 Field Documentation 38
5.9.2.1 func 38
5.9.2.2 length 39
5.9.2.3 mode 39
5.9.2.4 priority 39
5.9.2.5 reserved 39
5.9.2.6 type 39
5.9.2.7 version 40

6 File Documentation 41
6.1 clientAPI.c File Reference 41
6.2 clientAPI.h File Reference 42
6.3 commonAPI.h File Reference 43

6.4 internalMacros.h File Reference 45
6.5 Macros.h File Reference 46
6.6 main.c File Reference 47
6.6.1 Function Documentation 47
6.6.1.1 main 47
6.7 PacketLib.c File Reference 48
6.8 PacketLib.h File Reference 48
6.8.1 Variable Documentation 51
6.8.1.1 blockID 51
6.8.1.2 clientID 51
6.8.1.3 data 51
6.8.1.4 errCode 51
6.8.1.5 firstElement 52
6.8.1.6 func 52
6.8.1.7 generator 52
6.8.1.8 header 52
6.8.1.9 length 52
6.8.1.10 mode 52
6.8.1.11 priority 53
6.8.1.12 reserved 53
6.8.1.13 serverIP 53
6.8.1.14 type 53
6.8.1.15 version 53
6.8.1.16 wordCount 53
6.9 serverAPI.c File Reference 53
6.10 serverAPI.h File Reference 55

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Macros for internal use only 7
General API functions 10
Macros 12
Structures 16
Internal Functions 18
Client Functions 19
Server Functions 24

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

dat_broadcast_response
Broadcast response Each server responds to a broadcast sending its IP address 31
dat_decrypt_request
Decrypt data Request to decrypt data. Polynome has to be set first 31
dat_decrypt_response
Return decrypted data Returns the data from successfull decryption 32
dat_polynom_request
Set polynome request This function tries to set the polynome if the server is free or the current client has lower priority 33
dat_status_response
Response to a status request Servers respond with their current status 34
dat_unlock_request
Unlocks the server After the server is not needed anymore you can unlock it with this function. The server can then be used by another slave 35
error
Error frame An error message frame 36
msg
Structure for a message This structure holds pointers for the message header and the data structure 37
msg_header
A structure for the message header You can easily type cast the first 8 Byte of a message to this struct. The internal structure holds all the values then 38

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

clientAPI.c 41
clientAPI.h 42
commonAPI.h 43
internalMacros.h 45
Macros.h 46
main.c 47
PacketLib.c 48
PacketLib.h 48
serverAPI.c 53
serverAPI.h 55

Chapter 4

Module Documentation

4.1 Macros for internal use only

Macros

- #define HEADER_LENGTH 8

  The header length in bytes.
- #define VALUE_RESERVED 0

  Standard value for reserved fields.
- #define MAX_PACKET_LENGTH 60000

  The maximal packet length (60kB)
- #define NO_BLOCK_ID 0

  No block ID is present.
- #define PROTOCOL_VERSION 14

  The version of the protocol.
- #define MODE_STATUS 1

  The status script is the message source.
- #define MODE_SERVER 2

  The message originated from a server.
- #define MODE_CLIENT 3

  A client sent the message.
- #define FNC_POLYNOME 0

  Sets the polynome in the server.
- #define FNC_DECRYPT 1

  Decrypts a chunk of the file.
- #define FNC_UNLOCK 2

  Unlocks the server to make it available for other clients.
- #define FNC_BROADCAST 5

  Broadcast to discover all available servers.
- #define FNC_STATUS 6

  Status request for that node.
- #define MSG_REQUEST 3

  Request the specified function.
- #define MSG_RESPONSE 4

  Response to an earlier request.
- #define MSG_ERROR 15

  An error occurred decoding or executing.

4.1.1 Detailed Description

4.1.2 Macro Definition Documentation

4.1.2.1 #define FNC_BROADCAST 5

Broadcast to discover all available servers.

Definition at line 33 of file internalMacros.h.

4.1.2.2 #define FNC_DECRYPT 1

Decrypts a chunk of the file.

Definition at line 31 of file internalMacros.h.

4.1.2.3 #define FNC_POLYNOME 0

Sets the polynome in the server.

Definition at line 30 of file internalMacros.h.

4.1.2.4 #define FNC_STATUS 6

Status request for that node.

Definition at line 34 of file internalMacros.h.

4.1.2.5 #define FNC_UNLOCK 2

Unlocks the server to make it available for other clients.

Definition at line 32 of file internalMacros.h.

4.1.2.6 #define HEADER_LENGTH 8

The header length in bytes.

Definition at line 16 of file internalMacros.h.

4.1.2.7 #define MAX_PACKET_LENGTH 60000

The maximal packet length (60kB)

Definition at line 18 of file internalMacros.h.

4.1.2.8 #define MODE_CLIENT 3

A client sent the message.

Definition at line 27 of file internalMacros.h.

4.1.2.9 #define MODE_SERVER 2

The message originated from a server.

Definition at line 26 of file internalMacros.h.

4.1.2.10 #define MODE_STATUS 1

The status script is the message source.

Definition at line 25 of file internalMacros.h.

4.1.2.11 #define MSG_ERROR 15

An error occurred decoding or executing.

Definition at line 39 of file internalMacros.h.

4.1.2.12 #define MSG_REQUEST 3

Request the specified function.

Definition at line 37 of file internalMacros.h.

4.1.2.13 #define MSG_RESPONSE 4

Response to an earlier request.

Definition at line 38 of file internalMacros.h.

4.1.2.14 #define NO_BLOCK_ID 0

No block ID is present.

Definition at line 19 of file internalMacros.h.

4.1.2.15 #define PROTOCOL_VERSION 14

The version of the protocol.

Definition at line 22 of file internalMacros.h.

4.1.2.16 #define VALUE_RESERVED 0

Standard value for reserved fields.

Definition at line 17 of file internalMacros.h.

4.2 General API functions

Functions

- uint8_t recv_msg (msg ∗packet)

  Receive a message This function receives a message if there is one present on the Server. It also does a syntax check on the message to find badly generated packets.
- FID get_msg_type (msg ∗packet)

  get_msg_type This function returns the type of the current message.

4.2.1 Detailed Description

API Functions that apply to both sides, client and server

4.2.2 Function Documentation

4.2.2.1 FID get_msg_type (msg ∗ *packet*)

get_msg_type This function returns the type of the current message.

Author

<Author name="" here>="">

Parameters

`in`	*packet*	: The packet to get the type of.

Returns

The error code that occurred.

See also

FID
Macros

Definition at line 122 of file PacketLib.c.

4.2.2.2 uint8_t recv_msg (msg ∗ *packet*)

Receive a message This function receives a message if there is one present on the Server. It also does a syntax check on the message to find badly generated packets.

Author

<Author name="" here>="">

Parameters

`out`	*packet*	: The received packet

Returns

The error code of the message. ERR_NO_PACKET if there is no message.

See also

msg
Macros

Definition at line 117 of file PacketLib.c.

4.3 Macros

Macros

- #define NO_ERROR 0

  No error detected.
- #define ERR_PACKETLENGTH 1

  The packet length is invalid or does not match the actual length.
- #define ERR_INVALIDVERSION 2

  The version does not match the one defined in PACKET_LENGTH.
- #define ERR_INVALIDMODE 3

  The mode does not exist.
- #define ERR_NOSUCHFUNCTION 4

  The requested function does not exist (on this node)
- #define ERR_INVALIDTYPE 5

  The type is not specified.
- #define ERR_HEADER_DATA 6

  Inconsistent header data. Header is not valid.
- #define ERR_DATA 8

  Error in the data field detected.
- #define ERR_SERVERINUSE 16

  The server is currently used by another client.
- #define ERR_FUNCTIONTIMEOUT 32

  The called function timed out.
- #define ERR_FUNCTIONEXEC 33

  An error executing this function was detected.
- #define ERR_DECRYPT 64

  The data could not be decrypted due to an error.
- #define ERR_ALLOC 128

  Not enough free space to allocate data.
- #define ERR_NO_PACKET 254

  No Packet was on the socket.
- #define ERR_UNKNOWN 255

  An error occurred that does not match any of the other ones (this should never happen)
- #define ERROR -1

  An error occurred during excecution.
- #define SUCCESS 1

  Function ran without problems.

Enumerations

- enum FID {
  POLYNOME_REQ, POLYNOME_RSP, DECRYPT_REQ, DECRYPT_RSP,
  UNLOCK_REQ, UNLOCK_RSP, BROADCAST_REQ, BROADCAST_RSP,
  STATUS_REQ, STATUS_RSP, ERROR_RSP }

  An enumeration of all possible functions This is used as function ID reference.

4.3.1 Detailed Description

Macros and Enumerations used for the API

4.3.2 Macro Definition Documentation

4.3.2.1 #define ERR_ALLOC 128

Not enough free space to allocate data.

Definition at line 28 of file Macros.h.

4.3.2.2 #define ERR_DATA 8

Error in the data field detected.

Definition at line 23 of file Macros.h.

4.3.2.3 #define ERR_DECRYPT 64

The data could not be decrypted due to an error.

Definition at line 27 of file Macros.h.

4.3.2.4 #define ERR_FUNCTIONEXEC 33

An error executing this function was detected.

Definition at line 26 of file Macros.h.

4.3.2.5 #define ERR_FUNCTIONTIMEOUT 32

The called function timed out.

Definition at line 25 of file Macros.h.

4.3.2.6 #define ERR_HEADER_DATA 6

Inconsistent header data. Header is not valid.

Definition at line 22 of file Macros.h.

4.3.2.7 #define ERR_INVALIDMODE 3

The mode does not exist.

Definition at line 19 of file Macros.h.

4.3.2.8 #define ERR_INVALIDTYPE 5

The type is not specified.

Definition at line 21 of file Macros.h.

4.3.2.9 #define ERR_INVALIDVERSION 2

The version does not match the one defined in PACKET_LENGTH.

Definition at line 18 of file Macros.h.

4.3.2.10 #define ERR_NO_PACKET 254

No Packet was on the socket.

Definition at line 29 of file Macros.h.

4.3.2.11 #define ERR_NOSUCHFUNCTION 4

The requested function does not exist (on this node)

Definition at line 20 of file Macros.h.

4.3.2.12 #define ERR_PACKETLENGTH 1

The packet length is invalid or does not match the actual length.

Definition at line 17 of file Macros.h.

4.3.2.13 #define ERR_SERVERINUSE 16

The server is currently used by another client.

Definition at line 24 of file Macros.h.

4.3.2.14 #define ERR_UNKNOWN 255

An error occurred that does not match any of the other ones (this should never happen)

Definition at line 30 of file Macros.h.

4.3.2.15 #define ERROR -1

An error occurred during excecution.

Definition at line 33 of file Macros.h.

4.3.2.16 #define NO_ERROR 0

No error detected.

Definition at line 16 of file Macros.h.

4.3.2.17 #define SUCCESS 1

Function ran without problems.

Definition at line 34 of file Macros.h.

4.3.3 Enumeration Type Documentation

4.3.3.1 enum FID

An enumeration of all possible functions This is used as function ID reference.

Enumerator

POLYNOME_REQ Function : set polynome; Type : Request.

POLYNOME_RSP Function : set polynome; Type : Response.

DECRYPT_REQ Function : decrypt data; Type : Request.

DECRYPT_RSP Function : decrypt data; Type : Response.

UNLOCK_REQ Function : unlock server; Type : Request.

UNLOCK_RSP Function : unlock server; Type : Response.

BROADCAST_REQ Function : broadcast; Type : Request.

BROADCAST_RSP Function : broadcast; Type : Response.

STATUS_REQ Function : status check; Type : Request.

STATUS_RSP Function : status check; Type : Response.

ERROR_RSP Function : any; Type : Error.

Definition at line 39 of file Macros.h.

4.4 Structures

Data Structures

- struct msg_header

  A structure for the message header You can easily type cast the first 8 Byte of a message to this struct. The internal structure holds all the values then.
- struct dat_polynom_request

  Set polynome request This function tries to set the polynome if the server is free or the current client has lower priority.
- struct dat_decrypt_request

  Decrypt data Request to decrypt data. Polynome has to be set first.
- struct dat_decrypt_response

  Return decrypted data Returns the data from successfull decryption.
- struct dat_unlock_request

  Unlocks the server After the server is not needed anymore you can unlock it with this function. The server can then be used by another slave.
- struct dat_broadcast_response

  Broadcast response Each server responds to a broadcast sending its IP address.
- struct dat_status_response

  Response to a status request Servers respond with their current status.
- struct error

  Error frame An error message frame.
- struct msg

  Structure for a message This structure holds pointers for the message header and the data structure.

Functions

- struct msg_header __attribute__ ((__packed__)) msg_header

  A structure for the message header You can easily type cast the first 8 Byte of a message to this struct. The internal structure holds all the values then.

4.4.1 Detailed Description

Data Structures for internal use

4.4.2 Function Documentation

4.4.2.1 struct msg __attribute__ ((__packed__))

A structure for the message header You can easily type cast the first 8 Byte of a message to this struct. The internal structure holds all the values then.

Structure for a message This structure holds pointers for the message header and the data structure.

Error frame An error message frame.

Response to a status request Servers respond with their current status.

Broadcast response Each server responds to a broadcast sending its IP address.

Unlocks the server After the server is not needed anymore you can unlock it with this function. The server can then be used by another slave.

Return decrypted data Returns the data from successfull decryption.

Decrypt data Request to decrypt data. Polynome has to be set first.

Set polynome request This function tries to set the polynome if the server is free or the current client has lower priority.

See also

dat_polynom_request
Macros

4.5 Internal Functions

Functions

- uint8_t check_packet (msg ∗packet)
  Check a packet for internal errors.
- uint8_t send_msg (msg ∗packet)
  Sends a message via UDP.

4.5.1 Detailed Description

Functions for internal use only

4.5.2 Function Documentation

4.5.2.1 uint8_t check_packet (msg ∗ *packet*)

Check a packet for internal errors.

Author

Michel Schmidt

Parameters

`in`	*packet*	: The packet structure

Returns

The error code that occurred

See also

Macros

Definition at line 77 of file PacketLib.c.

4.5.2.2 uint8_t send_msg (msg ∗ *packet*)

Sends a message via UDP.

Author

<Author name="" here>="">

Parameters

`in`	*packet*	The packet to send

Returns

The error code that occurred

See also

msg
Macros

Definition at line 127 of file PacketLib.c.

4.6 Client Functions

Functions

- int init_client ()

  Initiates the lib with the permanent client data.
- uint8_t send_gp_req (uint16_t gp, uint32_t target_server_ip)

  Send a generator polynome This function sets a generator polynome to lock a server.
- uint8_t send_dec_req (uint16_t BID, uint16_t ∗data, uint32_t data_len, uint32_t target_server_ip)

  Send a decryption request Requests the decryption of a block.
- uint8_t send_unlock_req (uint32_t target_server_ip)

  Send an unlock request Unlock a connected server.
- uint8_t send_brdcst_req ()

  Send a broadcast request.
- uint8_t extract_gp_rsp (msg ∗packet, uint32_t ∗src_server_ip)

  Extract a generator polynome response Extract the data from the polynome extract response.
- uint8_t extract_dec_rsp (msg ∗packet, uint16_t ∗BID, uint8_t ∗data, uint32_t ∗data_len, uint32_t ∗src_↩ server_ip)

  Extract the decrypted data response This function extracts the decrypted data from the message.
- uint8_t extract_unlock_rsp (msg ∗packet, uint32_t ∗src_client_ip)

  Extracts the unlock confirmation This extracts the unlock confirmation.
- uint8_t extract_brdcst_rsp (msg ∗packet, uint32_t ∗src_server_ip)

  This extracts broadcast response.
- uint8_t extract_error_rsp (msg ∗packet, uint8_t ∗error_code, uint16_t ∗BID, uint32_t ∗src_server_ip)

  Extract an error message Extract an error message from a server.

4.6.1 Detailed Description

API Functions that only apply to the client

4.6.2 Function Documentation

4.6.2.1 uint8_t extract_brdcst_rsp (msg ∗ *packet*, uint32_t ∗ *src_server_ip*)

This extracts broadcast response.

Author

<Author name="" here>="">

Parameters

in	*packet*	: the packet to extract
out	*src_server_ip*	: the IP of the sourch server

Returns

The error code that occurred.

See also

Macros

Definition at line 53 of file clientAPI.c.

4.6.2.2 uint8_t extract_dec_rsp (msg ∗ *packet,* uint16_t ∗ *BID,* uint8_t ∗ *data,* uint32_t ∗ *data_len,* uint32_t ∗ *src_server_ip*)

Extract the decrypted data response This function extracts the decrypted data from the message.

Author

<Author name="" here>="">

Parameters

in	*packet*	: the packet to extract
out	*BID*	: the block ID of the decrypted packet
out	*data*	: the decrypted data
out	*data_len*	: the length of the decrypted data
out	*src_server_ip*	: the IP of the sourch server

Returns

The error code that occurred.

See also

Macros

Definition at line 43 of file clientAPI.c.

4.6.2.3 uint8_t extract_error_rsp (msg ∗ *packet,* uint8_t ∗ *error_code,* uint16_t ∗ *BID,* uint32_t ∗ *src_server_ip*)

Extract an error message Extract an error message from a server.

Author

<Author name="" here>="">

Parameters

in	*packet*	: the packet to extract
out	*error_code*	: The error code that occurred
out	*BID*	: the block ID of the decrypted packet (if present)
out	*src_server_ip*	: the IP of the sourch server

Returns

The error code that occurred.

See also

Macros

Definition at line 58 of file clientAPI.c.

4.6.2.4 uint8_t extract_gp_rsp (msg ∗ *packet,* uint32_t ∗ *src_server_ip*)

Extract a generator polynome response Extract the data from the polynome extract response.

Author

<Author name="" here>="">

Parameters

`in`	*packet*	: the packet to extract
`out`	*src_server_ip*	: the IP of the sourch server

Returns

The error code that occurred.

See also

Macros

Definition at line 38 of file clientAPI.c.

4.6.2.5 uint8_t extract_unlock_rsp (msg ∗ *packet,* uint32_t ∗ *src_client_ip*)

Extracts the unlock confirmation This extracts the unlock confirmation.

Author

<Author name="" here>="">

Parameters

`in`	*packet*	: the packet to extract
`out`	*src_server_ip*	: the IP of the sourch server

Returns

The error code that occurred.

See also

Macros

Definition at line 48 of file clientAPI.c.

4.6.2.6 int init_client ( )

Initiates the lib with the permanent client data.

Author

<Author name="" here>="">

Returns

Error code planed as return value

Definition at line 13 of file clientAPI.c.

4.6.2.7 uint8_t send_brdcst_req ( )

Send a broadcast request.

Author

<Author name="" here>="">

Returns

The error code that occurred.

See also

Macros

Definition at line 33 of file clientAPI.c.

4.6.2.8 uint8_t send_dec_req (uint16_t *BID*, uint16_t ∗ *data*, uint32_t *data_len*, uint32_t *target_server_ip*)

Send a decryption request Requests the decryption of a block.

Author

<Author name="" here>="">

Parameters

in	*BID*	: the id of the block to decrypt
in	*data*	: the data to decrypt
in	*data_len*	: the amount of words in data
in	*target_server_ip*	: the IP address of the target server

Returns

The error code that occurred.

See also

Macros

Definition at line 23 of file clientAPI.c.

4.6.2.9 uint8_t send_gp_req (uint16_t *gp*, uint32_t *target_server_ip*)

Send a generator polynome This function sets a generator polynome to lock a server.

Author

<Author name="" here>="">

Parameters

in	*gp*	: the generator polynome
in	*target_server_ip*	: the IP address of the target server

Returns

The error code that occurred.

See also

Macros

Definition at line 18 of file clientAPI.c.

4.6.2.10 uint8_t send_unlock_req (uint32_t *target_server_ip*)

Send an unlock request Unlock a connected server.

Author

<Author name="" here>="">

Parameters

in	*target_server_ip*	: the IP address of the target server

Returns

The error code that occurred.

See also

Macros

Definition at line 28 of file clientAPI.c.

4.7 Server Functions

Functions

- int init_server ()

  Initiates the lib with the permanent server data.
- uint8_t send_gp_rsp (uint32_t target_client_ip)

  Send a generator polynome response Confirm the successfull setting of the generator polynome.
- uint8_t send_dec_rsp (uint16_t BID, uint8_t ∗data, uint32_t data_len, uint32_t target_client_ip)

  Send the decrypted data Return the decrypted data to the client.
- uint8_t send_unlock_rsp (uint32_t target_client_ip)

  Send the unlock confirmation.
- uint8_t send_brdcst_rsp (uint32_t target_client_ip)

  Send a broadcast response.
- uint8_t send_status_rsp (uint16_t CID, uint32_t sequence_number)

  Send a status response Send the current status to the status script.
- uint8_t send_error_rsp (uint8_t err_code, uint32_t BID, uint32_t target_client_ip, FID fid)

  Send an error message.
- uint8_t extract_gp_req (msg ∗packet, uint16_t ∗gp, uint16_t ∗CID, uint8_t ∗prio, uint32_t ∗src_client_ip)

  Extract the generator polynome Extract the generator polynome from the packet.
- uint8_t extract_dec_req (msg ∗packet, uint16_t ∗CID, uint16_t ∗BID, uint16_t ∗data, uint32_t ∗data_len, uint32_t ∗src_client_ip)

  Extract data to decrypt.
- uint8_t extract_unlock_req (msg ∗packet, uint16_t ∗CID, uint32_t ∗src_client_ip)

  Extract the unlock command extract the command to unlock the server.
- uint8_t extract_brdcst_req (msg ∗packet, uint32_t ∗src_client_ip)

  Extract a broadcast request.
- uint8_t extract_status_req (msg ∗packet)

  Extract a status request.

4.7.1 Detailed Description

API Functions that only apply to the server

4.7.2 Function Documentation

4.7.2.1 uint8_t extract_brdcst_req (msg ∗ *packet*, uint32_t ∗ *src_client_ip*)

Extract a broadcast request.

Author

<Author name="" here>="">

Parameters

in	*packet*	: the packet to extract
out	*src_client_ip*	: the IP of the source client

Returns

The error code that occurred.

See also

Macros

Definition at line 63 of file serverAPI.c.

4.7.2.2 uint8_t extract_dec_req (msg ∗ *packet*, uint16_t ∗ *CID*, uint16_t ∗ *BID*, uint16_t ∗ *data*, uint32_t ∗ *data_len*, uint32_t ∗ *src_client_ip*)

Extract data to decrypt.

Author

<Author name="" here>="">

Parameters

in	*packet*	: the packet to extract
out	*CID*	: the client ID
out	*BID*	: the Block ID of this Block
out	*data*	: the data to decrypt
out	*data_len*	: the amount of data words to decrypt
out	*src_client_ip*	: the IP of the source client

Returns

The error code that occurred.

See also

Macros

Definition at line 53 of file serverAPI.c.

4.7.2.3 uint8_t extract_gp_req (msg ∗ *packet*, uint16_t ∗ *gp*, uint16_t ∗ *CID*, uint8_t ∗ *prio*, uint32_t ∗ *src_client_ip*)

Extract the generator polynome Extract the generator polynome from the packet.

Author

<Author name="" here>="">

Parameters

in	*packet*	: the packet to extract
out	*gp*	: the generator polynome
out	*CID*	: the client ID
out	*prio*	: the priority of the client
out	*src_client_ip*	: the IP of the source client

Returns

The error code that occurred.

See also

Macros

Definition at line 48 of file serverAPI.c.

4.7.2.4 uint8_t extract_status_req (msg ∗ *packet*)

Extract a status request.

Author

<Author name="" here>="">

Parameters

`in`	*packet*	: the packet to extract

Returns

The error code that occurred.

See also

Macros

Definition at line 68 of file serverAPI.c.

4.7.2.5 uint8_t extract_unlock_req (msg ∗ *packet,* uint16_t ∗ *CID,* uint32_t ∗ *src_client_ip*)

Extract the unlock command extract the command to unlock the server.

Author

<Author name="" here>="">

Parameters

`in`	*packet*	: the packet to extract
`out`	*CID*	: the client ID
`out`	*src_client_ip*	: the IP of the source client

Returns

The error code that occurred.

See also

Macros

Definition at line 58 of file serverAPI.c.

4.7.2.6 int init_server ( )

Initiates the lib with the permanent server data.

Returns

Error code planed as return value

Definition at line 13 of file serverAPI.c.

4.7.2.7 uint8_t send_brdcst_rsp (uint32_t *target_client_ip*)

Send a broadcast response.

Author

<Author name="" here>="">

Parameters

in	*target_client_ip*	: the IP address of the target client

Returns

The error code that occurred.

See also

Macros

Definition at line 33 of file serverAPI.c.

4.7.2.8 uint8_t send_dec_rsp (uint16_t *BID,* uint8_t ∗ *data,* uint32_t *data_len,* uint32_t *target_client_ip*)

Send the decrypted data Return the decrypted data to the client.

Author

<Author name="" here>="">

Parameters

in	*BID*	: The Block ID of this Block
in	*data*	: The data to send
in	*data_len*	: The length of the data field
in	*target_client_ip*	: the IP address of the target client

Returns

The error code that occurred.

See also

Macros

Definition at line 23 of file serverAPI.c.

4.7.2.9 uint8_t send_error_rsp (uint8_t *err_code,* uint32_t *BID,* uint32_t *target_client_ip,* FID *fid*)

Send an error message.

Author

<Author name="" here>="">

Parameters

in	*err_code*	: the error that occurred
in	*BID*	: The Block ID of this Block
in	*target_client_ip*	: the IP address of the target client
in	*fid*	: the function ID that was called

Returns

The error code that occurred during excecution.

See also

Macros
FID

Definition at line 43 of file serverAPI.c.

4.7.2.10 uint8_t send_gp_rsp (uint32_t *target_client_ip*)

Send a generator polynome response Confirm the successfull setting of the generator polynome.

Author

<Author name="" here>="">

Parameters

in	*target_client_ip*	: the IP address of the target client

Returns

The error code that occurred.

See also

Macros

Definition at line 18 of file serverAPI.c.

4.7.2.11 uint8_t send_status_rsp (uint16_t *CID*, uint32_t *sequence_number*)

Send a status response Send the current status to the status script.

Author

<Author name="" here>="">

Parameters

in	*CID*	: The client ID
in	*sequence_↩ number*	: The sequence number for the current client

Returns

The error code that occurred.

See also

Macros

Definition at line 38 of file serverAPI.c.

4.7.2.12 uint8_t send_unlock_rsp (uint32_t *target_client_ip*)

Send the unlock confirmation.

Author

<Author name="" here>="">

Parameters

`in`	*target_client_ip*	: the IP address of the target client

Returns

The error code that occurred.

See also

Macros

Definition at line 28 of file serverAPI.c.

Chapter 5

Data Structure Documentation

5.1 dat_broadcast_response Struct Reference

Broadcast response Each server responds to a broadcast sending its IP address.

```
#include <PacketLib.h>
```

Data Fields

- uint32_t serverIP

  4 times 1 Byte IP V4 address

5.1.1 Detailed Description

Broadcast response Each server responds to a broadcast sending its IP address.

Definition at line 69 of file PacketLib.h.

5.1.2 Field Documentation

5.1.2.1 uint32_t dat_broadcast_response::serverIP

4 times 1 Byte IP V4 address

Definition at line 71 of file PacketLib.h.

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

- PacketLib.h

5.2 dat_decrypt_request Struct Reference

Decrypt data Request to decrypt data. Polynome has to be set first.

```
#include <PacketLib.h>
```

Data Fields

- int16_t clientID

The ID of the requesting client.
- uint16_t blockID

  A (random) Block ID to tell the packets apart.
- uint16_t firstElement

  First element of the data structure (16 Bit Chunks)

5.2.1 Detailed Description

Decrypt data Request to decrypt data. Polynome has to be set first.

See also

dat_polynom_request

Definition at line 42 of file PacketLib.h.

5.2.2 Field Documentation

5.2.2.1 uint16_t dat_decrypt_request::blockID

A (random) Block ID to tell the packets apart.

Definition at line 45 of file PacketLib.h.

5.2.2.2 int16_t dat_decrypt_request::clientID

The ID of the requesting client.

Definition at line 44 of file PacketLib.h.

5.2.2.3 uint16_t dat_decrypt_request::firstElement

First element of the data structure (16 Bit Chunks)

Definition at line 46 of file PacketLib.h.

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

- PacketLib.h

5.3 dat_decrypt_response Struct Reference

Return decrypted data Returns the data from successfull decryption.

```
#include <PacketLib.h>
```

Data Fields

- int16_t clientID

  The ID of the requesting client.
- uint16_t blockID

  The Block ID set by the client.
- uint8_t firstElement

  First element of the data structure (8 Bit Chunks)

5.3.1 Detailed Description

Return decrypted data Returns the data from successfull decryption.

Definition at line 51 of file PacketLib.h.

5.3.2 Field Documentation

5.3.2.1 uint16_t dat_decrypt_response::blockID

The Block ID set by the client.

Definition at line 54 of file PacketLib.h.

5.3.2.2 int16_t dat_decrypt_response::clientID

The ID of the requesting client.

Definition at line 53 of file PacketLib.h.

5.3.2.3 uint8_t dat_decrypt_response::firstElement

First element of the data structure (8 Bit Chunks)

Definition at line 55 of file PacketLib.h.

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

- PacketLib.h

5.4 dat_polynom_request Struct Reference

Set polynome request This function tries to set the polynome if the server is free or the current client has lower priority.

```
#include <PacketLib.h>
```

Data Fields

- int16_t clientID

  The ID of the requesting client.
- uint16_t generator

  The generator polynome.

5.4.1 Detailed Description

Set polynome request This function tries to set the polynome if the server is free or the current client has lower priority.

Definition at line 34 of file PacketLib.h.

5.4.2 Field Documentation

5.4.2.1 int16_t dat_polynom_request::clientID

The ID of the requesting client.

Definition at line 36 of file PacketLib.h.

5.4.2.2 uint16_t dat_polynom_request::generator

The generator polynome.

Definition at line 37 of file PacketLib.h.

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

- PacketLib.h

5.5 dat_status_response Struct Reference

Response to a status request Servers respond with their current status.

```
#include <PacketLib.h>
```

Data Fields

- int16_t clientID

  The ID of the currently connected client.
- uint16_t reserved

  Reserved.
- uint32_t wordCount

  Amount of Decrypted data words for this client.

5.5.1 Detailed Description

Response to a status request Servers respond with their current status.

Definition at line 76 of file PacketLib.h.

5.5.2 Field Documentation

5.5.2.1 int16_t dat_status_response::clientID

The ID of the currently connected client.

Definition at line 78 of file PacketLib.h.

5.5.2.2 uint16_t dat_status_response::reserved

Reserved.

See also

VALUE_RESERVED

Definition at line 79 of file PacketLib.h.

5.5.2.3 uint32_t dat_status_response::wordCount

Amount of Decrypted data words for this client.

Definition at line 80 of file PacketLib.h.

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

- PacketLib.h

5.6 dat_unlock_request Struct Reference

Unlocks the server After the server is not needed anymore you can unlock it with this function. The server can then be used by another slave.

```
#include <PacketLib.h>
```

Data Fields

- int16_t clientID

  The ID of the current client.
- uint16_t reserved

  Reserved.

5.6.1 Detailed Description

Unlocks the server After the server is not needed anymore you can unlock it with this function. The server can then be used by another slave.

See also

dat_polynom_request

Definition at line 61 of file PacketLib.h.

5.6.2 Field Documentation

5.6.2.1 int16_t dat_unlock_request::clientID

The ID of the current client.

Definition at line 63 of file PacketLib.h.

5.6.2.2 uint16_t dat_unlock_request::reserved

Reserved.

See also

VALUE_RESERVED

Definition at line 64 of file PacketLib.h.

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

- PacketLib.h

5.7 error Struct Reference

Error frame An error message frame.

```
#include <PacketLib.h>
```

Data Fields

- uint8_t errCode

    The error code of the occurring error.
- uint16_t blockID

    Block ID where the error occurred (for ERR_DECRYPT and ERR_SERVERINUSE). Else 0.

5.7.1 Detailed Description

Error frame An error message frame.

See also

Macros

Definition at line 86 of file PacketLib.h.

5.7.2 Field Documentation

5.7.2.1 uint16_t error::blockID

Block ID where the error occurred (for ERR_DECRYPT and ERR_SERVERINUSE). Else 0.

Definition at line 89 of file PacketLib.h.

5.7.2.2 uint8_t error::errCode

The error code of the occurring error.

See also

NO_ERROR
ERR_PACKETLENGTH
ERR_INVALIDVERSION
ERR_INVALIDMODE
ERR_NOSUCHFUNCTION
ERR_INVALIDTYPE
ERR_HEADER_DATA
ERR_DATA
ERR_SERVERINUSE
ERR_FUNCTIONTIMEOUT
ERR_FUNCTIONEXEC
ERR_DECRYPT
ERR_ALLOC
ERR_NO_PACKET
ERR_UNKNOWN

Definition at line 88 of file PacketLib.h.

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

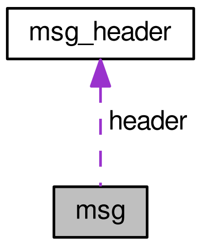
- PacketLib.h

5.8 msg Struct Reference

Structure for a message This structure holds pointers for the message header and the data structure.

`#include <PacketLib.h>`

Collaboration diagram for msg:



Data Fields

- msg_header ∗ header

  A pointer to the header of the structure.
- void ∗ data

  A pounter to the data field of the structure. Nullpointer for no data field.

5.8.1 Detailed Description

Structure for a message This structure holds pointers for the message header and the data structure.

Definition at line 94 of file PacketLib.h.

5.8.2 Field Documentation

5.8.2.1 void∗ msg::data

A pounter to the data field of the structure. Nullpointer for no data field.

See also

dat_polynom_request
dat_decrypt_request
dat_decrypt_response
dat_unlock_request
dat_broadcast_response
dat_status_response
error

Definition at line 97 of file PacketLib.h.

5.8.2.2 msg_header* msg::header

A pointer to the header of the structure.

See also

msg_header

Definition at line 96 of file PacketLib.h.

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

- PacketLib.h

5.9 msg_header Struct Reference

A structure for the message header You can easily type cast the first 8 Byte of a message to this struct. The internal structure holds all the values then.

```
#include <PacketLib.h>
```

Data Fields

- uint8_t priority

  The priority of the message (0 = HIGH, 255 = LOW)
- uint8_t version

  The current version of the script.
- uint8_t mode

  The mode of the message (sender type)
- uint8_t func:4

  The called function of this message.
- uint8_t type:4

  The message Type.
- uint16_t length

  The Length of the message data field.
- uint16_t reserved

  Reserved.

5.9.1 Detailed Description

A structure for the message header You can easily type cast the first 8 Byte of a message to this struct. The internal structure holds all the values then.

Definition at line 21 of file PacketLib.h.

5.9.2 Field Documentation

5.9.2.1 uint8_t msg_header::func

The called function of this message.

See also

FNC_POLYNOME
FNC_DECRYPT
FNC_UNLOCK
FNC_BROADCAST
FNC_STATUS

Definition at line 26 of file PacketLib.h.

5.9.2.2 uint16_t msg_header::length

The Length of the message data field.

Definition at line 28 of file PacketLib.h.

5.9.2.3 uint8_t msg_header::mode

The mode of the message (sender type)

See also

MODE_STATUS
MODE_SERVER
MODE_CLIENT

Definition at line 25 of file PacketLib.h.

5.9.2.4 uint8_t msg_header::priority

The priority of the message (0 = HIGH, 255 = LOW)

Definition at line 23 of file PacketLib.h.

5.9.2.5 uint16_t msg_header::reserved

Reserved.

See also

VALUE_RESERVED

Definition at line 29 of file PacketLib.h.

5.9.2.6 uint8_t msg_header::type

The message Type.

See also

MSG_REQUEST
MSG_RESPONSE
MSG_ERROR

Definition at line 27 of file PacketLib.h.

5.9.2.7 uint8_t msg_header::version

The current version of the script.

See also

PROTOCOL_VERSION

Definition at line 24 of file PacketLib.h.

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

- PacketLib.h

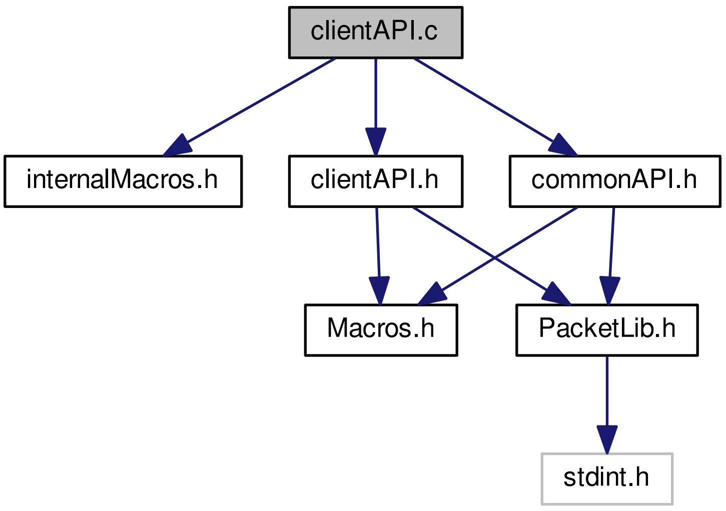
Chapter 6

File Documentation

6.1 clientAPI.c File Reference

```
#include "internalMacros.h"
#include "commonAPI.h"
#include "clientAPI.h"
```

Include dependency graph for clientAPI.c:



Functions

- int init_client ()

  Initiates the lib with the permanent client data.
- uint8_t send_gp_req (uint16_t gp, uint32_t target_server_ip)

  Send a generator polynome This function sets a generator polynome to lock a server.
- uint8_t send_dec_req (uint16_t BID, uint16_t ∗data, uint32_t data_len, uint32_t target_server_ip)

  Send a decryption request Requests the decryption of a block.

- uint8_t send_unlock_req (uint32_t target_server_ip)

  Send an unlock request Unlock a connected server.

- uint8_t send_brdcst_req ()

  Send a broadcast request.

- uint8_t extract_gp_rsp (msg ∗packet, uint32_t ∗src_server_ip)

  Extract a generator polynome response Extract the data from the polynome extract response.

- uint8_t extract_dec_rsp (msg ∗packet, uint16_t ∗BID, uint8_t ∗data, uint32_t ∗data_len, uint32_t ∗src_←
  server_ip)

  Extract the decrypted data response This function extracts the decrypted data from the message.

- uint8_t extract_unlock_rsp (msg ∗packet, uint32_t ∗src_client_ip)

  Extracts the unlock confirmation This extracts the unlock confirmation.

- uint8_t extract_brdcst_rsp (msg ∗packet, uint32_t ∗src_server_ip)

  This extracts broadcast response.

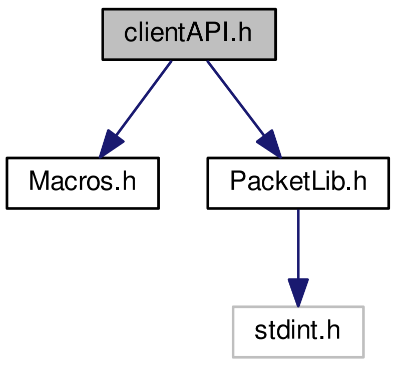
- uint8_t extract_error_rsp (msg ∗packet, uint8_t ∗error_code, uint16_t ∗BID, uint32_t ∗src_server_ip)

  Extract an error message Extract an error message from a server.

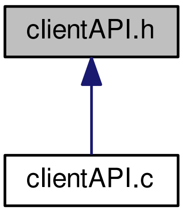
6.2 clientAPI.h File Reference

```
#include "Macros.h"
#include "PacketLib.h"
```

Include dependency graph for clientAPI.h:



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



Functions

- int init_client ()

  Initiates the lib with the permanent client data.
- uint8_t send_gp_req (uint16_t gp, uint32_t target_server_ip)

  Send a generator polynome This function sets a generator polynome to lock a server.
- uint8_t send_dec_req (uint16_t BID, uint16_t ∗data, uint32_t data_len, uint32_t target_server_ip)

  Send a decryption request Requests the decryption of a block.
- uint8_t send_unlock_req (uint32_t target_server_ip)

  Send an unlock request Unlock a connected server.
- uint8_t send_brdcst_req ()

  Send a broadcast request.
- uint8_t extract_gp_rsp (msg ∗packet, uint32_t ∗src_server_ip)

  Extract a generator polynome response Extract the data from the polynome extract response.
- uint8_t extract_dec_rsp (msg ∗packet, uint16_t ∗BID, uint8_t ∗data, uint32_t ∗data_len, uint32_t ∗src_↩
  server_ip)

  Extract the decrypted data response This function extracts the decrypted data from the message.
- uint8_t extract_unlock_rsp (msg ∗packet, uint32_t ∗src_client_ip)

  Extracts the unlock confirmation This extracts the unlock confirmation.
- uint8_t extract_brdcst_rsp (msg ∗packet, uint32_t ∗src_server_ip)

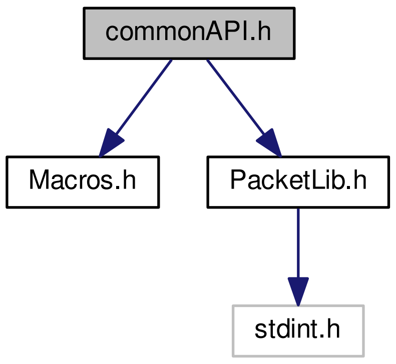
  This extracts broadcast response.
- uint8_t extract_error_rsp (msg ∗packet, uint8_t ∗error_code, uint16_t ∗BID, uint32_t ∗src_server_ip)

  Extract an error message Extract an error message from a server.

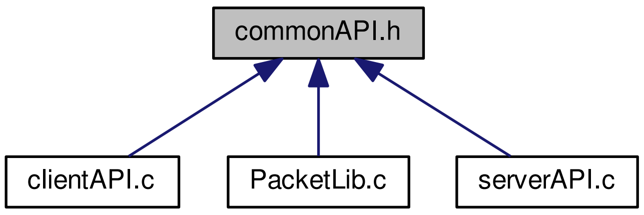
6.3 commonAPI.h File Reference

```
#include "Macros.h"
#include "PacketLib.h"
```

Include dependency graph for commonAPI.h:



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



Functions

- uint8_t recv_msg (msg ∗packet)

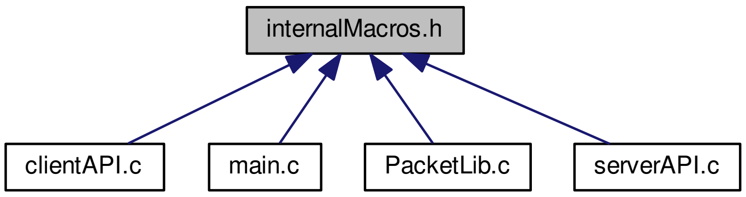
    Receive a message This function receives a message if there is one present on the Server. It also does a syntax check on the message to find badly generated packets.

- FID get_msg_type (msg ∗packet)

    get_msg_type This function returns the type of the current message.

6.4 internalMacros.h File Reference

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



Macros

- #define HEADER_LENGTH 8

  The header length in bytes.
- #define VALUE_RESERVED 0

  Standard value for reserved fields.
- #define MAX_PACKET_LENGTH 60000

  The maximal packet length (60kB)
- #define NO_BLOCK_ID 0

  No block ID is present.
- #define PROTOCOL_VERSION 14

  The version of the protocol.
- #define MODE_STATUS 1

  The status script is the message source.
- #define MODE_SERVER 2

  The message originated from a server.
- #define MODE_CLIENT 3

  A client sent the message.
- #define FNC_POLYNOME 0

  Sets the polynome in the server.
- #define FNC_DECRYPT 1

  Decrypts a chunk of the file.
- #define FNC_UNLOCK 2

  Unlocks the server to make it available for other clients.
- #define FNC_BROADCAST 5

  Broadcast to discover all available servers.
- #define FNC_STATUS 6

  Status request for that node.
- #define MSG_REQUEST 3

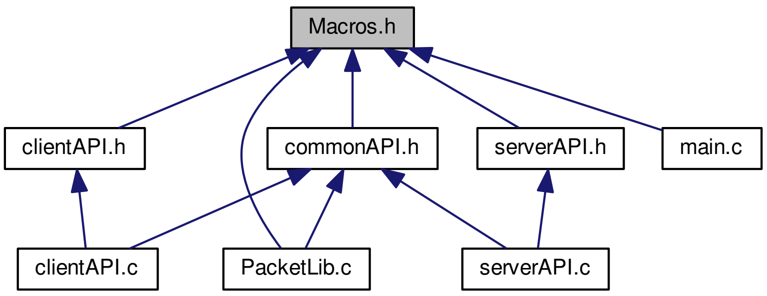
  Request the specified function.
- #define MSG_RESPONSE 4

  Response to an earlier request.
- #define MSG_ERROR 15

  An error occurred decoding or executing.

6.5 Macros.h File Reference

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



Macros

- #define NO_ERROR 0

  No error detected.
- #define ERR_PACKETLENGTH 1

  The packet length is invalid or does not match the actual length.
- #define ERR_INVALIDVERSION 2

  The version does not match the one defined in PACKET_LENGTH.
- #define ERR_INVALIDMODE 3

  The mode does not exist.
- #define ERR_NOSUCHFUNCTION 4

  The requested function does not exist (on this node)
- #define ERR_INVALIDTYPE 5

  The type is not specified.
- #define ERR_HEADER_DATA 6

  Inconsistent header data. Header is not valid.
- #define ERR_DATA 8

  Error in the data field detected.
- #define ERR_SERVERINUSE 16

  The server is currently used by another client.
- #define ERR_FUNCTIONTIMEOUT 32

  The called function timed out.
- #define ERR_FUNCTIONEXEC 33

  An error executing this function was detected.
- #define ERR_DECRYPT 64

  The data could not be decrypted due to an error.
- #define ERR_ALLOC 128

  Not enough free space to allocate data.
- #define ERR_NO_PACKET 254

  No Packet was on the socket.
- #define ERR_UNKNOWN 255

  An error occurred that does not match any of the other ones (this should never happen)

- #define ERROR -1

  An error occurred during excecution.

- #define SUCCESS 1

  Function ran without problems.

Enumerations

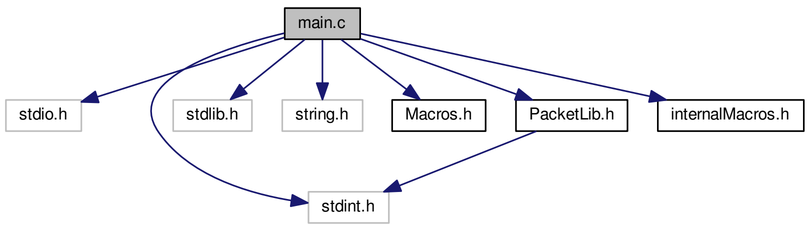
- enum FID {
  POLYNOME_REQ, POLYNOME_RSP, DECRYPT_REQ, DECRYPT_RSP,
  UNLOCK_REQ, UNLOCK_RSP, BROADCAST_REQ, BROADCAST_RSP,
  STATUS_REQ, STATUS_RSP, ERROR_RSP }

  An enumeration of all possible functions This is used as function ID reference.

6.6 main.c File Reference

```
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <string.h>
#include "Macros.h"
#include "PacketLib.h"
#include "internalMacros.h"
```

Include dependency graph for main.c:



Functions

- int main ()

6.6.1 Function Documentation

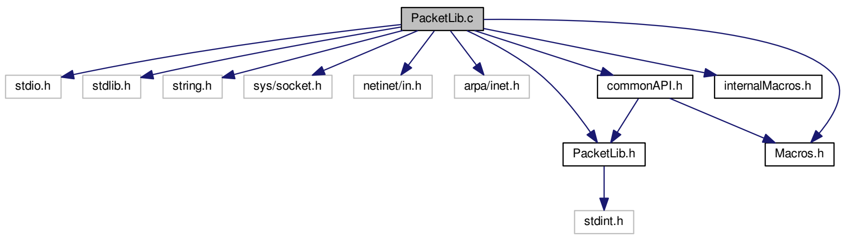
6.6.1.1 int main ( )

Definition at line 18 of file main.c.

6.7 PacketLib.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include "PacketLib.h"
#include "Macros.h"
#include "internalMacros.h"
#include "commonAPI.h"
```

Include dependency graph for PacketLib.c:



Functions

- uint8_t check_packet (msg ∗packet)

  Check a packet for internal errors.

- uint8_t recv_msg (msg ∗packet)

  Receive a message This function receives a message if there is one present on the Server. It also does a syntax check on the message to find badly generated packets.

- FID get_msg_type (msg ∗packet)

  get_msg_type This function returns the type of the current message.

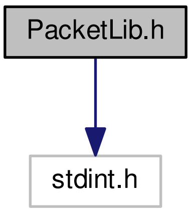
- uint8_t send_msg (msg ∗packet)

  Sends a message via UDP.

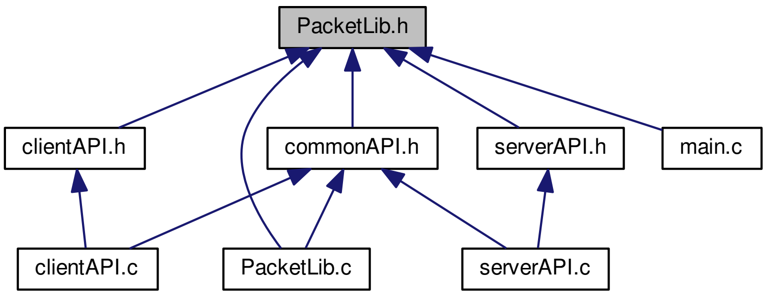
6.8 PacketLib.h File Reference

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

Include dependency graph for PacketLib.h:



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



Data Structures

- struct msg_header

    A structure for the message header You can easily type cast the first 8 Byte of a message to this struct. The internal structure holds all the values then.
- struct dat_polynom_request

    Set polynome request This function tries to set the polynome if the server is free or the current client has lower priority.
- struct dat_decrypt_request

    Decrypt data Request to decrypt data. Polynome has to be set first.
- struct dat_decrypt_response

    Return decrypted data Returns the data from successfull decryption.
- struct dat_unlock_request

    Unlocks the server After the server is not needed anymore you can unlock it with this function. The server can then be used by another slave.
- struct dat_broadcast_response

    Broadcast response Each server responds to a broadcast sending its IP address.
- struct dat_status_response

    Response to a status request Servers respond with their current status.
- struct error

Error frame An error message frame.

- struct msg

  Structure for a message This structure holds pointers for the message header and the data structure.

Functions

- struct msg_header __attribute__ ((__packed__)) msg_header

  A structure for the message header You can easily type cast the first 8 Byte of a message to this struct. The internal structure holds all the values then.
- uint8_t check_packet (msg ∗packet)

  Check a packet for internal errors.
- uint8_t send_msg (msg ∗packet)

  Sends a message via UDP.

Variables

- uint8_t priority

  The priority of the message (0 = HIGH, 255 = LOW)
- uint8_t version

  The current version of the script.
- uint8_t mode

  The mode of the message (sender type)
- uint8_t func

  The called function of this message.
- uint8_t type

  The message Type.
- uint16_t length

  The Length of the message data field.
- uint16_t reserved

  Reserved.
- int16_t clientID

  The ID of the requesting client.
- uint16_t generator

  The generator polynome.
- uint16_t blockID

  A (random) Block ID to tell the packets apart.
- uint16_t firstElement

  First element of the data structure (16 Bit Chunks)
- uint32_t serverIP

  4 times 1 Byte IP V4 address
- uint32_t wordCount

  Amount of Decrypted data words for this client.
- uint8_t errCode

  The error code of the occurring error.
- msg_header ∗ header

  A pointer to the header of the structure.
- void ∗ data

  A pounter to the data field of the structure. Nullpointer for no data field.

6.8.1 Variable Documentation

6.8.1.1 uint16_t blockID

A (random) Block ID to tell the packets apart.

Block ID where the error occurred (for ERR_DECRYPT and ERR_SERVERINUSE). Else 0.

The Block ID set by the client.

Definition at line 38 of file PacketLib.h.

6.8.1.2 int16_t clientID

The ID of the requesting client.

The ID of the currently connected client.

The ID of the current client.

Definition at line 37 of file PacketLib.h.

6.8.1.3 void∗ data

A pounter to the data field of the structure. Nullpointer for no data field.

See also

dat_polynom_request
dat_decrypt_request
dat_decrypt_response
dat_unlock_request
dat_broadcast_response
dat_status_response
error

Definition at line 38 of file PacketLib.h.

6.8.1.4 uint8_t errCode

The error code of the occurring error.

See also

NO_ERROR
ERR_PACKETLENGTH
ERR_INVALIDVERSION
ERR_INVALIDMODE
ERR_NOSUCHFUNCTION
ERR_INVALIDTYPE
ERR_HEADER_DATA
ERR_DATA
ERR_SERVERINUSE
ERR_FUNCTIONTIMEOUT
ERR_FUNCTIONEXEC
ERR_DECRYPT
ERR_ALLOC
ERR_NO_PACKET
ERR_UNKNOWN

Definition at line 37 of file PacketLib.h.

6.8.1.5 uint8_t firstElement

First element of the data structure (16 Bit Chunks)

First element of the data structure (8 Bit Chunks)

Definition at line 39 of file PacketLib.h.

6.8.1.6 uint8_t func

The called function of this message.

See also

FNC_POLYNOME
FNC_DECRYPT
FNC_UNLOCK
FNC_BROADCAST
FNC_STATUS

Definition at line 40 of file PacketLib.h.

6.8.1.7 uint16_t generator

The generator polynome.

Definition at line 38 of file PacketLib.h.

6.8.1.8 msg_header∗ header

A pointer to the header of the structure.

See also

msg_header

Definition at line 37 of file PacketLib.h.

6.8.1.9 uint16_t length

The Length of the message data field.

Definition at line 42 of file PacketLib.h.

6.8.1.10 uint8_t mode

The mode of the message (sender type)

See also

MODE_STATUS
MODE_SERVER
MODE_CLIENT

Definition at line 39 of file PacketLib.h.

6.8.1.11 uint8_t priority

The priority of the message (0 = HIGH, 255 = LOW)

Definition at line 37 of file PacketLib.h.

6.8.1.12 uint16_t reserved

Reserved.

See also

VALUE_RESERVED

Definition at line 43 of file PacketLib.h.

6.8.1.13 uint32_t serverIP

4 times 1 Byte IP V4 address

Definition at line 37 of file PacketLib.h.

6.8.1.14 uint8_t type

The message Type.

See also

MSG_REQUEST
MSG_RESPONSE
MSG_ERROR

Definition at line 41 of file PacketLib.h.

6.8.1.15 uint8_t version

The current version of the script.

See also

PROTOCOL_VERSION

Definition at line 38 of file PacketLib.h.

6.8.1.16 uint32_t wordCount

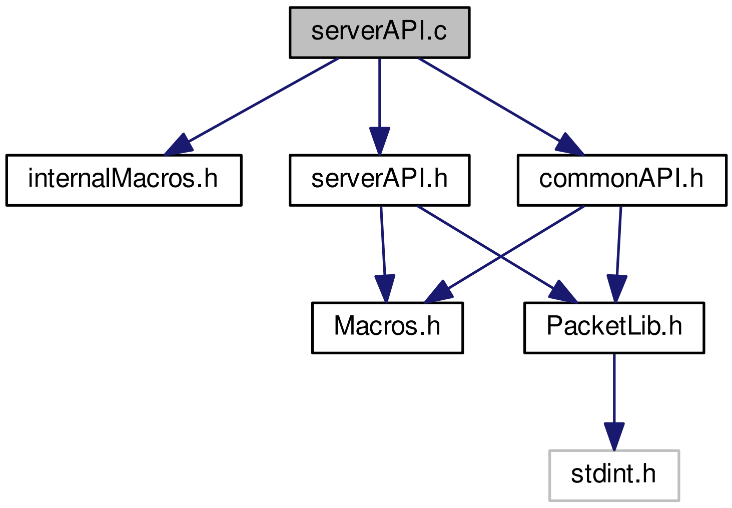
Amount of Decrypted data words for this client.

Definition at line 39 of file PacketLib.h.

6.9 serverAPI.c File Reference

```
#include "internalMacros.h"
#include "commonAPI.h"
#include "serverAPI.h"
```

Include dependency graph for serverAPI.c:



Functions

- int init_server ()

  Initiates the lib with the permanent server data.
- uint8_t send_gp_rsp (uint32_t target_client_ip)

  Send a generator polynome response Confirm the successfull setting of the generator polynome.
- uint8_t send_dec_rsp (uint16_t BID, uint8_t ∗data, uint32_t data_len, uint32_t target_client_ip)

  Send the decrypted data Return the decrypted data to the client.
- uint8_t send_unlock_rsp (uint32_t target_client_ip)

  Send the unlock confirmation.
- uint8_t send_brdcst_rsp (uint32_t target_client_ip)

  Send a broadcast response.
- uint8_t send_status_rsp (uint16_t CID, uint32_t sequence_number)

  Send a status response Send the current status to the status script.
- uint8_t send_error_rsp (uint8_t err_code, uint32_t blk_ID, uint32_t target_client_ip, FID fid)

  Send an error message.
- uint8_t extract_gp_req (msg ∗packet, uint16_t ∗gp, uint16_t ∗CID, uint8_t ∗prio, uint32_t ∗src_client_ip)

  Extract the generator polynome Extract the generator polynome from the packet.
- uint8_t extract_dec_req (msg ∗packet, uint16_t ∗CID, uint16_t ∗BID, uint16_t ∗data, uint32_t ∗data_len, uint32_t ∗src_client_ip)

  Extract data to decrypt.
- uint8_t extract_unlock_req (msg ∗packet, uint16_t ∗CID, uint32_t ∗src_client_ip)

  Extract the unlock command extract the command to unlock the server.
- uint8_t extract_brdcst_req (msg ∗packet, uint32_t ∗src_client_ip)

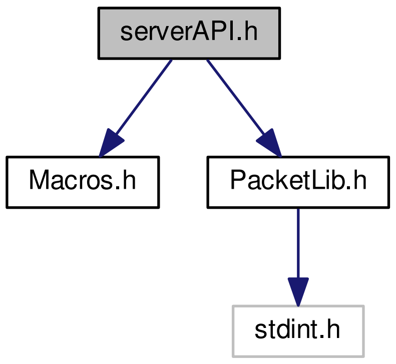
  Extract a broadcast request.
- uint8_t extract_status_req (msg ∗packet)

  Extract a status request.

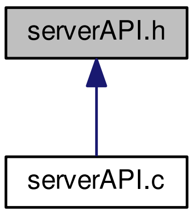
6.10 serverAPI.h File Reference

```
#include "Macros.h"
#include "PacketLib.h"
```

Include dependency graph for serverAPI.h:



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



Functions

- int init_server ()

  Initiates the lib with the permanent server data.
- uint8_t send_gp_rsp (uint32_t target_client_ip)

  Send a generator polynome response Confirm the successfull setting of the generator polynome.
- uint8_t send_dec_rsp (uint16_t BID, uint8_t ∗data, uint32_t data_len, uint32_t target_client_ip)

  Send the decrypted data Return the decrypted data to the client.
- uint8_t send_unlock_rsp (uint32_t target_client_ip)

  Send the unlock confirmation.
- uint8_t send_brdcst_rsp (uint32_t target_client_ip)

  Send a broadcast response.

- uint8_t send_status_rsp (uint16_t CID, uint32_t sequence_number)

    Send a status response Send the current status to the status script.
- uint8_t send_error_rsp (uint8_t err_code, uint32_t BID, uint32_t target_client_ip, FID fid)

    Send an error message.
- uint8_t extract_gp_req (msg ∗packet, uint16_t ∗gp, uint16_t ∗CID, uint8_t ∗prio, uint32_t ∗src_client_ip)

    Extract the generator polynome Extract the generator polynome from the packet.
- uint8_t extract_dec_req (msg ∗packet, uint16_t ∗CID, uint16_t ∗BID, uint16_t ∗data, uint32_t ∗data_len, uint32_t ∗src_client_ip)

    Extract data to decrypt.
- uint8_t extract_unlock_req (msg ∗packet, uint16_t ∗CID, uint32_t ∗src_client_ip)

    Extract the unlock command extract the command to unlock the server.
- uint8_t extract_brdcst_req (msg ∗packet, uint32_t ∗src_client_ip)

    Extract a broadcast request.
- uint8_t extract_status_req (msg ∗packet)

    Extract a status request.