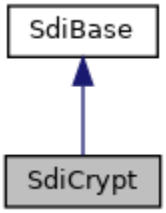
SdiCrypt Class Reference

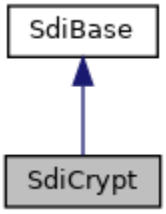
`#include <sdi_if.h>`

Inheritance diagram for SdiCrypt:



[legend]

Collaboration diagram for SdiCrypt:



[legend]

Data Structures	
struct	Placeholder

Public Member Functions	
	SdiCrypt ()
	~SdiCrypt ()
bool	open (const char *hostName)
	call SDI Crypto Open (70-00). More...
bool	close ()
	call SDI Crypto Open Close (70-01). More...
bool	isOpen () const
	Check if a crypto handle is present. More...
uint32_t	getCryptoHandle ()

	Read the crypto handle obtained by SDI Crypto Open. More...
bool	encrypt (const std::vector< unsigned char > &data, std::vector< unsigned char > &encrypted)
	SDI Crypto Encrypt (70-02). More...
bool	decrypt (const std::vector< unsigned char > &encrypted, std::vector< unsigned char > &decrypted)
	SDI Crypto Decrypt (70-03). More...
bool	sign (const std::vector< unsigned char > &data, std::vector< unsigned char > &signature)
	SDI Crypto Sign (70-04). More...
bool	verify (const std::vector< unsigned char > &data, const std::vector< unsigned char > &signature)
	SDI Crypto Verify (70-05). More...
bool	updateKey (unsigned char keyType, const std::vector< unsigned char > &keyData, std::vector< unsigned char > *proprietaryData=NULL, const unsigned char AS2805=0, std::vector< unsigned char > *KCV=NULL)
	SDI Crypto Update Key (70-06). More...
bool	setKeySetId (uint32_t ksid, uint32_t mksid=0, bool asAttribute=false)
	Set Key Set (70-07) or setter for getEncData(), getEncMsgData() and getMsgSignature() depending on parameter `asAttribute`. More...
bool	getEncryptedPin (unsigned char pinBlockFormat, std::vector< unsigned char > &pinBlock, bool requestZeroPinBlock=false)
	Get encrypted PIN (70-08). More...
std::string	getKeyInventory ()
	Get Key Inventory (70-09). More...
bool	getKeyData (unsigned char keyType, std::vector< unsigned char > &keyData, unsigned char kekFlag=0)
	Get Key Data (70-0A). More...

std::string	getStatus ()
	Get Status of security module (70-0B) with handle received by open(). More...
std::string	getStatus (std::string hostName)
	Get Status of security module (70-0B) for given host name or for all security modules. More...
void	setInitialVector (const std::vector< unsigned char > &iv)
	Set Initial Vector for various cipher commands (DFA403) More...
void	getInitialVector (std::vector< unsigned char > &iv) const
	Get Initial Vector returned by various cipher commands. More...
void	getKeySerialNumber (std::vector< unsigned char > &ksn) const
	Get key serial number returned by various cipher commands. More...
bool	getEncData (const Placeholder &descriptor, std::vector< unsigned char > &encrypted, bool useStoredData=false, bool incrementKSN=false)
	Encryption command for card holder sensitive data (29-00). More...
bool	getEncMsgData (const std::vector< unsigned char > &messageTemplate, const std::vector< Placeholder > &placeholder, std::vector< unsigned char > &encrypted, bool useStoredData=false, bool incrementKSN=false)
	Encryption command for card holder sensitive data in host messages (29-01) More...
bool	getMsgSignature (const std::vector< unsigned char > &messageTemplate, const std::vector< Placeholder > &placeholder, std::vector< unsigned char > &signature, bool useStoredData=false, bool incrementKSN=false)
	Singing of host messages (29-04). More...
bool	getEncTrxData (const std::vector< unsigned long > tags, std::vector< unsigned char > &data)
	Perform the command to get encrypted transaction data for later use (29-07). More...

bool	setEncTrxData (const std::vector< unsigned char > data)
	Perform the command to send encrypted transaction data back to the SDI Server (29-08). More...
bool	endEncTrxData ()
	Perform the command to finish the encrypted transaction data procedure inside the SDI Server (29-09). More...
Public Member Functions inherited from SdiBase	
	SdiBase ()
enum SDI_SW12	getSdiSw12 ()
int	getAdditionalResultValue ()
	Access Additional Result Value if returned in SDI response. More...
SDICLIENT_ERROR	getClientError ()
	Access client side error codes. More...
enum SDI_SW12	receiveSW12 ()
	Receive SDI server response with no data. More...
void	clear ()
	clear result data obtained from SDI communication More...
void	importResults (const SdiBase &intermediate)
	set result data obtained from intermediate SDI communication More...

Static Public Member Functions

static std::string	getVersions (int &additionalResult)
	Get versions of SEC ADK components (70-0C). More...

Additional Inherited Members

▸ Protected Member Functions inherited from SdiBase	
void	setSdiSw12 (enum SDI_SW12 s)
void	setClientError (int libsdiprotocol_result)
▸ Protected Attributes inherited from SdiBase	
unsigned short	sw12
int	additionalResultValue
SDICLIENT_ERROR	clientErr

Detailed Description

SdiCrypt holding the crypto handle from Crypto Open. General notes for most functions in this class:

- In case of error the SDI server's SW1 SW2 can be read with getSdiSw12()
- If present, additional error info can be read with getAdditionalResultValue()
- If a Initial Vector is required or received (random mode) it is managed by getInitialVector() and setInitialVector().
- If an KSN has been returned by the SDI server it can be read with getKeySerialNumber()

Data Structure Documentation

◆ libsdi::SdiCrypt::Placeholder

struct libsdi::SdiCrypt::Placeholder

Data descriptor for getEncData(), getEncMsgData() and getMsgSignature()

Data Fields		
vector< unsigned char >	applicationData	data that can be referenced in Placeholder::tagList (DFA120)
vector< unsigned char >	dataOptions	data formatting options (DFA121) - see SDI programmers guide, getEncData (29-00)
vector< unsigned char >	tagList	DOL format with length 0 for variable lengths (DF8F30)

Constructor & Destructor Documentation

◆ SdiCrypt()

SdiCrypt	(		)	

◆ ~SdiCrypt()

~SdiCrypt	(		)	

Member Function Documentation

◆ close()

bool close	(		)	

call SDI Crypto Open Close (70-01).

This funtion is called by SdiCrypt's destructor and open()

Returns

true in case of success

◆ decrypt()

bool decrypt	(	const std::vector< unsigned char > &	*encrypted*,
		std::vector< unsigned char > &	*decrypted*
	)		

SDI Crypto Decrypt (70-03).

Parameters

[in]	encrypted	encrypted data
[out]	decrypted	buffer for decrypted data

Setters

setInitialVector() (DFA403) **Note** Will be sent only once

Returns

true in case of success

◆ encrypt()

bool encrypt	(	const std::vector< unsigned char > &	*data*,
		std::vector< unsigned char > &	*encrypted*
	)		

SDI Crypto Encrypt (70-02).

Parameters

[in]	data	data for encryption
[out]	encrypted	buffer for encrypted data

Setters

setInitialVector() (DFA403) **Note** Will be sent only once

Returns

true in case of success

◆ endEncTrxData()

bool endEncTrxData	(		)	

Perform the command to finish the encrypted transaction data procedure inside the SDI Server (29-09).

Returns

true in case of success

◆ getCryptoHandle()

uint32_t getCryptoHandle	(		)	

Read the crypto handle obtained by SDI Crypto Open.

Intended for commands that are not handled by this object.

Returns
crypto handle, -1 when open() was not called or failed

◆ getEncData()

bool getEncData	(	const Placeholder &	*descriptor*,
		std::vector< unsigned char > &	*encrypted*,
		bool	*useStoredData* = `false`,
		bool	*incrementKSN* = `false`
	)		

Encryption command for card holder sensitive data (29-00).

Parameters

[in]	descriptor	Description of data to be encrypted
[out]	encrypted	Encrypted data
[in]	useStoredData	flag for using stored transaction data
[in]	incrementKSN	activate call to secIncrementKSN()

Setters
setInitialVector() (DFA403) **Note** Will be sent only once

setKeySetId() (DFA409, DFA415)

Returns
true for success

◆ getEncMsgData()

bool getEncMsgData	(	const std::vector< unsigned char > &	*messageTemplate*,
		const std::vector< Placeholder > &	*placeholder*,
		std::vector< unsigned char > &	*encrypted*,

		bool	*useStoredData* = `false`,
		bool	*incrementKSN* = `false`
	)		

Encryption command for card holder sensitive data in host messages (29-01)

Parameters

[in]	messageTemplate	Message Template including place holders for sensitive data elements
[in]	placeholder	Descriptions of data to be encrypted
[out]	encrypted	Encrypted data
[in]	useStoredData	flag for using stored transaction data
[in]	incrementKSN	activate call to secIncrementKSN()

Setters

setInitialVector() (DFA403) **Note** Will be sent only once

setKeySetId() (DFA409, DFA415)

Returns

true for success

◆ getEncryptedPin()

bool getEncryptedPin	(	unsigned char	*pinBlockFormat*,
		std::vector< unsigned char > &	*pinBlock*,
		bool	*requestZeroPinBlock* = `false`
	)		

Get encrypted PIN (70-08).

Parameters

[in]	pinBlockFormat	PIN block format, 0:ISO0, 1:ISO1, 2:ISO2, 3:ISO3

[out]	pinBlock	encrypted PIN
[in]	requestZeroPinBlock	true for zero PIN block

Returns

true in case of success

◆ getEncTrxData()

bool getEncTrxData	(	const std::vector< unsigned long >	*tags*,
		std::vector< unsigned char > &	*data*
	)		

Perform the command to get encrypted transaction data for later use (29-07).

Parameters

[in]	tags	tag list including all requested data elements
[out]	data	encrypted block containing the requested tags with the transaction data

Returns

true in case of success

◆ getInitialVector()

void getInitialVector	(	std::vector< unsigned char > &	*iv*	)	const	inline

Get Initial Vector returned by various cipher commands.

Parameters

[out]	iv	Initial Vector

◆ getKeyData()

bool getKeyData	(	unsigned char	*keyType*,
		std::vector< unsigned char > &	*keyData*,
		unsigned char	*kekFlag* = `0`
	)		

Get Key Data (70-0A).

Parameters

[in]	keyType	Key Type
[out]	keyData	information about key data
[in]	kekFlag	Bendigo KEK Flag 1: KEK1, 2: KEK2

Returns

true in case of success

◆ getKeyInventory()

std::string getKeyInventory	(		)	

Get Key Inventory (70-09).

Returns

json string with information about keys of the opened security module.

◆ getKeySerialNumber()

void getKeySerialNumber	(	std::vector< unsigned char > &	*ksn*	)	const	inline

Get key serial number returned by various cipher commands.

Parameters

[out]	ksn	Key Serial Number

◆ getMsgSignature()

bool getMsgSignature	(	const std::vector< unsigned char > &	*messageTemplate*,
		const std::vector< Placeholder > &	*placeholder*,
		std::vector< unsigned char > &	*signature*,
		bool	*useStoredData* = `false`,
		bool	*incrementKSN* = `false`
	)		

Singing of host messages (29-04).

Parameters

[in]	messageTemplate	Message Template including place holders for sensitive data elements
[in]	placeholder	Descriptions of data to be encrypted
[out]	signature	Signature
[in]	useStoredData	flag for using stored transaction data
[in]	incrementKSN	activate call to secIncrementKSN()

Setters

setInitialVector() (DFA403) **Note** Will be sent only once

setKeySetId() (DFA409, DFA415)

Returns

true for success

◆ getStatus() [1/2]

std::string getStatus	(		)	

Get Status of security module (70-0B) with handle received by open().

Returns
status information as JSON string

◆ getStatus() [2/2]

std::string getStatus	(	std::string	*hostName*	)	

Get Status of security module (70-0B) for given host name or for all security modules.

Parameters

[in]	hostName	configuration name. Use empty string to address all modules.

Returns
status information as JSON string

◆ getVersions()

static std::string getVersions	(	int &	*additionalResult*	)		static

Get versions of SEC ADK components (70-0C).

Parameters

[out]	additionalResult	SDI Additional Result Value in case of error

Returns
version information

◆ isOpen()

bool isOpen	(		)	const

Check if a crypto handle is present.

Returns
false if crypto handle has not been obtain or if close has been called

◆ open()

bool open	(	const char *	*hostName*	)	

call SDI Crypto Open (70-00).

The crypto handle received from SDI server is stored and used inside this object.

Parameters

[in]	hostName	host name

Returns
true in case of success

◆ setEncTrxData()

bool setEncTrxData	(	const std::vector< unsigned char >	*data*	)	

Perform the command to send encrypted transaction data back to the SDI Server (29-08).

Parameters

[in]	data	encrypted block containing the transaction data to be sent to the SDI Server

Returns
true in case of success

◆ setInitialVector()

void setInitialVector	(	const std::vector< unsigned char > &	*iv*	)		inline

Set Initial Vector for various cipher commands (DFA403)

Setter for encrypt(), decrypt(), sign(), verify(), getEncData(), getEncMsgData() and getMsgSignature()

> Sent once only, that is any call to methods listed above will clear the initial vector.

Parameters

[in]	iv	Initial Vector

◆ setKeySetId()

bool setKeySetId	(	uint32_t	*ksid*,
		uint32_t	*mksid* = `0`,
		bool	*asAttribute* = `false`
	)		

Set Key Set (70-07) or setter for getEncData(), getEncMsgData() and getMsgSignature() depending on parameter `asAttribute`.

Parameters

[in]	ksid	Key Set Id (DF409)
[in]	mksid	Master Key Set Id (DFA415)
[in]	asAttribute	key set ids are stored in this object and sent with each getEncData(), getEncMsgData() and getMsgSignature(). Behavior to be reset by providing zero values.

Returns
true in case of success

◆ sign()

bool sign	(	const std::vector< unsigned char > &	*data*,
		std::vector< unsigned char > &	*signature*
	)		

SDI Crypto Sign (70-04).

Parameters

[in]	data	data to sign
[out]	signature	MAC or signature

Setters
setInitialVector() (DFA403) **Note** Will be sent only once

Returns
true in case of success

◆ updateKey()

bool updateKey	(	unsigned char	*keyType*,
		const std::vector< unsigned char > &	*keyData*,
		std::vector< unsigned char > *	*proprietaryData* = `NULL`,
		const unsigned char	*AS2805* = `0`,
		std::vector< unsigned char > *	*KCV* = `NULL`
	)		

SDI Crypto Update Key (70-06).

Parameters

[in]	keyType	SEC ADK key type
[in]	keyData	Key Data or DUKPT Initial Key or 'KSN incrementation'
[in]	proprietaryData	Proprietary Data (e.g. KSN)
[in]	AS2805	AS2805 Tag, possible valid values: 1 ... 255
[in,out]	KCV	Key Check Value

Setters

setInitialVector() (DFA403) **Note** Will be sent only once

Returns

true in case of success

◆ verify()

bool verify	(	const std::vector< unsigned char > &	*data*,
		const std::vector< unsigned char > &	*signature*
	)		

SDI Crypto Verify (70-05).

Parameters

[in]	data	signed data
[in]	signature	MAC or signature

Setters

setInitialVector() (DFA403) **Note** Will be sent only once

Returns

true in case of successful positive verification

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

- sdiclient/client/sdi_if.h