osnma_core

Release 0.0.1

Aleix Galan

Sep 01, 2020

Table of Contents

OSNMACore Class

Module that implements the basic OSNMA actions

class `osnma_core.`**`OSNMACore`** (*svid=1, pubk_path=None*)

Class that handle all the atributes and methods related to the OSNMA protocol. It stores data fields, process messages and perform verification of the different OSNMA structures. It also provides information about the internal stucture of the OSNMA message and auxiliat data such as bitmasks and tables in case the receiver wants to implement them by itself.

`dms_kroot_process` (*dms_kroot, pubk_path=None, nma_header=None, ds_length=None*)

Process the message from the OSNMA DMS-KROOT. Reads and disfragment the fields from the dms_kroot message and then proceeds with the KROOT verification calling self.kroot_verification(). Allows to load custom NMA Header, DS length and pubk path in case they are not already saved in the object.

Parameters

- **`dms_kroot`** (*`BitArray; formated String for bin, oct or hex`*) – Bits from the DMS-KROOT OSNMA message.
- **`pubk_path`** (*`String`*) – Path to the public key that will be used
- **`nma_header`** (*`BitArray; formated String for bin, oct or hex`*) – NMA Header to be used in the verification
- **`ds_length`** (*`int`*) – Length of the DS field in DMS-KROOT message

Returns True if the verification of the KROOT is positive, False otherwise.

Return type bool

`dms_pkr_process` (*dms_pkr*)

Fragment the dms_pkr message in its fields and autenticates the new pkr key calling to self.pkr_verification

Parameters **`dms_pkr`** (*`BitArray; formated String for bin, oct or hex`*) – Raw DMS-PRK message

Returns True if the verification of the Public Key is positive, False otherwise.

Return type bool

`filter_nav_data_by_adkd` (*nav_data, adkd*)

Filters nav_data depending on the adkd parameter. Return all the nav_data to verifiy concatenated.

Parameters

- **`nav_data`** (*`list`*) – List with 15 BitArray objects containing full pages of the sub frame

- **adkd** (*int*) – Authentication Data and Key Delay that indicates the data to authenticate

Returns Concatenated navigation data to be authenticated.

Return type BitArray

format_mack_data (*raw_mack_data*)

Returns a list of lists with the mack blocks being the outer list and each entry in the mack block an entry in the inner list.

Parameters **raw_mack_data** (*BitArray*) – Subframe mack data with keys

Returns List of mack blocks with every mack block being a list of mac entries.

Return type list

get_data (*field_name*, *format=None*)

Get the data of the Field correspondant to the field_name parameter in the desired format. Possible formats: uint, bytes. If no format is specified a BitArray object will be returned.

Parameters

- **field_name** (*string*) – Name of the field.
- **format** (*string*) – Format for the data (None, 'uint' or 'bytes').

Returns Data contained in the Field object

Return type Object

get_description (*field_name*)

Get the field description.

Parameters **field_name** (*string*) – Name of the field.

Returns Description of the Field object.

Return type string

get_field (*field_name*)

Returns the Field object related to the name received as parameter.

Parameters **field_name** – Name of the field to be retrieved.

Type string

Returns Field object that corresponds to the field_name.

Return type Field

get_key_table ( )

Get key table dictionary with the current verified TESLA keys from the keychain.

Returns TESLA verified keys dictionary.

Return type dict

get_meaning (*field_name*)

Get the meaning of the Field checked with its current value. The value of the Field is returned if there is no meaning function associated.

Parameters **field_name** (*string*) – Name of the field.

Returns The meaning of the field or its value.

Return type Object

get_merkle_root ( )

Returns the Merkle root value.

Returns Merkle root value.

Return type BitArray

get_repr (*field_name*)

Get the default way of representing the Field received as parameter.

Parameters **field_name** (*string*) – Name of the field.

Returns Printable object with the default way of representing the object.

Return type string

get_size (*field_name*)

Get the size in bits of the Field correspondant to the field_name parameter.

Parameters **field_name** (*string*) – Name of the field.

Returns Size of the field in bits.

Return type int

kroot_verification (*pub_key=None, hash_name=None*)

Authenticates the saved KROOT with the current Public Key or the path for the one passed as parameter.

Parameters

- **pub_key** (*String*) – Path to the pem file with the pub_key used for the authentication
- **hash_name** (*String*) – OpenSSL hash name to override the loaded one

Returns The result of the verification of KROOT

Return type bool

load (*field_name, data*)

Load data to the OSNMa Field indicated. Also triggers secondary actions related to certain fields that modify the size of other fields or the use of certain funcions.

Parameters

- **field_name** (*String*) – Name of the OSNMA Field
- **data** (*BitArray; formated String for bin, oct or hex*) – Data of the field

load_batch (*data_dict*)

Load a dictionary with OSNMA Fields to the object.

Parameters **data_dict** (*dict*) – Dictionary with key = field_name and data the data for the field

load_floating_key (*index, gst_WN, gst_TOW, key*)

Loads a floating key of the chain to speed up chain authentication

Parameters

- **index** (*int*) – TESLA key index in the current chain.
- **gst_WN** (*BitArray*) – GST Week Number where the key was transmited.
- **gst_TOW** (*BitArray*) – GST Time of the Week where the key was transmited.

- **key** (*BitArray*) – TESLA key

mac0_verification (*mac_entry*, *nav_data*, *key*)

Compute the mac0 verification from it's entry in the first mack block, the navigation data of the subframe and it's correspondant key.

Parameters
- **mac_entry** (*BitArray*) – First MAC entry from the first mack block.
- **nav_data** (*list*) – List with 15 BitArray objects containing full pages of the sub frame
- **key** (*BitArray*) – Key from the first mack block.

Returns Tuple with (bool, computed mac0 tag, received mac0 tag).

Return type tuple

mac_seq_verification (*mack_block*, *key*)

Verify mac seq field of the first mac entry on the first mack block.

Parameters
- **mack_block** (*list*) – List with mac entries as BitArray
- **key** (*BitArray*) – Key of the first mack_block.

Returns Tuple with (bool, computed seq tag, received seq tag).

Return type tuple

mac_verification (*mac_entry*, *nav_data*, *key*, *counter*)

Not implemented yed.

mack_verification (*tesla_keys*, *mack_subframe*, *nav_data*, *gst_wn=None*, *gst_tow=None*)

Authenticates a full MACK message with the correspondant keys. Allows the authentication of past MACK messages with the parameters gst_wn and gst_tow. Note: Current version does not support cross-authentication.

Parameters
- **tesla_keys** (*list*) – List with the tesla keys for the MACK message in the same order as macs.
- **mack_subframe** (*BitArray*) – Raw MACK message to be authenticated in BitArray format.
- **nav_data** (*list*) – Navigation data of current satellite. Sorted in a list of 15 entries (one for each page) in BitArray format.
- **gst_wn** (*BitArray*) – Galileo Satellite Time Week Number to overwrite the current one only for this MACK.
- **gst_tow** (*BitArray*) – Galileo Satellite Time Time of Week to overwrite the current one only for this MACK.

Returns A dictionary with the keys 'mac0' and 'seq' and the values as a tuple (bool, computed_mac, received_mac)

Return type dict

pkr_verification ( )

Craft and authenticates the new public key message with the saved merkle root.

Returns True if the computed Merkle root is the same that the one saved.

Return type bool

`set_merkle_root` (*merkle_root*)

Change the value of the Merkle Tree root node.

Parameters **`merkle_root`** (*`BitArray`*) – The hash correspondant to the root node of the Merkle Tree

`set_size` (*field_name*, *size*)

Modify the size of the field with the field_name received as parameter.

Parameters

- **`field_name`** (*`string`*) – Name of the field.
- **`size`** (*`int`*) – New size of the field.

`tesla_key_verification` (*key*, *gst_wn*, *gst_tow*, *position*, *svid=None*)

Authenticates a TESLA key with the gst_wn and gst_tow from when it has been received. It also needs the position of the key in the mack block. The rest of the necessary data must be uploaded to the object before

Parameters

- **`key`** (*`BitArray`*) – TESLA key to be authenticated
- **`gst_wn`** (*`BitArray`*) – Galileo Satellite Time Week Number of the TESLA key
- **`gst_wn`** – Galileo Satellite Time Time of Week of the TESLA key
- **`position`** (*`int`*) – Position of the TESLA key inside the MACK keys
- **`svid`** – Override the current svid
- **`svid`** – int

Returns Tuple with a bool that indicates if the key has been verified and its key index

Return type tuple

- *Index*
- *Module Index*
- *Search Page*

Python Module Index

Index