Verteilte Systeme Labor 2.7

Generated by Doxygen 1.8.11

# **Contents**

1	Mod	ule Inde	ex		1
	1.1	Module	es		1
2	Data	Structi	ure Index		3
	2.1	Data S	tructures		3
3	File	Index			5
	3.1	File Lis	st		5
4	Mod	ule Doc	umentatio	on .	7
	4.1	Macros	s for interna	al use only	7
		4.1.1	Detailed	Description	8
		4.1.2	Macro De	efinition Documentation	8
			4.1.2.1	BROADCAST_ADDRESS	8
			4.1.2.2	CLIENT_PORT	8
			4.1.2.3	FNC_BROADCAST	8
			4.1.2.4	FNC_DECRYPT	8
			4.1.2.5	FNC_GP	8
			4.1.2.6	FNC_STATUS	8
			4.1.2.7	FNC_UNLOCK	9
			4.1.2.8	MAX_PACKET_LENGTH	9
			4.1.2.9	MODE_CLIENT	9
			4.1.2.10	MODE_SERVER	9
			4.1.2.11	MODE_STATUS	9
			41212	MSG FRROR	q

iv CONTENTS

		4.1.2.13	MSG_REQUEST	9
		4.1.2.14	MSG_RESPONSE	9
		4.1.2.15	NO_BLOCK_ID	10
		4.1.2.16	PROTOCOL_VERSION	10
		4.1.2.17	SERVER_PORT	10
		4.1.2.18	SERVER_PRIO	10
		4.1.2.19	VALUE_RESERVED	10
4.2	Genera	al API func	tions	11
	4.2.1	Detailed	Description	11
	4.2.2	Function	Documentation	11
		4.2.2.1	free_data(uint8_t *ptr)	11
		4.2.2.2	free_msg(msg *packet)	11
		4.2.2.3	get_msg_type(msg *packet)	12
		4.2.2.4	recv_msg(msg *packet, uint32_t *src_ip, uint16_t *src_port)	12
4.3	Macros	3		14
	4.3.1	Detailed	Description	15
	4.3.2	Macro De	efinition Documentation	15
		4.3.2.1	ERR_ALLOC	15
		4.3.2.2	ERR_DATA	15
		4.3.2.3	ERR_DECRYPT	15
		4.3.2.4	ERR_FUNCTIONEXEC	15
		4.3.2.5	ERR_FUNCTIONTIMEOUT	15
		4.3.2.6	ERR_HEADER_DATA	16
		4.3.2.7	ERR_INVALID_PTR	16
		4.3.2.8	ERR_INVALIDMODE	16
		4.3.2.9	ERR_INVALIDTYPE	16
		4.3.2.10	ERR_INVALIDVERSION	16
		4.3.2.11	ERR_LOCK_TIMEOUT	16
		4.3.2.12	ERR_NO_GP	16
		4.3.2.13	ERR_NO_INIT	16

CONTENTS

		4.3.2.14	ERR_NO_PACKET	17
		4.3.2.15	ERR_NOSUCHFUNCTION	17
		4.3.2.16	ERR_NOTFORME	17
		4.3.2.17	ERR_PACKETLENGTH	17
		4.3.2.18	ERR_SEND_ERROR	17
		4.3.2.19	ERR_SERVERINUSE	17
		4.3.2.20	ERR_UNKNOWN	17
		4.3.2.21	ERROR	17
		4.3.2.22	NO_ERROR	18
		4.3.2.23	SUCCESS	18
	4.3.3	Enumera	ation Type Documentation	18
		4.3.3.1	FID	18
4.4	Structu	ıres		19
	4.4.1	Detailed	Description	19
	4.4.2	Function	Documentation	20
		4.4.2.1	attribute((packed)) msg_header	20
	4.4.3	Variable	Documentation	20
		4.4.3.1	my_addr	20
		4.4.3.2	socketDscp	20
		4.4.3.2 4.4.3.3	socketDscp	20
4.5	Interna	4.4.3.3		
4.5	Interna 4.5.1	4.4.3.3	target_addr	20
4.5		4.4.3.3  Al Function  Detailed	target_addr	20 21
4.5	4.5.1	4.4.3.3  Al Function  Detailed	target_addr	20 21 21
4.5	4.5.1	4.4.3.3  Al Function  Detailed  Function	target_addr	20 21 21 21
4.5	4.5.1	4.4.3.3  Al Function  Detailed  Function  4.5.2.1	target_addr	20 21 21 21 21
4.5	4.5.1 4.5.2	4.4.3.3  Al Function  Detailed  Function  4.5.2.1  4.5.2.2  4.5.2.3	target_addr	20 21 21 21 21 21
	4.5.1 4.5.2	4.4.3.3  Al Function  Detailed  Function  4.5.2.1  4.5.2.2  4.5.2.3  Functions	target_addr	200 211 211 211 211 222
	4.5.1 4.5.2 Client	4.4.3.3  Al Function  Detailed  Function  4.5.2.1  4.5.2.2  4.5.2.3  Functions  Detailed	target_addr	20 21 21 21 21 21 22 23

vi

		4.6.2.2	extract_brdcst_rsp(msg *packet)	23
		4.6.2.3	extract_dec_rsp(msg *packet, uint16_t *BID, uint8_t **data, uint32_t *data_len)	24
		4.6.2.4	extract_error_rsp(msg *packet, uint8_t *error_code, uint16_t *BID)	24
		4.6.2.5	extract_gp_rsp(msg *packet)	25
		4.6.2.6	extract_unlock_rsp(msg *packet)	25
		4.6.2.7	init_client(int16_t p_cID, uint8_t p_prio, uint32_t p_bca)	26
		4.6.2.8	send_brdcst_req(uint16_t target_server_port)	26
		4.6.2.9	send_dec_req(uint16_t BID, uint16_t *data, uint32_t data_len, uint32_t target_← server_ip, uint16_t target_server_port)	27
		4.6.2.10	send_gp_req(uint16_t gp, uint32_t target_server_ip, uint16_t target_server_port)	27
		4.6.2.11	send_unlock_req(uint32_t target_server_ip, uint16_t target_server_port)	28
4.7	Server	Functions		29
	4.7.1	Detailed	Description	29
	4.7.2	Function	Documentation	29
		4.7.2.1	deinit_server()	29
		4.7.2.2	extract_brdcst_req(msg *packet)	30
		4.7.2.3	extract_dec_req(msg *packet, int16_t *CID, uint16_t *BID, uint16_t **data, uint32_t *data_len)	30
		4.7.2.4	extract_gp_req(msg *packet, uint16_t *gp, int16_t *CID, uint8_t *prio)	31
		4.7.2.5	extract_status_req(msg *packet)	31
		4.7.2.6	extract_unlock_req(msg *packet, int16_t *CID)	32
		4.7.2.7	init_server()	32
		4.7.2.8	send_brdcst_rsp(uint32_t target_client_ip, uint16_t target_client_port)	32
		4.7.2.9	send_dec_rsp(uint16_t BID, int16_t clientID, uint8_t *data, uint32_t data_len, uint32_t target_client_ip, uint16_t target_client_port)	33
		4.7.2.10	send_error_rsp(uint8_t err_code, uint32_t BID, FID fid, uint32_t target_client_ip, uint16_t target_client_port)	33
		4.7.2.11	send_gp_rsp(uint32_t target_client_ip, uint16_t target_client_port)	34
		4.7.2.12	send_status_rsp(int16_t CID, uint32_t sequence_number, uint32_t target_← status_ip, uint16_t target_client_port)	34
		4.7.2.13	send_unlock_rsp(uint32_t target_client_ip, uint16_t target_client_port)	35

CONTENTS vii

5	Data	Structi	ture Documentation	37
	5.1	dat_de	ecrypt_request Struct Reference	37
		5.1.1	Detailed Description	37
		5.1.2	Field Documentation	37
			5.1.2.1 blockID	37
			5.1.2.2 clientID	38
			5.1.2.3 firstElement	38
	5.2	dat_de	ecrypt_response Struct Reference	38
		5.2.1	Detailed Description	38
		5.2.2	Field Documentation	38
			5.2.2.1 blockID	38
			5.2.2.2 clientID	39
			5.2.2.3 firstElement	39
	5.3	dat_gp	p_request Struct Reference	39
		5.3.1	Detailed Description	39
		5.3.2	Field Documentation	39
			5.3.2.1 clientID	39
			5.3.2.2 generator	40
	5.4	dat_sta	atus_response Struct Reference	40
		5.4.1	Detailed Description	40
		5.4.2	Field Documentation	40
			5.4.2.1 clientID	40
			5.4.2.2 reserved	40
			5.4.2.3 wordCount	41
	5.5	dat_un	nlock_request Struct Reference	41
		5.5.1	Detailed Description	41
		5.5.2	Field Documentation	41
			5.5.2.1 clientID	41
			5.5.2.2 reserved	42
	5.6	error S	Struct Reference	42

viii CONTENTS

		5.6.1	Detailed Description	42
		5.6.2	Field Documentation	42
			5.6.2.1 blockID	42
			5.6.2.2 errCode	43
	5.7	msg St	truct Reference	43
		5.7.1	Detailed Description	44
		5.7.2	Field Documentation	44
			5.7.2.1 data	44
			5.7.2.2 header	44
	5.8	msg_h	neader Struct Reference	45
		5.8.1	Detailed Description	45
		5.8.2	Field Documentation	45
			5.8.2.1 func	45
			5.8.2.2 length	46
			5.8.2.3 mode	46
			5.8.2.4 priority	46
			5.8.2.5 reserved	46
			5.8.2.6 type	46
			5.8.2.7 version	46
	Eile I	Deaum		47
,				<b>47</b>
	6.1			
		6.1.1		48
				48
				48
				48
				48
	6.2	clientA	PI.h File Reference	49
	6.3	commo	onAPI.h File Reference	50
	6.4	interna	alMacros.h File Reference	51
	6.5	Macros	s.h File Reference	52

CONTENTS

6.6	main.c	File Refere	ence	. 54
	6.6.1	Function [	Documentation	. 54
		6.6.1.1	main()	. 54
6.7	Packet	Lib.c File R	eference	. 55
	6.7.1	Variable D	Occumentation	. 56
		6.7.1.1	buffer	. 56
6.8	Packet	Lib.h File R	deference	. 56
	6.8.1	Macro De	finition Documentation	. 58
		6.8.1.1	DEBUG_PRINTF	. 58
		6.8.1.2	DEBUG_PRINTF1	. 58
		6.8.1.3	DEBUG_PRINTF2	. 58
		6.8.1.4	DEBUG_PRINTF3	. 58
	6.8.2	Variable D	Documentation	. 58
		6.8.2.1	blockID	. 58
		6.8.2.2	clientID	. 59
		6.8.2.3	data	. 59
		6.8.2.4	errCode	. 59
		6.8.2.5	firstElement	. 60
		6.8.2.6	func	. 60
		6.8.2.7	generator	. 60
		6.8.2.8	header	. 60
		6.8.2.9	length	. 60
		6.8.2.10	mode	. 61
		6.8.2.11	priority	. 61
		6.8.2.12	reserved	. 61
		6.8.2.13	type	. 61
		6.8.2.14	version	. 61
		6.8.2.15	wordCount	. 62
6.9	server	API.c File R	Reference	. 62
	6.9.1	Variable D	Documentation	. 63
		6.9.1.1	initialized	. 63
6.10	server/	API.h File R	Reference	. 63

# **Chapter 1**

# **Module Index**

### 1.1 Modules

#### Here is a list of all modules:

acros for internal use only	7
eneral API functions	11
acros	14
tructures	19
ternal Functions	21
lient Functions	23
erver Functions	29

2 Module Index

# Chapter 2

# **Data Structure Index**

### 2.1 Data Structures

Here are the data structures with brief descriptions:

dat_dec	crypt_request	
	Decrypt data Request to decrypt data. Polynome has to be set first	37
dat_dec	crypt_response	
	Return decrypted data Returns the data from successfull decryption	38
dat_gp_	_request	
	Set polynome request This function tries to set the polynome if the server is free or the current	
	client has lower priority	39
dat_stat	tus_response	
	Response to a status request Servers respond with their current status	40
dat_unlo	ock_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	41
error		
	Error frame An error message frame	42
msg		
	Structure for a message This structure holds pointers for the message header and the data	
	structure	43
msg_he	eader	
	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	45

Data Structure Index

# **Chapter 3**

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

clientAPI.c																 										47
clientAPI.h																 										49
commonAPI.	h															 										50
internalMacro	os.	h														 										51
Macros.h .																 										52
main.c																 										54
PacketLib.c																 										55
PacketLib.h																 										56
serverAPI.c																 										62
serverAPI h																								_		63

6 File Index

### Chapter 4

### **Module Documentation**

### 4.1 Macros for internal use only

#### **Macros**

```
• #define SERVER_PORT 11111
```

The port on the Server side.

• #define CLIENT\_PORT 11111

The port on the Client side.

#define BROADCAST\_ADDRESS "127.0.0.1"

The Placeholder for broadcast address.

• #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 SERVER\_PRIO 0

Priority of server messages.

#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\_GP 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 BROADCAST\_ADDRESS "127.0.0.1"

The Placeholder for broadcast address.

Definition at line 17 of file internal Macros.h.

#### 4.1.2.2 #define CLIENT\_PORT 11111

The port on the Client side.

Definition at line 16 of file internalMacros.h.

#### 4.1.2.3 #define FNC\_BROADCAST 5

Broadcast to discover all available servers.

Definition at line 38 of file internalMacros.h.

#### 4.1.2.4 #define FNC\_DECRYPT 1

Decrypts a chunk of the file.

Definition at line 36 of file internal Macros.h.

#### 4.1.2.5 #define FNC\_GP 0

Sets the polynome in the server.

Definition at line 35 of file internalMacros.h.

#### 4.1.2.6 #define FNC\_STATUS 6

Status request for that node.

Definition at line 39 of file internalMacros.h.

#### 4.1.2.7 #define FNC\_UNLOCK 2

Unlocks the server to make it available for other clients.

Definition at line 37 of file internalMacros.h.

#### 4.1.2.8 #define MAX\_PACKET\_LENGTH 60000

The maximal packet length (60kB)

Definition at line 22 of file internalMacros.h.

#### 4.1.2.9 #define MODE\_CLIENT 3

A client sent the message.

Definition at line 32 of file internalMacros.h.

#### 4.1.2.10 #define MODE\_SERVER 2

The message originated from a server.

Definition at line 31 of file internalMacros.h.

#### 4.1.2.11 #define MODE\_STATUS 1

The status script is the message source.

Definition at line 30 of file internalMacros.h.

#### 4.1.2.12 #define MSG\_ERROR 15

An error occurred decoding or executing.

Definition at line 44 of file internalMacros.h.

#### 4.1.2.13 #define MSG\_REQUEST 3

Request the specified function.

Definition at line 42 of file internalMacros.h.

#### 4.1.2.14 #define MSG\_RESPONSE 4

Response to an earlier request.

Definition at line 43 of file internalMacros.h.

4.1.2.15 #define NO\_BLOCK\_ID 0

No block ID is present.

Definition at line 23 of file internalMacros.h.

4.1.2.16 #define PROTOCOL\_VERSION 14

The version of the protocol.

Definition at line 27 of file internalMacros.h.

4.1.2.17 #define SERVER\_PORT 11111

The port on the Server side.

Definition at line 15 of file internalMacros.h.

4.1.2.18 #define SERVER\_PRIO 0

Priority of server messages.

Definition at line 24 of file internalMacros.h.

4.1.2.19 #define VALUE\_RESERVED 0

Standard value for reserved fields.

Definition at line 21 of file internalMacros.h.

4.2 General API functions 11

#### 4.2 General API functions

#### **Functions**

uint8\_t recv\_msg (msg \*packet, uint32\_t \*src\_ip, uint16\_t \*src\_port)

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 free\_msg (msg \*packet)

Deletes the subfields of a msg type (But not the msg itself!)

• uint8\_t free\_data (uint8\_t \*ptr)

Frees the data stream allocated by extract\_dec\_req / \_rsp.

#### 4.2.1 Detailed Description

API Functions that apply to both sides, client and server

#### 4.2.2 Function Documentation

```
4.2.2.1 uint8_t free_data ( uint8_t * ptr )
```

Frees the data stream allocated by extract\_dec\_req / \_rsp.

**Author** 

Philipp Duller

#### **Parameters**

in	ptr	: the data field to be deleted

Returns

NO\_ERROR

Definition at line 453 of file PacketLib.c.

4.2.2.2 uint8\_t free\_msg ( msg \* packet )

Deletes the subfields of a msg type (But not the msg itself!)

Author

Michel Schmidt

#### **Parameters**

in <i>packet</i>	: the message to delete
------------------	-------------------------

#### Returns

ERROR if the packet was not valid; SUCCESS if not

Definition at line 429 of file PacketLib.c.

```
4.2.2.3 FID get_msg_type ( msg * packet )
```

get\_msg\_type This function returns the type of the current message.

#### **Author**

Michel Schmidt

#### **Parameters**

	in	packet	: The packet to get the type of.
--	----	--------	----------------------------------

#### **Returns**

The error code that occurred.

#### See also

FID Macros

Definition at line 244 of file PacketLib.c.

```
4.2.2.4 uint8_t recv_msg ( msg * packet, uint32_t * src_ip, uint16_t * src_port )
```

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**

Cornelius Bott

#### **Parameters**

out	packet	: The received packet
out	src_ip	: Source-IP-Address
out	src_port	: Source Port number

4.2 General API functions 13

#### Returns

The error code of the message.  ${\sf ERR\_NO\_PACKET}$  if there is no message.

See also

msg Macros

Definition at line 319 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\_LOCK\_TIMEOUT 34

The Server lock timed out.

• #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\_INVALID\_PTR 129

The given pointer was not valid.

#define ERR\_NOTFORME 130

Client detected client ID miss match.

• #define ERR NO GP 131

No GP is set in Server.

#define ERR\_SEND\_ERROR 252

Could not send message.

• #define ERR\_NO\_INIT 253

The API lib was not initialized.

• #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.

4.3 Macros 15

#### **Enumerations**

```
    enum FID {
        GP_REQ, GP_RSP, DECRYPT_REQ, DECRYPT_RSP,
        UNLOCK_REQ, UNLOCK_RSP, BROADCAST_REQ, BROADCAST_RSP,
        STATUS_REQ, STATUS_RSP, UNKNOWN, 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 31 of file Macros.h.

4.3.2.2 #define ERR\_DATA 8

Error in the data field detected.

Definition at line 25 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 30 of file Macros.h.

4.3.2.4 #define ERR\_FUNCTIONEXEC 33

An error executing this function was detected.

Definition at line 28 of file Macros.h.

4.3.2.5 #define ERR\_FUNCTIONTIMEOUT 32

The called function timed out.

Definition at line 27 of file Macros.h.

4.3.2.6 #define ERR\_HEADER\_DATA 6

Inconsistent header data. Header is not valid.

Definition at line 24 of file Macros.h.

4.3.2.7 #define ERR\_INVALID\_PTR 129

The given pointer was not valid.

Definition at line 32 of file Macros.h.

4.3.2.8 #define ERR\_INVALIDMODE 3

The mode does not exist.

Definition at line 21 of file Macros.h.

4.3.2.9 #define ERR\_INVALIDTYPE 5

The type is not specified.

Definition at line 23 of file Macros.h.

4.3.2.10 #define ERR\_INVALIDVERSION 2

The version does not match the one defined in PACKET\_LENGTH.

Definition at line 20 of file Macros.h.

4.3.2.11 #define ERR\_LOCK\_TIMEOUT 34

The Server lock timed out.

Definition at line 29 of file Macros.h.

4.3.2.12 #define ERR\_NO\_GP 131

No GP is set in Server.

Definition at line 34 of file Macros.h.

4.3.2.13 #define ERR\_NO\_INIT 253

The API lib was not initialized.

Definition at line 36 of file Macros.h.

4.3 Macros 17

4.3.2.14 #define ERR\_NO\_PACKET 254 No Packet was on the socket. Definition at line 37 of file Macros.h. 4.3.2.15 #define ERR\_NOSUCHFUNCTION 4 The requested function does not exist (on this node) Definition at line 22 of file Macros.h. 4.3.2.16 #define ERR\_NOTFORME 130 Client detected client ID miss match. Definition at line 33 of file Macros.h. 4.3.2.17 #define ERR\_PACKETLENGTH 1 The packet length is invalid or does not match the actual length. Definition at line 19 of file Macros.h. 4.3.2.18 #define ERR\_SEND\_ERROR 252 Could not send message. Definition at line 35 of file Macros.h. 4.3.2.19 #define ERR\_SERVERINUSE 16 The server is currently used by another client. Definition at line 26 of file Macros.h. 4.3.2.20 #define ERR\_UNKNOWN 255 An error occurred that does not match any of the other ones (this should never happen) Definition at line 38 of file Macros.h. 4.3.2.21 #define ERROR -1 An error occurred during excecution.

Definition at line 41 of file Macros.h.

#### 4.3.2.22 #define NO\_ERROR 0

No error detected.

Definition at line 18 of file Macros.h.

4.3.2.23 #define SUCCESS 1

Function ran without problems.

Definition at line 42 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

```
GP_REQ Function: set polynome; Type: Request.
GP_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.

UNKNOWN Unknown function. This should not happen.

ERROR_RSP Function: any; Type: Error.
```

Definition at line 47 of file Macros.h.

4.4 Structures 19

#### 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\_gp\_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\_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.

#### **Variables**

- int socketDscp
- struct sockaddr\_in my\_addr
- struct sockaddr\_in target\_addr

#### 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.

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

#### 4.4.3 Variable Documentation

4.4.3.1 struct sockaddr\_in my\_addr

Definition at line 23 of file PacketLib.c.

4.4.3.2 int socketDscp

Definition at line 22 of file PacketLib.c.

4.4.3.3 struct sockaddr\_in target\_addr

Definition at line 24 of file PacketLib.c.

4.5 Internal Functions 21

### 4.5 Internal Functions

#### **Functions**

```
    uint8_t check_pointers (msg *packet)
    check_pointers
```

uint8\_t check\_packet (msg \*packet)

Check a packet for internal errors.

• uint8\_t send\_msg (msg \*packet, uint32\_t target\_ip, uint16\_t target\_port)

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**

ſ	in	packet	: The packet structure
---	----	--------	------------------------

Returns

The error code that occurred

See also

Macros

Definition at line 49 of file PacketLib.c.

4.5.2.2 uint8\_t check\_pointers ( msg \* packet )

check\_pointers

Author

Michel Schmidt

#### **Parameters**

packet	: the packet pointers to check
--------	--------------------------------

#### Returns

ERR\_INVALID\_PTR or NO\_ERROR

#### See also

```
ERR_INVALID_PTR
NO_ERROR
```

Definition at line 27 of file PacketLib.c.

4.5.2.3 uint8\_t send\_msg ( msg \* packet, uint32\_t target\_ip, uint16\_t target\_port )

Sends a message via UDP.

#### Author

Stefan Scharrenbach

#### **Parameters**

in	packet	The packet to send
in	target_ip	The IP to send to
in	target_port	The Port to send to

#### Returns

The error code that occurred

#### See also

msg Macros

Definition at line 393 of file PacketLib.c.

4.6 Client Functions 23

#### 4.6 Client Functions

#### **Functions**

int init\_client (int16\_t p\_cID, uint8\_t p\_prio, uint32\_t p\_bca)

Initiates the lib with the permanent client data.

• int deinit client ()

Deinitializes the client lib.

uint8\_t send\_gp\_req (uint16\_t gp, uint32\_t target\_server\_ip, uint16\_t target\_server\_port)

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, uint16\_t target server port)

Send a decryption request Requests the decryption of a block.

uint8\_t send\_unlock\_req (uint32\_t target\_server\_ip, uint16\_t target\_server\_port)

Send an unlock request Unlock a connected server.

uint8\_t send\_brdcst\_req (uint16\_t target\_server\_port)

Send a broadcast request.

uint8\_t extract\_gp\_rsp (msg \*packet)

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)

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

uint8\_t extract\_unlock\_rsp (msg \*packet)

Extracts the unlock confirmation This extracts the unlock confirmation.

uint8\_t extract\_brdcst\_rsp (msg \*packet)

This extracts broadcast response.

• uint8\_t extract\_error\_rsp (msg \*packet, uint8\_t \*error\_code, uint16\_t \*BID)

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 int deinit_client ( )
```

Deinitializes the client lib.

**Author** 

<ADD here>="">

Returns

Error or SUCCESS

Definition at line 66 of file clientAPI.c.

4.6.2.2 uint8\_t extract\_brdcst\_rsp ( msg \* packet )

This extracts broadcast response.

**Author** 

Philipp Duller

#### **Parameters**

in	packet	: the packet to extract
----	--------	-------------------------

#### Returns

The error code that occurred.

#### See also

Macros

Definition at line 286 of file clientAPI.c.

```
4.6.2.3 uint8_t extract_dec_rsp ( msg * packet, uint16_t * BID, uint8_t ** data, uint32_t * data_len )
```

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

#### Author

Philipp Duller

#### **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

#### Returns

The error code that occurred.

#### See also

Macros

Definition at line 241 of file clientAPI.c.

4.6.2.4 uint8\_t extract\_error\_rsp ( msg\*packet, uint8\_t \* error\_code, uint16\_t \* BID )

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

#### Author

Philipp Duller

4.6 Client Functions 25

#### **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)

#### Returns

The error code that occurred.

#### See also

Macros

Definition at line 301 of file clientAPI.c.

4.6.2.5 uint8\_t extract\_gp\_rsp ( msg \* packet )

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

#### Author

Philipp Duller

#### **Parameters**

i	n	packet	: the packet to extract
---	---	--------	-------------------------

#### Returns

The error code that occurred.

#### See also

Macros

Definition at line 226 of file clientAPI.c.

4.6.2.6 uint8\_t extract\_unlock\_rsp ( msg \* packet )

Extracts the unlock confirmation This extracts the unlock confirmation.

**Author** 

Philipp Duller

#### **Parameters**

in	packet	: the packet to extract

#### Returns

The error code that occurred.

#### See also

#### Macros

Definition at line 271 of file clientAPI.c.

4.6.2.7 int init\_client ( int16\_t p\_clD, uint8\_t p\_prio, uint32\_t p\_bca )

Initiates the lib with the permanent client data.

#### Author

Philipp Duller

#### **Parameters**

	in	p_cID	: the client ID
ſ	in	p_prio	: the client priority
ſ	in	p_bca	: the broadcast address

#### Returns

Error or SUCCESS

Definition at line 21 of file clientAPI.c.

4.6.2.8 uint8\_t send\_brdcst\_req ( uint16\_t target\_server\_port )

Send a broadcast request.

#### **Author**

Philipp Duller

#### **Parameters**

in target_server_port	The Port of the target server
-----------------------	-------------------------------

4.6 Client Functions 27

### Returns

The error code that occurred.

### See also

Macros

Definition at line 193 of file clientAPI.c.

```
4.6.2.9 uint8_t send_dec_req ( uint16_t BID, uint16_t * data, uint32_t data_len, uint32_t target_server_ip, uint16_t target_server_port )
```

Send a decryption request Requests the decryption of a block.

### Author

Simon Lauser

#### **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
in	target_server_port	The Port of the target server

# Returns

The error code that occurred.

### See also

Macros

Definition at line 109 of file clientAPI.c.

4.6.2.10 uint8\_t send\_gp\_req ( uint16\_t gp, uint32\_t target\_server\_ip, uint16\_t target\_server\_port )

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

### Author

Simon Lauser

#### **Parameters**

in	gp	: the generator polynome
in	target_server_ip	: the IP address of the target server
Generat	edtargetygenrver_port	The Port of the target server

28 Module Documentation

### Returns

The error code that occurred.

### See also

Macros

Definition at line 72 of file clientAPI.c.

```
4.6.2.11 uint8_t send_unlock_req ( uint32_t target_server_ip, uint16_t target_server_port )
```

Send an unlock request Unlock a connected server.

### Author

Philipp Duller

### **Parameters**

in	target_server_ip	: the IP address of the target server
in	target_server_port	The Port of the target server

### Returns

The error code that occurred.

## See also

Macros

Definition at line 151 of file clientAPI.c.

4.7 Server Functions 29

### 4.7 Server Functions

#### **Functions**

• int init server ()

Initiates the lib with the permanent server data.

• int deinit\_server ()

Deinitializes the server lib.

• uint8\_t send\_gp\_rsp (uint32\_t target\_client\_ip, uint16\_t target\_client\_port)

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

uint8\_t send\_dec\_rsp (uint16\_t BID, int16\_t clientID, uint8\_t \*data, uint32\_t data\_len, uint32\_t target\_client
 \_ip, uint16\_t target\_client\_port)

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

uint8\_t send\_unlock\_rsp (uint32\_t target\_client\_ip, uint16\_t target\_client\_port)

Send the unlock confirmation.

• uint8\_t send\_brdcst\_rsp (uint32\_t target\_client\_ip, uint16\_t target\_client\_port)

Send a broadcast response.

uint8\_t send\_status\_rsp (int16\_t CID, uint32\_t sequence\_number, uint32\_t target\_status\_ip, uint16\_t target
 — client\_port)

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

uint8\_t send\_error\_rsp (uint8\_t err\_code, uint32\_t BID, FID fid, uint32\_t target\_client\_ip, uint16\_t target\_
 client\_port)

Send an error message.

• uint8\_t extract\_gp\_req (msg \*packet, uint16\_t \*gp, int16\_t \*CID, uint8\_t \*prio)

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

- uint8\_t extract\_dec\_req (msg \*packet, int16\_t \*CID, uint16\_t \*BID, uint16\_t \*\*data, uint32\_t \*data\_len)
   Extract data to decrypt.
- uint8\_t extract\_unlock\_req (msg \*packet, int16\_t \*CID)

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

uint8\_t extract\_brdcst\_req (msg \*packet)

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 int deinit\_server ( )

Deinitializes the server lib.

Author

<ADD here>="">

Returns

Error or SUCCESS

Definition at line 62 of file serverAPI.c.

30 Module Documentation

4.7.2.2 uint8\_t extract\_brdcst\_req ( msg \* packet )

Extract a broadcast request.

Author

Michel Schmidt

### **Parameters**

in	packet	: the packet to extract
----	--------	-------------------------

### Returns

The error code that occurred.

### See also

Macros

Definition at line 386 of file serverAPI.c.

 $\textbf{4.7.2.3} \quad \textbf{uint8\_t extract\_dec\_req ( msg*\textit{packet}, int16\_t*\textit{CID}, uint16\_t*\textit{BID}, uint16\_t**\textit{data}, uint32\_t*\textit{data\_len} )}$ 

Extract data to decrypt.

Author

Michel Schmidt

### **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

### Returns

The error code that occurred.

See also

Macros

Definition at line 338 of file serverAPI.c.

4.7 Server Functions 31

4.7.2.4 uint8\_t extract\_gp\_req ( msg\*packet, uint16\_t \* gp, int16\_t \* CID, uint8\_t \* prio )

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

Author

Michel Schmidt

### **Parameters**

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

### Returns

The error code that occurred.

See also

Macros

Definition at line 319 of file serverAPI.c.

4.7.2.5 uint8\_t extract\_status\_req ( msg \* packet )

Extract a status request.

Author

Michel Schmidt

### **Parameters**

in packet: the packet to extract
----------------------------------

Returns

The error code that occurred.

See also

Macros

Definition at line 401 of file serverAPI.c.

32 Module Documentation

```
4.7.2.6 uint8_t extract_unlock_req ( msg * packet, int16_t * CID )
```

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

**Author** 

Michel Schmidt

### **Parameters**

in	packet	: the packet to extract
out	CID	: the client ID

### Returns

The error code that occurred.

See also

Macros

Definition at line 369 of file serverAPI.c.

```
4.7.2.7 int init_server ( )
```

Initiates the lib with the permanent server data.

**Author** 

Michel Schmidt

Returns

Error or SUCCESS

TODO: initialized = 0; -> sollte man vielleicht hier auch tun?

Definition at line 19 of file serverAPI.c.

4.7.2.8 uint8\_t send\_brdcst\_rsp ( uint32\_t target\_client\_ip, uint16\_t target\_client\_port )

Send a broadcast response.

Author

Michel Schmidt

4.7 Server Functions 33

### **Parameters**

in	target_client_ip	: the IP address of the target client
in	target_client_port	: the Port of the target client

### Returns

The error code that occurred.

### See also

Macros

Definition at line 164 of file serverAPI.c.

4.7.2.9 uint8\_t send\_dec\_rsp ( uint16\_t *BID*, int16\_t *clientID*, uint8\_t \* *data*, uint32\_t *data\_len*, uint32\_t *target\_client\_ip*, uint16\_t *target\_client\_port* )

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

### Author

Simon Lauser

# **Parameters**

in	BID	: The Block ID of this Block
in	clientID	: The Client ID of the requesting client
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
in	target_client_port	: the Port of the target client

# Returns

The error code that occurred.

# See also

### Macros

Definition at line 95 of file serverAPI.c.

4.7.2.10 uint8\_t send\_error\_rsp ( uint8\_t err\_code, uint32\_t BID, FID fid, uint32\_t target\_client\_ip, uint16\_t target\_client\_port )

Send an error message.

# Author

Michel Schmidt

34 Module Documentation

### **Parameters**

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

### Returns

The error code that occurred during excecution.

### See also

Macros

FID

Definition at line 232 of file serverAPI.c.

4.7.2.11 uint8\_t send\_gp\_rsp ( uint32\_t target\_client\_ip, uint16\_t target\_client\_port )

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

### Author

Simon Lauser

### **Parameters**

in	target_client_ip	: the IP address of the target client
in	target_client_port	: the Port of the target client

# Returns

The error code that occurred.

### See also

# Macros

Definition at line 68 of file serverAPI.c.

4.7.2.12 uint8\_t send\_status\_rsp ( int16\_t CID, uint32\_t sequence\_number, uint32\_t target\_status\_ip, uint16\_t target\_client\_port )

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

#### **Author**

Michel Schmidt

4.7 Server Functions 35

### **Parameters**

in	CID	: The client ID
in	sequence_number	: The sequence number for the current client
in	target_status_ip	: The IP address of the status script
in	target_client_port	: the Port of the target client

# Returns

The error code that occurred.

### See also

Macros

Definition at line 191 of file serverAPI.c.

4.7.2.13 uint8\_t send\_unlock\_rsp ( uint32\_t target\_client\_ip, uint16\_t target\_client\_port )

Send the unlock confirmation.

### **Author**

Michel Schmidt

# **Parameters**

in	target_client_ip	: the IP address of the target client
in	target_client_port	: the Port of the target client

# Returns

The error code that occurred.

# See also

Macros

Definition at line 137 of file serverAPI.c.

36 Module Documentation

# **Chapter 5**

# **Data Structure Documentation**

# 5.1 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)

# 5.1.1 Detailed Description

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

See also

dat\_polynom\_request

Definition at line 64 of file PacketLib.h.

# 5.1.2 Field Documentation

5.1.2.1 uint16\_t dat\_decrypt\_request::blockID

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

Definition at line 67 of file PacketLib.h.

5.1.2.2 int16\_t dat\_decrypt\_request::clientID

The ID of the requesting client.

Definition at line 66 of file PacketLib.h.

5.1.2.3 uint16\_t dat\_decrypt\_request::firstElement

First element of the data structure (16 Bit Chunks)

Definition at line 68 of file PacketLib.h.

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

· PacketLib.h

# 5.2 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)

# 5.2.1 Detailed Description

Return decrypted data Returns the data from successfull decryption.

Definition at line 73 of file PacketLib.h.

### 5.2.2 Field Documentation

5.2.2.1 uint16\_t dat\_decrypt\_response::blockID

The Block ID set by the client.

Definition at line 76 of file PacketLib.h.

5.2.2.2 int16\_t dat\_decrypt\_response::clientID

The ID of the requesting client.

Definition at line 75 of file PacketLib.h.

5.2.2.3 uint8\_t dat\_decrypt\_response::firstElement

First element of the data structure (8 Bit Chunks)

Definition at line 77 of file PacketLib.h.

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

· PacketLib.h

# 5.3 dat\_gp\_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.

### 5.3.1 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 56 of file PacketLib.h.

### 5.3.2 Field Documentation

5.3.2.1 int16\_t dat\_gp\_request::clientID

The ID of the requesting client.

Definition at line 58 of file PacketLib.h.

5.3.2.2 uint16\_t dat\_gp\_request::generator

The generator polynome.

Definition at line 59 of file PacketLib.h.

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

PacketLib.h

# 5.4 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.

# 5.4.1 Detailed Description

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

Definition at line 91 of file PacketLib.h.

### 5.4.2 Field Documentation

5.4.2.1 int16\_t dat\_status\_response::clientID

The ID of the currently connected client.

Definition at line 93 of file PacketLib.h.

5.4.2.2 uint16\_t dat\_status\_response::reserved

Reserved.

See also

VALUE\_RESERVED

Definition at line 94 of file PacketLib.h.

5.4.2.3 uint32\_t dat\_status\_response::wordCount

Amount of Decrypted data words for this client.

Definition at line 95 of file PacketLib.h.

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

PacketLib.h

# 5.5 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.

# 5.5.1 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 83 of file PacketLib.h.

# 5.5.2 Field Documentation

5.5.2.1 int16\_t dat\_unlock\_request::clientID

The ID of the current client.

Definition at line 85 of file PacketLib.h.

5.5.2.2 uint16\_t dat\_unlock\_request::reserved

Reserved.

See also

VALUE RESERVED

Definition at line 86 of file PacketLib.h.

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

· PacketLib.h

# 5.6 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.

### 5.6.1 Detailed Description

Error frame An error message frame.

See also

Macros

Definition at line 101 of file PacketLib.h.

# 5.6.2 Field Documentation

5.6.2.1 uint16\_t error::blockID

Block ID where the error occurred (for ERR\_DECRYPT and ERR\_SERVERINUSE). Else 0.

Definition at line 104 of file PacketLib.h.

5.6.2.2 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 103 of file PacketLib.h.

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

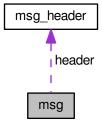
· PacketLib.h

# 5.7 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.

# 5.7.1 Detailed Description

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

Definition at line 109 of file PacketLib.h.

### 5.7.2 Field Documentation

```
5.7.2.1 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 112 of file PacketLib.h.

```
5.7.2.2 msg_header* msg::header
```

A pointer to the header of the structure.

#### See also

```
msg_header
```

Definition at line 111 of file PacketLib.h.

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

• PacketLib.h

# 5.8 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 type:4

The message Type.

• uint8\_t func:4

The called function of this message.

• uint16\_t length

The Length of the message data field.

uint16 t reserved

Reserved.

### 5.8.1 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 43 of file PacketLib.h.

# 5.8.2 Field Documentation

5.8.2.1 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 49 of file PacketLib.h.

5.8.2.2 uint16\_t msg\_header::length

The Length of the message data field.

Definition at line 50 of file PacketLib.h.

5.8.2.3 uint8\_t msg\_header::mode

The mode of the message (sender type)

See also

MODE\_STATUS MODE\_SERVER MODE\_CLIENT

Definition at line 47 of file PacketLib.h.

5.8.2.4 uint8\_t msg\_header::priority

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

Definition at line 45 of file PacketLib.h.

5.8.2.5 uint16\_t msg\_header::reserved

Reserved.

See also

VALUE RESERVED

Definition at line 51 of file PacketLib.h.

5.8.2.6 uint8\_t msg\_header::type

The message Type.

See also

MSG\_REQUEST MSG\_RESPONSE MSG\_ERROR

Definition at line 48 of file PacketLib.h.

5.8.2.7 uint8\_t msg\_header::version

The current version of the script.

See also

PROTOCOL\_VERSION

Definition at line 46 of file PacketLib.h.

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

· PacketLib.h

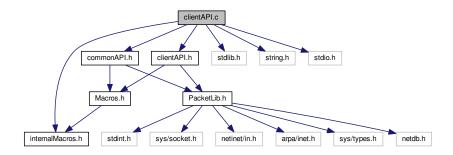
# **Chapter 6**

# **File Documentation**

# 6.1 clientAPI.c File Reference

```
#include "internalMacros.h"
#include "commonAPI.h"
#include "clientAPI.h"
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
```

Include dependency graph for clientAPI.c:



### **Functions**

- int init\_client (int16\_t p\_cID, uint8\_t p\_prio, uint32\_t p\_bca)
  - Initiates the lib with the permanent client data.
- int deinit\_client ()

Deinitializes the client lib.

- uint8\_t send\_gp\_req (uint16\_t gp, uint32\_t target\_server\_ip, uint16\_t target\_server\_port)
  - 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, uint16\_t target\_server\_port)

Send a decryption request Requests the decryption of a block.

uint8\_t send\_unlock\_req (uint32\_t target\_server\_ip, uint16\_t target\_server\_port)

Send an unlock request Unlock a connected server.

uint8\_t send\_brdcst\_req (uint16\_t target\_server\_port)

Send a broadcast request.

uint8\_t extract\_gp\_rsp (msg \*packet)

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)

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

• uint8\_t extract\_unlock\_rsp (msg \*packet)

Extracts the unlock confirmation This extracts the unlock confirmation.

• uint8\_t extract\_brdcst\_rsp (msg \*packet)

This extracts broadcast response.

uint8\_t extract\_error\_rsp (msg \*packet, uint8\_t \*error\_code, uint16\_t \*BID)

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

#### **Variables**

```
• static int16_t clientID = -1
```

- static uint8\_t prio = 255
- static uint32\_t broadcastAddress = 0
- static uint8\_t initialized = 0

# 6.1.1 Variable Documentation

```
6.1.1.1 uint32_t broadcastAddress = 0 [static]
```

Definition at line 18 of file clientAPI.c.

```
6.1.1.2 int16_t clientID = -1 [static]
```

Definition at line 16 of file clientAPI.c.

```
6.1.1.3 uint8_t initialized = 0 [static]
```

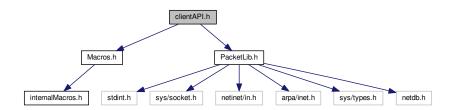
Definition at line 19 of file clientAPI.c.

```
6.1.1.4 uint8_t prio = 255 [static]
```

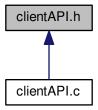
Definition at line 17 of file clientAPI.c.

### 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 (int16\_t p\_cID, uint8\_t p\_prio, uint32\_t p\_bca)

  Initiates the lib with the permanent client data.
- int deinit client ()

Deinitializes the client lib.

• uint8\_t send\_gp\_req (uint16\_t gp, uint32\_t target\_server\_ip, uint16\_t target\_server\_port)

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, uint16\_t target\_server\_port)

Send a decryption request Requests the decryption of a block.

uint8 t send unlock reg (uint32 t target server ip, uint16 t target server port)

Send an unlock request Unlock a connected server.

• uint8\_t send\_brdcst\_req (uint16\_t target\_server\_port)

Send a broadcast request.

uint8 t extract gp rsp (msg \*packet)

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)

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

uint8\_t extract\_unlock\_rsp (msg \*packet)

Extracts the unlock confirmation This extracts the unlock confirmation.

uint8\_t extract\_brdcst\_rsp (msg \*packet)

This extracts broadcast response.

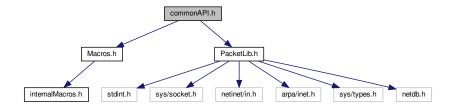
uint8\_t extract\_error\_rsp (msg \*packet, uint8\_t \*error\_code, uint16\_t \*BID)

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

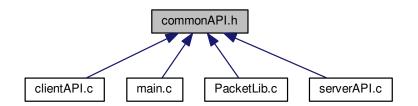
# 6.3 commonAPI.h File Reference

```
#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, uint32\_t \*src\_ip, uint16\_t \*src\_port)

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 free\_msg (msg \*packet)

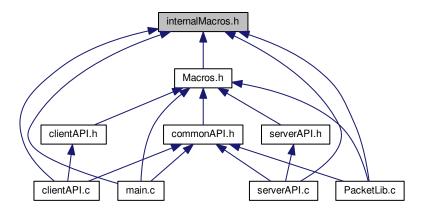
Deletes the subfields of a msg type (But not the msg itself!)

uint8\_t free\_data (uint8\_t \*ptr)

Frees the data stream allocated by extract\_dec\_req / \_rsp.

# 6.4 internal Macros.h File Reference

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



#### **Macros**

• #define SERVER\_PORT 11111

The port on the Server side.

• #define CLIENT\_PORT 11111

The port on the Client side.

• #define BROADCAST\_ADDRESS "127.0.0.1"

The Placeholder for broadcast address.

• #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 SERVER\_PRIO 0

Priority of server messages.

• #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\_GP 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

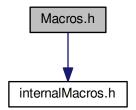
Response to an earlier request.

• #define MSG\_ERROR 15

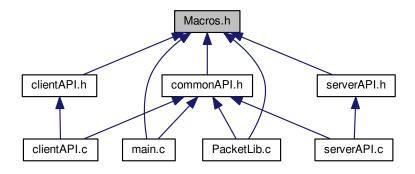
An error occurred decoding or executing.

# 6.5 Macros.h File Reference

#include "internalMacros.h"
Include dependency graph for Macros.h:



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\_LOCK\_TIMEOUT 34

The Server lock timed out.

• #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\_INVALID\_PTR 129

The given pointer was not valid.

• #define ERR NOTFORME 130

Client detected client ID miss match.

• #define ERR NO GP 131

No GP is set in Server.

#define ERR\_SEND\_ERROR 252

Could not send message.

#define ERR\_NO\_INIT 253

The API lib was not initialized.

• #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.

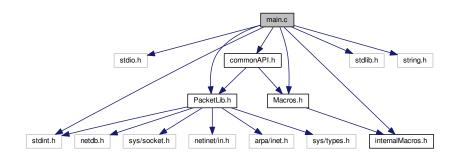
# **Enumerations**

enum FID {
 GP\_REQ, GP\_RSP, DECRYPT\_REQ, DECRYPT\_RSP,
 UNLOCK\_REQ, UNLOCK\_RSP, BROADCAST\_REQ, BROADCAST\_RSP,
 STATUS\_REQ, STATUS\_RSP, UNKNOWN, 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 "commonAPI.h"
Include dependency graph for main.c:
```



# **Functions**

• int main ()

# 6.6.1 Function Documentation

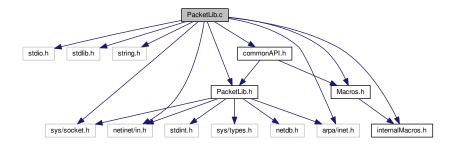
6.6.1.1 int main ( )

Definition at line 19 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\_pointers (msg \*packet)

check\_pointers

uint8\_t check\_packet (msg \*packet)

Check a packet for internal errors.

• FID get msg type (msg \*packet)

 $get\_msg\_type\ This\ function\ returns\ the\ type\ of\ the\ current\ message.$ 

uint8\_t recv\_msg (msg \*packet, uint32\_t \*src\_ip, uint16\_t \*src\_port)

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.

uint8 t send msg (msg \*packet, uint32 t target ip, uint16 t target port)

Sends a message via UDP.

uint8\_t free\_msg (msg \*packet)

Deletes the subfields of a msg type (But not the msg itself!)

uint8\_t free\_data (uint8\_t \*ptr)

Frees the data stream allocated by extract\_dec\_req / \_rsp.

#### **Variables**

- int socketDscp = 0
- struct sockaddr\_in my\_addr
- struct sockaddr\_in target\_addr
- uint8\_t buffer [MAX\_PACKET\_LENGTH]

### 6.7.1 Variable Documentation

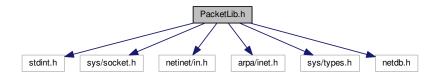
### 6.7.1.1 uint8\_t buffer[MAX\_PACKET\_LENGTH]

Definition at line 25 of file PacketLib.c.

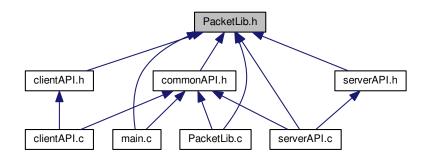
# 6.8 PacketLib.h File Reference

```
#include <stdint.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/types.h>
#include <netdb.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\_gp\_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\_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.

#### **Macros**

- #define DEBUG PRINTF(x)
- #define DEBUG\_PRINTF1(x, y)
- #define DEBUG\_PRINTF2(x, y, z)
- #define DEBUG\_PRINTF3(x, y, z, a)

### **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\_pointers (msg \*packet)

check\_pointers

• uint8\_t check\_packet (msg \*packet)

Check a packet for internal errors.

• uint8\_t send\_msg (msg \*packet, uint32\_t target\_ip, uint16\_t target\_port)

Sends a message via UDP.

### **Variables**

- int socketDscp
- · struct sockaddr in my addr
- struct sockaddr\_in target\_addr
- · 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 type

The message Type.

• uint8 t func

The called function of this message.

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 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 Macro Definition Documentation

```
6.8.1.1 #define DEBUG_PRINTF( x )
```

Definition at line 26 of file PacketLib.h.

6.8.1.2 #define DEBUG\_PRINTF1( x, y)

Definition at line 27 of file PacketLib.h.

6.8.1.3 #define DEBUG\_PRINTF2( x, y, z)

Definition at line 28 of file PacketLib.h.

6.8.1.4 #define DEBUG\_PRINTF3( x, y, z, a)

Definition at line 29 of file PacketLib.h.

# 6.8.2 Variable Documentation

6.8.2.1 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.

```
6.8.2.2 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.

```
6.8.2.3 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.

```
6.8.2.4 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.

```
6.8.2.5 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.
6.8.2.6 uint8_t func
The called function of this message.
See also
     FNC_POLYNOME
     FNC_DECRYPT
     FNC_UNLOCK
     FNC_BROADCAST
     FNC_STATUS
Definition at line 41 of file PacketLib.h.
6.8.2.7 uint16_t generator
The generator polynome.
Definition at line 38 of file PacketLib.h.
6.8.2.8 msg_header* header
A pointer to the header of the structure.
See also
     msg_header
Definition at line 37 of file PacketLib.h.
6.8.2.9 uint16_t length
The Length of the message data field.
```

Definition at line 42 of file PacketLib.h.

```
6.8.2.10 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.

```
6.8.2.11 uint8_t priority
```

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

Definition at line 37 of file PacketLib.h.

6.8.2.12 uint16\_t reserved

Reserved.

See also

```
VALUE_RESERVED
```

Definition at line 43 of file PacketLib.h.

6.8.2.13 uint8\_t type

The message Type.

See also

```
MSG_REQUEST
MSG_RESPONSE
MSG_ERROR
```

Definition at line 40 of file PacketLib.h.

6.8.2.14 uint8\_t version

The current version of the script.

See also

```
PROTOCOL_VERSION
```

Definition at line 38 of file PacketLib.h.

### 6.8.2.15 uint32\_t wordCount

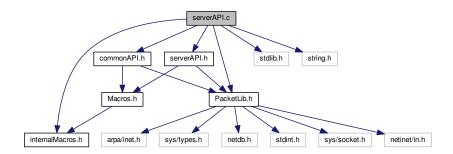
Amount of Decrypted data words for this client.

Definition at line 39 of file PacketLib.h.

### 6.9 serverAPI.c File Reference

```
#include "internalMacros.h"
#include "commonAPI.h"
#include "serverAPI.h"
#include <stdlib.h>
#include <string.h>
#include "PacketLib.h"
```

Include dependency graph for serverAPI.c:



### **Functions**

• int init\_server ()

Initiates the lib with the permanent server data.

• int deinit\_server ()

Deinitializes the server lib.

• uint8\_t send\_gp\_rsp (uint32\_t target\_client\_ip, uint16\_t target\_client\_port)

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

uint8\_t send\_dec\_rsp (uint16\_t BID, int16\_t clientID, uint8\_t \*data, uint32\_t data\_len, uint32\_t target\_client
 —ip, uint16\_t target\_client\_port)

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

• uint8\_t send\_unlock\_rsp (uint32\_t target\_client\_ip, uint16\_t target\_client\_port)

Send the unlock confirmation.

• uint8\_t send\_brdcst\_rsp (uint32\_t target\_client\_ip, uint16\_t target\_client\_port)

Send a broadcast response.

uint8\_t send\_status\_rsp (int16\_t CID, uint32\_t sequence\_number, uint32\_t target\_status\_ip, uint16\_t target
 client port)

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, FID fid, uint32\_t target\_client\_ip, uint16\_t target
 — client\_port)

Send an error message.

- uint8\_t extract\_gp\_req (msg \*packet, uint16\_t \*gp, int16\_t \*CID, uint8\_t \*prio)
  - Extract the generator polynome Extract the generator polynome from the packet.
- uint8\_t extract\_dec\_req (msg \*packet, int16\_t \*CID, uint16\_t \*BID, uint16\_t \*\*data, uint32\_t \*data\_len) Extract data to decrypt.
- uint8\_t extract\_unlock\_req (msg \*packet, int16\_t \*CID)

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

uint8\_t extract\_brdcst\_req (msg \*packet)

Extract a broadcast request.

• uint8\_t extract\_status\_req (msg \*packet)

Extract a status request.

### **Variables**

• static uint8\_t initialized = 0

### 6.9.1 Variable Documentation

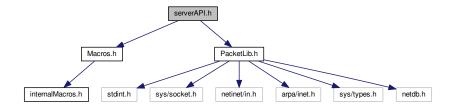
6.9.1.1 uint8\_t initialized = 0 [static]

Definition at line 17 of file serverAPI.c.

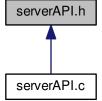
# 6.10 serverAPI.h File Reference

```
#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.

• int deinit\_server ()

Deinitializes the server lib.

• uint8\_t send\_gp\_rsp (uint32\_t target\_client\_ip, uint16\_t target\_client\_port)

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

uint8\_t send\_dec\_rsp (uint16\_t BID, int16\_t clientID, uint8\_t \*data, uint32\_t data\_len, uint32\_t target\_client
 \_ip, uint16\_t target\_client\_port)

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

• uint8\_t send\_unlock\_rsp (uint32\_t target\_client\_ip, uint16\_t target\_client\_port)

Send the unlock confirmation.

uint8\_t send\_brdcst\_rsp (uint32\_t target\_client\_ip, uint16\_t target\_client\_port)

Send a broadcast response.

uint8\_t send\_status\_rsp (int16\_t CID, uint32\_t sequence\_number, uint32\_t target\_status\_ip, uint16\_t target
 — client\_port)

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

uint8\_t send\_error\_rsp (uint8\_t err\_code, uint32\_t BID, FID fid, uint32\_t target\_client\_ip, uint16\_t target\_client\_port)

Send an error message.

• uint8 t extract gp req (msg \*packet, uint16 t \*gp, int16 t \*CID, uint8 t \*prio)

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

• uint8\_t extract\_dec\_req (msg \*packet, int16\_t \*CID, uint16\_t \*BID, uint16\_t \*\*data, uint32\_t \*data\_len)

Extract data to decrypt.

• uint8\_t extract\_unlock\_req (msg \*packet, int16\_t \*CID)

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

• uint8\_t extract\_brdcst\_req (msg \*packet)

Extract a broadcast request.

uint8\_t extract\_status\_req (msg \*packet)

Extract a status request.