
Pycryptoki Documentation

Release 2.2.1

Gemalto

Jul 25, 2019

1	Overview	1
1.1	Getting Started	2
1.1.1	Installation	2
1.1.2	Simple Example	2
1.2	Examples	2
1.2.1	Generating an RSA Key Pair	2
1.2.2	Encrypting data with AES-CBC-PAD	3
1.2.3	Finding a key and decrypting Data	4
1.3	Frequent Issues	5
1.3.1	Wrong data type	6
1.3.2	Internal Initialization Vectors	6
1.3.3	PKCS11 Calling Conventions	7
1.4	API Reference	8
1.4.1	Session/Token Management	8
1.4.1.1	Session Management	8
1.4.1.2	Token Management	17
1.4.2	Key Generation and Management	20
1.4.2.1	Key Generation	20
1.4.2.2	Key Management	23
1.4.2.3	Key Usage	24
1.4.3	Encryption/Decryption	25
1.4.3.1	Encryption	25
1.4.3.2	Decryption	26
1.4.3.3	Key Wrapping/Unwrapping	27
1.4.3.4	Multipart Helper	29
1.4.4	Sign/Verify operations	30
1.4.4.1	Sign	30
1.4.4.2	Verify	31
1.4.5	Attributes and Conversions	32
1.4.5.1	Conversions	34
1.4.6	Mechanisms	35
1.4.6.1	Helpers	36
1.4.6.2	AES Mechanisms	38
1.4.6.3	Generic Mechanisms	39
1.4.6.4	RC Mechanisms	40
1.4.6.5	RSA Mechanisms	41

1.4.7	Miscellaneous	41
1.4.7.1	RNG, Digest, Creating Objects	41
1.4.7.2	Find Objects, Attribute Setting/Getting	45
1.4.7.3	HSM Management	47
1.4.7.4	Audit Functions	55
1.4.7.5	Backup Functions	57
1.4.8	Pycryptoki Helpers	60
1.4.8.1	cryptoki_helpers	61
1.4.8.2	lookup_dicts	61
1.4.8.3	default_templates	61
1.4.8.4	defaults	62
1.4.8.5	cryptoki	62
1.4.9	Pycryptoki Daemon Package	152
1.4.9.1	daemon.rpyc_pycryptoki	152
1.4.9.2	pycryptoki.pycryptoki_client	152
Python Module Index		153
Index		155

CHAPTER 1

Overview

Pycryptoki is an open-source Python wrapper around Safenet's C PKCS11 library. Using python's ctypes library, we can simplify memory management, and provide easy, pythonic access to a PKCS11 shared library.

The primary function of pycryptoki is to *simplify* PKCS11 calls. Rather than needing to calculate data sizes, buffers, or other low-level memory manipulation, you simply need to pass in data.

It's highly recommended that you have the [PKCS11](#) documentation handy, as pycryptoki uses that as the underlying C interface. Session management, object management, and other concepts are unchanged from PKCS11.

```
from pycryptoki.default_templates import *
from pycryptoki.defines import *
from pycryptoki.key_generator import *
from pycryptoki.session_management import *

c_initialize_ex()
auth_session = c_open_session_ex(0)    # HSM slot # in this example is 0
login_ex(auth_session, 0, 'userpin')   # 0 is still the slot number, 'userpin' should
    ↪ be replaced by your password (None if PED or no challenge)

# Get some default templates
# They are simple python dictionaries, and can be modified to suit needs.
pub_template, priv_template = get_default_key_pair_template(CKM_RSA_PKCS_KEY_PAIR_GEN)

# Modifying template would look like:
pub_template[CKA_LABEL] = b"RSA PKCS Pub Key"
pub_template[CKA_MODULUS_BITS] = 2048    # 2048 key size

pubkey, privkey = c_generate_key_pair_ex(auth_session, CKM_RSA_PKCS_KEY_PAIR_GEN, pub_
    ↪ template, priv_template)
print("Generated Private key at %s and Public key at %s" % (privkey, pubkey))

c_logout_ex(auth_session)
c_close_session_ex(auth_session)
c_finalize_ex()
```

1.1 Getting Started

To use pycryptoki, you must have SafeNet LunaClient installed.

1.1.1 Installation

Pycryptoki can be installed on any machine that has Python installed. Python versions ≥ 2.7 are supported.:

```
pip install git+https://github.com/gemalto/pycryptoki
```

Pycryptoki will attempt to auto-locate the SafeNet Cryptoki shared library when pycryptoki is first called. It will use the configuration files as defined by the LunaClient documentation to determine which library to use.

1.1.2 Simple Example

This example will print out information about the given token slot.

```
from pycryptoki.session_management import (c_initialize_ex,
                                           c_get_info_ex,
                                           get_firmware_version,
                                           c_get_token_info_ex,
                                           c_finalize_ex)

c_initialize_ex()
print("C_GetInfo: ")
print("\n".join("\t{}: {}".format(x, y) for x, y in c_get_info_ex().items()))
token_info = c_get_token_info_ex(0)
print("C_GetTokenInfo:")
print("\n".join("\t{}: {}".format(x, y) for x, y in token_info.items()))
print("Firmware version: {}".format(get_firmware_version(0)))

c_finalize_ex()
```

1.2 Examples

1.2.1 Generating an RSA Key Pair

This example creates a 1024b RSA Key Pair.

```
from pycryptoki.session_management import (c_initialize_ex, c_finalize_ex,
                                           c_open_session_ex, c_close_
                                           ↪ session_ex,
                                           login_ex)
from pycryptoki.defines import CKM_RSA_PKCS_KEY_PAIR_GEN
from pycryptoki.key_generator import c_generate_key_pair_ex

c_initialize_ex()
session = c_open_session_ex(0)          # 0 -> slot number
login_ex(session, 0, 'userpin')        # 0 -> Slot number, 'userpin' -> token_
↪ password
```

(continues on next page)

(continued from previous page)

```

# Templates are dictionaries in pycryptoki
pub_template = {CKA_TOKEN: True,
                CKA_PRIVATE: True,
                CKA_MODIFIABLE: True,
                CKA_ENCRYPT: True,
                CKA_VERIFY: True,
                CKA_WRAP: True,
                CKA_MODULUS_BITS: 1024, # long 0 - MAX_RSA_KEY_NBITS
                CKA_PUBLIC_EXPONENT: 3, # byte
                CKA_LABEL: b"RSA Public Key"}
priv_template = {CKA_TOKEN: True,
                 CKA_PRIVATE: True,
                 CKA_SENSITIVE: True,
                 CKA_MODIFIABLE: True,
                 CKA_EXTRACTABLE: True,
                 CKA_DECRYPT: True,
                 CKA_SIGN: True,
                 CKA_UNWRAP: True,
                 CKA_LABEL: b"RSA Private Key"}

pub_key, priv_key = c_generate_key_pair_ex(session,
                                           mechanism=CKM_RSA_PKCS_KEY_PAIR_
↪GEN,
                                           pbkey_template=pub_template,
                                           prkey_template=priv_template)

c_close_session_ex(session)
c_finalize_ex()

```

1.2.2 Encrypting data with AES-CBC-PAD

This example generates a 24-byte AES key, then encrypts some data with that key using the AES-CBC-PAD mechanism.

```

from pycryptoki.session_management import (c_initialize_ex, c_finalize_ex,
                                           c_open_session_ex, c_close_
↪session_ex,
                                           login_ex)
from pycryptoki.defines import (CKM_AES_KEY_GEN,
                                CKA_LABEL,
                                CKA_ENCRYPT,
                                CKA_DECRYPT,
                                CKA_TOKEN,
                                CKA_CLASS,
                                CKA_KEY_TYPE,
                                CKK_AES,
                                CKO_SECRET_KEY,
                                CKA_SENSITIVE,
                                CKA_WRAP,
                                CKA_UNWRAP,
                                CKA_DERIVE,
                                CKA_VALUE_LEN,
                                CKA_EXTRACTABLE,
                                CKA_PRIVATE,

```

(continues on next page)

(continued from previous page)

```

                                CKM_AES_CBC_PAD)
from pycryptoki.key_generator import c_generate_key_ex
from pycryptoki.encryption import c_encrypt_ex
from pycryptoki.conversions import to_bytestring, from_hex
from pycryptoki.mechanism import Mechanism

c_initialize_ex()
session = c_open_session_ex(0)          # 0 = slot number
login_ex(session, 0, 'userpin')        # 'userpin' = token password

template = {CKA_LABEL: b"Sample AES Key",
            CKA_ENCRYPT: True,
            CKA_DECRYPT: True,
            CKA_TOKEN: False,
            CKA_CLASS: CKO_SECRET_KEY,
            CKA_KEY_TYPE: CKK_AES,
            CKA_SENSITIVE: True,
            CKA_PRIVATE: True,
            CKA_WRAP: True,
            CKA_UNWRAP: True,
            CKA_DERIVE: True,
            CKA_VALUE_LEN: 24,
            CKA_EXTRACTABLE: True,}
aes_key = c_generate_key_ex(session, CKM_AES_KEY_GEN, template)

# Data is in hex format here
raw_data = "d0d77c63ab61e75a5fd4719fa77cc2de1d817efedcbd43e7663736007672e8c7"

# Convert to raw bytes before passing into c_encrypt:
data_to_encrypt = to_bytestring(from_hex(raw_data))

# Note: this is *bad crypto practice*! DO NOT USE STATIC IVS!!
mechanism = Mechanism(mech_type=CKM_AES_CBC_PAD,
                      params={"iv": list(range(16))})
static_iv_encrypted_data = c_encrypt_ex(session, aes_key, data_to_encrypt,
    ↪mechanism)

c_close_session_ex(session)
c_finalize_ex()

```

1.2.3 Finding a key and decrypting Data

This example follows from the previous one, except instead of generating a key, we'll find one that was already used.

```

from pycryptoki.session_management import (c_initialize_ex, c_finalize_ex,
                                           c_open_session_ex, c_close_session_ex,
                                           login_ex)
from pycryptoki.object_attr_lookup import c_find_objects_ex
from pycryptoki.defines import (CKM_AES_KEY_GEN,
                                CKA_LABEL,
                                CKA_ENCRYPT,
                                CKA_DECRYPT,
                                CKA_TOKEN,

```

(continues on next page)

(continued from previous page)

```

        CKA_CLASS,
        CKA_KEY_TYPE,
        CKK_AES,
        CKO_SECRET_KEY,
        CKA_SENSITIVE,
        CKA_WRAP,
        CKA_UNWRAP,
        CKA_DERIVE,
        CKA_VALUE_LEN,
        CKA_EXTRACTABLE,
        CKA_PRIVATE,
        CKM_AES_CBC_PAD)
from pycryptoki.encryption import c_decrypt_ex
from pycryptoki.conversions import to_bytestring, from_hex
from pycryptoki.mechanism import Mechanism

c_initialize_ex()
session = c_open_session_ex(0)          # 0 = slot number
login_ex(session, 0, 'userpin')        # 'userpin' = token password

template = {CKA_LABEL: b"Sample AES key"}

keys = c_find_objects_ex(session, template, 1)
aes_key = keys.pop(0) # Use the first key found.

# Data is in hex format here
raw_data =
→ "95e28bc6da451f3064d688dd283c5c43a5dd374cb21064df836e2970e1024c2448f129062aacbae3e45abd098b893346
→ "

# Convert to raw bytes before passing into c_decrypt:
data_to_decrypt = to_bytestring(from_hex(raw_data))

# Note: this is *bad crypto practice*! DO NOT USE STATIC IVS!!
mechanism = Mechanism(mech_type=CKM_AES_CBC_PAD,
                      params={"iv": list(range(16))})
original_data = c_decrypt_ex(session, aes_key, data_to_decrypt, mechanism)

c_close_session_ex(session)
c_finalize_ex()

```

1.3 Frequent Issues

Contents

- *Frequent Issues*
 - *Wrong data type*
 - *Internal Initialization Vectors*
 - *PKCS11 Calling Conventions*

1.3.1 Wrong data type

Any cryptographic function working on data (ex. `c_encrypt`, `c_unwrap`) will expect a bytestring. A string object in Python2 is by default a *bytestring*, but in Python3 is a *unicode* string.

For example:

```
c_encrypt(session, key, "this is some test data", mechanism)
```

Will work in Python 2, but NOT Python 3. Instead, use the `pycryptoki.conversions` module to ensure that any data you pass into the cryptoki library is of the correct form.

Another ‘gotcha’ is that hex data represented as a string that is then used in an encrypt call would result in 2x the length of expected data:

```
from pycryptoki.conversions import to_bytestring, from_hex
hex_data = "deadbeef"
assert len(hex_data) == 8
raw_data = list(from_hex(hex_data))
assert len(raw_data) == 4
print (raw_data)
# Prints: [222, 173, 190, 239]
```

Another example:

```
from pycryptoki.conversions import to_bytestring, from_hex
some_hex_data = "06abde23df89"
data_to_encrypt = to_bytestring(from_hex(some_hex_data))
c_encrypt(session, key, data_to_encrypt, mechanism)
```

Note:

See this article for more details about the differences between unicode and bytestrings in python: <http://lucumr.pocoo.org/2014/1/5/unicode-in-2-and-3/>

1.3.2 Internal Initialization Vectors

When you use an internal IV for AES mechanisms, the IV is appended to the cipher text. This needs to be stripped off and used to create the mechanism for decryption:

```
from pycryptoki.encryption import c_encrypt_ex

data_to_encrypt = b"a" * 64
mech = Mechanism(CKM_AES_KW,
                 params={"iv": []}) # Uses an internal IV

enc_data = c_encrypt_ex(session, key, data_to_encrypt, mech)
iv = enc_data[-16:] # Strip off the last 16 bytes of the encrypted data.
decrypt_mech = Mechanism(CKM_AES_KW,
                        params={"iv": iv})
decrypted_data = c_decrypt_ex(session, key, enc_data[:-16], decrypt_mech)
```

1.3.3 PKCS11 Calling Conventions

The PKCS11 library has two main methods for returning data to the caller:

1. Allocate a large enough buffer for the resulting data and make the PKCS11 call with that buffer.
2. Call the function with a NULL pointer for the buffer. The PKCS11 library will then place the required buffer size in `*pulBufLen`.

Pycryptoki will let you perform either method for any function that returns data in a variable-length buffer with the `output_buffer` keyword argument. This argument takes either an integer, or a list of integers. The integer specifies the *size* of the buffer to use for the returned output. This means if you use a very small integer, you could get back `CKR_BUFFER_TOO_SMALL` (and you could also allocate a buffer that is incredibly large – limited by the memory of your system).

By default, pycryptoki will use method #2 (querying the library for buffer size):

```
data = b"deadbeef"
c_decrypt_ex(session, key, data, mechanism)
```

Will result in the raw underlying PKCS11 calls:

```
DEBUG: Cryptoki call: C_DecryptInit(8, <pycryptoki.cryptoki.CK_MECHANISM object at 0x7f693480c598>, c_ulong(26))
DEBUG: Cryptoki call: C_Decrypt(8, <pycryptoki.cryptoki.LP_c_ubyte object at 0x7f69347df598>, c_ulong(2056), None, <pycryptoki.cryptoki.LP_c_ulong object at 0x7f69347dfbf8>)
DEBUG: Allocating <class 'ctypes.c_ubyte'> buffer of size: 2048
DEBUG: Cryptoki call: C_Decrypt(8, <pycryptoki.cryptoki.LP_c_ubyte object at 0x7f69347df598>, c_ulong(2056), <pycryptoki.cryptoki.LP_c_ubyte object at 0x7f693498c9d8>, <pycryptoki.cryptoki.LP_c_ulong object at 0x7f693498c840>)
```

Note: None in python is the equivalent to NULL in C.

An example using a pre-allocated buffer:

```
data = b"deadbeef"
c_decrypt_ex(session, key, data, mechanism, output_buffer=0xffff)
```

And the resulting PKCS11 calls:

```
DEBUG: Cryptoki call: C_DecryptInit(8, <pycryptoki.cryptoki.CK_MECHANISM object at 0x7f693480c598>, c_ulong(26))
DEBUG: Allocating <class 'ctypes.c_ubyte'> buffer of size: 2048
DEBUG: Cryptoki call: C_Decrypt(8, <pycryptoki.cryptoki.LP_c_ubyte object at 0x7f69347df598>, c_ulong(2056), <pycryptoki.cryptoki.LP_c_ubyte object at 0x7f693498c9d8>, <pycryptoki.cryptoki.LP_c_ulong object at 0x7f693498c840>)
```

For multi-part operations, `output_buffer` should be a list of integers of equal size to the number of parts in the operation:

```
data = [b"a" * 8, b"b" * 8, b"c" * 8, b"d" * 8]
output_buffer = [0xffff] * len(data) # Equivalent to: [0xffff, 0xffff, 0xffff, 0xffff]
c_encrypt_ex(session, key, data, mechanism, output_buffer=output_buffer)
```

For a multi-part operation that returns data in the `C_*Final` function, the output buffer will be equivalent to the largest buffer size specified in the `output_buffer` list.

1.4 API Reference

There are some general guidelines to using pycryptoki:

1. If you want to perform a PKCS11 operation as a multi-part operation, provide the input data as a list or a tuple.
2. Data should always be passed into `c_` functions as raw byte data (bytestrings). Conversions are available to convert hex data or binary data to bytes at [pycryptoki.conversions](#)
3. Returned encrypted/decrypted data is always raw bytestrings.

1.4.1 Session/Token Management

Modules for Token and session creation and management.

1.4.1.1 Session Management

Methods responsible for managing a user's session and login/c_logout

`pycryptoki.session_management.c_close_all_sessions(slot)`

Closes all the sessions on a given slot

Parameters `slot` – The slot to close all sessions on

Returns `retcode`

Return type `int`

`pycryptoki.session_management.c_close_all_sessions_ex(slot)`

Executes `c_close_all_sessions()`, and checks the `retcode`; raising an exception if the return code is not `CKR_OK`.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the `retcode`, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.session_management.c_close_session(h_session)`

Closes a session

Parameters `h_session` (`int`) – Session handle

Returns `retcode`

Return type `int`

`pycryptoki.session_management.c_close_session_ex(h_session)`

Executes `c_close_session()`, and checks the `retcode`; raising an exception if the return code is not `CKR_OK`.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.session_management.c_finalize()`

Finalizes PKCS11 library.

Returns Cryptoki return code

`pycryptoki.session_management.c_finalize_ex()`

Executes `c_finalize()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.session_management.c_get_info()`

Get general information about the Cryptoki Library

Returns a dictionary containing the following keys:

- `cryptokiVersion`
- `manufacturerID`
- `flags`
- `libraryDescription`
- `libraryVersion`

`cryptokiVersion` and `libraryVersion` are `~pycryptoki.cryptoki.CK_VERSION` structs, and the major/minor values can be accessed directly (`info['cryptokiVersion'].major == 2`)

Returns (retcode, info dictionary)

`pycryptoki.session_management.c_get_info_ex()`

Executes `c_get_info()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.session_management.c_get_session_info(session)`

Get information about the given session.

Parameters `session` (*int*) – session handle

Returns (retcode, dictionary of session information)

Return type tuple

`pycryptoki.session_management.c_get_session_info_ex(session)`

Executes `c_get_session_info()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.session_management.c_get_slot_info(slot)`

Get information about the given slot number.

Parameters `slot` (*int*) – Target slot

Returns Dictionary of slot information

`pycryptoki.session_management.c_get_slot_info_ex(slot)`

Executes `c_get_slot_info()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.session_management.c_get_slot_list(token_present=True)`

Get a list of all slots.

Parameters `token_present` (*bool*) – If true, will only return slots that have a token present.

Returns List of slots

`pycryptoki.session_management.c_get_slot_list_ex(token_present=True)`

Executes `c_get_slot_list()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.session_management.c_get_token_info(slot_id, rstrip=True)`

Gets the token info for a given slot id

Parameters

- `slot_id` (*int*) – Token slot ID
- `rstrip` (*bool*) – If true, will strip trailing whitespace from char data.

Returns (retcode, A python dictionary representing the token info)

Return type tuple

`pycryptoki.session_management.c_get_token_info_ex(slot_id, rstrip=True)`

Executes `c_get_token_info()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.session_management.c_init_pin(h_session, pin)`

Initializes the PIN

Parameters

- **h_session** (*int*) – Session handle
- **pin** – pin to c_initialize

Returns The result code

`pycryptoki.session_management.c_init_pin_ex(h_session, pin)`

Executes `c_init_pin()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.session_management.c_initialize(flags=None, init_struct=None)`

Initializes current process for use with PKCS11.

Some sample flags:

CKF_LIBRARY_CANT_CREATE_OS_THREADS CKF_OS_LOCKING_OK

See the [PKCS11 documentation](#) for more details.

Parameters

- **flags** (*int*) – Flags to be set within InitArgs Struct. (Default = None)
- **init_struct** – InitArgs structure (Default = None)

Returns Cryptoki return code.

`pycryptoki.session_management.c_initialize_ex(flags=None, init_struct=None)`

Executes `c_initialize()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```


`pycryptoki.session_management.c_logout(h_session)`

Logs out of a given session

Parameters `h_session` (*int*) – Session handle

Returns retcode

Return type *int*

`pycryptoki.session_management.c_logout_ex(h_session)`

Executes `c_logout()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.session_management.c_open_session(slot_num, flags=6)`

Opens a session on the given slot

Parameters

- `slot_num` (*int*) – The slot to get a session on
- `flags` (*int*) – The flags to open the session with (Default value = (CKF_SERIAL_SESSION | CKF_RW_SESSION))

Returns (retcode, session handle)

Return type tuple

`pycryptoki.session_management.c_open_session_ex(slot_num, flags=6)`

Executes `c_open_session()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.session_management.c_set_pin(h_session, old_pass, new_pass)`

Allows a user to change their PIN

Parameters

- **h_session** (*int*) – Session handle
- **old_pass** – The user’s old password
- **new_pass** – The user’s desired new password

Returns The result code

`pycryptoki.session_management.c_set_pin_ex(h_session, old_pass, new_pass)`

Executes `c_set_pin()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.session_management.ca_closeapplicationID(slot, id_high, id_low)`

Close a given AppID on a slot.

Parameters

- **slot** (*int*) – Slot on which to close the APP ID
- **id_high** (*int*) – High value of App ID
- **id_low** (*int*) – Low value of App ID

Returns retcode

Return type `int`

`pycryptoki.session_management.ca_closeapplicationID_ex(slot, id_high, id_low)`

Executes `ca_closeapplicationID()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.session_management.ca_factory_reset(slot)`

Does a factory reset on a given slot

Parameters `slot` – The slot to do a factory reset on

Returns The result code

`pycryptoki.session_management.ca_factory_reset_ex(slot)`

Executes `ca_factory_reset()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.session_management.ca_openapplicationID(slot, id_high, id_low)`

Open an application ID on the given slot.

Parameters

- `slot` (*int*) – Slot on which to open the APP ID
- `id_high` (*int*) – High value of App ID
- `id_low` (*int*) – Low value of App ID

Returns retcode

Return type `int`

`pycryptoki.session_management.ca_openapplicationID_ex(slot, id_high, id_low)`

Executes `ca_openapplicationID()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.session_management.ca_restart(slot)`

Parameters `slot` –

`pycryptoki.session_management.ca_restart_ex(slot)`

Executes `ca_restart()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.session_management.ca_setapplicationID(id_high, id_low)`

Set the App ID for the current process.

Parameters

- **id_high** (*int*) – High value of App ID
- **id_low** (*int*) – Low value of App ID

Returns retcode

Return type *int*

`pycryptoki.session_management.ca_setapplicationID_ex(id_high, id_low)`

Executes `ca_setapplicationID()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.session_management.get_firmware_version(slot)`

Returns a string representing the firmware version of the given slot.

It will first try to call `CA_GetFirmwareVersion`, and if that fails (not present on older cryptoki libraries), will call `C_GetTokenInfo`.

Parameters **slot** (*int*) – Token slot number

Returns Firmware String in the format “X.Y.Z”, where X is major, Y is minor, Z is subminor.

Return type *str*

`pycryptoki.session_management.get_slot_dict(token_present=False)`

Compiles a dictionary of the available slots

Returns A python dictionary of the available slots

`pycryptoki.session_management.get_slot_dict_ex(token_present=False)`

Executes `get_slot_dict()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.session_management.login(h_session, slot_num=1, password=None, user_type=1)`

Login to the given session.

Parameters

- **h_session** (*int*) – Session handle
- **slot_num** (*int*) – Slot index to login on (Default value = 1)
- **password** (*bytes*) – Password to login with (Default value = “userpin”)
- **user_type** (*int*) – User type to login as (Default value = 1)

Returns retcode

Return type `int`

`pycryptoki.session_management.login_ex(h_session, slot_num=1, password=None, user_type=1)`

Executes `login()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

1.4.1.2 Token Management

Created on Aug 24, 2012

@author: mhughes

`pycryptoki.token_management.c_get_mechanism_info(slot, mechanism_type)`

Gets a mechanism’s info

Parameters

- **slot** – The slot to query
- **mechanism_type** – The type of the mechanism to get the information for

Returns The result code, The mechanism info

`pycryptoki.token_management.c_get_mechanism_info_ex(slot, mechanism_type)`

Executes `c_get_mechanism_info()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.token_management.c_get_mechanism_list(slot)`

Gets the list of mechanisms from the HSM

Parameters **slot** – The slot number to get the mechanism list on

Returns The result code, A python dictionary representing the mechanism list

`pycryptoki.token_management.c_get_mechanism_list_ex(slot)`

Executes `c_get_mechanism_list()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.token_management.c_init_token(slot_num, password, token_label='Main Token')`

Initializes at token at a given slot with the proper password and label

Parameters

- **slot_num** – The index of the slot to c_initialize a token in
- **password** – The password to c_initialize the slot with
- **token_label** – The label to c_initialize the slot with (Default value = 'Main Token')

Returns The result code

`pycryptoki.token_management.c_init_token_ex(slot_num, password, token_label='Main Token')`

Executes `c_init_token()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.token_management.ca_get_token_policies(slot)`

Get the policies of the given slot.

Parameters `slot` (*int*) – Target slot number

Returns retcode, {id: val} dict of policies (None if command failed)

`pycryptoki.token_management.ca_get_token_policies_ex(slot)`

Executes `ca_get_token_policies()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.token_management.get_token_by_label(label)`

Iterates through all the tokens and returns the first token that has a label that is identical to the one that is passed in

Parameters `label` – The label of the token to search for

Returns The result code, The slot of the token

`pycryptoki.token_management.get_token_by_label_ex(label)`

Executes `get_token_by_label()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

1.4.2 Key Generation and Management

1.4.2.1 Key Generation

Methods used to generate keys.

`pycryptoki.key_generator.c_copy_object(h_session, h_object, template=None)`

Method to call the C_CopyObject cryptoki command.

Parameters

- **h_session** (*int*) – Session handle
- **h_object** (*int*) – Handle to the object to be cloned
- **template** (*dict*) – Template for the new object. Defaults to None

Returns (retcode, Handle to the new cloned object)

Return type tuple

`pycryptoki.key_generator.c_copy_object_ex(h_session, h_object, template=None)`

Executes `c_copy_object()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.key_generator.c_derive_key(h_session, h_base_key, template, mechanism=None)`

Derives a key from another key.

Parameters

- **h_session** (*int*) – Session handle
- **h_base_key** (*int*) – The base key
- **template** (*dict*) – A python template of attributes to set on derived key

- **mechanism** – See the `parse_mechanism()` function for possible values.

Returns The result code, The derived key's handle

`pycryptoki.key_generator.c_derive_key_ex(h_session, h_base_key, template, mechanism=None)`

Executes `c_derive_key()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.key_generator.c_destroy_object(h_session, h_object_value)`

Deletes the object corresponding to the passed in object handle

Parameters

- **h_session** (*int*) – Session handle
- **h_object_value** (*int*) – The handle of the object to delete

Returns Return code

`pycryptoki.key_generator.c_destroy_object_ex(h_session, h_object_value)`

Executes `c_destroy_object()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.key_generator.c_generate_key(h_session, mechanism=None, template=None)`

Generates a symmetric key of a given flavor given the correct template.

Parameters

- **h_session** (*int*) – Session handle
- **template** (*dict*) – The template to use to generate the key
- **mechanism** – See the `parse_mechanism()` function for possible values.

Returns (retcode, generated key handle)

Rtype tuple

`pycryptoki.key_generator.c_generate_key_ex(h_session, mechanism=None, template=None)`

Executes `c_generate_key()`, and checks the retcode; raising an exception if the return code is not `CKR_OK`.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.key_generator.c_generate_key_pair(h_session, mechanism=None, pbkey_template=None, prkey_template=None)`

Generates a private and public key pair for a given flavor, and given public and private key templates. The return value will be the handle for the key.

Parameters

- **h_session** (*int*) – Session handle
- **pbkey_template** (*dict*) – The public key template to use for key generation
- **prkey_template** (*dict*) – The private key template to use for key generation
- **mechanism** – See the `parse_mechanism()` function for possible values.

Returns (retcode, public key handle, private key handle)

Return type tuple

`pycryptoki.key_generator.c_generate_key_pair_ex(h_session, mechanism=None, pbkey_template=None, prkey_template=None)`

Executes `c_generate_key_pair()`, and checks the retcode; raising an exception if the return code is not `CKR_OK`.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

```
pycryptoki.key_generator.clear_keys(h_session)
```

Quick hacked together function that can be used to clear the first 10 000 keys.

Parameters `h_session` (*int*) – Session handle

1.4.2.2 Key Management

Methods responsible for key management

```
pycryptoki.key_management.ca_generatemofn(h_session, m_value, vector_value, vector_count,
                                           is_secure_port_used)
```

Generates MofN secret information on a token.

Parameters

- `h_session` (*int*) – Session handle
- `m_value` – m
- `vector_count` – number of vectors
- `is_secure_port_used` – is secure port used
- `vector_value` –

Returns the result code

```
pycryptoki.key_management.ca_generatemofn_ex(h_session, m_value, vector_value, vector_count, is_secure_port_used)
```

Executes `ca_generatemofn()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

```
pycryptoki.key_management.ca_modifyusagecount(h_session, h_object, command_type, value)
```

Modifies CKA_USAGE_COUNT attribute of the object.

Parameters

- `h_session` (*int*) – Session handle
- `h_object` – object
- `command_type` – command type
- `value` – value

Returns the result code

`pycryptoki.key_management.ca_modifyusagecount_ex(h_session, h_object, command_type, value)`

Executes `ca_modifyusagecount()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

1.4.2.3 Key Usage

Methods responsible for key usage

`pycryptoki.key_usage.ca_clonemofn(h_session)`

Clones MofN secret from one token to another.

Parameters `h_session` (*int*) – Session handle

Returns the result code

`pycryptoki.key_usage.ca_clonemofn_ex(h_session)`

Executes `ca_clonemofn()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.key_usage.ca_duplicatemofn(h_session)`

Duplicates a set of M of N vectors.

Parameters `h_session` (*int*) – Session handle

Returns the result code

`pycryptoki.key_usage.ca_duplicatemofn_ex(h_session)`

Executes `ca_duplicatemofn()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

1.4.3 Encryption/Decryption

Contents

- *Encryption/Decryption*
 - *Encryption*
 - *Decryption*
 - *Key Wrapping/Unwrapping*
 - *Multipart Helper*

1.4.3.1 Encryption

`pycryptoki.encryption.c_encrypt(h_session, h_key, data, mechanism, output_buffer=None)`

Encrypts data with a given key and encryption flavor encryption flavors

Note: If data is a list or tuple of strings, multi-part encryption will be used.

Parameters

- **h_session** (*int*) – Current session
- **h_key** (*int*) – The key handle to encrypt the data with
- **data** – The data to encrypt, either a bytestring or a list of bytestrings. If this is a list a multipart operation will be used

Note: This will be converted to hexadecimal by calling:

```
to_hex(from_bytestring(data))
```

If you need to pass in raw hex data, call:

```
to_bytestring(from_hex(hex-data))
```

References:

- `to_hex()`
 - `from_hex()`
 - `to_bytestring()`
 - `from_bytestring()`
-

- **mechanism** – See the `parse_mechanism()` function for possible values.
- **output_buffer** (*list/int*) – Integer or list of integers that specify a size of output buffer to use for an operation. By default will query with NULL pointer buffer to get required size of buffer.

Returns (Retcode, Python bytestring of encrypted data)

Return type tuple

`pycryptoki.encryption.c_encrypt_ex(h_session, h_key, data, mechanism, output_buffer=None)`
 Executes `c_encrypt()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

1.4.3.2 Decryption

`pycryptoki.encryption.c_decrypt(h_session, h_key, encrypted_data, mechanism, output_buffer=None)`
 Decrypt given data with the given key and mechanism.

Note: If data is a list or tuple of strings, multi-part decryption will be used.

Parameters

- **h_session** (*int*) – The session to use
- **h_key** (*int*) – The handle of the key to use to decrypt
- **encrypted_data** (*bytes*) – Data to be decrypted

Note: Data will be converted to hexadecimal by calling:

```
to_hex(from_bytestring(data))
```

If you need to pass in raw hex data, call:

```
to_bytestring(from_hex(hex-data))
```

References:

- `to_hex()`
- `from_hex()`
- `to_bytestring()`
- `from_bytestring()`

- **mechanism** – See the `parse_mechanism()` function for possible values.
- **output_buffer** (*list/int*) – Integer or list of integers that specify a size of output buffer to use for an operation. By default will query with NULL pointer buffer to get required size of buffer.

Returns (Retcode, Python bytestring of decrypted data))

Return type tuple

`pycryptoki.encryption.c_decrypt_ex(h_session, h_key, encrypted_data, mechanism, output_buffer=None)`

Executes `c_decrypt()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

1.4.3.3 Key Wrapping/Unwrapping

`pycryptoki.encryption.c_wrap_key(h_session, h_wrapping_key, h_key, mechanism, output_buffer=None)`

Wrap a key off the HSM into an encrypted data blob.

Parameters

- **h_session** (*int*) – The session to use
- **h_wrapping_key** (*int*) – The handle of the key to use to wrap another key
- **h_key** (*int*) – The key to wrap based on the encryption flavor

- **mechanism** – See the `parse_mechanism()` function for possible values.

Returns (Retcode, python bytestring representing wrapped key)

Return type tuple

`pycryptoki.encryption.c_wrap_key_ex(h_session, h_wrapping_key, h_key, mechanism, output_buffer=None)`

Executes `c_wrap_key()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.encryption.c_unwrap_key(h_session, h_unwrapping_key, wrapped_key, key_template, mechanism)`

Unwrap a key from an encrypted data blob.

Parameters

- **h_session** (*int*) – The session to use
- **h_unwrapping_key** (*int*) – The wrapping key handle
- **wrapped_key** (*bytes*) – The wrapped key

Note: Data will be converted to hexadecimal by calling:

```
to_hex(from_bytestring(data))
```

If you need to pass in raw hex data, call:

```
to_bytestring(from_hex(hex-data))
```

References:

- `to_hex()`
- `from_hex()`
- `to_bytestring()`
- `from_bytestring()`

-
- **key_template** (*dict*) – The python template representing the new key's template
 - **mechanism** – See the `parse_mechanism()` function for possible values.

Returns (Retcode, unwrapped key handle)

Return type tuple

`pycryptoki.encryption.c_unwrap_key_ex(h_session, h_unwrapping_key, wrapped_key, key_template, mechanism)`

Executes `c_unwrap_key()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

1.4.3.4 Multipart Helper

`pycryptoki.encryption.do_multipart_operation(h_session, c_update_function, c_finalize_function, input_data_list, output_buffer=None)`

Some code which will do a multipart encrypt or decrypt since they are the same with just different functions called

Parameters

- **h_session** (*int*) – Session handle
- **c_update_function** – C_<NAME>Update function to call to update each operation.
- **c_finalize_function** – Function to call at end of multipart operation.
- **input_data_list** – List of data to call update function on.

Note: Data will be converted to hexadecimal by calling:

```
to_hex(from_bytestring(data))
```

If you need to pass in raw hex data, call:

```
to_bytestring(from_hex(hex-data))
```

References:

- `to_hex()`
- `from_hex()`
- `to_bytestring()`
- `from_bytestring()`

- **output_buffer** (*list*) – List of integers that specify a size of output buffers to use for multi-part operations. By default will query with NULL pointer buffer to get required size of buffer

1.4.4 Sign/Verify operations

Contents

- *Sign/Verify operations*
 - *Sign*
 - *Verify*

1.4.4.1 Sign

`pycryptoki.sign_verify.c_sign(h_session, h_key, data_to_sign, mechanism, out-put_buffer=None)`
Signs the given data with given key and mechanism.

Note: If data is a list or tuple of strings, multi-part operations will be used.

Parameters

- **h_session** (*int*) – Session handle
- **data_to_sign** – The data to sign, either a string or a list of strings. If this is a list a multipart operation will be used (using **C_...Update** and **C_...Final**)
ex:
 - "This is a proper argument of some data to use in the function"
 - ["This is another format of data this", "function will accept.", "It will operate on these strings in parts"]
- **h_key** (*int*) – The signing key
- **mechanism** – See the `parse_mechanism()` function for possible values.
- **output_buffer** (*list/int*) – Integer or list of integers that specify a size of output buffer to use for an operation. By default will query with NULL pointer buffer to get required size of buffer.

Returns (retcode, python string of signed data)

Return type tuple

`pycryptoki.sign_verify.c_sign_ex(h_session, h_key, data_to_sign, mechanism, output_buffer=None)`
Executes `c_sign()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

1.4.4.2 Verify

`pycryptoki.sign_verify.c_verify(h_session, h_key, data_to_verify, signature, mechanism)`

Verifies data with the given signature, key and mechanism.

Note: If data is a list or tuple of strings, multi-part operations will be used.

Parameters

- **h_session** (*int*) – Session handle
- **data_to_verify** – The data to sign, either a string or a list of strings. If this is a list a multipart operation will be used (using **C_...Update** and **C_...Final**)
ex:
– "This is a proper argument of some data to use in the function"
– ["This is another format of data this", "function will accept.", "It will operate on these strings in parts"]
- **signature** (*bytes*) – Signature with which to verify the data.
- **h_key** (*int*) – The verifying key
- **mechanism** – See the `parse_mechanism()` function for possible values.

Returns retcode of verify operation

`pycryptoki.sign_verify.c_verify_ex(h_session, h_key, data_to_verify, signature, mechanism)`

Executes `c_verify()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.sign_verify.do_multipart_verify(h_session, input_data_list, signature)`

Do a multipart verify operation

Parameters

- **h_session** (*int*) – Session handle
- **input_data_list** – list of data to verify with

- **signature** – signature to verify

Returns The result code

1.4.5 Attributes and Conversions

This module contains a wrapper around the key attributes and the template struct generation to make it possible to create templates in python and easily convert them into templates in C.

`pycryptoki.attributes.KEY_TRANSFORMS` `CK_ATTRIBUTE` Types mapped to Python->C transformation

class `pycryptoki.attributes.Attributes` (*args, **kwargs)

Python container for handling PKCS11 Attributes.

Provides `get_c_struct()`, that would returns a list of C Structs, each with the following structure:

```
class CK_ATTRIBUTE (Structure):
    '''
    Defines type, value and length of an attribute:

    c_ulong type;
    c_void_p pValue;
    c_ulong ulValueLen;
    '''
    pass
```

This list of structs can be used with `C_GetAttributeValue()` to get the length of the value that will be placed in `pValue` (will be set to `ulValueLen`), or if you already know the length required you can 'blank fill' `pValue` for direct use.

You can also provide new transformations in the form of a dictionary that will be preferred to the `KEY_TRANSFORMS` dictionary. This is passed in only as a keyword argument:

```
transform = {1L: lambda x: return x**2}`
attrs = Attributes({...}, new_transforms=transform)
# attrs.get_c_struct will use the lambda expression in the transform dictionary
# for key 1L
```

static `from_c_struct` (c_struct)

Build out a dictionary from a `c_struct`.

Parameters `c_struct` – Pointer to an array of `CK_ATTRIBUTE` structs

Returns dict

get_c_struct ()

Build an array of `CK_ATTRIBUTE` Structs & return it.

Returns `CK_ATTRIBUTE` array

`pycryptoki.attributes.c_struct_to_python` (c_struct)

Converts a C struct to a python dictionary.

Parameters `c_struct` – The c struct to convert into a dictionary in python

Returns Returns a python dictionary which represents the C struct passed in

`pycryptoki.attributes.convert_c_ubyte_array_to_string` (byte_array)

Converts a ctypes unsigned byte array into a string.

Parameters `byte_array` –

`pycryptoki.attributes.ret_type(c_type)`

Decorator to set a returned C Type so we can determine what type to use for an AutoCArray

Parameters `c_type` – Default return-type of the transform function.

`pycryptoki.attributes.to_bool(val, reverse=False)`

Convert a boolean-ish value to a pValue, ulValueLen tuple.

Parameters

- **val** – Value to convert
- **reverse** – Whether to convert from C -> Python

Returns (`ctypes.c_void_p` ptr to `pycryptoki.cryptoki.CK_BBOOL`,
`ctypes.c_ulong` size of bool value)

`pycryptoki.attributes.to_byte_array(val, reverse=False)`

Converts an arbitrarily sized integer, list, or byte array into a byte array.

It'll zero-pad the bit length so it's a multiple of 8, then convert the int to binary, split the binary string into sections of 8, then place each section into a slot in a `ctypes.c_ubyte` array (converting to small int).

Parameters

- **val** – Value to convert
- **reverse** – Whether to convert from C -> Python

Returns (`ctypes.c_void_p` ptr to `pycryptoki.cryptoki.CK_BYTE` array,
`ctypes.c_ulong` size of array)

`pycryptoki.attributes.to_char_array(val, reverse=False)`

Convert the given string or list of string values into a char array.

This is slightly different than `to_byte_array`, which has different assumptions as to the format of the input.

Parameters

- **val** – Value to convert
- **reverse** – Whether to convert from C -> Python

Returns (`ctypes.c_void_p` ptr to `pycryptoki.cryptoki.CK_CHAR` array,
`ctypes.c_ulong` size of array)

`pycryptoki.attributes.to_ck_date(val, reverse=False)`

Transform a date string, date dictionary, or date object into a PKCS11 readable form (YYYYMMDD)

Parameters

- **val** – Value to convert
- **reverse** – Whether to convert from C -> Python

Returns (`ctypes.c_void_p` ptr to `pycryptoki.cryptoki.CK_CHAR` array,
`ctypes.c_ulong` size of array)

`pycryptoki.attributes.to_long(val, reverse=False)`

Convert a integer/long value to a pValue, ulValueLen tuple

Parameters

- **val** – Value to convert

- **reverse** – Whether to convert from C -> Python

Returns (`ctypes.c_void_p` ptr to `ctypes.c_ulong`, `ctypes.c_ulong` size of long value)

`pycryptoki.attributes.to_pka_key_status(val, reverse=False)`

Transform a Per Key Authorization Key Status object into a PKCS11 readable byte string

Parameters

- **val** – Value to convert
- **reverse** – Whether to convert from C -> Python

Returns (`ctypes.c_void_p` ptr to `pycryptoki.cryptoki.CK_KEY_STATUS` object, `ctypes.c_ulong` size of array)

`pycryptoki.attributes.to_sub_attributes(val, reverse=False)`

Convert to another Attributes class & return the struct.

Parameters

- **val** – Value to convert
- **reverse** – Whether to convert from C -> Python

Returns (`ctypes.c_void_p` ptr to `pycryptoki.cryptoki.CK_ATTRIBUTE` array, `ctypes.c_ulong` size of array)

1.4.5.1 Conversions

Provide low-level conversions between common data types.

The `from_xyz` functions should all return an iterator over a list of integers, representing the individual bytes in the passed-in value.

The `to_xyz` functions take in an iterable of integers and convert it to the specified type.

Example 1

Listing 1: Convert a raw bytestring to hex

```
raw_bytes = from_bytestring(b"Some test data")
assert raw_bytes == [83, 111, 109, 101, 32, 116, 101, 115, 116, 32, 100, 97,
↳ 116, 97]

hex_data = to_hex(from_bytestring(b"Some test data"))
assert hex_data == b'536f6d6520746573742064617461'
```

Example 2

Listing 2: Convert hex data to a raw bytestring

```
bytestring_data = to_bytestring(from_hex(b'536f6d6520746573742064617461'))
assert bytestring_data == b"Some test data"

raw_bytes = list(from_hex(b'536f6d6520746573742064617461'))
assert raw_bytes == [83, 111, 109, 101, 32, 116, 101, 115, 116, 32, 100, 97,
↳ 116, 97]
```

`pycryptoki.conversions.from_bin(bin_)`

Convert a string-representation of binary into a list of integers.

Parameters `bin` (*str*) – String representation of binary data (ex: “10110111”)

Returns iterator over integers

`pycryptoki.conversions.from_bytestring(ascii_)`

Convert an iterable of strings into an iterable of integers.

Note: For bytestrings on python3, this does effectively nothing, since iterating over a bytestring in python 3 will return integers.

Parameters `ascii` – String to convert

Returns iterator

`pycryptoki.conversions.from_hex(hex_)`

Convert a hexademical string to an iterable of integers.

Parameters `hex` (*str*) – Hex string

Returns Iterator

`pycryptoki.conversions.to_bin(ascii_)`

Convert an iterable of integers to a binary representation.

Parameters `ascii` (*iterable*) – iterable of integers

Returns bytestring of the binary values

`pycryptoki.conversions.to_bytestring(ascii_)`

Convert an iterable of integers into a bytestring.

Parameters `ascii` (*iterable*) – Iterable of integers

Returns bytestring

`pycryptoki.conversions.to_hex(ints)`

Convert an iterable of integers to a hexadecimal string.

Parameters `ints` (*iterable*) – Iterable of integers

Returns bytestring representing the hex data.

1.4.6 Mechanisms

Conversions for pure-python dictionaries to C struct mechanisms.

To implement a new Mechanism:

1. Create a new mechanism class, deriving from *Mechanism*
2. Set `REQUIRED_PARAMS` as a class variable. `REQUIRED_PARAMS` should be a list of strings, defining required parameter keys.

```
class IvMechanism(Mechanism):
    REQUIRED_PARAMS = ['iv']
```

3. Override `to_c_mech()` on the new mechanism class. This function can access `self.params` to get passed-in parameters, and should create the C parameter struct required by the mechanism. This should also return `self.mech` (which is a `CK_MECHANISM` struct).

Listing 3: Simple Example

```
class IvMechanism(Mechanism):
    REQUIRED_PARAMS = ['iv']

    def to_c_mech(self):
        super(IvMechanism, self).to_c_mech()
        if len(self.params['iv']) == 0:
            LOG.debug("Setting IV to NULL (using internal)")
            iv_ba = None
            iv_len = 0
        else:
            iv_ba, iv_len = to_byte_array(self.params['iv'])
        self.mech.pParameter = iv_ba
        self.mech.usParameterLen = iv_len
        return self.mech
```

Listing 4: Example with a PARAMS struct

```
class AESXTSMechanism(Mechanism):
    REQUIRED_PARAMS = ['cb', 'hTweakKey']

    def to_c_mech(self):
        super(AESXTSMechanism, self).to_c_mech()
        xts_params = CK_AES_XTS_PARAMS()
        xts_params.cb = (CK_BYTE * 16)(*self.params['cb'])
        xts_params.hTweakKey = CK_ULONG(self.params['hTweakKey'])
        self.mech.pParameter = cast(pointer(xts_params), c_void_p)
        self.mech.usParameterLen = CK_ULONG(sizeof(xts_params))
        return self.mech
```

1.4.6.1 Helpers

Mechanism base class, as well as helper functions for parsing Mechanism arguments to pycryptoki functions.

class pycryptoki.mechanism.helpers.**Mechanism** (*mech_type='UNKNOWN', params=None*)
 Bases: `object`

Base class for pycryptoki mechanisms. Performs checks for missing parameters w/ created mechs, and creates the base Mechanism Struct for conversion to ctypes.

REQUIRED_PARAMS = []

to_c_mech()

Create the Mechanism structure & set the mech type to the passed-in flavor.

Returns `CK_MECHANISM`

exception pycryptoki.mechanism.helpers.**MechanismException**

Bases: `Exception`

Exception raised for mechanism errors. Ex: required parameters are missing

`pycryptoki.mechanism.helpers.get_c_struct_from_mechanism` (*python_dictionary*,
params_type_string)

Gets a c struct from a python dictionary representing that struct

Parameters

- **python_dictionary** – The python dictionary representing the C struct, see `CK_AES_CBC_PAD_EXTRACT_PARAMS` for an example
- **params_type_string** – A string representing the parameter struct. ex. for `CK_AES_CBC_PAD_EXTRACT_PARAMS` use the string `CK_AES_CBC_PAD_EXTRACT_PARAMS`

Returns A C struct

`pycryptoki.mechanism.helpers.get_python_dict_from_c_mechanism` (*c_mechanism*,
params_type_string)

Gets a python dictionary from a c mechanism's struct for serialization and easier test case writing

Parameters

- **c_mechanism** – The c mechanism to convert to a python dictionary
- **params_type_string** – A string representing the parameter struct. ex. for `CK_AES_CBC_PAD_EXTRACT_PARAMS` use the string `CK_AES_CBC_PAD_EXTRACT_PARAMS`

Returns A python dictionary representing the c struct

`pycryptoki.mechanism.helpers.parse_mechanism` (*mechanism_param*)

Designed for use with any function call that takes in a mechanism, this will handle a mechanism parameter that is one of the following:

1. CKM_ integer constant – will create a `CK_MECHANISM` with only `mech_type` set.

```
parse_mechanism(CKM_RSA_PKCS)
# Results in:
mech = CK_MECHANISM()
mech.mechanism = CK_MECHANISM_TYPE(CKM_RSA_PKCS)
mech.pParameter = None
mech.usParameterLen = 0
```

2. Dictionary with `mech_type` as a mandatory key, and `params` as an optional key. This will be passed into the `Mechanism` class for conversion to a `CK_MECHANISM`.

```
parse_mechanism({'mech_type': CKM_AES_CBC,
                 'params': {'iv': list(range(8))}})
# Results in:
mech = CK_MECHANISM()
mech.mechanism = CK_MECHANISM_TYPE(CKM_AES_CBC)
iv_ba, iv_len = to_byte_array(list(range(8)))
mech.pParameter = iv_ba
mech.usParameterLen = iv_len
```

3. `CK_MECHANISM` struct – passed directly into the raw C Call.
4. Mechanism class – will call `to_c_mech()` on the class, and use the results.

Warning: If you're using this with rpyc, you need to make sure the call `to_c_mech` occurs on the *server* (the machine with the HSM)! If you pass in a *Mechanism* class that was created on the client, the resulting call into `to_c_mech()` will *also* be on the client side!

Note: You can look at `REQUIRED_PARAMS` on each mechanism class to see what parameters are required.

Parameters `mechanism_param` – Parameter to convert to a C Mechanism.

Returns `CK_MECHANISM` struct.

1.4.6.2 AES Mechanisms

AES-specific mechanism implementations.

```
class pycryptoki.mechanism.aes.AESCBCEncryptDataMechanism (mech_type='UNKNOWN',  
                                                         params=None)
```

Bases: `pycryptoki.mechanism.helpers.Mechanism`

AES CBC mechanism for deriving keys from encrypted data.

```
REQUIRED_PARAMS = ['iv', 'data']
```

```
to_c_mech ()
```

Convert extra parameters to ctypes, then build out the mechanism.

Returns `CK_MECHANISM`

```
class pycryptoki.mechanism.aes.AESCTRMechanism (mech_type='UNKNOWN',  
                                                         params=None)
```

Bases: `pycryptoki.mechanism.helpers.Mechanism`

AES CTR Mechanism param conversion.

```
REQUIRED_PARAMS = ['cb', 'ulCounterBits']
```

```
to_c_mech ()
```

Convert extra parameters to ctypes, then build out the mechanism.

Returns `CK_MECHANISM`

```
class pycryptoki.mechanism.aes.AESECBEncryptDataMechanism (mech_type='UNKNOWN',  
                                                         params=None)
```

Bases: `pycryptoki.mechanism.helpers.Mechanism`

AES mechanism for deriving keys from encrypted data.

```
REQUIRED_PARAMS = ['data']
```

```
to_c_mech ()
```

Convert extra parameters to ctypes, then build out the mechanism.

Returns `CK_MECHANISM`

```
class pycryptoki.mechanism.aes.AESGCMMechanism (mech_type='UNKNOWN',  
                                                         params=None)
```

Bases: `pycryptoki.mechanism.helpers.Mechanism`

Creates the AES-GCM specific param structure & converts python types to C types.

```
REQUIRED_PARAMS = ['iv', 'AAD', 'ulTagBits']
```

```
to_c_mech()
```

Convert extra parameters to ctypes, then build out the mechanism.

Returns *CK_MECHANISM*

```
class pycryptoki.mechanism.aes.AESXTSMechanism(mech_type='UNKNOWN',
                                              params=None)
```

Bases: *pycryptoki.mechanism.helpers.Mechanism*

Creates the AES-XTS specific param structure & converts python types to C types.

```
REQUIRED_PARAMS = ['cb', 'hTweakKey']
```

```
to_c_mech()
```

Convert extra parameters to ctypes, then build out the mechanism.

Returns *CK_MECHANISM*

```
class pycryptoki.mechanism.aes.Iv16Mechanism(mech_type='UNKNOWN', params=None)
```

Bases: *pycryptoki.mechanism.helpers.Mechanism*

Mech class for flavors that require an IV set in the mechanism. Will default to [1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8] if no IV is passed in

```
to_c_mech()
```

Convert extra parameters to ctypes, then build out the mechanism.

Returns *CK_MECHANISM*

```
class pycryptoki.mechanism.aes.IvMechanism(mech_type='UNKNOWN', params=None)
```

Bases: *pycryptoki.mechanism.helpers.Mechanism*

Mech class for flavors that require an IV set in the mechanism. Will default to [0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38] if no IV is passed in

```
to_c_mech()
```

Convert extra parameters to ctypes, then build out the mechanism.

Returns *CK_MECHANISM*

1.4.6.3 Generic Mechanisms

Generic Mechanisms conversions.

```
class pycryptoki.mechanism.generic.AutoMech(mech_type='UNKNOWN', params=None)
```

Bases: *pycryptoki.mechanism.helpers.Mechanism*

An attempt to examine underlying C Struct and fill in the appropriate fields, making some assumptions about the data. This works best with parameter structs that only have CK_ULONGs within them (though there is a best-effort attempt to handle arrays).

Warning: Do not use this if the mechanism is already defined!

```
to_c_mech()
```

Attempt to handle generic mechanisms by introspection of the structure.

Returns *CK_MECHANISM*

```
class pycryptoki.mechanism.generic.ConcatenationDeriveMechanism(mech_type='UNKNOWN',
                                                                params=None)
```

Bases: *pycryptoki.mechanism.helpers.Mechanism*

Mechanism class for key derivations. This will take in a second key handle in the parameters, and use it in the resulting Structure.

Warning: This mechanism is disabled in later versions of PKCS11.

REQUIRED_PARAMS = ['h_second_key']

to_c_mech()

Add in a pointer to the second key in the resulting mech structure.

Returns *CK_MECHANISM*

class pycryptoki.mechanism.generic.**NullMech**(*mech_type='UNKNOWN', params=None*)

Bases: *pycryptoki.mechanism.helpers.Mechanism*

Class that creates a mechanism from a flavor with null parameters. Used mostly for signing mechanisms that really don't need anything else.

to_c_mech()

Simply set the pParameter to null pointer.

Returns *CK_MECHANISM*

class pycryptoki.mechanism.generic.**StringDataDerivationMechanism**(*mech_type='UNKNOWN', params=None*)

Bases: *pycryptoki.mechanism.helpers.Mechanism*

Mechanism class for key derivation using passed in string data.

REQUIRED_PARAMS = ['data']

to_c_mech()

Convert data to bytearray, then use in the resulting mech structure.

Returns *CK_MECHANISM*

1.4.6.4 RC Mechanisms

RC-related Mechanism implementations

class pycryptoki.mechanism.rc.**RC2CBCMechanism**(*mech_type='UNKNOWN', params=None*)

Bases: *pycryptoki.mechanism.helpers.Mechanism*

Creates required RC2CBC Param structure & converts python data to C data.

REQUIRED_PARAMS = ['usEffectiveBits', 'iv']

to_c_mech()

Convert extra parameters to ctypes, then build out the mechanism.

Returns *CK_MECHANISM*

class pycryptoki.mechanism.rc.**RC2Mechanism**(*mech_type='UNKNOWN', params=None*)

Bases: *pycryptoki.mechanism.helpers.Mechanism*

Sets the mechanism parameter to the usEffectiveBits

REQUIRED_PARAMS = ['usEffectiveBits']

to_c_mech()

Convert extra parameters to ctypes, then build out the mechanism.

Returns *CK_MECHANISM*

```
class pycryptoki.mechanism.rc.RC5CBCMechanism(mech_type='UNKNOWN',
                                             params=None)
```

Bases: *pycryptoki.mechanism.helpers.Mechanism*

Creates required RC5CBC Param structure & converts python data to C data.

```
REQUIRED_PARAMS = ['ulWordsize', 'ulRounds', 'iv']
```

```
to_c_mech()
```

Convert extra parameters to ctypes, then build out the mechanism.

Returns *CK_MECHANISM*

```
class pycryptoki.mechanism.rc.RC5Mechanism(mech_type='UNKNOWN', params=None)
```

Bases: *pycryptoki.mechanism.helpers.Mechanism*

Creates required RC5 Param structure & converts python data to C data.

```
REQUIRED_PARAMS = ['ulWordsize', 'ulRounds']
```

```
to_c_mech()
```

Convert extra parameters to ctypes, then build out the mechanism.

Returns *CK_MECHANISM*

1.4.6.5 RSA Mechanisms

RSA-related Mechanism implementations.

```
class pycryptoki.mechanism.rsa.RSAPKCSOAEPMechanism(mech_type='UNKNOWN',
                                                    params=None)
```

Bases: *pycryptoki.mechanism.helpers.Mechanism*

Create the required RSA_PKCS_OAEP param structure & convert python data to C data.

```
REQUIRED_PARAMS = ['hashAlg', 'mgf']
```

```
to_c_mech()
```

Convert extra parameters to ctypes, then build out the mechanism.

Returns *CK_MECHANISM*

```
class pycryptoki.mechanism.rsa.RSAPKCSPSSMechanism(mech_type='UNKNOWN',
                                                    params=None)
```

Bases: *pycryptoki.mechanism.helpers.Mechanism*

Create the required RSA_PKCS_PSS param structure & convert python data to C data.

```
REQUIRED_PARAMS = ['hashAlg', 'mgf']
```

```
to_c_mech()
```

Uses default salt length of 8. Can be overridden w/ a parameter though.

Returns *CK_MECHANISM*

1.4.7 Miscellaneous

1.4.7.1 RNG, Digest, Creating Objects

PKCS11 Interface to the following functions:

- `c_generate_random`
- `c_seed_random`
- `c_digest`
- `c_digestkey`
- `c_create_object`
- `c_set_ped_id` (**CA** function)
- `c_get_ped_id` (**CA** function)

`pycryptoki.misc.c_create_object(h_session, template)`

Creates an object based on a given python template

Parameters

- **h_session** (*int*) – Session handle
- **template** (*dict*) – The python template which the object will be based on

Returns (retcode, the handle of the object)

Return type tuple

`pycryptoki.misc.c_create_object_ex(h_session, template)`

Executes `c_create_object()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.misc.c_digest(h_session, data_to_digest, digest_flavor, mechanism=None, output_buffer=None)`

Digests some data

Parameters

- **h_session** (*int*) – Session handle
- **data_to_digest** (*bytes*) – The data to digest, either a string or a list of strings. If this is a list a multipart operation will be used
- **digest_flavor** (*int*) – The flavour of the mechanism to digest (MD2, SHA-1, HAS-160, SHA224, SHA256, SHA384, SHA512)
- **mechanism** – See the `parse_mechanism()` function for possible values. If None will use digest flavor.

- **output_buffer** (*list/int*) – Integer or list of integers that specify a size of output buffer to use for an operation. By default will query with NULL pointer buffer to get required size of buffer.

Returns (retcode, a python string of the digested data)

Return type tuple

`pycryptoki.misc.c_digest_ex(h_session, data_to_digest, digest_flavor, mechanism=None, output_buffer=None)`

Executes `c_digest()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.misc.c_digestkey(h_session, h_key, digest_flavor, mechanism=None)`

Digest a key

Parameters

- **h_session** (*int*) – Session handle
- **h_key** (*int*) – Key to digest
- **digest_flavor** (*int*) – Digest flavor
- **mechanism** – See the `parse_mechanism()` function for possible values. If None will use digest flavor.

`pycryptoki.misc.c_digestkey_ex(h_session, h_key, digest_flavor, mechanism=None)`

Executes `c_digestkey()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.misc.c_generate_random(h_session, length)`

Generates a sequence of random numbers

Parameters

- **h_session** (*int*) – Session handle
- **length** (*int*) – The length in bytes of the random number sequence

Returns (retcode, A string of random data)

Return type tuple

`pycryptoki.misc.c_generate_random_ex(h_session, length)`

Executes `c_generate_random()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.misc.c_get_ped_id(slot)`

Get the PED ID for the given slot.

Parameters **slot** – slot number

Returns The result code and ID

`pycryptoki.misc.c_get_ped_id_ex(slot)`

Executes `c_get_ped_id()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.misc.c_seed_random(h_session, seed)`

Seeds the random number generator

Parameters

- **h_session** (*int*) – Session handle
- **seed** (*bytes*) – A python string of some seed

Returns retcode

Return type `int`

`pycryptoki.misc.c_seed_random_ex(h_session, seed)`

Executes `c_seed_random()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.misc.c_set_ped_id(slot, id)`

Set the PED ID for the given slot.

Parameters

- **slot** – slot number
- **id** – PED ID to use

Returns The result code

`pycryptoki.misc.c_set_ped_id_ex(slot, id)`

Executes `c_set_ped_id()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

1.4.7.2 Find Objects, Attribute Setting/Getting

Functions for dealing with object attributes

`pycryptoki.object_attr_lookup.c_find_objects(h_session, template, num_entries)`

Calls `c_find_objects` and `c_find_objects_init` to get a python dictionary of the objects found.

Parameters

- **h_session** (*int*) – Session handle
- **template** – A python dictionary of the object template to look for
- **num_entries** – The max number of entries to return

Returns Returns a list of handles of objects found

`pycryptoki.object_attr_lookup.c_find_objects_ex(h_session, template, num_entries)`

Executes `c_find_objects()`, and checks the retcode; raising an exception if the return code is not `CKR_OK`.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.object_attr_lookup.c_get_attribute_value(h_session, h_object, template)`

Calls `C_GetAttributeValue` to get an attribute value based on a python template

Parameters

- **h_session** (*int*) – Session handle
- **h_object** – The handle of the object to get attributes for
- **template** – A python dictionary representing the template of the attributes to be retrieved

Returns A python dictionary representing the attributes returned from the HSM/library

`pycryptoki.object_attr_lookup.c_get_attribute_value_ex(h_session, h_object, template)`

Executes `c_get_attribute_value()`, and checks the retcode; raising an exception if the return code is not `CKR_OK`.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.object_attr_lookup.c_set_attribute_value(h_session, h_object, template)`

Calls `C_SetAttributeValue` to set an attribute value based on a python template

Parameters

- **h_session** (*int*) – Session handle
- **h_object** – The handle of the object to get attributes for

- **template** – A python dictionary representing the template of the attributes to be written

Returns A python dictionary representing the attributes returned from the HSM/library

`pycryptoki.object_attr_lookup.c_set_attribute_value_ex(h_session, h_object, template)`

Executes `c_set_attribute_value()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

1.4.7.3 HSM Management

Methods responsible for pycryptoki ‘hsm management’ set of commands.

`pycryptoki.hsm_management.c_performselftest(slot, test_type, input_data, input_data_len)`

Test: Performs a self test for specified test type on a given slot.

Parameters

- **slot** – slot number
- **test_type** – type of test CK_ULONG
- **input_data** – pointer to input data CK_BYTE_PTR
- **input_data_len** – input data length CK_ULONG

Returns

the result code

[CK_SLOT_ID, CK_ULONG, CK_BYTE_PTR, CK_ULONG, CK_BYTE_PTR, CK_ULONG_PTR]

`pycryptoki.hsm_management.c_performselftest_ex(slot, test_type, input_data, input_data_len)`

Executes `c_performselftest()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.hsm_management.ca_createloginchallenge(h_session, user_type, challenge)`

Creates a login challenge for the given user.

Parameters

- **h_session** (*int*) – Session handle
- **user_type** – user type
- **challenge** – challenge

Returns the result code

`pycryptoki.hsm_management.ca_createloginchallenge_ex(h_session, user_type, challenge)`

Executes `ca_createloginchallenge()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.hsm_management.ca_deleteremotepedvector(h_session)`

Deletes a remote PED vector

Parameters **h_session** (*int*) – Session handle

Returns the result code

`pycryptoki.hsm_management.ca_deleteremotepedvector_ex(h_session)`

Executes `ca_deleteremotepedvector()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.hsm_management.ca_get_hsm_capability_set(slot)`

Get the capabilities of the given slot.

Parameters `slot` (*int*) – Target slot number

Returns `retcode`, {`id`: `val`} dict of capabilities (None if command failed)

`pycryptoki.hsm_management.ca_get_hsm_capability_set_ex(slot)`

Executes `ca_get_hsm_capability_set()`, and checks the `retcode`; raising an exception if the return code is not `CKR_OK`.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the `retcode`, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.hsm_management.ca_get_hsm_capability_setting(slot, capability_id)`

Get the value of a single capability

Parameters

- `slot` – slot ID of slot to query
- `capability_id` – capability ID

Returns result code, `CK_ULONG` representing capability active or not

`pycryptoki.hsm_management.ca_get_hsm_capability_setting_ex(slot, capability_id)`

Executes `ca_get_hsm_capability_setting()`, and checks the `retcode`; raising an exception if the return code is not `CKR_OK`.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the `retcode`, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.hsm_management.ca_get_hsm_policy_set(slot)`

Get the policies of the given slot.

Parameters `slot` (*int*) – Target slot number

Returns `retcode`, {`id`: `val`} dict of policies (None if command failed)

`pycryptoki.hsm_management.ca_get_hsm_policy_set_ex(slot)`

Executes `ca_get_hsm_policy_set()`, and checks the `retcode`; raising an exception if the return code is not `CKR_OK`.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the `retcode`, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.hsm_management.ca_get_hsm_policy_setting(slot, policy_id)`

Get the value of a single policy

Parameters

- `slot` – slot ID of slot to query
- `policy_id` – policy ID

Returns result code, `CK_ULONG` representing policy active or not

`pycryptoki.hsm_management.ca_get_hsm_policy_setting_ex(slot, policy_id)`

Executes `ca_get_hsm_policy_setting()`, and checks the `retcode`; raising an exception if the return code is not `CKR_OK`.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the `retcode`, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.hsm_management.ca_hainit(h_session, h_key)`

Creates a login key pair on the primary token.

Parameters

- `h_session` (*int*) – Session handle

- **h_key** – the login private key

Returns the result code

`pycryptoki.hsm_management.ca_hainit_ex(h_session, h_key)`

Executes `ca_hainit()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.hsm_management.ca_initializeremotepedvector(h_session)`

Initializes a remote PED vector

Parameters **h_session** (*int*) – Session handle

Returns the result code

`pycryptoki.hsm_management.ca_initializeremotepedvector_ex(h_session)`

Executes `ca_initializeremotepedvector()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.hsm_management.ca_mtkresplit(slot)`

Resplit the MTK

Parameters **slot** – slot number

Returns the result code

`pycryptoki.hsm_management.ca_mtkresplit_ex(slot)`

Executes `ca_mtkresplit()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.hsm_management.ca_mtkrestore(slot)`

Restore the MTK

Parameters `slot` – slot number

Returns the result code

`pycryptoki.hsm_management.ca_mtkrestore_ex(slot)`

Executes `ca_mtkrestore()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.hsm_management.ca_mtkzeroize(slot)`

Zeroize the MTK

Parameters `slot` – slot number

Returns the result code

`pycryptoki.hsm_management.ca_mtkzeroize_ex(slot)`

Executes `ca_mtkzeroize()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```


`pycryptoki.hsm_management.ca_set_destructive_hsm_policies(h_session, policies)`
Set multiple HSM policies.

Parameters

- **h_session** (*int*) – Session handle
- **policies** – dict of policy ID ints and value ints

Returns result code

`pycryptoki.hsm_management.ca_set_destructive_hsm_policies_ex(h_session, policies)`

Executes `ca_set_destructive_hsm_policies()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.hsm_management.ca_set_destructive_hsm_policy(h_session, policy_id, policy_val)`

Sets the destructive HSM policies by calling CA_SetDestructiveHSMPolicy

Parameters

- **h_session** (*int*) – Session handle
- **policy_id** – The ID of the policy being set
- **policy_val** – The value of the policy being set

Returns The result code

`pycryptoki.hsm_management.ca_set_destructive_hsm_policy_ex(h_session, policy_id, policy_val)`

Executes `ca_set_destructive_hsm_policy()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.hsm_management.ca_set_hsm_policies(h_session, policies)`
Set multiple HSM policies.

Parameters

- **h_session** (*int*) – Session handle
- **policies** – dict of policy ID ints and value ints

Returns result code

`pycryptoki.hsm_management.ca_set_hsm_policies_ex(h_session, policies)`
Executes `ca_set_hsm_policies()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.hsm_management.ca_set_hsm_policy(h_session, policy_id, policy_val)`
Sets the HSM policies by calling CA_SetHSMPolicy

Parameters

- **h_session** (*int*) – Session handle
- **policy_id** – The ID of the policy being set
- **policy_val** – The value of the policy being set

Returns The result code

`pycryptoki.hsm_management.ca_set_hsm_policy_ex(h_session, policy_id, policy_val)`
Executes `ca_set_hsm_policy()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.hsm_management.ca_settokencertificatesignature` (*h_session*, *access_level*,
customer_id,
pub_template, *signature*, *signature_len*)

Completes the installation of a certificate on a token. The caller must supply a public key and a signature for token certificate. The public key is provided through the template; it must contain a key type, a modulus and a public exponent.

Parameters

- **h_session** (*int*) – Session handle
- **access_level** – the access level
- **customer_id** – the customer ID
- **pub_template** – the public template
- **signature** – the signature
- **signature_len** – the length in bytes of the signature

Returns the result code

`pycryptoki.hsm_management.ca_settokencertificatesignature_ex` (*h_session*, *access_level*,
customer_id,
pub_template,
signature, *signature_len*)

Executes `ca_settokencertificatesignature()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

1.4.7.4 Audit Functions

Methods responsible for managing a user's session and login/c_logout

`pycryptoki.audit_handling.ca_get_time` (*h_session*)

Parameters **h_session** (*int*) – Session handle

`pycryptoki.audit_handling.ca_get_time_ex` (*h_session*)

Executes `ca_get_time()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.audit_handling.ca_init_audit(slot, audit_pin, audit_label)`

Parameters

- **slot** –
- **audit_pin** –
- **audit_label** –

`pycryptoki.audit_handling.ca_init_audit_ex(slot, audit_pin, audit_label)`

Executes `ca_init_audit()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.audit_handling.ca_time_sync(h_session, ultime)`

Parameters

- **h_session** (*int*) – Session handle
- **ultime** –

`pycryptoki.audit_handling.ca_time_sync_ex(h_session, ultime)`

Executes `ca_time_sync()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

1.4.7.5 Backup Functions

Backup related commands

`pycryptoki.backup.ca_close_secure_token(h_session, h_ID)`

Parameters

- **h_session** (*int*) – Session handle
- **h_ID** –

`pycryptoki.backup.ca_close_secure_token_ex(h_session, h_ID)`

Executes `ca_close_secure_token()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.backup.ca_extract(h_session, mechanism)`

Parameters

- **h_session** (*int*) – Session handle
- **mechanism** – See the `parse_mechanism()` function for possible values.

`pycryptoki.backup.ca_extract_ex(h_session, mechanism)`

Executes `ca_extract()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.backup.ca_insert(h_session, mechanism)`

Parameters

- **h_session** (*int*) – Session handle
- **mechanism** – See the `parse_mechanism()` function for possible values.

`pycryptoki.backup.ca_insert_ex(h_session, mechanism)`

Executes `ca_insert()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.backup.ca_open_secure_token(h_session, storage_path, dev_ID, mode)`

Parameters

- **h_session** (*int*) – Session handle
- **storage_path** –
- **dev_ID** –
- **mode** –

`pycryptoki.backup.ca_open_secure_token_ex(h_session, storage_path, dev_ID, mode)`

Executes `ca_open_secure_token()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.backup.ca_sim_extract(h_session, key_handles, authform, auth_secrets=None, sub-
set_size=0, delete_after_extract=False)`

Extract multiple keys to a wrapped blob. The returned blob can then be written into a file.

Parameters

- **h_session** (*int*) – Session handle

- **key_handles** (*list[int]*) – List of key handles to extract
- **authform** (*int*) – Type of authentication to use. See `pycryptoki.backup.SIM_AUTH` for details
- **auth_secrets** (*list[str]*) – Authorization secrets to use (Length will correspond to the N value in `ckdemo`)
- **subset_size** (*int*) – Subset size required for key use (Corresponds to the M value in `ckdemo`)
- **delete_after_extract** (*bool*) – If true, will destroy the original keys after they have been extracted.

Returns `retcode`, `blob_data` tuple.

`pycryptoki.backup.ca_sim_extract_ex(h_session, key_handles, authform, auth_secrets=None, subset_size=0, delete_after_extract=False)`

Executes `ca_sim_extract()`, and checks the `retcode`; raising an exception if the return code is not `CKR_OK`.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the `retcode`, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.backup.ca_sim_insert(h_session, blob_data, authform, auth_secrets=None)`

Insert keys into the HSM from blob data that was wrapped off using SIM.

Parameters

- **h_session** (*int*) – Session handle
- **blob_data** (*str*) – Read in raw wrapped data. Typically read in from a file.
- **authform** (*int*) – Type of authentication to use. See `pycryptoki.backup.SIM_AUTH` for details
- **auth_secrets** (*list[str]*) – Authorization secrets to use (Length will correspond to the N value in `ckdemo`)

Returns `retcode`, `keys` tuple, where `keys` is a list of integers.

`pycryptoki.backup.ca_sim_insert_ex(h_session, blob_data, authform, auth_secrets=None)`

Executes `ca_sim_insert()`, and checks the `retcode`; raising an exception if the return code is not `CKR_OK`.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

`pycryptoki.backup.ca_sim_multisign(h_session, blob_data, data_to_sign, mechanism, authform, auth_secrets=None)`

Sign data using keys that were extracted to a SIM blob.

Parameters

- **h_session** (*int*) – Session handle
- **blob_data** (*str*) – Read in raw wrapped key data. Typically read in from a file.
- **data_to_sign** – List of bytestring data to sign
- **mechanism** – Mechanism to use with the Sign operation
- **authform** (*int*) – Type of authentication to use. See `pycryptoki.backup.SIM_AUTH` for details
- **auth_secrets** (*list[str]*) – Authorization secrets to use (Length will correspond to the N value in ckdemo)

Returns retcode, signature list

`pycryptoki.backup.ca_sim_multisign_ex(h_session, blob_data, data_to_sign, mechanism, authform, auth_secrets=None)`

Executes `ca_sim_multisign()`, and checks the retcode; raising an exception if the return code is not CKR_OK.

Note: By default, this will not return the return code if the function returns additional data.

Example:

```
retcode, key_handle = c_generate_key(...)
#vs
key_handle = c_generate_key_ex(...)
```

If the function *only* returns the retcode, then that will still be returned:

```
retcode = c_seed_random(...)
retcode = c_seed_random_ex(...)
```

1.4.8 Pycryptoki Helpers

These are various helper modules and functions. They contain constant definitions, C parameter structs, configuration parsing, and default templates.

1.4.8.1 cryptoki_helpers

Helper functions to get us access to the PKCS11 library.

exception pycryptoki.cryptoki_helpers.CryptokiConfigException

Bases: pycryptoki.exceptions.LunaException

Exception raised when we fail to determine the PKCS11 library location

exception pycryptoki.cryptoki_helpers.CryptokiDLLException (*additional_info*,
orig_error)

Bases: Exception

Custom exception class used to print an error when a call to the Cryptoki DLL failed. The late binding makes debugging a little bit more difficult because function calls have to pass through an additional layer of abstraction. This custom exception prints out a quick message detailing exactly what function failed.

class pycryptoki.cryptoki_helpers.CryptokiDLLSingleton

Bases: object

A singleton class which holds an instance of the loaded cryptoki DLL object.

classmethod from_path (*path*)

get_dll ()

Get the loaded library (parsed from crstoki.ini/Chrystoki.conf)

loaded_dll_library = None

pycryptoki.cryptoki_helpers.log_args (*funcname*, *args*)

Log function name & arguments for a cryptoki ctypes call.

Parameters

- **funcname** (*str*) – Function name
- **args** (*tuple*) – Arguments to be passed to ctypes function.

pycryptoki.cryptoki_helpers.make_late_binding_function (*function_name*)

A function factory for creating a function that will bind to the cryptoki DLL only when the function is called.

Parameters *function_name* –

pycryptoki.cryptoki_helpers.parse_chrstoki_conf ()

Parse the crstoki.ini/Chrystoki.conf file to find the library .so/.dll file so that we can use it.

1.4.8.2 lookup_dicts

Module that contains lookup dictionaries for easy logging of error codes and other constants within pycryptoki.

pycryptoki.lookup_dicts.ATTR_NAME_LOOKUP = {0: 'CKA_CLASS', 1: 'CKA_TOKEN', 2: 'CKA_PRIVATE'

pycryptoki.lookup_dicts.ret_vals_dictionary = {0: 'CKR_OK', 1: 'CKR_CANCEL', 2: 'CKR_HOST_NOT_AVAILABLE'

1.4.8.3 default_templates

File containing a number of templates taken from CKDemo and manually converted into python format. See the attributes.py file for methods to convert them into the proper C format.

pycryptoki.default_templates.CERTIFICATE_TEMPLATE = {0: 1, 1: True, 3: b'Created certificate'

The simple data object template taken from CKDemo when you select the Create Object option and choose data

`pycryptoki.default_templates.CKM_DH_PKCS_PARAMETER_GEN_TEMP = {1: True, 2: True, 3: b'S'`
 The simple certificate object taken from CKDemo when you select the Create Object option and choose certi-
 cate

`pycryptoki.default_templates.CKM_SSL3_PRE_MASTER_KEY_GEN_TEMP = {1: True, 3: b'SSL3 Pre M`
 Curve dictionary for ECDSA with oids as lists, taken from Components/tools/common/CommonData.cpp

`pycryptoki.default_templates.KEY_PAIR_GENERATOR_TEMPLATES = {0: ({1: True, 2: True, 368`
 This list is not complete

`pycryptoki.default_templates.get_default_key_pair_template(mechanism)`
 Gets the default template for the given key pair gen mechanism, returns a deep copy

Parameters *mechanism* –

`pycryptoki.default_templates.get_default_key_template(mechanism)`
 Gets a default template for the given key gen mechanism, returns a deep copy

Parameters *mechanism* –

1.4.8.4 defaults

A file containing commonly used strings or other data similar to a config file

1.4.8.5 cryptoki

This file contains all of the ctypes definitions for the cryptoki library. The ctypes definitions outline the structures for the cryptoki C API.

`pycryptoki.cryptoki.CA_InvokeService(*args)`

Parameters

- **args* –
- ***kwargs* –

`class pycryptoki.cryptoki.CK_OTP_SIGNATURE_INFO`

pParams

Structure/Union member

ulCount

Structure/Union member

`pycryptoki.cryptoki.CA_GetSlotIdForContainer(*args)`

Parameters

- **args* –
- ***kwargs* –

`pycryptoki.cryptoki.C_UnwrapKey(*args)`

Parameters

- **args* –
- ***kwargs* –

`pycryptoki.cryptoki.Int32`
 alias of `ctypes.c_long`

`pycryptoki.cryptoki.CA_PerformModuleCall(*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.CA_SetApplicationID(*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.CA_LoadEncryptedModule(*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.CA_MTKZeroize(*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.CK_CA_Restart`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.C_SetAttributeValue(*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.CA_STCSetDigestAlgorithm(*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.CK_C_SetOperationState`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.C_VerifyFinal(*args)`

Parameters

- `*args` –
- `**kwargs` –

`class pycryptoki.cryptoki.CK_DATE`

day

Structure/Union member

month

Structure/Union member

year
Structure/Union member

class pycryptoki.cryptoki.CK_WTLS_PRF_PARAMS

DigestMechanism
Structure/Union member

pLabel
Structure/Union member

pOutput
Structure/Union member

pSeed
Structure/Union member

pulOutputLen
Structure/Union member

ulLabelLen
Structure/Union member

ulSeedLen
Structure/Union member

pycryptoki.cryptoki.C_GetInfo(*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.CA-TokenZeroize(*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.CK_CA_GetConfigurationElementDescription
alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.CK_C_SetPIN
alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.CA_GetContainerList(*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.CK_CA_STCGetDigestID
alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.CK_ATTRIBUTE_PTR
alias of pycryptoki.cryptoki.LP_CK_ATTRIBUTE

class pycryptoki.cryptoki.CK_VERSION

major

Structure/Union member

minor

Structure/Union member

`pycryptoki.cryptoki.CK_CA_GetFPV`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_InitRolePIN`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_LockClusteredSlot(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.C_WaitForSlotEvent(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_VOID_PTR`

alias of `ctypes.c_void_p`

`pycryptoki.cryptoki.CK_CA_GetSlotIdForContainer`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_EncodeECParamsFromFile(*args)`

Parameters

- ***args** –
- ****kwargs** –

class `pycryptoki.cryptoki.CK_MECHANISM_INFO`

flags

Structure/Union member

ulMaxKeySize

Structure/Union member

ulMinKeySize

Structure/Union member

`pycryptoki.cryptoki.CK_CA_CloneAsTargetInit`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_STCSetMaxSessions`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_LoadModule(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_FirmwareRollback(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_GetSecureElementMeta`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_Deactivate(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.Float64`
alias of `ctypes.c_double`

`pycryptoki.cryptoki.CK_X9_42_DH_KDF_TYPE`
alias of `ctypes.c_ulong`

class `pycryptoki.cryptoki.CK_INFO`

cryptokiVersion

Structure/Union member

flags

Structure/Union member

libraryDescription

Structure/Union member

libraryVersion

Structure/Union member

manufacturerID

Structure/Union member

`pycryptoki.cryptoki.CA_HAActivateMofN(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_MultisignValue`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_GetHSMCapabilitySet(*args)`

Parameters

- ***args** –
- ****kwargs** –

class `pycryptoki.cryptoki.CK_KIP_PARAMS`

hKey

Structure/Union member

pMechanism
Structure/Union member

pSeed
Structure/Union member

ulSeedLen
Structure/Union member

`pycryptoki.cryptoki.CK_CA_SpRawWrite`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_OTP_PARAM_PTR`
alias of `pycryptoki.cryptoki.LP_CK_OTP_PARAM`

class `pycryptoki.cryptoki.CK_X9_42_MQV_DERIVE_PARAMS`

hPrivateData
Structure/Union member

kdf
Structure/Union member

pOtherInfo
Structure/Union member

pPublicData
Structure/Union member

pPublicData2
Structure/Union member

publicKey
Structure/Union member

ulOtherInfoLen
Structure/Union member

ulPrivateDataLen
Structure/Union member

ulPublicDataLen
Structure/Union member

ulPublicDataLen2
Structure/Union member

`pycryptoki.cryptoki.CK_CA_GetHState`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.C_CloseAllSessions(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_STCSetKeyActivationTimeOut`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.C_SignInit(*args)`

Parameters

- ***args** –
- ****kwargs** –

class pycryptoki.cryptoki.**CK_CMS_SIG_PARAMS**

certificateHandle
Structure/Union member

pContentType
Structure/Union member

pDigestMechanism
Structure/Union member

pRequestedAttributes
Structure/Union member

pRequiredAttributes
Structure/Union member

pSigningMechanism
Structure/Union member

ulRequestedAttributesLen
Structure/Union member

ulRequiredAttributesLen
Structure/Union member

pycryptoki.cryptoki.**CA_GetRemotePEDVectorStatus** (*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.**CK_C_GetFunctionList**
alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.**CK_CA_LogGetStatus**
alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.**CK_C_OpenSession**
alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.**CA_IndirectLogin** (*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.**CA_OpenSecureToken** (*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.**CA_GenerateMofN** (*args)

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_MTKResplit`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_HALogin(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_RetrieveLicenseList`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

class `pycryptoki.cryptoki.CK_ECMQV_DERIVE_PARAMS`

hPrivateData
Structure/Union member

kdf
Structure/Union member

pPublicData
Structure/Union member

pPublicData2
Structure/Union member

pSharedData
Structure/Union member

publicKey
Structure/Union member

ulPrivateDataLen
Structure/Union member

ulPublicDataLen
Structure/Union member

ulPublicDataLen2
Structure/Union member

ulSharedDataLen
Structure/Union member

`pycryptoki.cryptoki.CK_CA_LKMReceiverComplete`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_InitSlotRolePIN(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_MOFN_ACTIVATION_PTR`
alias of `pycryptoki.cryptoki.LP_CA_MOFN_ACTIVATION`

`pycryptoki.cryptoki.CA_GetFunctionList(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_TLS_PRF_PARAMS_PTR`
 alias of `pycryptoki.cryptoki.LP_CK_TLS_PRF_PARAMS`
`pycryptoki.cryptoki.CA_GetHSMStorageInformation(*args)`

Parameters

- ***args** –
- ****kwargs** –

class `pycryptoki.cryptoki.CA_MOFN_ACTIVATION`

pVector

Structure/Union member

ulVectorLen

Structure/Union member

`pycryptoki.cryptoki.CK_CA_GetTime`
 alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`
`pycryptoki.cryptoki.Word`
 alias of `ctypes.c_ulong`
`pycryptoki.cryptoki.CA_LogVerifyFile(*args)`

Parameters

- ***args** –
- ****kwargs** –

class `pycryptoki.cryptoki.CK_DES_CTR_PARAMS`

cb

Structure/Union member

ulCounterBits

Structure/Union member

`pycryptoki.cryptoki.CA_GetClusterState(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_STMGetState(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_STCGetCipherAlgorithm`
 alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_STCGetMaxSessions(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_OBJECT_HANDLE`
alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CK_MAC_GENERAL_PARAMS`
alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CK_C_VerifyUpdate`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_WriteCommonStore`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_GetPedId(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_Insert(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_EC_MAC_SCHEME`
alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CK_CA_STCGetClientInfo`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_ConfigureRemotePED(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_STCGetPubKey(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_C_WrapKey`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_SpRawWrite(*args)`

Parameters

- ***args** –
- ****kwargs** –

```
class pycryptoki.cryptoki.CK_KDF_PRF_PARAMS
```

```
    pContext  
        Structure/Union member
```

```
    pLabel  
        Structure/Union member
```

```
    prfType  
        Structure/Union member
```

```
    ulContextLen  
        Structure/Union member
```

```
    ulCounter  
        Structure/Union member
```

```
    ulEncodingScheme  
        Structure/Union member
```

```
    ulLabelLen  
        Structure/Union member
```

```
pycryptoki.cryptoki.CK_ULONG  
    alias of ctypes.c_ulong
```

```
pycryptoki.cryptoki.CA_GetPrimarySlot(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CK_SSL3_MASTER_KEY_DERIVE_PARAMS_PTR  
    alias of pycryptoki.cryptoki.LP_CK_SSL3_MASTER_KEY_DERIVE_PARAMS
```

```
pycryptoki.cryptoki.Float  
    alias of ctypes.c_double
```

```
pycryptoki.cryptoki.CK_DESTROYMUTEX  
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CK_ECMQV_DERIVE_PARAMS_PTR  
    alias of pycryptoki.cryptoki.LP_CK_ECMQV_DERIVE_PARAMS
```

```
pycryptoki.cryptoki.CA_InitIndirectPIN(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CK_CA_DeactivateMofN  
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CK_CA_STCGetPubKey  
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CA_SpRawRead(*args)
```

Parameters

- ***args** –

- ****kwargs** –

`pycryptoki.cryptoki.CA_DestroyMultipleObjects(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_C_GetSlotList`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_GetTokenStorageInformation(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.SInt8`

alias of `ctypes.c_byte`

`pycryptoki.cryptoki.CK_DES_CTR_PARAMS_PTR`

alias of `pycryptoki.cryptoki.LP_CK_DES_CTR_PARAMS`

`pycryptoki.cryptoki.CK_CA_STCGetState`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

class `pycryptoki.cryptoki.CK_RC5_MAC_GENERAL_PARAMS`

ulMacLength

Structure/Union member

ulRounds

Structure/Union member

ulWordsize

Structure/Union member

`pycryptoki.cryptoki.CK_CA_InvokeService`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_SEED_CTR_PARAMS`

alias of `pycryptoki.cryptoki.CK_AES_CTR_PARAMS`

`pycryptoki.cryptoki.CK_LKM_TOKEN_ID`

alias of `pycryptoki.cryptoki.CK_LKM_TOKEN_ID_S`

`pycryptoki.cryptoki.CA_CloneObjectToAllSessions(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_ListSecureTokenUpdate(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_MTKSetStorage(*args)`

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CA_SetMofN(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CK_CA_SetContainerPolicies
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
class pycryptoki.cryptoki.CK_CLUSTER_STATE
```

bMembers

Structure/Union member

ulMemberStatus

Structure/Union member

```
pycryptoki.cryptoki.CK_C_CreateObject
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CK_CA_STCIsEnabled
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CA_CloneModifyMofN(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CK_C_DecryptFinal
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CK_CA_SetContainerPolicy
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.eInitMsgs
    alias of ctypes.c_int
```

```
pycryptoki.cryptoki.CK_FLAGS
    alias of ctypes.c_ulong
```

```
pycryptoki.cryptoki.CK_CA_STCSetDigestAlgorithm
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CA_GetFPV(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CK_HA_MEMBER_PTR
    alias of pycryptoki.cryptoki.LP_CK_HA_MEMBER
```

```
pycryptoki.cryptoki.C_Digest(*args)
```

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_SetRDK`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_HAAnswerLoginChallenge(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.BYTE`

alias of `ctypes.c_ubyte`

`pycryptoki.cryptoki.CK_CA_GetSessionInfo`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.C_SignEncryptUpdate(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_MECHANISM_TYPE_PTR`

alias of `pycryptoki.cryptoki.LP_c_ulong`

`pycryptoki.cryptoki.CK_XOR_BASE_DATA_KDF_PARAMS_PTR`

alias of `pycryptoki.cryptoki.LP_CK_XOR_BASE_DATA_KDF_PARAMS`

class `pycryptoki.cryptoki.CK_SESSION_INFO`

flags

Structure/Union member

slotID

Structure/Union member

state

Structure/Union member

usDeviceError

Structure/Union member

`pycryptoki.cryptoki.CK_CA_CloneObjectToAllSessions`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_FirmwareRollback`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

class `pycryptoki.cryptoki.CK_WTLS_KEY_MAT_OUT`

hKey

Structure/Union member

hMacSecret

Structure/Union member

pIV

Structure/Union member

class pycryptoki.cryptoki.CK_WTLS_KEY_MAT_PARAMS

DigestMechanism

Structure/Union member

RandomInfo

Structure/Union member

bIsExport

Structure/Union member

pReturnedKeyMaterial

Structure/Union member

ulIVSizeInBits

Structure/Union member

ulKeySizeInBits

Structure/Union member

ulMacSizeInBits

Structure/Union member

ulSequenceNumber

Structure/Union member

pycryptoki.cryptoki.CA_ReplaceFastPathKEK(*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.CA_ExtractMaskedObject(*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.CK_CA_GetNumberOfAllowedContainers

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

pycryptoki.cryptoki.C_DigestEncryptUpdate(*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.UInt16

alias of `ctypes.c_ushort`

pycryptoki.cryptoki.CK_CA_HAAnswerLoginChallenge

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

pycryptoki.cryptoki.CK_RSA_PKCS_MGF_TYPE_PTR

alias of `pycryptoki.cryptoki.LP_c_ulong`

pycryptoki.cryptoki.CA_STCSetSequenceWindowSize(*args)

Parameters

- ***args** –
- ****kwargs** –

```
class pycryptoki.cryptoki.CK_SKIPJACK_RELAYX_PARAMS
```

pNewPassword

Structure/Union member

pNewPublicData

Structure/Union member

pNewRandomA

Structure/Union member

pOldPassword

Structure/Union member

pOldPublicData

Structure/Union member

pOldRandomA

Structure/Union member

pOldWrappedX

Structure/Union member

ulNewPasswordLen

Structure/Union member

ulNewPublicDataLen

Structure/Union member

ulNewRandomLen

Structure/Union member

ulOldPasswordLen

Structure/Union member

ulOldPublicDataLen

Structure/Union member

ulOldRandomLen

Structure/Union member

ulOldWrappedXLen

Structure/Union member

```
pycryptoki.cryptoki.CK_CA_InitIndirectPIN
```

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

```
pycryptoki.cryptoki.CA_GetTokenObjectHandle(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.C_EncryptFinal(*args)
```

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA-TokenZeroize`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_EC-KDF-TYPE`
alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CA_InitializeRemotePEDVector(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_LogVerify(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_SetDestructiveHSPolicies(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA-ManualKCV`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_LogGetStatus(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CREATEMUTEX`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA-LKMInitiatorComplete`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_SetPedId(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA-ModifyMofN`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_C-GetAttributeValue`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_CreateContainer(*args)`

Parameters

- ***args** –
- ****kwargs** –

class pycryptoki.cryptoki.**CK_KEY_WRAP_SET_OAEP_PARAMS**

bBC
Structure/Union member

pX
Structure/Union member

ulXLen
Structure/Union member

pycryptoki.cryptoki.**CK_C_SignInit**
alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.**CK_SESSION_INFO_PTR**
alias of pycryptoki.cryptoki.LP_CK_SESSION_INFO

pycryptoki.cryptoki.**CK_CA_STCSetCipherAlgorithm**
alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.**CK_CA_SIMInsert**
alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.**CK_CA_SIMExtract**
alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.**CA_GenerateCloneableMofN**(*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.**CK_CHAR_PTR**
alias of pycryptoki.cryptoki.LP_c_ubyte

pycryptoki.cryptoki.**CA_ResetDevice**(*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.**CA_STCSetCipherAlgorithm**(*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.**CA_GetModuleList**(*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.**CK_CA_GetModuleInfo**
alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

`pycryptoki.cryptoki.CK_CA_SetTPV`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_GetTime(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_GetTokenPolicies(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_InsertMaskedObject(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_CloneAllObjectsToSession(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_WriteCommonStore(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_CloseSecureToken(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_RC5_MAC_GENERAL_PARAMS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_RC5_MAC_GENERAL_PARAMS`

`pycryptoki.cryptoki.CK_CA_ClonePrivateKey`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_C_EncryptFinal`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_PKCS5_PBKD2_PSEUDO_RANDOM_FUNCTION_TYPE`
alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CK_C_VerifyInit`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_LKM_TOKEN_ID_PTR`
alias of `pycryptoki.cryptoki.LP_CK_LKM_TOKEN_ID_S`

```
pycryptoki.cryptoki.CA_SetContainerPolicy(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CK_CA_FactoryReset
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.PointerDifference
    alias of ctypes.c_long
```

```
pycryptoki.cryptoki.CA_CloseAllSecondarySessions(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CK_CA_CloseApplicationIDForContainer
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CA_STCGetKeyLifeTime(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
class pycryptoki.cryptoki.CK_RC2_MAC_GENERAL_PARAMS
```

```
    ulMacLength
        Structure/Union member
```

```
    usEffectiveBits
        Structure/Union member
```

```
pycryptoki.cryptoki.CK_SESSION_HANDLE
    alias of ctypes.c_ulong
```

```
class pycryptoki.cryptoki.CK_SLOT_INFO
```

```
    firmwareVersion
        Structure/Union member
```

```
    flags
        Structure/Union member
```

```
    hardwareVersion
        Structure/Union member
```

```
    manufacturerID
        Structure/Union member
```

```
    slotDescription
        Structure/Union member
```

```
pycryptoki.cryptoki.CA_STCGetDigestID(*args)
```

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_GetHSMPolicySetting`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_CreateContainerLoginChallenge`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_QueryLicense(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_STCGetCipherAlgorithm(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_SESSION_HANDLE_PTR`
alias of `pycryptoki.cryptoki.LP_c_ulong`

`pycryptoki.cryptoki.CK_RC2_MAC_GENERAL_PARAMS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_RC2_MAC_GENERAL_PARAMS`

`pycryptoki.cryptoki.CA_CloseApplicationID(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_C_Logout`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_HAGetLoginChallenge(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_SetRDK(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_C_WaitForSlotEvent`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST_PTR`
alias of `pycryptoki.cryptoki.LP_CK_SFNT_CA_FUNCTION_LIST`

`pycryptoki.cryptoki.CK_SKIPJACK_PRIVATE_WRAP_PTR`
alias of `pycryptoki.cryptoki.LP_CK_SKIPJACK_PRIVATE_WRAP_PARAMS`

`pycryptoki.cryptoki.CK_CA_GetObjectUID`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.C_SetOperationState(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_ConfigureRemotePED`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_C_SignRecoverInit`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_STCGetDigestAlgorithm`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CERTIFICATE_TYPE`
alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CA_Extract(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_OBJECT_CLASS`
alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CA_STCGetAdminPubKey(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.C_VerifyRecover(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_STCGetCipherNameByID(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_GetObjectHandle(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA-TokenDelete`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_CloneObject (*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_C_DigestEncryptUpdate`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.C_VerifyRecoverInit (*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.C_DigestKey (*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_LogExternal (*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_GetFunctionList`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_GetPartitionPolicyTemplate (*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_CloneAsTarget`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_HALogin`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_CloneAsSource (*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_KEY_TYPE`
alias of `ctypes.c_ulong`

class `pycryptoki.cryptoki.CK_RSA_PKCS_PSS_PARAMS`

hashAlg

Structure/Union member

mgf
Structure/Union member

usSaltLen
Structure/Union member

`pycryptoki.cryptoki.CK_CA_GetSecondarySlot`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.UInt32`
alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CA_OpenSessionWithAppID(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_InvokeServiceFinal(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_AES_XTS_PARAMS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_AES_XTS_PARAMS`

`pycryptoki.cryptoki.C_OpenSession(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_SetUserContainerName`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_WaitForSlotEvent(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_WTLS_RANDOM_DATA_PTR`
alias of `pycryptoki.cryptoki.LP_CK_WTLS_RANDOM_DATA`

`pycryptoki.cryptoki.CK_C_CloseAllSessions`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_RSA_PKCS_PSS_PARAMS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_RSA_PKCS_PSS_PARAMS`

class `pycryptoki.cryptoki.CK_RC2_CBC_PARAMS`

iv
Structure/Union member

usEffectiveBits
Structure/Union member

`pycryptoki.cryptoki.CK_CA_GetServerInstanceBySlotID`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_LoadModule`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_HAAnswerMofNChallenge`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_GetRemotePEDVectorStatus`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_ARIA_CTR_PARAMS`
alias of `pycryptoki.cryptoki.CK_AES_CTR_PARAMS`

class `pycryptoki.cryptoki.CK_CAMELLIA_CTR_PARAMS`

cb

Structure/Union member

ulCounterBits

Structure/Union member

`pycryptoki.cryptoki.CK_C_INITIALIZE_ARGS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_C_INITIALIZE_ARGS`

`pycryptoki.cryptoki.CK_CA_SetLKCV`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_AES_CBC_PAD_INSERT_PARAMS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_AES_CBC_PAD_INSERT_PARAMS`

`pycryptoki.cryptoki.CK_CA_CloseSecondarySession`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_DismantleRemotePED(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_ResetPIN(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_MTKRestore`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_WTLS_KEY_MAT_OUT_PTR`
alias of `pycryptoki.cryptoki.LP_CK_WTLS_KEY_MAT_OUT`

class `pycryptoki.cryptoki.CK_PKCS5_PBKD2_PARAMS`

iterations

Structure/Union member

pPassword

Structure/Union member

pPrfData
Structure/Union member

pSaltSourceData
Structure/Union member

prf
Structure/Union member

saltSource
Structure/Union member

ulPrfDataLen
Structure/Union member

ulSaltSourceDataLen
Structure/Union member

usPasswordLen
Structure/Union member

`pycryptoki.cryptoki.CK_C_Verify`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_CloneMofN(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_GenerateMofN`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_GetPrimarySlot`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.Int64`
alias of `ctypes.c_long`

class `pycryptoki.cryptoki.CK_X9_42_DH2_DERIVE_PARAMS`

hPrivateKey
Structure/Union member

kdf
Structure/Union member

pOtherInfo
Structure/Union member

pPublicData
Structure/Union member

pPublicData2
Structure/Union member

ulOtherInfoLen
Structure/Union member

ulPrivateKeyLen
Structure/Union member

ulPublicDataLen

Structure/Union member

ulPublicDataLen2

Structure/Union member

`pycryptoki.cryptoki.CA_ClonePrivateKey(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_CloseSecondarySession(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_OpenApplicationIDForContainer`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_DeleteContainer(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_DeleteRemotePEDVector(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_UnlockClusteredSlot`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_GetTokenCapabilities`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_CloneModifyMofN`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_STATE`

alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CK_CA_GetHSMCapabilitySetting`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_C_UnwrapKey`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_GetContainerList`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_MultisignValue(*args)`

Parameters

- ***args** –

- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_DisableUnauthTokenInsertion`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_SetPedId`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.C_Verify(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_Extract`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_C_SetAttributeValue`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_GetHSMCapabilitySet`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_SIMMultiSign(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.C_DecryptFinal(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.Byte`
alias of `ctypes.c_ubyte`

`pycryptoki.cryptoki.CK_CA_CloneAsSource`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_FUNCTION_LIST_PTR`
alias of `pycryptoki.cryptoki.LP_CK_FUNCTION_LIST`

`pycryptoki.cryptoki.CA_ActivateMofN(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_HAGetMasterPublic`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

class `pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST`

`CA_ActivateMofN`

Structure/Union member

CA_AuthorizeKey
Structure/Union member

CA_CapabilityUpdate
Structure/Union member

CA_CheckOperationState
Structure/Union member

CA_ChoosePrimarySlot
Structure/Union member

CA_ChooseSecondarySlot
Structure/Union member

CA_CloneAllObjectsToSession
Structure/Union member

CA_CloneAsSource
Structure/Union member

CA_CloneAsTarget
Structure/Union member

CA_CloneAsTargetInit
Structure/Union member

CA_CloneModifyMofN
Structure/Union member

CA_CloneMofN
Structure/Union member

CA_CloneObject
Structure/Union member

CA_CloneObjectToAllSessions
Structure/Union member

CA_ClonePrivateKey
Structure/Union member

CA_CloseAllSecondarySessions
Structure/Union member

CA_CloseApplicationID
Structure/Union member

CA_CloseApplicationIDForContainer
Structure/Union member

CA_CloseSecondarySession
Structure/Union member

CA_CloseSecureToken
Structure/Union member

CA_ConfigureRemotePED
Structure/Union member

CA_CreateContainer
Structure/Union member

CA_CreateContainerLoginChallenge

Structure/Union member

CA_CreateLoginChallenge

Structure/Union member

CA_Deactivate

Structure/Union member

CA_DeactivateMofN

Structure/Union member

CA_DeleteContainer

Structure/Union member

CA_DeleteContainerWithHandle

Structure/Union member

CA_DeleteRemotePEDVector

Structure/Union member

CA_DescribeUtilizationBinId

Structure/Union member

CA_DestroyMultipleObjects

Structure/Union member

CA_DisableUnauthTokenInsertion

Structure/Union member

CA_DismantleRemotePED

Structure/Union member

CA_DuplicateMofN

Structure/Union member

CA_EnableUnauthTokenInsertion

Structure/Union member

CA_EncodeECChar2Params

Structure/Union member

CA_EncodeECPParamsFromFile

Structure/Union member

CA_EncodeECPrimeParams

Structure/Union member

CA_Extract

Structure/Union member

CA_ExtractMaskedObject

Structure/Union member

CA_FactoryReset

Structure/Union member

CA_FindAdminSlotForSlot

Structure/Union member

CA_FirmwareRollback

Structure/Union member

CA_FirmwareUpdate
Structure/Union member

CA_GenerateCloneableMofN
Structure/Union member

CA_GenerateCloningKEV
Structure/Union member

CA_GenerateMofN
Structure/Union member

CA_GenerateTokenKeys
Structure/Union member

CA_GetClusterState
Structure/Union member

CA_GetConfigurationElementDescription
Structure/Union member

CA_GetContainerCapabilitySet
Structure/Union member

CA_GetContainerCapabilitySetting
Structure/Union member

CA_GetContainerList
Structure/Union member

CA_GetContainerName
Structure/Union member

CA_GetContainerPolicySet
Structure/Union member

CA_GetContainerPolicySetting
Structure/Union member

CA_GetContainerStatus
Structure/Union member

CA_GetContainerStorageInformation
Structure/Union member

CA_GetExtendedTPV
Structure/Union member

CA_GetFPV
Structure/Union member

CA_GetFirmwareVersion
Structure/Union member

CA_GetFunctionList
Structure/Union member

CA_GetHState
Structure/Union member

CA_GetHSMCapabilitySet
Structure/Union member

CA_GetHSMCapabilitySetting
Structure/Union member

CA_GetHSMPolicySet
Structure/Union member

CA_GetHSMPolicySetting
Structure/Union member

CA_GetHSMStats
Structure/Union member

CA_GetHSMStorageInformation
Structure/Union member

CA_GetModuleInfo
Structure/Union member

CA_GetModuleList
Structure/Union member

CA_GetMofNStatus
Structure/Union member

CA_GetNumberOfAllowedContainers
Structure/Union member

CA_GetObjectHandle
Structure/Union member

CA_GetObjectUID
Structure/Union member

CA_GetPedId
Structure/Union member

CA_GetPrimarySlot
Structure/Union member

CA_GetRemotePEDVectorStatus
Structure/Union member

CA_GetRollbackFirmwareVersion
Structure/Union member

CA_GetSecondarySlot
Structure/Union member

CA_GetSecureElementMeta
Structure/Union member

CA_GetServerInstanceBySlotID
Structure/Union member

CA_GetSessionInfo
Structure/Union member

CA_GetSlotIdForContainer
Structure/Union member

CA_GetSlotIdForPhysicalSlot
Structure/Union member

CA_GetSlotListFromServerInstance
Structure/Union member

CA_GetTPV
Structure/Union member

CA_GetTSV
Structure/Union member

CA_GetTime
Structure/Union member

CA_GetTokenCapabilities
Structure/Union member

CA_GetTokenCertificateInfo
Structure/Union member

CA_GetTokenCertificates
Structure/Union member

CA_GetTokenInsertionCount
Structure/Union member

CA_GetTokenObjectHandle
Structure/Union member

CA_GetTokenObjectUID
Structure/Union member

CA_GetTokenPolicies
Structure/Union member

CA_GetTokenStatus
Structure/Union member

CA_GetTokenStorageInformation
Structure/Union member

CA_GetTunnelSlotNumber
Structure/Union member

CA_GetUnauthTokenInsertionStatus
Structure/Union member

CA_GetUserContainerName
Structure/Union member

CA_GetUserContainerNumber
Structure/Union member

CA_HAActivateMofN
Structure/Union member

CA_HAAnswerLoginChallenge
Structure/Union member

CA_HAAnswerMofNChallenge
Structure/Union member

CA_HAGetLoginChallenge
Structure/Union member

CA_HAGetMasterPublic
Structure/Union member

CA_HAInit
Structure/Union member

CA_HALogin
Structure/Union member

CA_IndirectLogin
Structure/Union member

CA_InitAudit
Structure/Union member

CA_InitIndirectPIN
Structure/Union member

CA_InitIndirectToken
Structure/Union member

CA_InitRolePIN
Structure/Union member

CA_InitSlotRolePIN
Structure/Union member

CA_InitializeRemotePEDVector
Structure/Union member

CA_Insert
Structure/Union member

CA_InsertMaskedObject
Structure/Union member

CA_InvokeService
Structure/Union member

CA_InvokeServiceAsynch
Structure/Union member

CA_InvokeServiceFinal
Structure/Union member

CA_InvokeServiceInit
Structure/Union member

CA_InvokeServiceSinglePart
Structure/Union member

CA_IsMofNEnabled
Structure/Union member

CA_IsMofNRequired
Structure/Union member

CA_LKMInitiatorChallenge
Structure/Union member

CA_LKMInitiatorComplete
Structure/Union member

CA_LKMReceiverComplete
Structure/Union member

CA_LKMReceiverResponse
Structure/Union member

CA_ListSecureTokenInit
Structure/Union member

CA_ListSecureTokenUpdate
Structure/Union member

CA_LoadEncryptedModule
Structure/Union member

CA_LoadModule
Structure/Union member

CA_LockClusteredSlot
Structure/Union member

CA_LogExportSecret
Structure/Union member

CA_LogExternal
Structure/Union member

CA_LogGetConfig
Structure/Union member

CA_LogGetStatus
Structure/Union member

CA_LogImportSecret
Structure/Union member

CA_LogSetConfig
Structure/Union member

CA_LogVerify
Structure/Union member

CA_LogVerifyFile
Structure/Union member

CA_MTKGetState
Structure/Union member

CA_MTKResplit
Structure/Union member

CA_MTKRestore
Structure/Union member

CA_MTKSetStorage
Structure/Union member

CA_MTKZeroize
Structure/Union member

CA_ManualKCV
Structure/Union member

CA_ModifyMofN
Structure/Union member

CA_ModifyUsageCount
Structure/Union member

CA_MultisignValue
Structure/Union member

CA_OpenApplicationID
Structure/Union member

CA_OpenApplicationIDForContainer
Structure/Union member

CA_OpenSecureToken
Structure/Union member

CA_OpenSession
Structure/Union member

CA_OpenSessionWithAppID
Structure/Union member

CA_PerformModuleCall
Structure/Union member

CA_PerformSelfTest
Structure/Union member

CA_QueryLicense
Structure/Union member

CA_ReadAllUtilizationCounters
Structure/Union member

CA_ReadAndResetUtilizationMetrics
Structure/Union member

CA_ReadCommonStore
Structure/Union member

CA_ReadUtilizationMetrics
Structure/Union member

CA_ReplaceFastPathKEK
Structure/Union member

CA_ResetDevice
Structure/Union member

CA_ResetPIN
Structure/Union member

CA_Restart
Structure/Union member

CA_RestartForContainer
Structure/Union member

CA_RetrieveLicenseList
Structure/Union member

CA_RoleStateGet
Structure/Union member

CA_SIMExtract
Structure/Union member

CA_SIMInsert
Structure/Union member

CA_SIMMultiSign
Structure/Union member

CA_STCClearCipherAlgorithm
Structure/Union member

CA_STCClearDigestAlgorithm
Structure/Union member

CA_STCDeregister
Structure/Union member

CA_STCGetAdminPubKey
Structure/Union member

CA_STCGetChannelID
Structure/Union member

CA_STCGetCipherAlgorithm
Structure/Union member

CA_STCGetCipherID
Structure/Union member

CA_STCGetCipherIDs
Structure/Union member

CA_STCGetCipherNameByID
Structure/Union member

CA_STCGetClientInfo
Structure/Union member

CA_STCGetClientsList
Structure/Union member

CA_STCGetCurrentKeyLife
Structure/Union member

CA_STCGetDigestAlgorithm
Structure/Union member

CA_STCGetDigestID
Structure/Union member

CA_STCGetDigestIDs
Structure/Union member

CA_STCGetDigestNameByID
Structure/Union member

CA_STCGetKeyActivationTimeOut
Structure/Union member

CA_STCGetKeyLifeTime
Structure/Union member

CA_STCGetMaxSessions
Structure/Union member

CA_STCGetPartPubKey
Structure/Union member

CA_STCGetPubKey
Structure/Union member

CA_STCGetSequenceWindowSize
Structure/Union member

CA_STCGetState
Structure/Union member

CA_STCIsEnabled
Structure/Union member

CA_STCRegister
Structure/Union member

CA_STCSetCipherAlgorithm
Structure/Union member

CA_STCSetDigestAlgorithm
Structure/Union member

CA_STCSetKeyActivationTimeOut
Structure/Union member

CA_STCSetKeyLifeTime
Structure/Union member

CA_STCSetMaxSessions
Structure/Union member

CA_STCSetSequenceWindowSize
Structure/Union member

CA_STMGetState
Structure/Union member

CA_STMToggle
Structure/Union member

CA_SetApplicationID
Structure/Union member

CA_SetAuthorizationData
Structure/Union member

CA_SetCloningDomain
Structure/Union member

CA_SetContainerPolicies
Structure/Union member

CA_SetContainerPolicy
Structure/Union member

CA_SetContainerSize
Structure/Union member

CA_SetDestructiveHSMPolicies
Structure/Union member

CA_SetDestructiveHSMPolicy
Structure/Union member

CA_SetExtendedTPV
Structure/Union member

CA_SetHSMPolicies
Structure/Union member

CA_SetHSMPolicy
Structure/Union member

CA_SetKCV
Structure/Union member

CA_SetLKCV
Structure/Union member

CA_SetMofN
Structure/Union member

CA_SetPedId
Structure/Union member

CA_SetRDK
Structure/Union member

CA_SetTPV
Structure/Union member

CA_SetTokenCertificateSignature
Structure/Union member

CA_SetTokenPolicies
Structure/Union member

CA_SetUserContainerName
Structure/Union member

CA_SpRawRead
Structure/Union member

CA_SpRawWrite
Structure/Union member

CA_SwitchSecondarySlot
Structure/Union member

CA_TimeSync
Structure/Union member

CA_TokenDelete
Structure/Union member

CA_TokenInsert
Structure/Union member

CA-TokenInsertNoAuth
Structure/Union member

CA-TokenZeroize
Structure/Union member

CA_UnloadModule
Structure/Union member

CA_UnlockClusteredSlot
Structure/Union member

CA_WaitForSlotEvent
Structure/Union member

CA_WriteCommonStore
Structure/Union member

CA_Zeroize
Structure/Union member

version
Structure/Union member

`pycryptoki.cryptoki.CK_CA_OpenApplicationID`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_AES_GMAC_PARAMS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_AES_GCM_PARAMS`

`pycryptoki.cryptoki.CK_CAMELLIA_CTR_PARAMS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_CAMELLIA_CTR_PARAMS`

`pycryptoki.cryptoki.CA_RetrieveLicenseList(*args)`

Parameters

- ***args** –
- ****kwargs** –

class `pycryptoki.cryptoki.CK_SSL3_RANDOM_DATA`

pClientRandom
Structure/Union member

pServerRandom
Structure/Union member

ulClientRandomLen
Structure/Union member

ulServerRandomLen
Structure/Union member

`pycryptoki.cryptoki.CK_CA_SetContainerSize`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

class `pycryptoki.cryptoki.CK_SSL3_KEY_MAT_PARAMS`

RandomInfo
Structure/Union member

bIsExport

Structure/Union member

pReturnedKeyMaterial

Structure/Union member

ulIVSizeInBits

Structure/Union member

ulKeySizeInBits

Structure/Union member

ulMacSizeInBits

Structure/Union member

`pycryptoki.cryptoki.CK_KIP_PARAMS_PTR`

alias of `pycryptoki.cryptoki.LP_CK_KIP_PARAMS`

`pycryptoki.cryptoki.CK_CA_WaitForSlotEvent`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_OTP_SIGNATURE_INFO_PTR`

alias of `pycryptoki.cryptoki.LP_CK_OTP_SIGNATURE_INFO`

`pycryptoki.cryptoki.CA_GetSessionInfo(*args)`

Parameters

- ***args** –
- ****kwargs** –

class `pycryptoki.cryptoki.CK_WTLS_RANDOM_DATA`

pClientRandom

Structure/Union member

pServerRandom

Structure/Union member

ulClientRandomLen

Structure/Union member

ulServerRandomLen

Structure/Union member

`pycryptoki.cryptoki.CK_USHORT`

alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CK_CA_LoadEncryptedModule`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_SetTPV(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST_PTR_PTR`

alias of `pycryptoki.cryptoki.LP_LP_CK_SFNT_CA_FUNCTION_LIST`

`pycryptoki.cryptoki.CA_GetUserContainerName(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_C_GetFunctionStatus`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_STCGetCipherNameByID`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_PRF_KDF_PARAMS`
alias of `pycryptoki.cryptoki.CK_KDF_PRF_PARAMS`

`pycryptoki.cryptoki.CK_CA_Deactivate`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_X9_42_DH1_DERIVE_PARAMS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_X9_42_DH1_DERIVE_PARAMS`

`pycryptoki.cryptoki.CK_C_VerifyFinal`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_ChooseSecondarySlot(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_GetUserContainerNumber(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.UInt`
alias of `ctypes.c_uint`

`pycryptoki.cryptoki.CK_RSA_PKCS_OAEP_SOURCE_TYPE_PTR`
alias of `pycryptoki.cryptoki.LP_c_ulong`

`pycryptoki.cryptoki.CA_GetHSMCapabilitySetting(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_GetTokenObjectHandle`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_GetUnauthTokenInsertionStatus`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.fwResultCode`
alias of `ctypes.c_int`

`pycryptoki.cryptoki.CA_MTKGetState(*args)`

Parameters

- ***args** –

- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_CloneObject`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_MECHANISM_TYPE`
alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CA_SetKCV(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_GetContainerStorageInformation`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

class `pycryptoki.cryptoki.CK_ATTRIBUTE`

pValue
Structure/Union member

type
Structure/Union member

usValueLen
Structure/Union member

`pycryptoki.cryptoki.CK_CA_SetDestructiveHSM Policies`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_SwitchSecondarySlot`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

class `pycryptoki.cryptoki.CK_MECHANISM`

mechanism
Structure/Union member

pParameter
Structure/Union member

usParameterLen
Structure/Union member

`pycryptoki.cryptoki.CA_RoleStateGet(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.C_Encrypt(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_C_SignUpdate`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

```

pycryptoki.cryptoki.CK_INFO_PTR
    alias of pycryptoki.cryptoki.LP_CK_INFO
pycryptoki.cryptoki.CK_ARIA_CTR_PARAMS_PTR
    alias of pycryptoki.cryptoki.LP_CK_AES_CTR_PARAMS
pycryptoki.cryptoki.CK_C_SeedRandom
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
pycryptoki.cryptoki.CA_GetHState(*args)

```

Parameters

- ***args** –
- ****kwargs** –

```

pycryptoki.cryptoki.C_SignRecoverInit(*args)

```

Parameters

- ***args** –
- ****kwargs** –

```

pycryptoki.cryptoki.CK_CA_STCGetClientsList
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
pycryptoki.cryptoki.CK_C_FindObjectsFinal
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
pycryptoki.cryptoki.CK_CA_GetContainerPolicySetting
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
pycryptoki.cryptoki.CK_CA_STCSetKeyLifeTime
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
pycryptoki.cryptoki.CK_BYTE
    alias of ctypes.c_ubyte
class pycryptoki.cryptoki.CK_SSL3_KEY_MAT_OUT

```

hClientKey

Structure/Union member

hClientMacSecret

Structure/Union member

hServerKey

Structure/Union member

hServerMacSecret

Structure/Union member

pIVClient

Structure/Union member

pIVServer

Structure/Union member

```

pycryptoki.cryptoki.CK_GetTotalOperations
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
pycryptoki.cryptoki.CK_SLOT_INFO_PTR
    alias of pycryptoki.cryptoki.LP_CK_SLOT_INFO

```

`pycryptoki.cryptoki.CK_CA_GetObjectHandle`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_GetSlotListFromServerInstance`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CKCA_MODULE_INFO_PTR`
alias of `pycryptoki.cryptoki.LP_CKCA_MODULE_INFO`

`pycryptoki.cryptoki.CK_KEA_DERIVE_PARAMS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_KEA_DERIVE_PARAMS`

`pycryptoki.cryptoki.CK_BYTE_PTR`
alias of `pycryptoki.cryptoki.LP_c_ubyte`

`pycryptoki.cryptoki.CA_GetServerInstanceBySlotID(*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.CA_SetContainerSize(*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.CA_GetContainerPolicySetting(*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.CK_CA_CloseAllSecondarySessions`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_GenerateTokenKeys(*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.CA_TimeSync(*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.CA_LKMInitiatorChallenge(*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.HalfWord`
alias of `ctypes.c_ushort`

`pycryptoki.cryptoki.CK_C_GetMechanismList`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_EncodeECPrimeParams`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_VOID_PTR_PTR`
alias of `pycryptoki.cryptoki.LP_c_void_p`

`pycryptoki.cryptoki.CA_MOFN_STATUS`
alias of `pycryptoki.cryptoki.CA_M_OF_N_STATUS`

`pycryptoki.cryptoki.CK_CA_FindAdminSlotForSlot`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_C_CopyObject`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_CreateLoginChallenge`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_STCGetMaxSessions`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CT_TokenHndle`
alias of `pycryptoki.cryptoki.LP_CT_Token`

`pycryptoki.cryptoki.C_SetPIN(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_InitIndirectToken(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.C_GenerateKey(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_STCGetCipherIDs`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.C_InitPIN(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_C_GetSlotInfo`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

class `pycryptoki.cryptoki.CK_ECIES_PARAMS`

dhPrimitive
Structure/Union member

encScheme
Structure/Union member

kdf
Structure/Union member

macScheme
Structure/Union member

pSharedData1
Structure/Union member

pSharedData2
Structure/Union member

ulEncKeyLenInBits
Structure/Union member

ulMacKeyLenInBits
Structure/Union member

ulMacLenInBits
Structure/Union member

ulSharedDataLen1
Structure/Union member

ulSharedDataLen2
Structure/Union member

class pycryptoki.cryptoki.CK_AES_CTR_PARAMS

cb
Structure/Union member

ulCounterBits
Structure/Union member

class pycryptoki.cryptoki.CK_LKM_TOKEN_ID_S

id
Structure/Union member

pycryptoki.cryptoki.CA_SetExtendedTPV(*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.CK_CA_STCRegister
alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.CA_HAInit(*args)

Parameters

- ***args** –
- ****kwargs** –


```
pycryptoki.cryptoki.CK_X9_42_DH2_DERIVE_PARAMS_PTR
    alias of pycryptoki.cryptoki.LP_CK_X9_42_DH2_DERIVE_PARAMS
```

```
pycryptoki.cryptoki.CK_CA_TimeSync
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CK_CA_DeleteRemotePEDVector
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CA_LogImportSecret(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CK_KEY_WRAP_SET_OAEP_PARAMS_PTR
    alias of pycryptoki.cryptoki.LP_CK_KEY_WRAP_SET_OAEP_PARAMS
```

```
pycryptoki.cryptoki.CA_STCGetDigestNameByID(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CK_CA_GetTokenInsertionCount
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CK_PARAM_TYPE
    alias of ctypes.c_ulong
```

```
pycryptoki.cryptoki.CA_GetContainerName(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CK_CA_ChooseSecondarySlot
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CA_MOFN_STATUS_PTR
    alias of pycryptoki.cryptoki.LP_CA_M_OF_N_STATUS
```

```
pycryptoki.cryptoki.CA_FindAdminSlotForSlot(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.ResultCodeValue
    alias of ctypes.c_int
```

```
class pycryptoki.cryptoki.CK_ECDH1_DERIVE_PARAMS
```

kdf

Structure/Union member

pPublicData

Structure/Union member

pSharedData

Structure/Union member

ulPublicDataLen

Structure/Union member

ulSharedDataLen

Structure/Union member

`pycryptoki.cryptoki.CK_RC2_PARAMS_PTR`

alias of `pycryptoki.cryptoki.LP_c_ulong`

`pycryptoki.cryptoki.CK_WTLS_PRF_PARAMS_PTR`

alias of `pycryptoki.cryptoki.LP_CK_WTLS_PRF_PARAMS`

`pycryptoki.cryptoki.C_FindObjectsFinal(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_C_CancelFunction`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_GetContainerStatus`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_C_VerifyRecover`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_RC2_CBC_PARAMS_PTR`

alias of `pycryptoki.cryptoki.LP_CK_RC2_CBC_PARAMS`

`pycryptoki.cryptoki.CA_STCSetMaxSessions(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_GetContainerStatus(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_PerformSelfTest(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_STCGetCipherID(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_EncodeECPrimeParams(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_GetContainerCapabilitySet`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_GetConfigurationElementDescription(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.C_Login(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_CloneAllObjectsToSession`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_STCGetClientInfo(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.C_CreateObject(*args)`

Parameters

- ***args** –
- ****kwargs** –

`class pycryptoki.cryptoki.CK_KEA_DERIVE_PARAMS`

isSender

Structure/Union member

pPublicData

Structure/Union member

pRandomA

Structure/Union member

pRandomB

Structure/Union member

ulPublicDataLen

Structure/Union member

ulRandomLen

Structure/Union member

`pycryptoki.cryptoki.CK_CA_FirmwareUpdate`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

```
pycryptoki.cryptoki.CK_CA_OpenSession
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CA_Restart(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.UInt64
    alias of ctypes.c_ulong
```

```
pycryptoki.cryptoki.CK_C_Sign
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CK_CA_ReadCommonStore
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CKCA_MODULE_ID_PTR
    alias of pycryptoki.cryptoki.LP_c_ulong
```

```
pycryptoki.cryptoki.CK_LONG
    alias of ctypes.c_long
```

```
class pycryptoki.cryptoki.CA_MOFN_GENERATION
```

pVector

Structure/Union member

ulVectorLen

Structure/Union member

ulWeight

Structure/Union member

```
pycryptoki.cryptoki.CK_CA_ListSecureTokenUpdate
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CK_CA_SetKCV
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CK_OBJECT_HANDLE_PTR
    alias of pycryptoki.cryptoki.LP_c_ulong
```

```
pycryptoki.cryptoki.CA_LogExportSecret(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CK_CA_InvokeServiceFinal
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CK_CA_HAInit
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.Int
    alias of ctypes.c_int
```

```
pycryptoki.cryptoki.CA_LKMReceiverResponse(*args)
```

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_STCClearDigestAlgorithm`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_C_InitPIN`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

class `pycryptoki.cryptoki.CK_AES_CBC_PAD_EXTRACT_PARAMS`

ctxID

Structure/Union member

pBuffer

Structure/Union member

pbFileName

Structure/Union member

pedId

Structure/Union member

pulBufferLen

Structure/Union member

ulDeleteAfterExtract

Structure/Union member

ulHandle

Structure/Union member

ulStorage

Structure/Union member

ulType

Structure/Union member

`pycryptoki.cryptoki.CK_SKIPJACK_RELAYX_PARAMS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_SKIPJACK_RELAYX_PARAMS`

`pycryptoki.cryptoki.CA_GetModuleInfo(*args)`

Parameters

- ***args** –
- ****kwargs** –

class `pycryptoki.cryptoki.CK_TLS_PRF_PARAMS`

pLabel

Structure/Union member

pOutput

Structure/Union member

pSeed

Structure/Union member

pulOutputLen

Structure/Union member

ulLabelLen

Structure/Union member

ulSeedLen

Structure/Union member

`pycryptoki.cryptoki.CA_GetSecureElementMeta(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_SLOT_ID`

alias of `ctypes.c_ulong`

class `pycryptoki.cryptoki.CT_Token`

`pycryptoki.cryptoki.CK_CA_LogExternal`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_CloseSecureToken`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_ListSecureTokenInit`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_C_Digest`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_C_Finalize`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_GetTPV(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_EncodeECParamsFromFile`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_RestartForContainer`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_GetNumberOfAllowedContainers(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_ChoosePrimarySlot(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.C_VerifyInit(*args)`

Parameters

- ***args** –
- ****kwargs** –

```
class pycryptoki.cryptoki.CK_SKIPJACK_PRIVATE_WRAP_PARAMS
```

pBaseG

Structure/Union member

pPassword

Structure/Union member

pPrimeP

Structure/Union member

pPublicData

Structure/Union member

pRandomA

Structure/Union member

pSubprimeQ

Structure/Union member

ulPAndGLen

Structure/Union member

ulPublicDataLen

Structure/Union member

ulQLen

Structure/Union member

ulRandomLen

Structure/Union member

usPasswordLen

Structure/Union member

```
pycryptoki.cryptoki.CA_SetTokenPolicies(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CKCA_MODULE_ID
```

alias of `ctypes.c_ulong`

```
pycryptoki.cryptoki.CK_CA_DeleteContainerWithHandle
```

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

```
pycryptoki.cryptoki.CK_LOCKMUTEX
```

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

```
pycryptoki.cryptoki.CK_CA_GenerateCloningKEV
```

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

```
pycryptoki.cryptoki.CK_C_DecryptUpdate
```

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

```
class pycryptoki.cryptoki.CA_M_OF_N_STATUS
```

```
    ulFlag
        Structure/Union member
```

```
    ulID
        Structure/Union member
```

```
    ulM
        Structure/Union member
```

```
    ulN
        Structure/Union member
```

```
    ulSecretSize
        Structure/Union member
```

```
pycryptoki.cryptoki.CK_C_CloseSession
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CK_EC_ENC_SCHEME
    alias of ctypes.c_ulong
```

```
pycryptoki.cryptoki.CK_MECHANISM_INFO_PTR
    alias of pycryptoki.cryptoki.LP_CK_MECHANISM_INFO
```

```
pycryptoki.cryptoki.CK_OTP_PARAM_TYPE
    alias of ctypes.c_ulong
```

```
pycryptoki.cryptoki.CA_UnloadModule(*args)
```

Parameters

- **args* –
- ***kwargs* –

```
pycryptoki.cryptoki.CK_CA_PerformModuleCall
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CA_GetTokenInsertionCount(*args)
```

Parameters

- **args* –
- ***kwargs* –

```
pycryptoki.cryptoki.CK_CA_ResetDevice
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CK_CA_GetHSMPolicySet
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CA_TokenDelete(*args)
```

Parameters

- **args* –
- ***kwargs* –

```
pycryptoki.cryptoki.CK_AES_GMAC_PARAMS
    alias of pycryptoki.cryptoki.CK_AES_GCM_PARAMS
```



```

pycryptoki.cryptoki.CK_PBE_PARAMS_PTR
    alias of pycryptoki.cryptoki.LP_CK_PBE_PARAMS

class pycryptoki.cryptoki.CK_ARIA_CBC_ENCRYPT_DATA_PARAMS

    iv
        Structure/Union member

    length
        Structure/Union member

    pData
        Structure/Union member

pycryptoki.cryptoki.C_SeedRandom(*args)

    Parameters
        • *args –
        • **kwargs –

pycryptoki.cryptoki.CA_RestartForContainer(*args)

    Parameters
        • *args –
        • **kwargs –

pycryptoki.cryptoki.CA_STCClearDigestAlgorithm(*args)

    Parameters
        • *args –
        • **kwargs –

pycryptoki.cryptoki.CK_CA_STCDeregister
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.CK_CA_STCGetDigestIDs
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.CK_CA_GetTokenCertificates
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.HANDLE
    alias of ctypes.c_int

pycryptoki.cryptoki.CK_CA_TokenInsert
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.CA_ManualKCV(*args)

    Parameters
        • *args –
        • **kwargs –

pycryptoki.cryptoki.C_CancelFunction(*args)

    Parameters
        • *args –

```

- ****kwargs** –

class pycryptoki.cryptoki.**CK_HA_STATUS**

groupSerial

Structure/Union member

listSize

Structure/Union member

memberList

Structure/Union member

pycryptoki.cryptoki.**CK_C_DigestKey**

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

pycryptoki.cryptoki.**C_Initialize**(*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.**CK_RSA_PKCS_OAEP_PARAMS_PTR**

alias of `pycryptoki.cryptoki.LP_CK_RSA_PKCS_OAEP_PARAMS`

pycryptoki.cryptoki.**C_InitToken**(*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.**CK_CA_ActivateMofN**

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

pycryptoki.cryptoki.**C_GetSlotList**(*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.**C_GetMechanismInfo**(*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.**CA_STCGetChannelID**(*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.**CK_C_EncryptUpdate**

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

pycryptoki.cryptoki.**Boolean**

alias of `ctypes.c_ubyte`

`pycryptoki.cryptoki.CK_CA_Zeroize`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_WTLS_KEY_MAT_PARAMS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_WTLS_KEY_MAT_PARAMS`

`pycryptoki.cryptoki.CK_CA_InitIndirectToken`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_GetTSV(*args)`

Parameters

- ***args** –
- ****kwargs** –

class `pycryptoki.cryptoki.CK_RC5_PARAMS`

ulRounds

Structure/Union member

ulWordsize

Structure/Union member

`pycryptoki.cryptoki.C_SignFinal(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_C_GenerateKey`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_UnlockClusteredSlot(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_AES_CTR_PARAMS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_AES_CTR_PARAMS`

`pycryptoki.cryptoki.CA_FirmwareUpdate(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_USHORT_PTR`
alias of `pycryptoki.cryptoki.LP_c_ulong`

`pycryptoki.cryptoki.CK_PKCS5_PBKD2_PARAMS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_PKCS5_PBKD2_PARAMS`

`pycryptoki.cryptoki.CK_C_DeriveKey`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_GetContainerPolicySet`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_SIMInsert(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_SetHSMPolicies`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_AES_CBC_PAD_EXTRACT_PARAMS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_AES_CBC_PAD_EXTRACT_PARAMS`

`pycryptoki.cryptoki.CK_CA_RoleStateGet`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_C_VerifyRecoverInit`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA-TokenInsertNoAuth`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA-TokenInsertNoAuth(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_ECDH2_DERIVE_PARAMS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_ECDH2_DERIVE_PARAMS`

`pycryptoki.cryptoki.CK_CA-ChoosePrimarySlot`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA-IndirectLogin`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA-InvokeServiceAsynch`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA-GetTokenPolicies`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA-InitAudit`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA-SIMMultiSign`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

class `pycryptoki.cryptoki.CK_DES_CBC_ENCRYPT_DATA_PARAMS`

iv
Structure/Union member

length
Structure/Union member

pData
Structure/Union member

class `pycryptoki.cryptoki.CK_CAMELLIA_CBC_ENCRYPT_DATA_PARAMS`

iv
Structure/Union member

length
Structure/Union member

pData
Structure/Union member

`pycryptoki.cryptoki.CK_CA_GetHSMStats`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.C_GenerateKeyPair(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_MTKSetStorage`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CKA_SIM_AUTH_FORM`
alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CK_C_SignRecover`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_HW_FEATURE_TYPE`
alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CK_CA_ExtractMaskedObject`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

class `pycryptoki.cryptoki.CA_ROLE_STATE`

flags
Structure/Union member

loginAttemptsLeft
Structure/Union member

primaryAuthMech
Structure/Union member

secondaryAuthMech
Structure/Union member

`pycryptoki.cryptoki.CA_GetTokenCertificates(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CLUSTER_STATE_PTR`
alias of `pycryptoki.cryptoki.LP_CK_CLUSTER_STATE`

`pycryptoki.cryptoki.CA_CheckOperationState(*args)`

Parameters

- ***args** –

- ****kwargs** –

`pycryptoki.cryptoki.C_GetTokenInfo(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_SetTokenPolicies`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_CloneAsTarget(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_SetCloningDomain(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_GetObjectUID(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_VERSION_PTR`

alias of `pycryptoki.cryptoki.LP_CK_VERSION`

`pycryptoki.cryptoki.CK_CA_OpenSessionWithAppID`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.C_DecryptInit(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_STCDeregister(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.SInt32`

alias of `ctypes.c_long`

`pycryptoki.cryptoki.CK_CA_STCGetCipherID`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_STCSetKeyActivationTimeOut(*args)`

Parameters

- ***args** –

- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_CreateContainer`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_ULONG_PTR`
alias of `pycryptoki.cryptoki.LP_c_ulong`

`pycryptoki.cryptoki.CK_KDF_PRF_PARAMS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_KDF_PRF_PARAMS`

`pycryptoki.cryptoki.CA_STCGetDigestAlgorithm(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_GetTokenStatus`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_CreateContainerWithPolicy(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_GetContainerCapabilitySetting(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_DeriveKeyAndWrap(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_InvokeServiceAsynch(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_C_GetInfo`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_SetApplicationID`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

class `pycryptoki.cryptoki.CK_AES_CBC_ENCRYPT_DATA_PARAMS`

iv

Structure/Union member

length

Structure/Union member

pData

Structure/Union member

```

pycryptoki.cryptoki.CK_C_DecryptVerifyUpdate
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.CK_CA_STMToggle
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.CK_CA_DestroyMultipleObjects
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.CK_KEY_DERIVATION_STRING_DATA_PTR
    alias of pycryptoki.cryptoki.LP_CK_KEY_DERIVATION_STRING_DATA

pycryptoki.cryptoki.CK_CA_ReplaceFastPathKEK
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.CK_CA_InvokeServiceSinglePart
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.ULong
    alias of ctypes.c_ulong

pycryptoki.cryptoki.CK_CA_QueryLicense
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.CK_DES_CBC_ENCRYPT_DATA_PARAMS_PTR
    alias of pycryptoki.cryptoki.LP_CK_DES_CBC_ENCRYPT_DATA_PARAMS

pycryptoki.cryptoki.CK_SLOT_ID_PTR
    alias of pycryptoki.cryptoki.LP_c_ulong

pycryptoki.cryptoki.CA-TokenInsert(*args)

```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CA_IsMofNRRequired(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CA_HAGetMasterPublic(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CK_CA_LogVerifyFile
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CA_SetContainerPolicies(*args)
```

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_HAGetLoginChallenge`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_GenerateCloningKEV(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_SetHSMPolicy(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_GetTPV`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_SetTokenCertificateSignature`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_RV`
alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CK_NOTIFY`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_C_FindObjects`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_ModifyUsageCount(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.C_VerifyUpdate(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_LogImportSecret`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_X9_42_MQV_DERIVE_PARAMS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_X9_42_MQV_DERIVE_PARAMS`

`pycryptoki.cryptoki.CK_CA_ResetPIN`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_X9_42_DH_KDF_TYPE_PTR`
alias of `pycryptoki.cryptoki.LP_c_ulong`

`pycryptoki.cryptoki.CK_ARIA_CBC_ENCRYPT_DATA_PARAMS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_ARIA_CBC_ENCRYPT_DATA_PARAMS`

`pycryptoki.cryptoki.CK_C_Decrypt`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_CloneMofN`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_IsMofNRequired`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.C_Sign(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_STCClearCipherAlgorithm(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_C_DigestUpdate`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

class `pycryptoki.cryptoki.CK_X9_42_DH1_DERIVE_PARAMS`

kdf
Structure/Union member

pOtherInfo
Structure/Union member

pPublicData
Structure/Union member

ulOtherInfoLen
Structure/Union member

ulPublicDataLen
Structure/Union member

`pycryptoki.cryptoki.C_GetFunctionList(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_SetCloningDomain`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_C_Initialize`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_GetTokenObjectUID(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.C_GetOperationState(*args)`

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CK_C_GetSessionInfo
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.CK_BBOOL
    alias of ctypes.c_ubyte

pycryptoki.cryptoki.CK_CA_EncodeECChar2Params
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.CK_CA_STCGetChannelID
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.CK_CA_LogSetConfig
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.CA_GetRollbackFirmwareVersion(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CK_C_GenerateKeyPair
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.CK_CA_STCGetCurrentKeyLife
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.CK_CA_GetTokenObjectUID
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.CK_CA_PerformSelfTest
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

class pycryptoki.cryptoki.CK_ECDH2_DERIVE_PARAMS
```

```
hPrivateKey
    Structure/Union member

kdf
    Structure/Union member

pPublicKey
    Structure/Union member

pPublicKey2
    Structure/Union member

pSharedData
    Structure/Union member

ulPrivateKeyLen
    Structure/Union member

ulPublicKeyLen
    Structure/Union member

ulPublicKeyLen2
    Structure/Union member
```

ulSharedDataLen

Structure/Union member

`pycryptoki.cryptoki.CK_CA_CheckOperationState`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_SetMofN`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_DismantleRemotePED`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_GetTokenCapabilities(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_OBJECT_CLASS_PTR`

alias of `pycryptoki.cryptoki.LP_c_ulong`

`pycryptoki.cryptoki.CK_RC2_PARAMS`

alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CA_GetSecondarySlot(*args)`

Parameters

- ***args** –
- ****kwargs** –

class `pycryptoki.cryptoki.HSM_STATS_PARAMS`

ulHighValue

Structure/Union member

ulId

Structure/Union member

ulLowValue

Structure/Union member

`pycryptoki.cryptoki.CK_CA_SetDestructiveHSMPolicy`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_InvokeServiceInit`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_GetMofNStatus(*args)`

Parameters

- ***args** –
- ****kwargs** –

class `pycryptoki.cryptoki.CK_OTP_PARAM`

pValue

Structure/Union member

type
Structure/Union member

usValueLen
Structure/Union member

`pycryptoki.cryptoki.CA_STCRegister(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_SpRawRead`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_C_GetOperationState`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

class `pycryptoki.cryptoki.CK_TOKEN_INFO`

firmwareVersion
Structure/Union member

flags
Structure/Union member

hardwareVersion
Structure/Union member

label
Structure/Union member

manufacturerID
Structure/Union member

model
Structure/Union member

serialNumber
Structure/Union member

ulFreePrivateMemory
Structure/Union member

ulFreePublicMemory
Structure/Union member

ulTotalPrivateMemory
Structure/Union member

ulTotalPublicMemory
Structure/Union member

usMaxPinLen
Structure/Union member

usMaxRwSessionCount
Structure/Union member

usMaxSessionCount
Structure/Union member

usMinPinLen
Structure/Union member

usRwSessionCount
Structure/Union member

usSessionCount
Structure/Union member

utcTime
Structure/Union member

class pycryptoki.cryptoki.**CK_RSA_PKCS_OAEP_PARAMS**

hashAlg
Structure/Union member

mgf
Structure/Union member

pSourceData
Structure/Union member

source
Structure/Union member

ulSourceDataLen
Structure/Union member

pycryptoki.cryptoki.**CK_SSL3_KEY_MAT_PARAMS_PTR**
alias of pycryptoki.cryptoki.LP_CK_SSL3_KEY_MAT_PARAMS

pycryptoki.cryptoki.**CA_ReadCommonStore**(*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.**CK_CA_CapabilityUpdate**
alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.**CK_C_EncryptInit**
alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

pycryptoki.cryptoki.**C_Logout**(*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.**CA_MTKRestore**(*args)

Parameters

- ***args** –
- ****kwargs** –

pycryptoki.cryptoki.**CA_LKMInitiatorComplete**(*args)

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_CloneAsTargetInit(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_OpenApplicationIDForContainer(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_GetTunnelSlotNumber`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_LogSetConfig(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.SizeType`
alias of `ctypes.c_uint`

`pycryptoki.cryptoki.CK_CA_LogExportSecret`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_STCGetCipherIDs(*args)`

Parameters

- ***args** –
- ****kwargs** –

`class pycryptoki.cryptoki.CK_C_INITIALIZE_ARGS`

CreateMutex

Structure/Union member

DestroyMutex

Structure/Union member

LockMutex

Structure/Union member

UnlockMutex

Structure/Union member

flags

Structure/Union member

pReserved

Structure/Union member

`pycryptoki.cryptoki.CA_GetTokenCertificateInfo(*args)`

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.C_Decrypt(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CK_C_SignEncryptUpdate
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CA_GetExtendedTPV(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CA_GetContainerPolicySet(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CK_EXTRACT_PARAMS_PTR
    alias of pycryptoki.cryptoki.LP_c_ulong
```

```
pycryptoki.cryptoki.CK_CAMELLIA_CBC_ENCRYPT_DATA_PARAMS_PTR
    alias of pycryptoki.cryptoki.LP_CK_CAMELLIA_CBC_ENCRYPT_DATA_PARAMS
```

```
pycryptoki.cryptoki.CK_PKCS5_PBKDF2_SALT_SOURCE_TYPE_PTR
    alias of pycryptoki.cryptoki.LP_c_ulong
```

```
pycryptoki.cryptoki.C_DecryptDigestUpdate(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
class pycryptoki.cryptoki.CK_AES_XTS_PARAMS
```

cb

Structure/Union member

hTweakKey

Structure/Union member

```
pycryptoki.cryptoki.CA_Get(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
class pycryptoki.cryptoki.CK_AES_GCM_PARAMS
```


pAAD
Structure/Union member

pIv
Structure/Union member

ulAADLen
Structure/Union member

ulIvBits
Structure/Union member

ulIvLen
Structure/Union member

ulTagBits
Structure/Union member

`pycryptoki.cryptoki.CK_HA_STATE_PTR`
alias of `pycryptoki.cryptoki.LP_CK_HA_STATUS`

`pycryptoki.cryptoki.CA_LogGetConfig(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_STCGetCurrentKeyLife(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_SetHSMPolicies(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_STMToggle(*args)`

Parameters

- ***args** –
- ****kwargs** –

`class pycryptoki.cryptoki.CK_XOR_BASE_DATA_KDF_PARAMS`

kdf
Structure/Union member

pSharedData
Structure/Union member

ulSharedDataLen
Structure/Union member

`pycryptoki.cryptoki.C_Finalize(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_InitAudit(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_SetHSMPolicy`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_OpenSecureToken`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_CapabilityUpdate(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.C_GetSlotInfo(*args)`

Parameters

- ***args** –
- ****kwargs** –

class `pycryptoki.cryptoki.CK_HA_MEMBER`

memberSerial
Structure/Union member

memberStatus
Structure/Union member

`pycryptoki.cryptoki.CA_STCGetDigestIDs(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.C_FindObjectsInit(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_SIMExtract(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_GetExtendedTPV`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_DisableUnauthTokenInsertion(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_C_FindObjectsInit`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_STCGetSequenceWindowSize`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_RSA_PKCS_OAEP_SOURCE_TYPE`
alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CK_UNLOCKMUTEX`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

class `pycryptoki.cryptoki.CKCA_MODULE_INFO`

developerName
Structure/Union member

moduleDescription
Structure/Union member

moduleVersion
Structure/Union member

ulModuleSize
Structure/Union member

class `pycryptoki.cryptoki.CK_RC5_CBC_PARAMS`

pIv
Structure/Union member

ulIvLen
Structure/Union member

ulRounds
Structure/Union member

ulWordsize
Structure/Union member

`pycryptoki.cryptoki.CA_InvokeServiceSinglePart(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_KDF_PRF_ENCODING_SCHEME`
alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CK_C_DecryptInit`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_UnloadModule`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_OpenSession(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_DeleteContainerWithHandle(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_FactoryReset(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_SetUserContainerName(*args)`

Parameters

- ***args** –
- ****kwargs** –

`class pycryptoki.cryptoki.CK_PBE_PARAMS`

pInitVector

Structure/Union member

pPassword

Structure/Union member

pSalt

Structure/Union member

usIteration

Structure/Union member

usPasswordLen

Structure/Union member

usSaltLen

Structure/Union member

`pycryptoki.cryptoki.CK_CA_InsertMaskedObject`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_STCGetState(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_STCSetSequenceWindowSize`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_USER_TYPE`

alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.C_GetMechanismList(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_WTLS_MASTER_KEY_DERIVE_PARAMS_PTR`

alias of `pycryptoki.cryptoki.LP_CK_WTLS_MASTER_KEY_DERIVE_PARAMS`

`pycryptoki.cryptoki.C_GetAttributeValue(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_ListSecureTokenInit(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_OpenApplicationID(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_STCGetKeyActivationTimeOut(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_DuplicateMofN(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_GetModuleList`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.C_GetFunctionStatus(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_OTP_PARAMS_PTR`

alias of `pycryptoki.cryptoki.LP_CK_OTP_PARAMS`

`pycryptoki.cryptoki.CK_CA_SetExtendedTPV`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_C_SignFinal`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`
`pycryptoki.cryptoki.CA_SetDestructiveHSMPolicy(*args)`

Parameters

- ***args** –
- ****kwargs** –

class `pycryptoki.cryptoki.CK_SSL3_MASTER_KEY_DERIVE_PARAMS`

RandomInfo

Structure/Union member

pVersion

Structure/Union member

`pycryptoki.cryptoki.CK_UTF8CHAR_PTR`
alias of `pycryptoki.cryptoki.LP_c_ubyte`

class `pycryptoki.cryptoki.swapper`

bytes

Structure/Union member

words

Structure/Union member

`pycryptoki.cryptoki.C_WrapKey(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_GetSlotListFromServerInstance(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_ATTRIBUTE_TYPE`
alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CK_AES_CBC_ENCRYPT_DATA_PARAMS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_AES_CBC_ENCRYPT_DATA_PARAMS`

`pycryptoki.cryptoki.CK_CA_GetMofNStatus`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_GetRollbackFirmwareVersion`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_GetContainerCapabilitySetting`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.SInt16`
alias of `ctypes.c_short`

`pycryptoki.cryptoki.CK_C_GetMechanismInfo`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_C_DigestFinal`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_GetTokenCertificateInfo`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_DeleteContainer`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_STCGetPartPubKey(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.C_DestroyObject(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_EncodeECChar2Params(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_PKCS5_PBKD2_PSEUDO_RANDOM_FUNCTION_TYPE_PTR`
alias of `pycryptoki.cryptoki.LP_c_ulong`

`pycryptoki.cryptoki.C_GetSessionInfo(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.Int16`
alias of `ctypes.c_short`

`pycryptoki.cryptoki.CK_CA_GetUserContainerNumber`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_SSL3_KEY_MAT_OUT_PTR`
alias of `pycryptoki.cryptoki.LP_CK_SSL3_KEY_MAT_OUT`

`pycryptoki.cryptoki.CK_C_GenerateRandom`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_ModifyUsageCount`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_MTKResplit(*args)`

Parameters

- ***args** –

- ****kwargs** –

`pycryptoki.cryptoki.CK_CHAR`
alias of `ctypes.c_ubyte`

`pycryptoki.cryptoki.CA_GetHSMStats(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.UInt8`
alias of `ctypes.c_ubyte`

`pycryptoki.cryptoki.CA_GetUnauthTokenInsertionStatus(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CMS_SIG_PARAMS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_CMS_SIG_PARAMS`

`pycryptoki.cryptoki.C_DeriveKey(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.C_DigestUpdate(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_GetHSMStorageInformation`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.C_FindObjects(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.SInt64`
alias of `ctypes.c_long`

`pycryptoki.cryptoki.SInt`
alias of `ctypes.c_int`

`pycryptoki.cryptoki.CK_PKCS5_PBKDF2_SALT_SOURCE_TYPE`
alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CK_RSA_PKCS_MGF_TYPE`
alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CK_EXTRACT_PARAMS`
alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CK_RC5_CBC_PARAMS_PTR`
 alias of `pycryptoki.cryptoki.LP_CK_RC5_CBC_PARAMS`

`pycryptoki.cryptoki.CA_STCSetKeyLifeTime(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_GetHSMPolicySetting(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_CreateContainerLoginChallenge(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_ResetTotalOperations`
 alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_MOFN_GENERATION_PTR`
 alias of `pycryptoki.cryptoki.LP_CA_MOFN_GENERATION`

`pycryptoki.cryptoki.CK_CA_InitSlotRolePIN`
 alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_AES_GCM_PARAMS_PTR`
 alias of `pycryptoki.cryptoki.CK_AES_GCM_PARAMS`

`pycryptoki.cryptoki.CK_CA_STMGetState`
 alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_EnableUnauthTokenInsertion`
 alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_C_DecryptDigestUpdate`
 alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_Zeroize(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_HAAnswerMofNChallenge(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_MAC_GENERAL_PARAMS_PTR`
 alias of `pycryptoki.cryptoki.LP_c_ulong`

```

pycryptoki.cryptoki.CK_TOKEN_INFO_PTR
    alias of pycryptoki.cryptoki.LP_CK_TOKEN_INFO
pycryptoki.cryptoki.CK_CA_STCGetDigestNameByID
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
class pycryptoki.cryptoki.CK_AES_CBC_PAD_INSERT_PARAMS

```

```

    ctxID
        Structure/Union member

```

```

    pBuffer
        Structure/Union member

```

```

    pbFileName
        Structure/Union member

```

```

    pedId
        Structure/Union member

```

```

    pulHandle
        Structure/Union member

```

```

    pulType
        Structure/Union member

```

```

    ulBufferLen
        Structure/Union member

```

```

    ulContainerState
        Structure/Union member

```

```

    ulStorage
        Structure/Union member

```

```

    ulStorageType
        Structure/Union member

```

```

pycryptoki.cryptoki.CK_CA_GetSlotIdForPhysicalSlot
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

```

```

pycryptoki.cryptoki.CK_CA_LKMInitiatorChallenge
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

```

```

pycryptoki.cryptoki.CK_CA_HAActivateMofN
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType

```

```

class pycryptoki.cryptoki.CK_KEY_DERIVATION_STRING_DATA

```

```

    pData
        Structure/Union member

```

```

    ulLen
        Structure/Union member

```

```

pycryptoki.cryptoki.CK_MECHANISM_PTR
    alias of pycryptoki.cryptoki.LP_CK_MECHANISM

```

```

pycryptoki.cryptoki.CA_SetTokenCertificateSignature(*args)

```

Parameters

- **args* –

- ****kwargs** –

`pycryptoki.cryptoki.CK_C_GetTokenInfo`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_SwitchSecondarySlot(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_STCGetAdminPubKey`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

class `pycryptoki.cryptoki.CK_FUNCTION_LIST`

C_CancelFunction

Structure/Union member

C_CloseAllSessions

Structure/Union member

C_CloseSession

Structure/Union member

C_CopyObject

Structure/Union member

C_CreateObject

Structure/Union member

C_Decrypt

Structure/Union member

C_DecryptDigestUpdate

Structure/Union member

C_DecryptFinal

Structure/Union member

C_DecryptInit

Structure/Union member

C_DecryptUpdate

Structure/Union member

C_DecryptVerifyUpdate

Structure/Union member

C_DeriveKey

Structure/Union member

C_DestroyObject

Structure/Union member

C_Digest

Structure/Union member

C_DigestEncryptUpdate

Structure/Union member

C_DigestFinal
Structure/Union member

C_DigestInit
Structure/Union member

C_DigestKey
Structure/Union member

C_DigestUpdate
Structure/Union member

C_Encrypt
Structure/Union member

C_EncryptFinal
Structure/Union member

C_EncryptInit
Structure/Union member

C_EncryptUpdate
Structure/Union member

C_Finalize
Structure/Union member

C_FindObjects
Structure/Union member

C_FindObjectsFinal
Structure/Union member

C_FindObjectsInit
Structure/Union member

C_GenerateKey
Structure/Union member

C_GenerateKeyPair
Structure/Union member

C_GenerateRandom
Structure/Union member

C_GetAttributeValue
Structure/Union member

C_GetFunctionList
Structure/Union member

C_GetFunctionStatus
Structure/Union member

C_GetInfo
Structure/Union member

C_GetMechanismInfo
Structure/Union member

C_GetMechanismList
Structure/Union member

C_GetObjectSize
Structure/Union member

C_GetOperationState
Structure/Union member

C_GetSessionInfo
Structure/Union member

C_GetSlotInfo
Structure/Union member

C_GetSlotList
Structure/Union member

C_GetTokenInfo
Structure/Union member

C_InitPIN
Structure/Union member

C_InitToken
Structure/Union member

C_Initialize
Structure/Union member

C_Login
Structure/Union member

C_Logout
Structure/Union member

C_OpenSession
Structure/Union member

C_SeedRandom
Structure/Union member

C_SetAttributeValue
Structure/Union member

C_SetOperationState
Structure/Union member

C_SetPIN
Structure/Union member

C_Sign
Structure/Union member

C_SignEncryptUpdate
Structure/Union member

C_SignFinal
Structure/Union member

C_SignInit
Structure/Union member

C_SignRecover
Structure/Union member

C_SignRecoverInit
Structure/Union member

C_SignUpdate
Structure/Union member

C_UnwrapKey
Structure/Union member

C_Verify
Structure/Union member

C_VerifyFinal
Structure/Union member

C_VerifyInit
Structure/Union member

C_VerifyRecover
Structure/Union member

C_VerifyRecoverInit
Structure/Union member

C_VerifyUpdate
Structure/Union member

C_WaitForSlotEvent
Structure/Union member

C_WrapKey
Structure/Union member

version
Structure/Union member

`pycryptoki.cryptoki.CK_CA_DuplicateMofN`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_RC5_PARAMS_PTR`
alias of `pycryptoki.cryptoki.LP_CK_RC5_PARAMS`

`pycryptoki.cryptoki.CK_C_DigestInit`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_ModifyMofN(*args)`

Parameters

- ***args** –
- ****kwargs** –

class `pycryptoki.cryptoki.CK_WTLS_MASTER_KEY_DERIVE_PARAMS`

DigestMechanism
Structure/Union member

RandomInfo
Structure/Union member

pVersion
Structure/Union member

```
pycryptoki.cryptoki.CA_InvokeServiceInit(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CK_CA_STCGetKeyActivationTimeOut
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CA_GetTokenStatus(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.C_SignUpdate(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.C_EncryptInit(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
class pycryptoki.cryptoki.CK_OTP_PARAMS
```

pParams

Structure/Union member

ulCount

Structure/Union member

```
pycryptoki.cryptoki.CK_SEED_CTR_PARAMS_PTR
    alias of pycryptoki.cryptoki.LP_CK_AES_CTR_PARAMS
```

```
pycryptoki.cryptoki.CA_GetContainerCapabilitySet(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.C_DigestFinal(*args)
```

Parameters

- ***args** –
- ****kwargs** –

```
pycryptoki.cryptoki.CK_CA_LockClusteredSlot
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
```

```
pycryptoki.cryptoki.CA_EnableUnauthTokenInsertion(*args)
```

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_LogVerify`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_GetTSV`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_LKMReceiverComplete(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_STCIsEnabled(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.C_CloseSession(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_EC_DH_PRIMITIVE`
alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CK_C_Login`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_IsMofNEEnabled(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_LogGetConfig`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_FUNCTION_LIST_PTR_PTR`
alias of `pycryptoki.cryptoki.LP_LP_CK_FUNCTION_LIST`

`pycryptoki.cryptoki.CK_CA_GenerateTokenKeys`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.C_DecryptVerifyUpdate(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_UTF8CHAR`
alias of `ctypes.c_ubyte`

`pycryptoki.cryptoki.CK_CA_InitializeRemotePEDVector`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.C_DigestInit (*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.CA_GetContainerStorageInformation (*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.CA_GetHSMPolicySet (*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.C_CopyObject (*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.CK_CA_CloseApplicationID`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_Insert`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_NOTIFICATION`
alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.C_SignRecover (*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.CA_CreateLoginChallenge (*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.C_EncryptUpdate (*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.CK_CA_GetContainerName`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_KDF_PRF_TYPE`

alias of `ctypes.c_ulong`

`pycryptoki.cryptoki.CK_C_GetObjectSize`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_STCGetPartPubKey`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_DeactivateMofN(*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.CK_ECDH1_DERIVE_PARAMS_PTR`

alias of `pycryptoki.cryptoki.LP_CK_ECDH1_DERIVE_PARAMS`

`pycryptoki.cryptoki.CK_C_Encrypt`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_CloseApplicationIDForContainer(*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.C_DecryptUpdate(*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.Int8`

alias of `ctypes.c_char`

`pycryptoki.cryptoki.CA_TamperClear(*args)`

Parameters

- `*args` –
- `**kwargs` –

`pycryptoki.cryptoki.CK_C_DestroyObject`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_STCGetKeyLifeTime`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CK_CA_GetTokenStorageInformation`

alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_GetSlotIdForPhysicalSlot(*args)`

Parameters

- `*args` –
- `**kwargs` –

```

pycryptoki.cryptoki.CK_C_InitToken
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
pycryptoki.cryptoki.CK_CA_LKMReceiverResponse
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
pycryptoki.cryptoki.CK_CA_GetUserContainerName
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
pycryptoki.cryptoki.CK_CA_MTKZeroize
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
pycryptoki.cryptoki.CK_CA_GetClusterState
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
pycryptoki.cryptoki.CK_CA_STCClearCipherAlgorithm
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
pycryptoki.cryptoki.Float32
    alias of ctypes.c_float
pycryptoki.cryptoki.CK_ECIES_PARAMS_PTR
    alias of pycryptoki.cryptoki.LP_CK_ECIES_PARAMS
pycryptoki.cryptoki.CK_CA_GetPedId
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
pycryptoki.cryptoki.CK_CA_MTKGetState
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
pycryptoki.cryptoki.C_GetObjectSize(*args)

```

Parameters

- ***args** –
- ****kwargs** –

```

pycryptoki.cryptoki.CK_CA_GenerateCloneableMofN
    alias of ctypes.CFUNCTYPE.<locals>.CFunctionType
pycryptoki.cryptoki.CA_STCGetClientsList(*args)

```

Parameters

- ***args** –
- ****kwargs** –

```

pycryptoki.cryptoki.CA_STCGetSequenceWindowSize(*args)

```

Parameters

- ***args** –
- ****kwargs** –

```

pycryptoki.cryptoki.CA_InitRolePIN(*args)

```

Parameters

- ***args** –
- ****kwargs** –

```

pycryptoki.cryptoki.C_GenerateRandom(*args)

```

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CA_GetTunnelSlotNumber(*args)`

Parameters

- ***args** –
- ****kwargs** –

`pycryptoki.cryptoki.CK_CA_IsMofNEEnabled`
alias of `ctypes.CFUNCTYPE.<locals>.CFunctionType`

`pycryptoki.cryptoki.CA_SetLKCVC(*args)`

Parameters

- ***args** –
- ****kwargs** –

1.4.9 Pycryptoki Daemon Package

Start `pycryptoki.daemon.rpyc_pycryptoki.py` on your remote client, then connect to it using `RemotePycryptokiClient`. You can then use the `RemotePycryptokiClient` as if it were local:

```
pycryptoki = RemotePycryptokiClient('10.2.96.130', port=8001)
pycryptoki.c_initialize_ex() # Executed on the daemon!
session = pycryptoki.c_open_session_ex(0)
#etc
```

1.4.9.1 daemon.rpyc_pycryptoki

1.4.9.2 pycryptoki.pycryptoki_client

p

- `pycryptoki.attributes`, 32
- `pycryptoki.audit_handling`, 55
- `pycryptoki.backup`, 57
- `pycryptoki.conversions`, 34
- `pycryptoki.cryptoki`, 62
- `pycryptoki.cryptoki_helpers`, 61
- `pycryptoki.default_templates`, 61
- `pycryptoki.defaults`, 62
- `pycryptoki.hsm_management`, 47
- `pycryptoki.key_generator`, 20
- `pycryptoki.key_management`, 23
- `pycryptoki.key_usage`, 24
- `pycryptoki.lookup_dicts`, 61
- `pycryptoki.mechanism`, 35
 - `pycryptoki.mechanism.aes`, 38
 - `pycryptoki.mechanism.generic`, 39
 - `pycryptoki.mechanism.helpers`, 36
 - `pycryptoki.mechanism.rc`, 40
 - `pycryptoki.mechanism.rsa`, 41
- `pycryptoki.misc`, 41
- `pycryptoki.object_attr_lookup`, 45
- `pycryptoki.session_management`, 8
- `pycryptoki.token_management`, 17

A

AESCBCEncryptDataMechanism (class in pycryptoki.mechanism.aes), 38
AESCTRMechanism (class in pycryptoki.mechanism.aes), 38
AESECBEncryptDataMechanism (class in pycryptoki.mechanism.aes), 38
AESGCMMechanism (class in pycryptoki.mechanism.aes), 38
AESXTSMechanism (class in pycryptoki.mechanism.aes), 39
ATTR_NAME_LOOKUP (in module pycryptoki.lookup_dicts), 61
Attributes (class in pycryptoki.attributes), 32
AutoMech (class in pycryptoki.mechanism.generic), 39

B

bBC (pycryptoki.cryptoki.CK_KEY_WRAP_SET_OAEP_PARAMS attribute), 79
bIsExport (pycryptoki.cryptoki.CK_SSL3_KEY_MAT_PARAMS attribute), 101
bIsExport (pycryptoki.cryptoki.CK_WTLS_KEY_MAT_PARAMS attribute), 76
bMembers (pycryptoki.cryptoki.CK_CLUSTER_STATE attribute), 74
Boolean (in module pycryptoki.cryptoki), 118
BYTE (in module pycryptoki.cryptoki), 75
Byte (in module pycryptoki.cryptoki), 89
bytes (pycryptoki.cryptoki.swapper attribute), 138

C

C_CancelFunction (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 143
C_CancelFunction() (in module pycryptoki.cryptoki), 117
c_close_all_sessions() (in module pycryptoki.session_management), 8

c_close_all_sessions_ex() (in module pycryptoki.session_management), 8
c_close_session() (in module pycryptoki.session_management), 8
c_close_session_ex() (in module pycryptoki.session_management), 8
C_CloseAllSessions (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 143
C_CloseAllSessions() (in module pycryptoki.cryptoki), 67
C_CloseSession (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 143
C_CloseSession() (in module pycryptoki.cryptoki), 148
c_copy_object() (in module pycryptoki.key_generator), 20
c_copy_object_ex() (in module pycryptoki.key_generator), 20
C_CopyObject (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 143
C_CopyObject() (in module pycryptoki.cryptoki), 149
c_create_object() (in module pycryptoki.misc), 42
c_create_object_ex() (in module pycryptoki.misc), 42
C_CreateObject (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 143
C_CreateObject() (in module pycryptoki.cryptoki), 111
C_Decrypt (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 143
C_Decrypt() (in module pycryptoki.cryptoki), 132
c_decrypt() (in module pycryptoki.encryption), 26
c_decrypt_ex() (in module pycryptoki.encryption), 27
C_DecryptDigestUpdate (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 143

<code>toki.cryptoki.CK_FUNCTION_LIST</code> attribute), 143	<code>C_DigestFinal()</code> (in module <code>pycryptoki.cryptoki</code>), 147
<code>C_DecryptDigestUpdate()</code> (in module <code>pycryptoki.cryptoki</code>), 132	<code>C_DigestInit</code> (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 144
<code>C_DecryptFinal</code> (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 143	<code>C_DigestInit()</code> (in module <code>pycryptoki.cryptoki</code>), 149
<code>C_DecryptFinal()</code> (in module <code>pycryptoki.cryptoki</code>), 89	<code>C_DigestKey</code> (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 144
<code>C_DecryptInit</code> (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 143	<code>C_DigestKey()</code> (in module <code>pycryptoki.cryptoki</code>), 84
<code>C_DecryptInit()</code> (in module <code>pycryptoki.cryptoki</code>), 122	<code>c_digestkey()</code> (in module <code>pycryptoki.misc</code>), 43
<code>C_DecryptUpdate</code> (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 143	<code>c_digestkey_ex()</code> (in module <code>pycryptoki.misc</code>), 43
<code>C_DecryptUpdate()</code> (in module <code>pycryptoki.cryptoki</code>), 150	<code>C_DigestUpdate</code> (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 144
<code>C_DecryptVerifyUpdate</code> (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 143	<code>C_DigestUpdate()</code> (in module <code>pycryptoki.cryptoki</code>), 140
<code>C_DecryptVerifyUpdate()</code> (in module <code>pycryptoki.cryptoki</code>), 148	<code>C_Encrypt</code> (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 144
<code>c_derive_key()</code> (in module <code>pycryptoki.key_generator</code>), 20	<code>C_Encrypt()</code> (in module <code>pycryptoki.cryptoki</code>), 104
<code>c_derive_key_ex()</code> (in module <code>pycryptoki.key_generator</code>), 21	<code>c_encrypt()</code> (in module <code>pycryptoki.encryption</code>), 25
<code>C_DeriveKey</code> (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 143	<code>c_encrypt_ex()</code> (in module <code>pycryptoki.encryption</code>), 26
<code>C_DeriveKey()</code> (in module <code>pycryptoki.cryptoki</code>), 140	<code>C_EncryptFinal</code> (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 144
<code>c_destroy_object()</code> (in module <code>pycryptoki.key_generator</code>), 21	<code>C_EncryptFinal()</code> (in module <code>pycryptoki.cryptoki</code>), 77
<code>c_destroy_object_ex()</code> (in module <code>pycryptoki.key_generator</code>), 21	<code>C_EncryptInit</code> (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 144
<code>C_DestroyObject</code> (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 143	<code>C_EncryptInit()</code> (in module <code>pycryptoki.cryptoki</code>), 147
<code>C_DestroyObject()</code> (in module <code>pycryptoki.cryptoki</code>), 139	<code>C_EncryptUpdate</code> (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 144
<code>C_Digest</code> (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 143	<code>C_EncryptUpdate()</code> (in module <code>pycryptoki.cryptoki</code>), 149
<code>C_Digest()</code> (in module <code>pycryptoki.cryptoki</code>), 74	<code>C_Finalize</code> (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 144
<code>c_digest()</code> (in module <code>pycryptoki.misc</code>), 42	<code>C_Finalize()</code> (in module <code>pycryptoki.cryptoki</code>), 133
<code>c_digest_ex()</code> (in module <code>pycryptoki.misc</code>), 43	<code>c_finalize()</code> (in module <code>pycryptoki.session_management</code>), 9
<code>C_DigestEncryptUpdate</code> (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 143	<code>c_finalize_ex()</code> (in module <code>pycryptoki.session_management</code>), 9
<code>C_DigestEncryptUpdate()</code> (in module <code>pycryptoki.cryptoki</code>), 76	<code>c_find_objects()</code> (in module <code>pycryptoki.object_attr_lookup</code>), 45
<code>C_DigestFinal</code> (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 143	<code>c_find_objects_ex()</code> (in module <code>pycryptoki.object_attr_lookup</code>), 46
	<code>C_FindObjects</code> (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 144

[C_FindObjects\(\)](#) (in module `pycryptoki.cryptoki`), 140
[C_FindObjectsFinal](#) (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 144
[C_FindObjectsFinal\(\)](#) (in module `pycryptoki.cryptoki`), 110
[C_FindObjectsInit](#) (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 144
[C_FindObjectsInit\(\)](#) (in module `pycryptoki.cryptoki`), 134
[c_generate_key\(\)](#) (in module `pycryptoki.key_generator`), 21
[c_generate_key_ex\(\)](#) (in module `pycryptoki.key_generator`), 22
[c_generate_key_pair\(\)](#) (in module `pycryptoki.key_generator`), 22
[c_generate_key_pair_ex\(\)](#) (in module `pycryptoki.key_generator`), 22
[c_generate_random\(\)](#) (in module `pycryptoki.misc`), 43
[c_generate_random_ex\(\)](#) (in module `pycryptoki.misc`), 44
[C_GenerateKey](#) (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 144
[C_GenerateKey\(\)](#) (in module `pycryptoki.cryptoki`), 107
[C_GenerateKeyPair](#) (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 144
[C_GenerateKeyPair\(\)](#) (in module `pycryptoki.cryptoki`), 121
[C_GenerateRandom](#) (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 144
[C_GenerateRandom\(\)](#) (in module `pycryptoki.cryptoki`), 151
[c_get_attribute_value\(\)](#) (in module `pycryptoki.object_attr_lookup`), 46
[c_get_attribute_value_ex\(\)](#) (in module `pycryptoki.object_attr_lookup`), 46
[c_get_info\(\)](#) (in module `pycryptoki.session_management`), 9
[c_get_info_ex\(\)](#) (in module `pycryptoki.session_management`), 9
[c_get_mechanism_info\(\)](#) (in module `pycryptoki.token_management`), 17
[c_get_mechanism_info_ex\(\)](#) (in module `pycryptoki.token_management`), 18
[c_get_mechanism_list\(\)](#) (in module `pycryptoki.token_management`), 18
[c_get_mechanism_list_ex\(\)](#) (in module `pycryptoki.token_management`), 18
[c_get_ped_id\(\)](#) (in module `pycryptoki.misc`), 44
[c_get_ped_id_ex\(\)](#) (in module `pycryptoki.misc`), 44
[c_get_session_info\(\)](#) (in module `pycryptoki.session_management`), 10
[c_get_session_info_ex\(\)](#) (in module `pycryptoki.session_management`), 10
[c_get_slot_info\(\)](#) (in module `pycryptoki.session_management`), 10
[c_get_slot_info_ex\(\)](#) (in module `pycryptoki.session_management`), 10
[c_get_slot_list\(\)](#) (in module `pycryptoki.session_management`), 11
[c_get_slot_list_ex\(\)](#) (in module `pycryptoki.session_management`), 11
[c_get_token_info\(\)](#) (in module `pycryptoki.session_management`), 11
[c_get_token_info_ex\(\)](#) (in module `pycryptoki.session_management`), 11
[C_GetAttributeValue](#) (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 144
[C_GetAttributeValue\(\)](#) (in module `pycryptoki.cryptoki`), 137
[C_GetFunctionList](#) (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 144
[C_GetFunctionList\(\)](#) (in module `pycryptoki.cryptoki`), 126
[C_GetFunctionStatus](#) (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 144
[C_GetFunctionStatus\(\)](#) (in module `pycryptoki.cryptoki`), 137
[C_GetInfo](#) (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 144
[C_GetInfo\(\)](#) (in module `pycryptoki.cryptoki`), 64
[C_GetMechanismInfo](#) (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 144
[C_GetMechanismInfo\(\)](#) (in module `pycryptoki.cryptoki`), 118
[C_GetMechanismList](#) (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 144
[C_GetMechanismList\(\)](#) (in module `pycryptoki.cryptoki`), 137
[C_GetObjectSize](#) (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 144
[C_GetObjectSize\(\)](#) (in module `pycryptoki.cryptoki`), 151
[C_GetOperationState](#) (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute),

- 145
- `C_GetOperationState()` (in module `pycryptoki.cryptoki`), 126
- `C_GetSessionInfo` (pycryptoki.CK_FUNCTION_LIST attribute), 145
- `C_GetSessionInfo()` (in module `pycryptoki.cryptoki`), 139
- `C_GetSlotInfo` (pycryptoki.CK_FUNCTION_LIST attribute), 145
- `C_GetSlotInfo()` (in module `pycryptoki.cryptoki`), 134
- `C_GetSlotList` (pycryptoki.CK_FUNCTION_LIST attribute), 145
- `C_GetSlotList()` (in module `pycryptoki.cryptoki`), 118
- `C_GetTokenInfo` (pycryptoki.CK_FUNCTION_LIST attribute), 145
- `C_GetTokenInfo()` (in module `pycryptoki.cryptoki`), 122
- `c_init_pin()` (in module `pycryptoki.session_management`), 11
- `c_init_pin_ex()` (in module `pycryptoki.session_management`), 12
- `c_init_token()` (in module `pycryptoki.token_management`), 18
- `c_init_token_ex()` (in module `pycryptoki.token_management`), 19
- `C_Initialize` (pycryptoki.CK_FUNCTION_LIST attribute), 145
- `C_Initialize()` (in module `pycryptoki.cryptoki`), 118
- `c_initialize()` (in module `pycryptoki.session_management`), 12
- `c_initialize_ex()` (in module `pycryptoki.session_management`), 12
- `C_InitPIN` (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 145
- `C_InitPIN()` (in module `pycryptoki.cryptoki`), 107
- `C_InitToken` (pycryptoki.CK_FUNCTION_LIST attribute), 145
- `C_InitToken()` (in module `pycryptoki.cryptoki`), 118
- `C_Login` (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 145
- `C_Login()` (in module `pycryptoki.cryptoki`), 111
- `C_Logout` (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 145
- `C_Logout()` (in module `pycryptoki.cryptoki`), 130
- `c_logout()` (in module `pycryptoki.session_management`), 12
- `c_logout_ex()` (in module `pycryptoki.session_management`), 13
- `c_open_session()` (in module `pycryptoki.session_management`), 13
- `c_open_session_ex()` (in module `pycryptoki.session_management`), 13
- `C_OpenSession` (pycryptoki.CK_FUNCTION_LIST attribute), 145
- `C_OpenSession()` (in module `pycryptoki.cryptoki`), 85
- `c_performselftest()` (in module `pycryptoki.hsm_management`), 47
- `c_performselftest_ex()` (in module `pycryptoki.hsm_management`), 47
- `c_seed_random()` (in module `pycryptoki.misc`), 44
- `c_seed_random_ex()` (in module `pycryptoki.misc`), 45
- `C_SeedRandom` (pycryptoki.CK_FUNCTION_LIST attribute), 145
- `C_SeedRandom()` (in module `pycryptoki.cryptoki`), 117
- `c_set_attribute_value()` (in module `pycryptoki.object_attr_lookup`), 46
- `c_set_attribute_value_ex()` (in module `pycryptoki.object_attr_lookup`), 47
- `c_set_ped_id()` (in module `pycryptoki.misc`), 45
- `c_set_ped_id_ex()` (in module `pycryptoki.misc`), 45
- `c_set_pin()` (in module `pycryptoki.session_management`), 13
- `c_set_pin_ex()` (in module `pycryptoki.session_management`), 14
- `C_SetAttributeValue` (pycryptoki.CK_FUNCTION_LIST attribute), 145
- `C_SetAttributeValue()` (in module `pycryptoki.cryptoki`), 63
- `C_SetOperationState` (pycryptoki.CK_FUNCTION_LIST attribute), 145
- `C_SetOperationState()` (in module `pycryptoki.cryptoki`), 83
- `C_SetPIN` (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 145
- `C_SetPIN()` (in module `pycryptoki.cryptoki`), 107
- `C_Sign` (pycryptoki.cryptoki.CK_FUNCTION_LIST attribute), 145
- `C_Sign()` (in module `pycryptoki.cryptoki`), 126
- `c_sign()` (in module `pycryptoki.sign_verify`), 30
- `c_sign_ex()` (in module `pycryptoki.sign_verify`), 30
- `C_SignEncryptUpdate` (pycryptoki.CK_FUNCTION_LIST attribute),

- 145
- `C_SignEncryptUpdate()` (in module `pycryptoki.cryptoki`), 75
- `C_SignFinal` (`pycryptoki.cryptoki.CK_FUNCTION_LIST` attribute), 145
- `C_SignFinal()` (in module `pycryptoki.cryptoki`), 119
- `C_SignInit` (`pycryptoki.cryptoki.CK_FUNCTION_LIST` attribute), 145
- `C_SignInit()` (in module `pycryptoki.cryptoki`), 67
- `C_SignRecover` (`pycryptoki.cryptoki.CK_FUNCTION_LIST` attribute), 145
- `C_SignRecover()` (in module `pycryptoki.cryptoki`), 149
- `C_SignRecoverInit` (`pycryptoki.cryptoki.CK_FUNCTION_LIST` attribute), 145
- `C_SignRecoverInit()` (in module `pycryptoki.cryptoki`), 105
- `C_SignUpdate` (`pycryptoki.cryptoki.CK_FUNCTION_LIST` attribute), 146
- `C_SignUpdate()` (in module `pycryptoki.cryptoki`), 147
- `c_struct_to_python()` (in module `pycryptoki.attributes`), 32
- `c_unwrap_key()` (in module `pycryptoki.encryption`), 28
- `c_unwrap_key_ex()` (in module `pycryptoki.encryption`), 29
- `C_UnwrapKey` (`pycryptoki.cryptoki.CK_FUNCTION_LIST` attribute), 146
- `C_UnwrapKey()` (in module `pycryptoki.cryptoki`), 62
- `C_Verify` (`pycryptoki.cryptoki.CK_FUNCTION_LIST` attribute), 146
- `C_Verify()` (in module `pycryptoki.cryptoki`), 89
- `c_verify()` (in module `pycryptoki.sign_verify`), 31
- `c_verify_ex()` (in module `pycryptoki.sign_verify`), 31
- `C_VerifyFinal` (`pycryptoki.cryptoki.CK_FUNCTION_LIST` attribute), 146
- `C_VerifyFinal()` (in module `pycryptoki.cryptoki`), 63
- `C_VerifyInit` (`pycryptoki.cryptoki.CK_FUNCTION_LIST` attribute), 146
- `C_VerifyInit()` (in module `pycryptoki.cryptoki`), 114
- `C_VerifyRecover` (`pycryptoki.cryptoki.CK_FUNCTION_LIST` attribute), 146
- `C_VerifyRecover()` (in module `pycryptoki.cryptoki`), 83
- `C_VerifyRecoverInit` (`pycryptoki.cryptoki.CK_FUNCTION_LIST` attribute), 146
- `C_VerifyRecoverInit()` (in module `pycryptoki.cryptoki`), 84
- `C_VerifyUpdate` (`pycryptoki.cryptoki.CK_FUNCTION_LIST` attribute), 146
- `C_VerifyUpdate()` (in module `pycryptoki.cryptoki`), 125
- `C_WaitForSlotEvent` (`pycryptoki.cryptoki.CK_FUNCTION_LIST` attribute), 146
- `C_WaitForSlotEvent()` (in module `pycryptoki.cryptoki`), 65
- `c_wrap_key()` (in module `pycryptoki.encryption`), 27
- `c_wrap_key_ex()` (in module `pycryptoki.encryption`), 28
- `C_WrapKey` (`pycryptoki.cryptoki.CK_FUNCTION_LIST` attribute), 146
- `C_WrapKey()` (in module `pycryptoki.cryptoki`), 138
- `CA_ActivateMofN` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 89
- `CA_ActivateMofN()` (in module `pycryptoki.cryptoki`), 89
- `CA_AuthorizeKey` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 89
- `CA_CapabilityUpdate` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 90
- `CA_CapabilityUpdate()` (in module `pycryptoki.cryptoki`), 134
- `CA_CheckOperationState` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 90
- `CA_CheckOperationState()` (in module `pycryptoki.cryptoki`), 121
- `CA_ChoosePrimarySlot` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 90
- `CA_ChoosePrimarySlot()` (in module `pycryptoki.cryptoki`), 114
- `CA_ChooseSecondarySlot` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 90
- `CA_ChooseSecondarySlot()` (in module `pycryptoki.cryptoki`), 103
- `CA_CloneAllObjectsToSession` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST`

- attribute*), 90
- CA_CloneAllObjectsToSession() (in module *pycryptoki.cryptoki*), 80
- CA_CloneAsSource (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 90
- CA_CloneAsSource() (in module *pycryptoki.cryptoki*), 84
- CA_CloneAsTarget (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 90
- CA_CloneAsTarget() (in module *pycryptoki.cryptoki*), 122
- CA_CloneAsTargetInit (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 90
- CA_CloneAsTargetInit() (in module *pycryptoki.cryptoki*), 131
- CA_CloneModifyMofN (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 90
- CA_CloneModifyMofN() (in module *pycryptoki.cryptoki*), 74
- CA_CloneMofN (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 90
- CA_CloneMofN() (in module *pycryptoki.cryptoki*), 87
- ca_clonemofn() (in module *pycryptoki.key_usage*), 24
- ca_clonemofn_ex() (in module *pycryptoki.key_usage*), 24
- CA_CloneObject (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 90
- CA_CloneObject() (in module *pycryptoki.cryptoki*), 83
- CA_CloneObjectToAllSessions (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 90
- CA_CloneObjectToAllSessions() (in module *pycryptoki.cryptoki*), 73
- CA_ClonePrivateKey (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 90
- CA_ClonePrivateKey() (in module *pycryptoki.cryptoki*), 88
- ca_close_secure_token() (in module *pycryptoki.backup*), 57
- ca_close_secure_token_ex() (in module *pycryptoki.backup*), 57
- CA_CloseAllSecondarySessions (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 90
- CA_CloseAllSecondarySessions() (in module *pycryptoki.cryptoki*), 81
- CA_CloseApplicationID (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 90
- CA_CloseApplicationID() (in module *pycryptoki.cryptoki*), 82
- ca_closeapplicationID() (in module *pycryptoki.session_management*), 14
- ca_closeapplicationID_ex() (in module *pycryptoki.session_management*), 14
- CA_CloseApplicationIDForContainer (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 90
- CA_CloseApplicationIDForContainer() (in module *pycryptoki.cryptoki*), 150
- CA_CloseSecondarySession (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 90
- CA_CloseSecondarySession() (in module *pycryptoki.cryptoki*), 88
- CA_CloseSecureToken (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 90
- CA_CloseSecureToken() (in module *pycryptoki.cryptoki*), 80
- CA_ConfigureRemotePED (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 90
- CA_ConfigureRemotePED() (in module *pycryptoki.cryptoki*), 71
- CA_CreateContainer (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 90
- CA_CreateContainer() (in module *pycryptoki.cryptoki*), 78
- CA_CreateContainerLoginChallenge (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 90
- CA_CreateContainerLoginChallenge() (in module *pycryptoki.cryptoki*), 141
- CA_CreateContainerWithPolicy() (in module *pycryptoki.cryptoki*), 123
- CA_CreateLoginChallenge (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 91
- CA_CreateLoginChallenge() (in module *pycryptoki.cryptoki*), 149
- ca_createloginchallenge() (in module *pycryptoki.hsm_management*), 48
- ca_createloginchallenge_ex() (in module *pycryptoki.hsm_management*), 48
- CA_Deactivate (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 91

- CA_Deactivate() (in module *pycryptoki.cryptoki*), 66
- CA_DeactivateMofN (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 91
- CA_DeactivateMofN() (in module *pycryptoki.cryptoki*), 150
- CA_DeleteContainer (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 91
- CA_DeleteContainer() (in module *pycryptoki.cryptoki*), 88
- CA_DeleteContainerWithHandle (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 91
- CA_DeleteContainerWithHandle() (in module *pycryptoki.cryptoki*), 136
- CA_DeleteRemotePEDVector (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 91
- CA_DeleteRemotePEDVector() (in module *pycryptoki.cryptoki*), 88
- ca_deleteremotepedvector() (in module *pycryptoki.hsm_management*), 48
- ca_deleteremotepedvector_ex() (in module *pycryptoki.hsm_management*), 48
- CA_DeriveKeyAndWrap() (in module *pycryptoki.cryptoki*), 123
- CA_DescribeUtilizationBinId (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 91
- CA_DestroyMultipleObjects (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 91
- CA_DestroyMultipleObjects() (in module *pycryptoki.cryptoki*), 73
- CA_DisableUnauthTokenInsertion (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 91
- CA_DisableUnauthTokenInsertion() (in module *pycryptoki.cryptoki*), 134
- CA_DismantleRemotePED (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 91
- CA_DismantleRemotePED() (in module *pycryptoki.cryptoki*), 86
- CA_DuplicateMofN (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 91
- CA_DuplicateMofN() (in module *pycryptoki.cryptoki*), 137
- ca_duplicatemofn() (in module *pycryptoki.key_usage*), 24
- ca_duplicatemofn_ex() (in module *pycryptoki.key_usage*), 24
- CA_EnableUnauthTokenInsertion (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 91
- CA_EnableUnauthTokenInsertion() (in module *pycryptoki.cryptoki*), 147
- CA_EncodeECChar2Params (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 91
- CA_EncodeECChar2Params() (in module *pycryptoki.cryptoki*), 139
- CA_EncodeECParamsFromFile (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 91
- CA_EncodeECParamsFromFile() (in module *pycryptoki.cryptoki*), 65
- CA_EncodeECPrimeParams (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 91
- CA_EncodeECPrimeParams() (in module *pycryptoki.cryptoki*), 110
- CA_Extract (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 91
- ca_extract() (in module *pycryptoki.backup*), 57
- CA_Extract() (in module *pycryptoki.cryptoki*), 83
- ca_extract_ex() (in module *pycryptoki.backup*), 57
- CA_ExtractMaskedObject (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 91
- CA_ExtractMaskedObject() (in module *pycryptoki.cryptoki*), 76
- ca_factory_reset() (in module *pycryptoki.session_management*), 14
- ca_factory_reset_ex() (in module *pycryptoki.session_management*), 15
- CA_FactoryReset (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 91
- CA_FactoryReset() (in module *pycryptoki.cryptoki*), 136
- CA_FindAdminSlotForSlot (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 91
- CA_FindAdminSlotForSlot() (in module *pycryptoki.cryptoki*), 109
- CA_FirmwareRollback (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 91
- CA_FirmwareRollback() (in module *pycryptoki.cryptoki*), 65
- CA_FirmwareUpdate (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 91

<code>CA_FirmwareUpdate()</code> (in module <code>pycryptoki.cryptoki</code>), 119	<code>CA_GetClusterState()</code> (in module <code>pycryptoki.cryptoki</code>), 70
<code>CA_GenerateCloneableMofN</code> (<code>pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST</code> attribute), 92	<code>CA_GetConfigurationElementDescription</code> (<code>pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST</code> attribute), 92
<code>CA_GenerateCloneableMofN()</code> (in module <code>pycryptoki.cryptoki</code>), 79	<code>CA_GetConfigurationElementDescription()</code> (in module <code>pycryptoki.cryptoki</code>), 111
<code>CA_GenerateCloningKEV</code> (<code>pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST</code> attribute), 92	<code>CA_GetContainerCapabilitySet</code> (<code>pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST</code> attribute), 92
<code>CA_GenerateCloningKEV()</code> (in module <code>pycryptoki.cryptoki</code>), 125	<code>CA_GetContainerCapabilitySet()</code> (in module <code>pycryptoki.cryptoki</code>), 147
<code>CA_GenerateMofN</code> (<code>pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST</code> attribute), 92	<code>CA_GetContainerCapabilitySetting</code> (<code>pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST</code> attribute), 92
<code>CA_GenerateMofN()</code> (in module <code>pycryptoki.cryptoki</code>), 68	<code>CA_GetContainerCapabilitySetting()</code> (in module <code>pycryptoki.cryptoki</code>), 123
<code>ca_generatemofn()</code> (in module <code>pycryptoki.key_management</code>), 23	<code>CA_GetContainerList</code> (<code>pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST</code> attribute), 92
<code>ca_generatemofn_ex()</code> (in module <code>pycryptoki.key_management</code>), 23	<code>CA_GetContainerList()</code> (in module <code>pycryptoki.cryptoki</code>), 64
<code>CA_GenerateTokenKeys</code> (<code>pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST</code> attribute), 92	<code>CA_GetContainerName</code> (<code>pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST</code> attribute), 92
<code>CA_GenerateTokenKeys()</code> (in module <code>pycryptoki.cryptoki</code>), 106	<code>CA_GetContainerName()</code> (in module <code>pycryptoki.cryptoki</code>), 109
<code>CA_Get()</code> (in module <code>pycryptoki.cryptoki</code>), 132	<code>CA_GetContainerPolicySet</code> (<code>pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST</code> attribute), 92
<code>ca_get_hsm_capability_set()</code> (in module <code>pycryptoki.hsm_management</code>), 49	<code>CA_GetContainerPolicySet()</code> (in module <code>pycryptoki.cryptoki</code>), 132
<code>ca_get_hsm_capability_set_ex()</code> (in module <code>pycryptoki.hsm_management</code>), 49	<code>CA_GetContainerPolicySetting</code> (<code>pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST</code> attribute), 92
<code>ca_get_hsm_capability_setting()</code> (in module <code>pycryptoki.hsm_management</code>), 49	<code>CA_GetContainerPolicySetting()</code> (in module <code>pycryptoki.cryptoki</code>), 106
<code>ca_get_hsm_capability_setting_ex()</code> (in module <code>pycryptoki.hsm_management</code>), 49	<code>CA_GetContainerStatus</code> (<code>pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST</code> attribute), 92
<code>ca_get_hsm_policy_set()</code> (in module <code>pycryptoki.hsm_management</code>), 49	<code>CA_GetContainerStatus()</code> (in module <code>pycryptoki.cryptoki</code>), 110
<code>ca_get_hsm_policy_set_ex()</code> (in module <code>pycryptoki.hsm_management</code>), 50	<code>CA_GetContainerStorageInformation</code> (<code>pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST</code> attribute), 92
<code>ca_get_hsm_policy_setting()</code> (in module <code>pycryptoki.hsm_management</code>), 50	<code>CA_GetContainerStorageInformation()</code> (in module <code>pycryptoki.cryptoki</code>), 149
<code>ca_get_hsm_policy_setting_ex()</code> (in module <code>pycryptoki.hsm_management</code>), 50	<code>CA_GetExtendedTPV</code> (<code>pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST</code> attribute), 92
<code>ca_get_time()</code> (in module <code>pycryptoki.audit_handling</code>), 55	<code>CA_GetExtendedTPV()</code> (in module <code>pycryptoki.cryptoki</code>), 132
<code>ca_get_time_ex()</code> (in module <code>pycryptoki.audit_handling</code>), 55	<code>CA_GetFirmwareVersion</code> (<code>pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST</code> attribute), 92
<code>ca_get_token_policies()</code> (in module <code>pycryptoki.token_management</code>), 19	
<code>ca_get_token_policies_ex()</code> (in module <code>pycryptoki.token_management</code>), 19	
<code>CA_GetClusterState</code> (<code>pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST</code> attribute), 92	

- attribute*), 92
- CA_GetFPV (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *token.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 92
- CA_GetFPV () (in module *pycryptoki.cryptoki*), 74
- CA_GetFunctionList (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 92
- CA_GetFunctionList () (in module *pycryptoki.cryptoki*), 69
- CA_GetHState (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 92
- CA_GetHState () (in module *pycryptoki.cryptoki*), 105
- CA_GetHSMCapabilitySet (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 92
- CA_GetHSMCapabilitySet () (in module *pycryptoki.cryptoki*), 66
- CA_GetHSMCapabilitySetting (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 92
- CA_GetHSMCapabilitySetting () (in module *pycryptoki.cryptoki*), 103
- CA_GetHSMPolicySet (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 93
- CA_GetHSMPolicySet () (in module *pycryptoki.cryptoki*), 149
- CA_GetHSMPolicySetting (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 93
- CA_GetHSMPolicySetting () (in module *pycryptoki.cryptoki*), 141
- CA_GetHSMStats (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 93
- CA_GetHSMStats () (in module *pycryptoki.cryptoki*), 140
- CA_GetHSMStorageInformation (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 93
- CA_GetHSMStorageInformation () (in module *pycryptoki.cryptoki*), 70
- CA_GetModuleInfo (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 93
- CA_GetModuleInfo () (in module *pycryptoki.cryptoki*), 113
- CA_GetModuleList (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 93
- CA_GetModuleList () (in module *pycryptoki.cryptoki*), 79
- CA_GetMofNStatus (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 93
- CA_GetMofNStatus () (in module *pycryptoki.cryptoki*), 128
- CA_GetNumberOfAllowedContainers (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 93
- CA_GetNumberOfAllowedContainers () (in module *pycryptoki.cryptoki*), 114
- CA_GetObjectHandle (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 93
- CA_GetObjectHandle () (in module *pycryptoki.cryptoki*), 83
- CA_GetObjectUID (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 93
- CA_GetObjectUID () (in module *pycryptoki.cryptoki*), 122
- CA_GetPartitionPolicyTemplate () (in module *pycryptoki.cryptoki*), 84
- CA_GetPedId (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 93
- CA_GetPedId () (in module *pycryptoki.cryptoki*), 71
- CA_GetPrimarySlot (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 93
- CA_GetPrimarySlot () (in module *pycryptoki.cryptoki*), 72
- CA_GetRemotePEDVectorStatus (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 93
- CA_GetRemotePEDVectorStatus () (in module *pycryptoki.cryptoki*), 68
- CA_GetRollbackFirmwareVersion (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 93
- CA_GetRollbackFirmwareVersion () (in module *pycryptoki.cryptoki*), 127
- CA_GetSecondarySlot (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 93
- CA_GetSecondarySlot () (in module *pycryptoki.cryptoki*), 128
- CA_GetSecureElementMeta (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 93
- CA_GetSecureElementMeta () (in module *pycryptoki.cryptoki*), 114
- CA_GetServerInstanceBySlotID (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST* *attribute*), 93

CA_GetServerInstanceBySlotID() (in module <i>pycryptoki.cryptoki</i>), 106	CA_GetTokenObjectUID() (in module <i>pycryptoki.cryptoki</i>), 126
CA_GetSessionInfo (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 93	CA_GetTokenPolicies (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 94
CA_GetSessionInfo() (in module <i>pycryptoki.cryptoki</i>), 102	CA_GetTokenPolicies() (in module <i>pycryptoki.cryptoki</i>), 80
CA_GetSlotIdForContainer (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 93	CA_GetTokenStatus (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 94
CA_GetSlotIdForContainer() (in module <i>pycryptoki.cryptoki</i>), 62	CA_GetTokenStatus() (in module <i>pycryptoki.cryptoki</i>), 147
CA_GetSlotIdForPhysicalSlot (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 93	CA_GetTokenStorageInformation (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 94
CA_GetSlotIdForPhysicalSlot() (in module <i>pycryptoki.cryptoki</i>), 150	CA_GetTokenStorageInformation() (in module <i>pycryptoki.cryptoki</i>), 73
CA_GetSlotListFromServerInstance (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 93	CA_GetTPV (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 94
CA_GetSlotListFromServerInstance() (in module <i>pycryptoki.cryptoki</i>), 138	CA_GetTPV() (in module <i>pycryptoki.cryptoki</i>), 114
CA_GetTime (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 94	CA_GetTSV (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 94
CA_GetTime() (in module <i>pycryptoki.cryptoki</i>), 80	CA_GetTSV() (in module <i>pycryptoki.cryptoki</i>), 119
CA_GetTokenCapabilities (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 94	CA_GetTunnelSlotNumber (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 94
CA_GetTokenCapabilities() (in module <i>pycryptoki.cryptoki</i>), 128	CA_GetTunnelSlotNumber() (in module <i>pycryptoki.cryptoki</i>), 152
CA_GetTokenCertificateInfo (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 94	CA_GetUnauthTokenInsertionStatus (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 94
CA_GetTokenCertificateInfo() (in module <i>pycryptoki.cryptoki</i>), 131	CA_GetUnauthTokenInsertionStatus() (in module <i>pycryptoki.cryptoki</i>), 140
CA_GetTokenCertificates (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 94	CA_GetUserContainerName (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 94
CA_GetTokenCertificates() (in module <i>pycryptoki.cryptoki</i>), 121	CA_GetUserContainerName() (in module <i>pycryptoki.cryptoki</i>), 102
CA_GetTokenInsertionCount (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 94	CA_GetUserContainerNumber (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 94
CA_GetTokenInsertionCount() (in module <i>pycryptoki.cryptoki</i>), 116	CA_GetUserContainerNumber() (in module <i>pycryptoki.cryptoki</i>), 103
CA_GetTokenObjectHandle (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 94	CA_HAActivateMofN (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 94
CA_GetTokenObjectHandle() (in module <i>pycryptoki.cryptoki</i>), 77	CA_HAActivateMofN() (in module <i>pycryptoki.cryptoki</i>), 66
CA_GetTokenObjectUID (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 94	CA_HAAnswerLoginChallenge (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 94
	CA_HAAnswerLoginChallenge() (in module <i>pycryptoki.cryptoki</i>), 75
	CA_HAAnswerMofNChallenge (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 94

- toki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 94
- CA_HAAnswerMofNChallenge() (in module *pycryptoki.cryptoki*), 141
- CA_HAGetLoginChallenge (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 94
- CA_HAGetLoginChallenge() (in module *pycryptoki.cryptoki*), 82
- CA_HAGetMasterPublic (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 94
- CA_HAGetMasterPublic() (in module *pycryptoki.cryptoki*), 124
- CA_HAInit (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 95
- CA_HAInit() (in module *pycryptoki.cryptoki*), 108
- ca_hainit() (in module *pycryptoki.hsm_management*), 50
- ca_hainit_ex() (in module *pycryptoki.hsm_management*), 51
- CA_HALogin (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 95
- CA_HALogin() (in module *pycryptoki.cryptoki*), 69
- CA_IndirectLogin (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 95
- CA_IndirectLogin() (in module *pycryptoki.cryptoki*), 68
- ca_init_audit() (in module *pycryptoki.audit_handling*), 56
- ca_init_audit_ex() (in module *pycryptoki.audit_handling*), 56
- CA_InitAudit (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 95
- CA_InitAudit() (in module *pycryptoki.cryptoki*), 134
- CA_InitializeRemotePEDVector (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 95
- CA_InitializeRemotePEDVector() (in module *pycryptoki.cryptoki*), 78
- ca_initializeremotepedvector() (in module *pycryptoki.hsm_management*), 51
- ca_initializeremotepedvector_ex() (in module *pycryptoki.hsm_management*), 51
- CA_InitIndirectPIN (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 95
- CA_InitIndirectPIN() (in module *pycryptoki.cryptoki*), 72
- CA_InitIndirectToken (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 95
- CA_InitIndirectToken() (in module *pycryptoki.cryptoki*), 107
- CA_InitRolePIN (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 95
- CA_InitRolePIN() (in module *pycryptoki.cryptoki*), 151
- CA_InitSlotRolePIN (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 95
- CA_InitSlotRolePIN() (in module *pycryptoki.cryptoki*), 69
- CA_Insert (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 95
- ca_insert() (in module *pycryptoki.backup*), 57
- CA_Insert() (in module *pycryptoki.cryptoki*), 71
- ca_insert_ex() (in module *pycryptoki.backup*), 58
- CA_InsertMaskedObject (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 95
- CA_InsertMaskedObject() (in module *pycryptoki.cryptoki*), 80
- CA_InvokeService (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 95
- CA_InvokeService() (in module *pycryptoki.cryptoki*), 62
- CA_InvokeServiceAsynch (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 95
- CA_InvokeServiceAsynch() (in module *pycryptoki.cryptoki*), 123
- CA_InvokeServiceFinal (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 95
- CA_InvokeServiceFinal() (in module *pycryptoki.cryptoki*), 85
- CA_InvokeServiceInit (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 95
- CA_InvokeServiceInit() (in module *pycryptoki.cryptoki*), 146
- CA_InvokeServiceSinglePart (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 95
- CA_InvokeServiceSinglePart() (in module *pycryptoki.cryptoki*), 135
- CA_IsMofNEEnabled (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 95
- CA_IsMofNEEnabled() (in module *pycryptoki.cryptoki*), 148

- `CA_IsMofNRequired` (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 95
- `CA_IsMofNRequired()` (in module *pycryptoki.cryptoki*), 124
- `CA_ListSecureTokenInit` (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 96
- `CA_ListSecureTokenInit()` (in module *pycryptoki.cryptoki*), 137
- `CA_ListSecureTokenUpdate` (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 96
- `CA_ListSecureTokenUpdate()` (in module *pycryptoki.cryptoki*), 73
- `CA_LKMInitiatorChallenge` (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 95
- `CA_LKMInitiatorChallenge()` (in module *pycryptoki.cryptoki*), 106
- `CA_LKMInitiatorComplete` (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 95
- `CA_LKMInitiatorComplete()` (in module *pycryptoki.cryptoki*), 130
- `CA_LKMReceiverComplete` (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 95
- `CA_LKMReceiverComplete()` (in module *pycryptoki.cryptoki*), 148
- `CA_LKMReceiverResponse` (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 96
- `CA_LKMReceiverResponse()` (in module *pycryptoki.cryptoki*), 112
- `CA_LoadEncryptedModule` (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 96
- `CA_LoadEncryptedModule()` (in module *pycryptoki.cryptoki*), 63
- `CA_LoadModule` (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 96
- `CA_LoadModule()` (in module *pycryptoki.cryptoki*), 65
- `CA_LockClusteredSlot` (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 96
- `CA_LockClusteredSlot()` (in module *pycryptoki.cryptoki*), 65
- `CA_LogExportSecret` (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 96
- `CA_LogExportSecret()` (in module *pycryptoki.cryptoki*), 112
- `CA_LogExternal` (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 96
- `CA_LogExternal()` (in module *pycryptoki.cryptoki*), 84
- `CA_LogGetConfig` (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 96
- `CA_LogGetConfig()` (in module *pycryptoki.cryptoki*), 133
- `CA_LogGetStatus` (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 96
- `CA_LogGetStatus()` (in module *pycryptoki.cryptoki*), 78
- `CA_LogImportSecret` (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 96
- `CA_LogImportSecret()` (in module *pycryptoki.cryptoki*), 109
- `CA_LogSetConfig` (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 96
- `CA_LogSetConfig()` (in module *pycryptoki.cryptoki*), 131
- `CA_LogVerify` (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 96
- `CA_LogVerify()` (in module *pycryptoki.cryptoki*), 78
- `CA_LogVerifyFile` (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 96
- `CA_LogVerifyFile()` (in module *pycryptoki.cryptoki*), 70
- `CA_M_OF_N_STATUS` (class in *pycryptoki.cryptoki*), 115
- `CA_ManualKCV` (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 96
- `CA_ManualKCV()` (in module *pycryptoki.cryptoki*), 117
- `CA_ModifyMofN` (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 96
- `CA_ModifyMofN()` (in module *pycryptoki.cryptoki*), 146
- `CA_ModifyUsageCount` (*pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute*), 97
- `CA_ModifyUsageCount()` (in module *pycryptoki.cryptoki*), 125
- `ca_modifyusagecount()` (in module *pycryptoki.key_management*), 23

- `ca_modifyusagecount_ex()` (in module `pycryptoki.key_management`), 23
- `CA_MOFN_ACTIVATION` (class in `pycryptoki.cryptoki`), 70
- `CA_MOFN_ACTIVATION_PTR` (in module `pycryptoki.cryptoki`), 69
- `CA_MOFN_GENERATION` (class in `pycryptoki.cryptoki`), 112
- `CA_MOFN_GENERATION_PTR` (in module `pycryptoki.cryptoki`), 141
- `CA_MOFN_STATUS` (in module `pycryptoki.cryptoki`), 107
- `CA_MOFN_STATUS_PTR` (in module `pycryptoki.cryptoki`), 109
- `CA_MTKGetState` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 96
- `CA_MTKGetState()` (in module `pycryptoki.cryptoki`), 103
- `CA_MTKResplit` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 96
- `CA_MTKResplit()` (in module `pycryptoki.cryptoki`), 139
- `ca_mtkresplit()` (in module `pycryptoki.hsm_management`), 51
- `ca_mtkresplit_ex()` (in module `pycryptoki.hsm_management`), 51
- `CA_MTKRestore` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 96
- `CA_MTKRestore()` (in module `pycryptoki.cryptoki`), 130
- `ca_mtkrestore()` (in module `pycryptoki.hsm_management`), 52
- `ca_mtkrestore_ex()` (in module `pycryptoki.hsm_management`), 52
- `CA_MTKSetStorage` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 96
- `CA_MTKSetStorage()` (in module `pycryptoki.cryptoki`), 73
- `CA_MTKZeroize` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 96
- `CA_MTKZeroize()` (in module `pycryptoki.cryptoki`), 63
- `ca_mtkzeroize()` (in module `pycryptoki.hsm_management`), 52
- `ca_mtkzeroize_ex()` (in module `pycryptoki.hsm_management`), 52
- `CA_MultisignValue` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 97
- `CA_MultisignValue()` (in module `pycryptoki.cryptoki`), 88
- `ca_open_secure_token()` (in module `pycryptoki.backup`), 58
- `ca_open_secure_token_ex()` (in module `pycryptoki.backup`), 58
- `CA_OpenApplicationID` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 97
- `CA_OpenApplicationID()` (in module `pycryptoki.cryptoki`), 137
- `ca_openapplicationID()` (in module `pycryptoki.session_management`), 15
- `ca_openapplicationID_ex()` (in module `pycryptoki.session_management`), 15
- `CA_OpenApplicationIDForContainer` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 97
- `CA_OpenApplicationIDForContainer()` (in module `pycryptoki.cryptoki`), 131
- `CA_OpenSecureToken` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 97
- `CA_OpenSecureToken()` (in module `pycryptoki.cryptoki`), 68
- `CA_OpenSession` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 97
- `CA_OpenSession()` (in module `pycryptoki.cryptoki`), 135
- `CA_OpenSessionWithAppID` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 97
- `CA_OpenSessionWithAppID()` (in module `pycryptoki.cryptoki`), 85
- `CA_PerformModuleCall` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 97
- `CA_PerformModuleCall()` (in module `pycryptoki.cryptoki`), 63
- `CA_PerformSelfTest` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 97
- `CA_PerformSelfTest()` (in module `pycryptoki.cryptoki`), 110
- `CA_QueryLicense` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 97
- `CA_QueryLicense()` (in module `pycryptoki.cryptoki`), 82
- `CA_ReadAllUtilizationCounters` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 97
- `CA_ReadAndResetUtilizationMetrics` (`py-`

cryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 97

CA_ReadCommonStore (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 97

CA_ReadCommonStore() (in module *pycryptoki.cryptoki*), 130

CA_ReadUtilizationMetrics (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 97

CA_ReplaceFastPathKEK (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 97

CA_ReplaceFastPathKEK() (in module *pycryptoki.cryptoki*), 76

CA_ResetDevice (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 97

CA_ResetDevice() (in module *pycryptoki.cryptoki*), 79

CA_ResetPIN (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 97

CA_ResetPIN() (in module *pycryptoki.cryptoki*), 86

CA_Restart (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 97

CA_Restart() (in module *pycryptoki.cryptoki*), 112

ca_restart() (in module *pycryptoki.session_management*), 15

ca_restart_ex() (in module *pycryptoki.session_management*), 15

CA_RestartForContainer (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 97

CA_RestartForContainer() (in module *pycryptoki.cryptoki*), 117

CA_RetrieveLicenseList (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 97

CA_RetrieveLicenseList() (in module *pycryptoki.cryptoki*), 101

CA_ROLE_STATE (class in *pycryptoki.cryptoki*), 121

CA_RoleStateGet (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 97

CA_RoleStateGet() (in module *pycryptoki.cryptoki*), 104

ca_set_destructive_hsm_policies() (in module *pycryptoki.hsm_management*), 52

ca_set_destructive_hsm_policies_ex() (in module *pycryptoki.hsm_management*), 53

ca_set_destructive_hsm_policy() (in module *pycryptoki.hsm_management*), 53

ca_set_destructive_hsm_policy_ex() (in module *pycryptoki.hsm_management*), 53

ca_set_hsm_policies() (in module *pycryptoki.hsm_management*), 54

ca_set_hsm_policies_ex() (in module *pycryptoki.hsm_management*), 54

ca_set_hsm_policy() (in module *pycryptoki.hsm_management*), 54

ca_set_hsm_policy_ex() (in module *pycryptoki.hsm_management*), 54

CA_SetApplicationID (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 99

CA_SetApplicationID() (in module *pycryptoki.cryptoki*), 63

ca_setapplicationID() (in module *pycryptoki.session_management*), 16

ca_setapplicationID_ex() (in module *pycryptoki.session_management*), 16

CA_SetAuthorizationData (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 99

CA_SetCloningDomain (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 99

CA_SetCloningDomain() (in module *pycryptoki.cryptoki*), 122

CA_SetContainerPolicies (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 99

CA_SetContainerPolicies() (in module *pycryptoki.cryptoki*), 124

CA_SetContainerPolicy (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 99

CA_SetContainerPolicy() (in module *pycryptoki.cryptoki*), 80

CA_SetContainerSize (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 99

CA_SetContainerSize() (in module *pycryptoki.cryptoki*), 106

CA_SetDestructiveHSM Policies (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 100

CA_SetDestructiveHSM Policies() (in module *pycryptoki.cryptoki*), 78

CA_SetDestructiveHSM Policy (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 100

CA_SetDestructiveHSM Policy() (in module *pycryptoki.cryptoki*), 138

CA_SetExtendedTPV (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST

- attribute*), 100
- CA_SetExtendedTPV() (in module *pycryptoki.cryptoki*), 108
- CA_SetHSMPolicies (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST *attribute*), 100
- CA_SetHSMPolicies() (in module *pycryptoki.cryptoki*), 133
- CA_SetHSMPolicy (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST *attribute*), 100
- CA_SetHSMPolicy() (in module *pycryptoki.cryptoki*), 125
- CA_SetKCV (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST *attribute*), 100
- CA_SetKCV() (in module *pycryptoki.cryptoki*), 104
- CA_SetLKCV (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST *attribute*), 100
- CA_SetLKCV() (in module *pycryptoki.cryptoki*), 152
- CA_SetMofN (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST *attribute*), 100
- CA_SetMofN() (in module *pycryptoki.cryptoki*), 74
- CA_SetPedId (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST *attribute*), 100
- CA_SetPedId() (in module *pycryptoki.cryptoki*), 78
- CA_SetRDk (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST *attribute*), 100
- CA_SetRDk() (in module *pycryptoki.cryptoki*), 82
- CA_SetTokenCertificateSignature (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST *attribute*), 100
- CA_SetTokenCertificateSignature() (in module *pycryptoki.cryptoki*), 142
- ca_settokencertificatecertificate() (in module *pycryptoki.hsm_management*), 54
- ca_settokencertificatecertificate_ex() (in module *pycryptoki.hsm_management*), 55
- CA_SetTokenPolicies (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST *attribute*), 100
- CA_SetTokenPolicies() (in module *pycryptoki.cryptoki*), 115
- CA_SetTPV (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST *attribute*), 100
- CA_SetTPV() (in module *pycryptoki.cryptoki*), 102
- CA_SetUserContainerName (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST *attribute*), 100
- CA_SetUserContainerName() (in module *pycryptoki.cryptoki*), 136
- ca_sim_extract() (in module *pycryptoki.backup*), 58
- ca_sim_extract_ex() (in module *pycryptoki.backup*), 59
- ca_sim_insert() (in module *pycryptoki.backup*), 59
- ca_sim_insert_ex() (in module *pycryptoki.backup*), 59
- ca_sim_multisign() (in module *pycryptoki.backup*), 60
- ca_sim_multisign_ex() (in module *pycryptoki.backup*), 60
- CA_SIMExtract (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST *attribute*), 98
- CA_SIMExtract() (in module *pycryptoki.cryptoki*), 134
- CA_SIMInsert (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST *attribute*), 98
- CA_SIMInsert() (in module *pycryptoki.cryptoki*), 119
- CA_SIMMultiSign (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST *attribute*), 98
- CA_SIMMultiSign() (in module *pycryptoki.cryptoki*), 89
- CA_SpRawRead (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST *attribute*), 100
- CA_SpRawRead() (in module *pycryptoki.cryptoki*), 72
- CA_SpRawWrite (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST *attribute*), 100
- CA_SpRawWrite() (in module *pycryptoki.cryptoki*), 71
- CA_STCClearCipherAlgorithm (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST *attribute*), 98
- CA_STCClearCipherAlgorithm() (in module *pycryptoki.cryptoki*), 126
- CA_STCClearDigestAlgorithm (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST *attribute*), 98
- CA_STCClearDigestAlgorithm() (in module *pycryptoki.cryptoki*), 117
- CA_STCDeregister (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST *attribute*), 98
- CA_STCDeregister() (in module *pycryptoki.cryptoki*), 122
- CA_STCGetAdminPubKey (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST *attribute*), 98
- CA_STCGetAdminPubKey() (in module *pycryptoki.cryptoki*), 83

CA_STCGetChannelID	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 98	CA_STCGetDigestNameByID	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 98
CA_STCGetChannelID()	(in module pycryptoki.cryptoki), 118	CA_STCGetDigestNameByID()	(in module pycryptoki.cryptoki), 109
CA_STCGetCipherAlgorithm	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 98	CA_STCGetKeyActivationTimeOut	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 98
CA_STCGetCipherAlgorithm()	(in module pycryptoki.cryptoki), 82	CA_STCGetKeyActivationTimeOut()	(in module pycryptoki.cryptoki), 137
CA_STCGetCipherID	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 98	CA_STCGetKeyLifeTime	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 98
CA_STCGetCipherID()	(in module pycryptoki.cryptoki), 110	CA_STCGetKeyLifeTime()	(in module pycryptoki.cryptoki), 81
CA_STCGetCipherIDs	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 98	CA_STCGetMaxSessions	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 99
CA_STCGetCipherIDs()	(in module pycryptoki.cryptoki), 131	CA_STCGetMaxSessions()	(in module pycryptoki.cryptoki), 70
CA_STCGetCipherNameByID	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 98	CA_STCGetPartPubKey	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 99
CA_STCGetCipherNameByID()	(in module pycryptoki.cryptoki), 83	CA_STCGetPartPubKey()	(in module pycryptoki.cryptoki), 139
CA_STCGetClientInfo	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 98	CA_STCGetPubKey	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 99
CA_STCGetClientInfo()	(in module pycryptoki.cryptoki), 111	CA_STCGetPubKey()	(in module pycryptoki.cryptoki), 71
CA_STCGetClientsList	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 98	CA_STCGetSequenceWindowSize	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 99
CA_STCGetClientsList()	(in module pycryptoki.cryptoki), 151	CA_STCGetSequenceWindowSize()	(in module pycryptoki.cryptoki), 151
CA_STCGetCurrentKeyLife	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 98	CA_STCGetState	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 99
CA_STCGetCurrentKeyLife()	(in module pycryptoki.cryptoki), 133	CA_STCGetState()	(in module pycryptoki.cryptoki), 136
CA_STCGetDigestAlgorithm	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 98	CA_STCIsEnabled	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 99
CA_STCGetDigestAlgorithm()	(in module pycryptoki.cryptoki), 123	CA_STCIsEnabled()	(in module pycryptoki.cryptoki), 148
CA_STCGetDigestID	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 98	CA_STCRegister	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 99
CA_STCGetDigestID()	(in module pycryptoki.cryptoki), 81	CA_STCRegister()	(in module pycryptoki.cryptoki), 129
CA_STCGetDigestIDs	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 98	CA_STCSetCipherAlgorithm	(pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 99
CA_STCGetDigestIDs()	(in module pycryptoki.cryptoki), 134		

- CA_STCSetCipherAlgorithm() (in module *pycryptoki.cryptoki*), 79
- CA_STCSetDigestAlgorithm (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 99
- CA_STCSetDigestAlgorithm() (in module *pycryptoki.cryptoki*), 63
- CA_STCSetKeyActivationTimeOut (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 99
- CA_STCSetKeyActivationTimeOut() (in module *pycryptoki.cryptoki*), 122
- CA_STCSetKeyLifeTime (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 99
- CA_STCSetKeyLifeTime() (in module *pycryptoki.cryptoki*), 141
- CA_STCSetMaxSessions (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 99
- CA_STCSetMaxSessions() (in module *pycryptoki.cryptoki*), 110
- CA_STCSetSequenceWindowSize (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 99
- CA_STCSetSequenceWindowSize() (in module *pycryptoki.cryptoki*), 76
- CA_STMGetState (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 99
- CA_STMGetState() (in module *pycryptoki.cryptoki*), 70
- CA_STMToggle (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 99
- CA_STMToggle() (in module *pycryptoki.cryptoki*), 133
- CA_SwitchSecondarySlot (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 100
- CA_SwitchSecondarySlot() (in module *pycryptoki.cryptoki*), 143
- CA_TamperClear() (in module *pycryptoki.cryptoki*), 150
- ca_time_sync() (in module *pycryptoki.audit_handling*), 56
- ca_time_sync_ex() (in module *pycryptoki.audit_handling*), 56
- CA_TimeSync (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 100
- CA_TimeSync() (in module *pycryptoki.cryptoki*), 106
- CA-TokenDelete (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 100
- CA-TokenDelete() (in module *pycryptoki.cryptoki*), 116
- CA-TokenInsert (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 100
- CA-TokenInsert() (in module *pycryptoki.cryptoki*), 124
- CA-TokenInsertNoAuth (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 100
- CA-TokenInsertNoAuth() (in module *pycryptoki.cryptoki*), 120
- CA-TokenZeroize (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 101
- CA-TokenZeroize() (in module *pycryptoki.cryptoki*), 64
- CA_UnloadModule (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 101
- CA_UnloadModule() (in module *pycryptoki.cryptoki*), 116
- CA_UnlockClusteredSlot (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 101
- CA_UnlockClusteredSlot() (in module *pycryptoki.cryptoki*), 119
- CA_WaitForSlotEvent (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 101
- CA_WaitForSlotEvent() (in module *pycryptoki.cryptoki*), 85
- CA_WriteCommonStore (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 101
- CA_WriteCommonStore() (in module *pycryptoki.cryptoki*), 80
- CA_Zeroize (pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST attribute), 101
- CA_Zeroize() (in module *pycryptoki.cryptoki*), 141
- cb (pycryptoki.cryptoki.CK_AES_CTR_PARAMS attribute), 108
- cb (pycryptoki.cryptoki.CK_AES_XTS_PARAMS attribute), 132
- cb (pycryptoki.cryptoki.CK_CAMELLIA_CTR_PARAMS attribute), 86
- cb (pycryptoki.cryptoki.CK_DES_CTR_PARAMS attribute), 70
- CERTIFICATE_TEMPLATE (in module *pycryptoki.default_templates*), 61
- certificateHandle (pycryptoki.cryptoki.CK_CMS_SIG_PARAMS attribute), 100

- tribute*), 68
- CK_AES_CBC_ENCRYPT_DATA_PARAMS (class in *pycryptoki.cryptoki*), 123
- CK_AES_CBC_ENCRYPT_DATA_PARAMS_PTR (in module *pycryptoki.cryptoki*), 138
- CK_AES_CBC_PAD_EXTRACT_PARAMS (class in *pycryptoki.cryptoki*), 113
- CK_AES_CBC_PAD_EXTRACT_PARAMS_PTR (in module *pycryptoki.cryptoki*), 120
- CK_AES_CBC_PAD_INSERT_PARAMS (class in *pycryptoki.cryptoki*), 142
- CK_AES_CBC_PAD_INSERT_PARAMS_PTR (in module *pycryptoki.cryptoki*), 86
- CK_AES_CTR_PARAMS (class in *pycryptoki.cryptoki*), 108
- CK_AES_CTR_PARAMS_PTR (in module *pycryptoki.cryptoki*), 119
- CK_AES_GCM_PARAMS (class in *pycryptoki.cryptoki*), 132
- CK_AES_GCM_PARAMS_PTR (in module *pycryptoki.cryptoki*), 141
- CK_AES_GMAC_PARAMS (in module *pycryptoki.cryptoki*), 116
- CK_AES_GMAC_PARAMS_PTR (in module *pycryptoki.cryptoki*), 101
- CK_AES_XTS_PARAMS (class in *pycryptoki.cryptoki*), 132
- CK_AES_XTS_PARAMS_PTR (in module *pycryptoki.cryptoki*), 85
- CK_ARIA_CBC_ENCRYPT_DATA_PARAMS (class in *pycryptoki.cryptoki*), 117
- CK_ARIA_CBC_ENCRYPT_DATA_PARAMS_PTR (in module *pycryptoki.cryptoki*), 125
- CK_ARIA_CTR_PARAMS (in module *pycryptoki.cryptoki*), 86
- CK_ARIA_CTR_PARAMS_PTR (in module *pycryptoki.cryptoki*), 105
- CK_ATTRIBUTE (class in *pycryptoki.cryptoki*), 104
- CK_ATTRIBUTE_PTR (in module *pycryptoki.cryptoki*), 64
- CK_ATTRIBUTE_TYPE (in module *pycryptoki.cryptoki*), 138
- CK_BBOOL (in module *pycryptoki.cryptoki*), 127
- CK_BYTE (in module *pycryptoki.cryptoki*), 105
- CK_BYTE_PTR (in module *pycryptoki.cryptoki*), 106
- CK_C_CancelFunction (in module *pycryptoki.cryptoki*), 110
- CK_C_CloseAllSessions (in module *pycryptoki.cryptoki*), 85
- CK_C_CloseSession (in module *pycryptoki.cryptoki*), 116
- CK_C_CopyObject (in module *pycryptoki.cryptoki*), 107
- CK_C_CreateObject (in module *pycryptoki.cryptoki*), 74
- CK_C_Decrypt (in module *pycryptoki.cryptoki*), 125
- CK_C_DecryptDigestUpdate (in module *pycryptoki.cryptoki*), 141
- CK_C_DecryptFinal (in module *pycryptoki.cryptoki*), 74
- CK_C_DecryptInit (in module *pycryptoki.cryptoki*), 135
- CK_C_DecryptUpdate (in module *pycryptoki.cryptoki*), 115
- CK_C_DecryptVerifyUpdate (in module *pycryptoki.cryptoki*), 124
- CK_C_DeriveKey (in module *pycryptoki.cryptoki*), 119
- CK_C_DestroyObject (in module *pycryptoki.cryptoki*), 150
- CK_C_Digest (in module *pycryptoki.cryptoki*), 114
- CK_C_DigestEncryptUpdate (in module *pycryptoki.cryptoki*), 84
- CK_C_DigestFinal (in module *pycryptoki.cryptoki*), 139
- CK_C_DigestInit (in module *pycryptoki.cryptoki*), 146
- CK_C_DigestKey (in module *pycryptoki.cryptoki*), 118
- CK_C_DigestUpdate (in module *pycryptoki.cryptoki*), 126
- CK_C_Encrypt (in module *pycryptoki.cryptoki*), 150
- CK_C_EncryptFinal (in module *pycryptoki.cryptoki*), 80
- CK_C_EncryptInit (in module *pycryptoki.cryptoki*), 130
- CK_C_EncryptUpdate (in module *pycryptoki.cryptoki*), 118
- CK_C_Finalize (in module *pycryptoki.cryptoki*), 114
- CK_C_FindObjects (in module *pycryptoki.cryptoki*), 125
- CK_C_FindObjectsFinal (in module *pycryptoki.cryptoki*), 105
- CK_C_FindObjectsInit (in module *pycryptoki.cryptoki*), 135
- CK_C_GenerateKey (in module *pycryptoki.cryptoki*), 119
- CK_C_GenerateKeyPair (in module *pycryptoki.cryptoki*), 127
- CK_C_GenerateRandom (in module *pycryptoki.cryptoki*), 139
- CK_C_GetAttributeValue (in module *pycryptoki.cryptoki*), 78
- CK_C_GetFunctionList (in module *pycryptoki.cryptoki*), 68
- CK_C_GetFunctionStatus (in module *pycryptoki.cryptoki*), 103
- CK_C_GetInfo (in module *pycryptoki.cryptoki*), 123

- CK_C_GetMechanismInfo (in module *pycryptoki.cryptoki*), 138
- CK_C_GetMechanismList (in module *pycryptoki.cryptoki*), 106
- CK_C_GetObjectSize (in module *pycryptoki.cryptoki*), 150
- CK_C_GetOperationState (in module *pycryptoki.cryptoki*), 129
- CK_C_GetSessionInfo (in module *pycryptoki.cryptoki*), 127
- CK_C_GetSlotInfo (in module *pycryptoki.cryptoki*), 107
- CK_C_GetSlotList (in module *pycryptoki.cryptoki*), 73
- CK_C_GetTokenInfo (in module *pycryptoki.cryptoki*), 143
- CK_C_Initialize (in module *pycryptoki.cryptoki*), 126
- CK_C_INITIALIZE_ARGS (class in *pycryptoki.cryptoki*), 131
- CK_C_INITIALIZE_ARGS_PTR (in module *pycryptoki.cryptoki*), 86
- CK_C_InitPIN (in module *pycryptoki.cryptoki*), 113
- CK_C_InitToken (in module *pycryptoki.cryptoki*), 150
- CK_C_Login (in module *pycryptoki.cryptoki*), 148
- CK_C_Logout (in module *pycryptoki.cryptoki*), 82
- CK_C_OpenSession (in module *pycryptoki.cryptoki*), 68
- CK_C_SeedRandom (in module *pycryptoki.cryptoki*), 105
- CK_C_SetAttributeValue (in module *pycryptoki.cryptoki*), 89
- CK_C_SetOperationState (in module *pycryptoki.cryptoki*), 63
- CK_C_SetPIN (in module *pycryptoki.cryptoki*), 64
- CK_C_Sign (in module *pycryptoki.cryptoki*), 112
- CK_C_SignEncryptUpdate (in module *pycryptoki.cryptoki*), 132
- CK_C_SignFinal (in module *pycryptoki.cryptoki*), 137
- CK_C_SignInit (in module *pycryptoki.cryptoki*), 79
- CK_C_SignRecover (in module *pycryptoki.cryptoki*), 121
- CK_C_SignRecoverInit (in module *pycryptoki.cryptoki*), 83
- CK_C_SignUpdate (in module *pycryptoki.cryptoki*), 104
- CK_C_UnwrapKey (in module *pycryptoki.cryptoki*), 88
- CK_C_Verify (in module *pycryptoki.cryptoki*), 87
- CK_C_VerifyFinal (in module *pycryptoki.cryptoki*), 103
- CK_C_VerifyInit (in module *pycryptoki.cryptoki*), 80
- CK_C_VerifyRecover (in module *pycryptoki.cryptoki*), 110
- CK_C_VerifyRecoverInit (in module *pycryptoki.cryptoki*), 120
- CK_C_VerifyUpdate (in module *pycryptoki.cryptoki*), 71
- CK_C_WaitForSlotEvent (in module *pycryptoki.cryptoki*), 82
- CK_C_WrapKey (in module *pycryptoki.cryptoki*), 71
- CK_CA_ActivateMofN (in module *pycryptoki.cryptoki*), 118
- CK_CA_CapabilityUpdate (in module *pycryptoki.cryptoki*), 130
- CK_CA_CheckOperationState (in module *pycryptoki.cryptoki*), 128
- CK_CA_ChoosePrimarySlot (in module *pycryptoki.cryptoki*), 120
- CK_CA_ChooseSecondarySlot (in module *pycryptoki.cryptoki*), 109
- CK_CA_CloneAllObjectsToSession (in module *pycryptoki.cryptoki*), 111
- CK_CA_CloneAsSource (in module *pycryptoki.cryptoki*), 89
- CK_CA_CloneAsTarget (in module *pycryptoki.cryptoki*), 84
- CK_CA_CloneAsTargetInit (in module *pycryptoki.cryptoki*), 65
- CK_CA_CloneModifyMofN (in module *pycryptoki.cryptoki*), 88
- CK_CA_CloneMofN (in module *pycryptoki.cryptoki*), 125
- CK_CA_CloneObject (in module *pycryptoki.cryptoki*), 104
- CK_CA_CloneObjectToAllSessions (in module *pycryptoki.cryptoki*), 75
- CK_CA_ClonePrivateKey (in module *pycryptoki.cryptoki*), 80
- CK_CA_CloseAllSecondarySessions (in module *pycryptoki.cryptoki*), 106
- CK_CA_CloseApplicationID (in module *pycryptoki.cryptoki*), 149
- CK_CA_CloseApplicationIDForContainer (in module *pycryptoki.cryptoki*), 81
- CK_CA_CloseSecondarySession (in module *pycryptoki.cryptoki*), 86
- CK_CA_CloseSecureToken (in module *pycryptoki.cryptoki*), 114
- CK_CA_ConfigureRemotePED (in module *pycryptoki.cryptoki*), 83
- CK_CA_CreateContainer (in module *pycryptoki.cryptoki*), 123
- CK_CA_CreateContainerLoginChallenge (in module *pycryptoki.cryptoki*), 82
- CK_CA_CreateLoginChallenge (in module *py-*

- cryptoki.cryptoki*), 107
- CK_CA_Deactivate (in module *pycryptoki.cryptoki*), 103
- CK_CA_DeactivateMofN (in module *pycryptoki.cryptoki*), 72
- CK_CA_DeleteContainer (in module *pycryptoki.cryptoki*), 139
- CK_CA_DeleteContainerWithHandle (in module *pycryptoki.cryptoki*), 115
- CK_CA_DeleteRemotePEDVector (in module *pycryptoki.cryptoki*), 109
- CK_CA_DestroyMultipleObjects (in module *pycryptoki.cryptoki*), 124
- CK_CA_DisableUnauthTokenInsertion (in module *pycryptoki.cryptoki*), 89
- CK_CA_DismantleRemotePED (in module *pycryptoki.cryptoki*), 128
- CK_CA_DuplicateMofN (in module *pycryptoki.cryptoki*), 146
- CK_CA_EnableUnauthTokenInsertion (in module *pycryptoki.cryptoki*), 141
- CK_CA_EncodeECChar2Params (in module *pycryptoki.cryptoki*), 127
- CK_CA_EncodeECParamsFromFile (in module *pycryptoki.cryptoki*), 114
- CK_CA_EncodeECPrimeParams (in module *pycryptoki.cryptoki*), 107
- CK_CA_Extract (in module *pycryptoki.cryptoki*), 89
- CK_CA_ExtractMaskedObject (in module *pycryptoki.cryptoki*), 121
- CK_CA_FactoryReset (in module *pycryptoki.cryptoki*), 81
- CK_CA_FindAdminSlotForSlot (in module *pycryptoki.cryptoki*), 107
- CK_CA_FirmwareRollback (in module *pycryptoki.cryptoki*), 75
- CK_CA_FirmwareUpdate (in module *pycryptoki.cryptoki*), 111
- CK_CA_GenerateCloneableMofN (in module *pycryptoki.cryptoki*), 151
- CK_CA_GenerateCloningKEV (in module *pycryptoki.cryptoki*), 115
- CK_CA_GenerateMofN (in module *pycryptoki.cryptoki*), 87
- CK_CA_GenerateTokenKeys (in module *pycryptoki.cryptoki*), 148
- CK_CA_GetClusterState (in module *pycryptoki.cryptoki*), 151
- CK_CA_GetConfigurationElementDescription (in module *pycryptoki.cryptoki*), 64
- CK_CA_GetContainerCapabilitySet (in module *pycryptoki.cryptoki*), 111
- CK_CA_GetContainerCapabilitySetting (in module *pycryptoki.cryptoki*), 138
- CK_CA_GetContainerList (in module *pycryptoki.cryptoki*), 88
- CK_CA_GetContainerName (in module *pycryptoki.cryptoki*), 149
- CK_CA_GetContainerPolicySet (in module *pycryptoki.cryptoki*), 119
- CK_CA_GetContainerPolicySetting (in module *pycryptoki.cryptoki*), 105
- CK_CA_GetContainerStatus (in module *pycryptoki.cryptoki*), 110
- CK_CA_GetContainerStorageInformation (in module *pycryptoki.cryptoki*), 104
- CK_CA_GetExtendedTPV (in module *pycryptoki.cryptoki*), 134
- CK_CA_GetFPV (in module *pycryptoki.cryptoki*), 65
- CK_CA_GetFunctionList (in module *pycryptoki.cryptoki*), 84
- CK_CA_GetHASTate (in module *pycryptoki.cryptoki*), 67
- CK_CA_GetHSMCapabilitySet (in module *pycryptoki.cryptoki*), 89
- CK_CA_GetHSMCapabilitySetting (in module *pycryptoki.cryptoki*), 88
- CK_CA_GetHSMPolicySet (in module *pycryptoki.cryptoki*), 116
- CK_CA_GetHSMPolicySetting (in module *pycryptoki.cryptoki*), 82
- CK_CA_GetHSMStats (in module *pycryptoki.cryptoki*), 121
- CK_CA_GetHSMStorageInformation (in module *pycryptoki.cryptoki*), 140
- CK_CA_GetModuleInfo (in module *pycryptoki.cryptoki*), 79
- CK_CA_GetModuleList (in module *pycryptoki.cryptoki*), 137
- CK_CA_GetMofNStatus (in module *pycryptoki.cryptoki*), 138
- CK_CA_GetNumberOfAllowedContainers (in module *pycryptoki.cryptoki*), 76
- CK_CA_GetObjectHandle (in module *pycryptoki.cryptoki*), 105
- CK_CA_GetObjectUID (in module *pycryptoki.cryptoki*), 82
- CK_CA_GetPedId (in module *pycryptoki.cryptoki*), 151
- CK_CA_GetPrimarySlot (in module *pycryptoki.cryptoki*), 87
- CK_CA_GetRemotePEDVectorStatus (in module *pycryptoki.cryptoki*), 86
- CK_CA_GetRollbackFirmwareVersion (in module *pycryptoki.cryptoki*), 138
- CK_CA_GetSecondarySlot (in module *pycryptoki.cryptoki*), 85
- CK_CA_GetSecureElementMeta (in module *py-*

- cryptoki.cryptoki*), 66
- CK_CA_GetServerInstanceBySlotID (in module *pycryptoki.cryptoki*), 85
- CK_CA_GetSessionInfo (in module *pycryptoki.cryptoki*), 75
- CK_CA_GetSlotIdForContainer (in module *pycryptoki.cryptoki*), 65
- CK_CA_GetSlotIdForPhysicalSlot (in module *pycryptoki.cryptoki*), 142
- CK_CA_GetSlotListFromServerInstance (in module *pycryptoki.cryptoki*), 106
- CK_CA_GetTime (in module *pycryptoki.cryptoki*), 70
- CK_CA_GetTokenCapabilities (in module *pycryptoki.cryptoki*), 88
- CK_CA_GetTokenCertificateInfo (in module *pycryptoki.cryptoki*), 139
- CK_CA_GetTokenCertificates (in module *pycryptoki.cryptoki*), 117
- CK_CA_GetTokenInsertionCount (in module *pycryptoki.cryptoki*), 109
- CK_CA_GetTokenObjectHandle (in module *pycryptoki.cryptoki*), 103
- CK_CA_GetTokenObjectUID (in module *pycryptoki.cryptoki*), 127
- CK_CA_GetTokenPolicies (in module *pycryptoki.cryptoki*), 120
- CK_CA_GetTokenStatus (in module *pycryptoki.cryptoki*), 123
- CK_CA_GetTokenStorageInformation (in module *pycryptoki.cryptoki*), 150
- CK_CA_GetTPV (in module *pycryptoki.cryptoki*), 125
- CK_CA_GetTSV (in module *pycryptoki.cryptoki*), 148
- CK_CA_GetTunnelSlotNumber (in module *pycryptoki.cryptoki*), 131
- CK_CA_GetUnauthTokenInsertionStatus (in module *pycryptoki.cryptoki*), 103
- CK_CA_GetUserContainerName (in module *pycryptoki.cryptoki*), 151
- CK_CA_GetUserContainerNumber (in module *pycryptoki.cryptoki*), 139
- CK_CA_HAActivateMofN (in module *pycryptoki.cryptoki*), 142
- CK_CA_HAAnswerLoginChallenge (in module *pycryptoki.cryptoki*), 76
- CK_CA_HAAnswerMofNChallenge (in module *pycryptoki.cryptoki*), 86
- CK_CA_HAGetLoginChallenge (in module *pycryptoki.cryptoki*), 124
- CK_CA_HAGetMasterPublic (in module *pycryptoki.cryptoki*), 89
- CK_CA_HAInit (in module *pycryptoki.cryptoki*), 112
- CK_CA_HALogin (in module *pycryptoki.cryptoki*), 84
- CK_CA_IndirectLogin (in module *pycryptoki.cryptoki*), 120
- CK_CA_InitAudit (in module *pycryptoki.cryptoki*), 120
- CK_CA_InitializeRemotePEDVector (in module *pycryptoki.cryptoki*), 148
- CK_CA_InitIndirectPIN (in module *pycryptoki.cryptoki*), 77
- CK_CA_InitIndirectToken (in module *pycryptoki.cryptoki*), 119
- CK_CA_InitRolePIN (in module *pycryptoki.cryptoki*), 65
- CK_CA_InitSlotRolePIN (in module *pycryptoki.cryptoki*), 141
- CK_CA_Insert (in module *pycryptoki.cryptoki*), 149
- CK_CA_InsertMaskedObject (in module *pycryptoki.cryptoki*), 136
- CK_CA_InvokeService (in module *pycryptoki.cryptoki*), 73
- CK_CA_InvokeServiceAsynch (in module *pycryptoki.cryptoki*), 120
- CK_CA_InvokeServiceFinal (in module *pycryptoki.cryptoki*), 112
- CK_CA_InvokeServiceInit (in module *pycryptoki.cryptoki*), 128
- CK_CA_InvokeServiceSinglePart (in module *pycryptoki.cryptoki*), 124
- CK_CA_IsMofNEnabled (in module *pycryptoki.cryptoki*), 152
- CK_CA_IsMofNRequired (in module *pycryptoki.cryptoki*), 126
- CK_CA_ListSecureTokenInit (in module *pycryptoki.cryptoki*), 114
- CK_CA_ListSecureTokenUpdate (in module *pycryptoki.cryptoki*), 112
- CK_CA_LKMInitiatorChallenge (in module *pycryptoki.cryptoki*), 142
- CK_CA_LKMInitiatorComplete (in module *pycryptoki.cryptoki*), 78
- CK_CA_LKMReceiverComplete (in module *pycryptoki.cryptoki*), 69
- CK_CA_LKMReceiverResponse (in module *pycryptoki.cryptoki*), 151
- CK_CA_LoadEncryptedModule (in module *pycryptoki.cryptoki*), 102
- CK_CA_LoadModule (in module *pycryptoki.cryptoki*), 86
- CK_CA_LockClusteredSlot (in module *pycryptoki.cryptoki*), 147
- CK_CA_LogExportSecret (in module *pycryptoki.cryptoki*), 131
- CK_CA_LogExternal (in module *pycryptoki.cryptoki*), 114
- CK_CA_LogGetConfig (in module *pycryptoki.cryptoki*), 148
- CK_CA_LogGetStatus (in module *pycryptoki.cryptoki*), 148

- toki.cryptoki*), 68
- CK_CA_LogImportSecret (in module *pycryptoki.cryptoki*), 125
- CK_CA_LogSetConfig (in module *pycryptoki.cryptoki*), 127
- CK_CA_LogVerify (in module *pycryptoki.cryptoki*), 148
- CK_CA_LogVerifyFile (in module *pycryptoki.cryptoki*), 124
- CK_CA_ManualKCV (in module *pycryptoki.cryptoki*), 78
- CK_CA_ModifyMofN (in module *pycryptoki.cryptoki*), 78
- CK_CA_ModifyUsageCount (in module *pycryptoki.cryptoki*), 139
- CK_CA_MTKGetState (in module *pycryptoki.cryptoki*), 151
- CK_CA_MTKResplit (in module *pycryptoki.cryptoki*), 69
- CK_CA_MTKRestore (in module *pycryptoki.cryptoki*), 86
- CK_CA_MTKSetStorage (in module *pycryptoki.cryptoki*), 121
- CK_CA_MTKZeroize (in module *pycryptoki.cryptoki*), 151
- CK_CA_MultisignValue (in module *pycryptoki.cryptoki*), 66
- CK_CA_OpenApplicationID (in module *pycryptoki.cryptoki*), 101
- CK_CA_OpenApplicationIDForContainer (in module *pycryptoki.cryptoki*), 88
- CK_CA_OpenSecureToken (in module *pycryptoki.cryptoki*), 134
- CK_CA_OpenSession (in module *pycryptoki.cryptoki*), 111
- CK_CA_OpenSessionWithAppID (in module *pycryptoki.cryptoki*), 122
- CK_CA_PerformModuleCall (in module *pycryptoki.cryptoki*), 116
- CK_CA_PerformSelfTest (in module *pycryptoki.cryptoki*), 127
- CK_CA_QueryLicense (in module *pycryptoki.cryptoki*), 124
- CK_CA_ReadCommonStore (in module *pycryptoki.cryptoki*), 112
- CK_CA_ReplaceFastPathKEK (in module *pycryptoki.cryptoki*), 124
- CK_CA_ResetDevice (in module *pycryptoki.cryptoki*), 116
- CK_CA_ResetPIN (in module *pycryptoki.cryptoki*), 125
- CK_CA_Restart (in module *pycryptoki.cryptoki*), 63
- CK_CA_RestartForContainer (in module *pycryptoki.cryptoki*), 114
- CK_CA_RetrieveLicenseList (in module *pycryptoki.cryptoki*), 69
- CK_CA_RoleStateGet (in module *pycryptoki.cryptoki*), 120
- CK_CA_SetApplicationID (in module *pycryptoki.cryptoki*), 123
- CK_CA_SetCloningDomain (in module *pycryptoki.cryptoki*), 126
- CK_CA_SetContainerPolicies (in module *pycryptoki.cryptoki*), 74
- CK_CA_SetContainerPolicy (in module *pycryptoki.cryptoki*), 74
- CK_CA_SetContainerSize (in module *pycryptoki.cryptoki*), 101
- CK_CA_SetDestructiveHSMPolicies (in module *pycryptoki.cryptoki*), 104
- CK_CA_SetDestructiveHSMPolicy (in module *pycryptoki.cryptoki*), 128
- CK_CA_SetExtendedTPV (in module *pycryptoki.cryptoki*), 137
- CK_CA_SetHSMPolicies (in module *pycryptoki.cryptoki*), 120
- CK_CA_SetHSMPolicy (in module *pycryptoki.cryptoki*), 134
- CK_CA_SetKCV (in module *pycryptoki.cryptoki*), 112
- CK_CA_SetLKCV (in module *pycryptoki.cryptoki*), 86
- CK_CA_SetMofN (in module *pycryptoki.cryptoki*), 128
- CK_CA_SetPedId (in module *pycryptoki.cryptoki*), 89
- CK_CA_SetRDK (in module *pycryptoki.cryptoki*), 75
- CK_CA_SetTokenCertificateSignature (in module *pycryptoki.cryptoki*), 125
- CK_CA_SetTokenPolicies (in module *pycryptoki.cryptoki*), 122
- CK_CA_SetTPV (in module *pycryptoki.cryptoki*), 79
- CK_CA_SetUserContainerName (in module *pycryptoki.cryptoki*), 85
- CK_CA_SIMExtract (in module *pycryptoki.cryptoki*), 79
- CK_CA_SIMInsert (in module *pycryptoki.cryptoki*), 79
- CK_CA_SIMMultiSign (in module *pycryptoki.cryptoki*), 120
- CK_CA_SpRawRead (in module *pycryptoki.cryptoki*), 129
- CK_CA_SpRawWrite (in module *pycryptoki.cryptoki*), 67
- CK_CA_STCClearCipherAlgorithm (in module *pycryptoki.cryptoki*), 151
- CK_CA_STCClearDigestAlgorithm (in module *pycryptoki.cryptoki*), 113
- CK_CA_STCDeregister (in module *pycryptoki.cryptoki*), 117
- CK_CA_STCGetAdminPubKey (in module *pycryptoki.cryptoki*), 143

- CK_CA_STCGetChannelID (in module *pycryptoki.cryptoki*), 127
- CK_CA_STCGetCipherAlgorithm (in module *pycryptoki.cryptoki*), 70
- CK_CA_STCGetCipherID (in module *pycryptoki.cryptoki*), 122
- CK_CA_STCGetCipherIDs (in module *pycryptoki.cryptoki*), 107
- CK_CA_STCGetCipherNameByID (in module *pycryptoki.cryptoki*), 103
- CK_CA_STCGetClientInfo (in module *pycryptoki.cryptoki*), 71
- CK_CA_STCGetClientsList (in module *pycryptoki.cryptoki*), 105
- CK_CA_STCGetCurrentKeyLife (in module *pycryptoki.cryptoki*), 127
- CK_CA_STCGetDigestAlgorithm (in module *pycryptoki.cryptoki*), 83
- CK_CA_STCGetDigestID (in module *pycryptoki.cryptoki*), 64
- CK_CA_STCGetDigestIDs (in module *pycryptoki.cryptoki*), 117
- CK_CA_STCGetDigestNameByID (in module *pycryptoki.cryptoki*), 142
- CK_CA_STCGetKeyActivationTimeOut (in module *pycryptoki.cryptoki*), 147
- CK_CA_STCGetKeyLifeTime (in module *pycryptoki.cryptoki*), 150
- CK_CA_STCGetMaxSessions (in module *pycryptoki.cryptoki*), 107
- CK_CA_STCGetPartPubKey (in module *pycryptoki.cryptoki*), 150
- CK_CA_STCGetPubKey (in module *pycryptoki.cryptoki*), 72
- CK_CA_STCGetSequenceWindowSize (in module *pycryptoki.cryptoki*), 135
- CK_CA_STCGetState (in module *pycryptoki.cryptoki*), 73
- CK_CA_STCIsEnabled (in module *pycryptoki.cryptoki*), 74
- CK_CA_STCRegister (in module *pycryptoki.cryptoki*), 108
- CK_CA_STCSetCipherAlgorithm (in module *pycryptoki.cryptoki*), 79
- CK_CA_STCSetDigestAlgorithm (in module *pycryptoki.cryptoki*), 74
- CK_CA_STCSetKeyActivationTimeOut (in module *pycryptoki.cryptoki*), 67
- CK_CA_STCSetKeyLifeTime (in module *pycryptoki.cryptoki*), 105
- CK_CA_STCSetMaxSessions (in module *pycryptoki.cryptoki*), 65
- CK_CA_STCSetSequenceWindowSize (in module *pycryptoki.cryptoki*), 136
- CK_CA_STMGetState (in module *pycryptoki.cryptoki*), 141
- CK_CA_STMToggle (in module *pycryptoki.cryptoki*), 124
- CK_CA_SwitchSecondarySlot (in module *pycryptoki.cryptoki*), 104
- CK_CA_TimeSync (in module *pycryptoki.cryptoki*), 109
- CK_CA-TokenDelete (in module *pycryptoki.cryptoki*), 83
- CK_CA-TokenInsert (in module *pycryptoki.cryptoki*), 117
- CK_CA-TokenInsertNoAuth (in module *pycryptoki.cryptoki*), 120
- CK_CA-TokenZeroize (in module *pycryptoki.cryptoki*), 78
- CK_CA_UnloadModule (in module *pycryptoki.cryptoki*), 135
- CK_CA_UnlockClusteredSlot (in module *pycryptoki.cryptoki*), 88
- CK_CA_WaitForSlotEvent (in module *pycryptoki.cryptoki*), 102
- CK_CA_WriteCommonStore (in module *pycryptoki.cryptoki*), 71
- CK_CA_Zeroize (in module *pycryptoki.cryptoki*), 118
- CK_CAMELLIA_CBC_ENCRYPT_DATA_PARAMS (class in *pycryptoki.cryptoki*), 120
- CK_CAMELLIA_CBC_ENCRYPT_DATA_PARAMS_PTR (in module *pycryptoki.cryptoki*), 132
- CK_CAMELLIA_CTR_PARAMS (class in *pycryptoki.cryptoki*), 86
- CK_CAMELLIA_CTR_PARAMS_PTR (in module *pycryptoki.cryptoki*), 101
- CK_CERTIFICATE_TYPE (in module *pycryptoki.cryptoki*), 83
- CK_CHAR (in module *pycryptoki.cryptoki*), 140
- CK_CHAR_PTR (in module *pycryptoki.cryptoki*), 79
- CK_CLUSTER_STATE (class in *pycryptoki.cryptoki*), 74
- CK_CLUSTER_STATE_PTR (in module *pycryptoki.cryptoki*), 121
- CK_CMS_SIG_PARAMS (class in *pycryptoki.cryptoki*), 68
- CK_CMS_SIG_PARAMS_PTR (in module *pycryptoki.cryptoki*), 140
- CK_CREATEMUTEX (in module *pycryptoki.cryptoki*), 78
- CK_DATE (class in *pycryptoki.cryptoki*), 63
- CK_DES_CBC_ENCRYPT_DATA_PARAMS (class in *pycryptoki.cryptoki*), 120
- CK_DES_CBC_ENCRYPT_DATA_PARAMS_PTR (in module *pycryptoki.cryptoki*), 124
- CK_DES_CTR_PARAMS (class in *pycryptoki.cryptoki*), 70
- CK_DES_CTR_PARAMS_PTR (in module *pycryptoki.cryptoki*), 73

- CK_DESTROYMUTEX (in module *pycryptoki.cryptoki*), 72
- CK_EC_DH_PRIMITIVE (in module *pycryptoki.cryptoki*), 148
- CK_EC_ENC_SCHEME (in module *pycryptoki.cryptoki*), 116
- CK_EC_KDF_TYPE (in module *pycryptoki.cryptoki*), 78
- CK_EC_MAC_SCHEME (in module *pycryptoki.cryptoki*), 71
- CK_ECDH1_DERIVE_PARAMS (class in *pycryptoki.cryptoki*), 109
- CK_ECDH1_DERIVE_PARAMS_PTR (in module *pycryptoki.cryptoki*), 150
- CK_ECDH2_DERIVE_PARAMS (class in *pycryptoki.cryptoki*), 127
- CK_ECDH2_DERIVE_PARAMS_PTR (in module *pycryptoki.cryptoki*), 120
- CK_ECIES_PARAMS (class in *pycryptoki.cryptoki*), 107
- CK_ECIES_PARAMS_PTR (in module *pycryptoki.cryptoki*), 151
- CK_ECMQV_DERIVE_PARAMS (class in *pycryptoki.cryptoki*), 69
- CK_ECMQV_DERIVE_PARAMS_PTR (in module *pycryptoki.cryptoki*), 72
- CK_EXTRACT_PARAMS (in module *pycryptoki.cryptoki*), 140
- CK_EXTRACT_PARAMS_PTR (in module *pycryptoki.cryptoki*), 132
- CK_FLAGS (in module *pycryptoki.cryptoki*), 74
- CK_FUNCTION_LIST (class in *pycryptoki.cryptoki*), 143
- CK_FUNCTION_LIST_PTR (in module *pycryptoki.cryptoki*), 89
- CK_FUNCTION_LIST_PTR_PTR (in module *pycryptoki.cryptoki*), 148
- CK_GetTotalOperations (in module *pycryptoki.cryptoki*), 105
- CK_HA_MEMBER (class in *pycryptoki.cryptoki*), 134
- CK_HA_MEMBER_PTR (in module *pycryptoki.cryptoki*), 74
- CK_HA_STATE_PTR (in module *pycryptoki.cryptoki*), 133
- CK_HA_STATUS (class in *pycryptoki.cryptoki*), 118
- CK_HW_FEATURE_TYPE (in module *pycryptoki.cryptoki*), 121
- CK_INFO (class in *pycryptoki.cryptoki*), 66
- CK_INFO_PTR (in module *pycryptoki.cryptoki*), 104
- CK_KDF_PRF_ENCODING_SCHEME (in module *pycryptoki.cryptoki*), 135
- CK_KDF_PRF_PARAMS (class in *pycryptoki.cryptoki*), 71
- CK_KDF_PRF_PARAMS_PTR (in module *pycryptoki.cryptoki*), 123
- CK_KDF_PRF_TYPE (in module *pycryptoki.cryptoki*), 149
- CK_KEA_DERIVE_PARAMS (class in *pycryptoki.cryptoki*), 111
- CK_KEA_DERIVE_PARAMS_PTR (in module *pycryptoki.cryptoki*), 106
- CK_KEY_DERIVATION_STRING_DATA (class in *pycryptoki.cryptoki*), 142
- CK_KEY_DERIVATION_STRING_DATA_PTR (in module *pycryptoki.cryptoki*), 124
- CK_KEY_TYPE (in module *pycryptoki.cryptoki*), 84
- CK_KEY_WRAP_SET_OAEP_PARAMS (class in *pycryptoki.cryptoki*), 79
- CK_KEY_WRAP_SET_OAEP_PARAMS_PTR (in module *pycryptoki.cryptoki*), 109
- CK_KIP_PARAMS (class in *pycryptoki.cryptoki*), 66
- CK_KIP_PARAMS_PTR (in module *pycryptoki.cryptoki*), 102
- CK_LKM_TOKEN_ID (in module *pycryptoki.cryptoki*), 73
- CK_LKM_TOKEN_ID_PTR (in module *pycryptoki.cryptoki*), 80
- CK_LKM_TOKEN_ID_S (class in *pycryptoki.cryptoki*), 108
- CK_LOCKMUTEX (in module *pycryptoki.cryptoki*), 115
- CK_LONG (in module *pycryptoki.cryptoki*), 112
- CK_MAC_GENERAL_PARAMS (in module *pycryptoki.cryptoki*), 71
- CK_MAC_GENERAL_PARAMS_PTR (in module *pycryptoki.cryptoki*), 141
- CK_MECHANISM (class in *pycryptoki.cryptoki*), 104
- CK_MECHANISM_INFO (class in *pycryptoki.cryptoki*), 65
- CK_MECHANISM_INFO_PTR (in module *pycryptoki.cryptoki*), 116
- CK_MECHANISM_PTR (in module *pycryptoki.cryptoki*), 142
- CK_MECHANISM_TYPE (in module *pycryptoki.cryptoki*), 104
- CK_MECHANISM_TYPE_PTR (in module *pycryptoki.cryptoki*), 75
- CK_NOTIFICATION (in module *pycryptoki.cryptoki*), 149
- CK_NOTIFY (in module *pycryptoki.cryptoki*), 125
- CK_OBJECT_CLASS (in module *pycryptoki.cryptoki*), 83
- CK_OBJECT_CLASS_PTR (in module *pycryptoki.cryptoki*), 128
- CK_OBJECT_HANDLE (in module *pycryptoki.cryptoki*), 71
- CK_OBJECT_HANDLE_PTR (in module *pycryptoki.cryptoki*), 112
- CK_OTP_PARAM (class in *pycryptoki.cryptoki*), 128
- CK_OTP_PARAM_PTR (in module *pycryptoki.cryptoki*), 67

CK_OTP_PARAM_TYPE (in module *pycryptoki.cryptoki*), 116
 CK_OTP_PARAMS (class in *pycryptoki.cryptoki*), 147
 CK_OTP_PARAMS_PTR (in module *pycryptoki.cryptoki*), 137
 CK_OTP_SIGNATURE_INFO (class in *pycryptoki.cryptoki*), 62
 CK_OTP_SIGNATURE_INFO_PTR (in module *pycryptoki.cryptoki*), 102
 CK_PARAM_TYPE (in module *pycryptoki.cryptoki*), 109
 CK_PBE_PARAMS (class in *pycryptoki.cryptoki*), 136
 CK_PBE_PARAMS_PTR (in module *pycryptoki.cryptoki*), 116
 CK_PKCS5_PBKD2_PARAMS (class in *pycryptoki.cryptoki*), 86
 CK_PKCS5_PBKD2_PARAMS_PTR (in module *pycryptoki.cryptoki*), 119
 CK_PKCS5_PBKD2_PSEUDO_RANDOM_FUNCTION_TYPE (in module *pycryptoki.cryptoki*), 80
 CK_PKCS5_PBKD2_PSEUDO_RANDOM_FUNCTION_TYPE_PTR (in module *pycryptoki.cryptoki*), 139
 CK_PKCS5_PBKDF2_SALT_SOURCE_TYPE (in module *pycryptoki.cryptoki*), 140
 CK_PKCS5_PBKDF2_SALT_SOURCE_TYPE_PTR (in module *pycryptoki.cryptoki*), 132
 CK_PRF_KDF_PARAMS (in module *pycryptoki.cryptoki*), 103
 CK_RC2_CBC_PARAMS (class in *pycryptoki.cryptoki*), 85
 CK_RC2_CBC_PARAMS_PTR (in module *pycryptoki.cryptoki*), 110
 CK_RC2_MAC_GENERAL_PARAMS (class in *pycryptoki.cryptoki*), 81
 CK_RC2_MAC_GENERAL_PARAMS_PTR (in module *pycryptoki.cryptoki*), 82
 CK_RC2_PARAMS (in module *pycryptoki.cryptoki*), 128
 CK_RC2_PARAMS_PTR (in module *pycryptoki.cryptoki*), 110
 CK_RC5_CBC_PARAMS (class in *pycryptoki.cryptoki*), 135
 CK_RC5_CBC_PARAMS_PTR (in module *pycryptoki.cryptoki*), 140
 CK_RC5_MAC_GENERAL_PARAMS (class in *pycryptoki.cryptoki*), 73
 CK_RC5_MAC_GENERAL_PARAMS_PTR (in module *pycryptoki.cryptoki*), 80
 CK_RC5_PARAMS (class in *pycryptoki.cryptoki*), 119
 CK_RC5_PARAMS_PTR (in module *pycryptoki.cryptoki*), 146
 CK_ResetTotalOperations (in module *pycryptoki.cryptoki*), 141
 CK_RSA_PKCS_MGF_TYPE (in module *pycryptoki.cryptoki*), 140
 CK_RSA_PKCS_MGF_TYPE_PTR (in module *pycryptoki.cryptoki*), 76
 CK_RSA_PKCS_OAEP_PARAMS (class in *pycryptoki.cryptoki*), 130
 CK_RSA_PKCS_OAEP_PARAMS_PTR (in module *pycryptoki.cryptoki*), 118
 CK_RSA_PKCS_OAEP_SOURCE_TYPE (in module *pycryptoki.cryptoki*), 135
 CK_RSA_PKCS_OAEP_SOURCE_TYPE_PTR (in module *pycryptoki.cryptoki*), 103
 CK_RSA_PKCS_PSS_PARAMS (class in *pycryptoki.cryptoki*), 84
 CK_RSA_PKCS_PSS_PARAMS_PTR (in module *pycryptoki.cryptoki*), 85
 CK_RV (in module *pycryptoki.cryptoki*), 125
 CK_SEED_CTR_PARAMS (in module *pycryptoki.cryptoki*), 73
 CK_SEED_CTR_PARAMS_PTR (in module *pycryptoki.cryptoki*), 147
 CK_SESSION_HANDLE (in module *pycryptoki.cryptoki*), 81
 CK_SESSION_HANDLE_PTR (in module *pycryptoki.cryptoki*), 82
 CK_SESSION_INFO (class in *pycryptoki.cryptoki*), 75
 CK_SESSION_INFO_PTR (in module *pycryptoki.cryptoki*), 79
 CK_SFNT_CA_FUNCTION_LIST (class in *pycryptoki.cryptoki*), 89
 CK_SFNT_CA_FUNCTION_LIST_PTR (in module *pycryptoki.cryptoki*), 82
 CK_SFNT_CA_FUNCTION_LIST_PTR_PTR (in module *pycryptoki.cryptoki*), 102
 CK_SKIPJACK_PRIVATE_WRAP_PARAMS (class in *pycryptoki.cryptoki*), 115
 CK_SKIPJACK_PRIVATE_WRAP_PTR (in module *pycryptoki.cryptoki*), 82
 CK_SKIPJACK_RELAYX_PARAMS (class in *pycryptoki.cryptoki*), 77
 CK_SKIPJACK_RELAYX_PARAMS_PTR (in module *pycryptoki.cryptoki*), 113
 CK_SLOT_ID (in module *pycryptoki.cryptoki*), 114
 CK_SLOT_ID_PTR (in module *pycryptoki.cryptoki*), 124
 CK_SLOT_INFO (class in *pycryptoki.cryptoki*), 81
 CK_SLOT_INFO_PTR (in module *pycryptoki.cryptoki*), 105
 CK_SSL3_KEY_MAT_OUT (class in *pycryptoki.cryptoki*), 105
 CK_SSL3_KEY_MAT_OUT_PTR (in module *pycryptoki.cryptoki*), 139
 CK_SSL3_KEY_MAT_PARAMS (class in *pycryptoki.cryptoki*), 101
 CK_SSL3_KEY_MAT_PARAMS_PTR (in module *pycryptoki.cryptoki*), 130
 CK_SSL3_MASTER_KEY_DERIVE_PARAMS (class in

- pycryptoki.cryptoki*), 138
 - CK_SSL3_MASTER_KEY_DERIVE_PARAMS_PTR (in module *pycryptoki.cryptoki*), 72
 - CK_SSL3_RANDOM_DATA (class in *pycryptoki.cryptoki*), 101
 - CK_STATE (in module *pycryptoki.cryptoki*), 88
 - CK_TLS_PRF_PARAMS (class in *pycryptoki.cryptoki*), 113
 - CK_TLS_PRF_PARAMS_PTR (in module *pycryptoki.cryptoki*), 70
 - CK_TOKEN_INFO (class in *pycryptoki.cryptoki*), 129
 - CK_TOKEN_INFO_PTR (in module *pycryptoki.cryptoki*), 141
 - CK_ULONG (in module *pycryptoki.cryptoki*), 72
 - CK_ULONG_PTR (in module *pycryptoki.cryptoki*), 123
 - CK_UNLOCKMUTEX (in module *pycryptoki.cryptoki*), 135
 - CK_USER_TYPE (in module *pycryptoki.cryptoki*), 136
 - CK_USHORT (in module *pycryptoki.cryptoki*), 102
 - CK_USHORT_PTR (in module *pycryptoki.cryptoki*), 119
 - CK_UTF8CHAR (in module *pycryptoki.cryptoki*), 148
 - CK_UTF8CHAR_PTR (in module *pycryptoki.cryptoki*), 138
 - CK_VERSION (class in *pycryptoki.cryptoki*), 64
 - CK_VERSION_PTR (in module *pycryptoki.cryptoki*), 122
 - CK_VOID_PTR (in module *pycryptoki.cryptoki*), 65
 - CK_VOID_PTR_PTR (in module *pycryptoki.cryptoki*), 107
 - CK_WTLS_KEY_MAT_OUT (class in *pycryptoki.cryptoki*), 75
 - CK_WTLS_KEY_MAT_OUT_PTR (in module *pycryptoki.cryptoki*), 86
 - CK_WTLS_KEY_MAT_PARAMS (class in *pycryptoki.cryptoki*), 76
 - CK_WTLS_KEY_MAT_PARAMS_PTR (in module *pycryptoki.cryptoki*), 119
 - CK_WTLS_MASTER_KEY_DERIVE_PARAMS (class in *pycryptoki.cryptoki*), 146
 - CK_WTLS_MASTER_KEY_DERIVE_PARAMS_PTR (in module *pycryptoki.cryptoki*), 137
 - CK_WTLS_PRF_PARAMS (class in *pycryptoki.cryptoki*), 64
 - CK_WTLS_PRF_PARAMS_PTR (in module *pycryptoki.cryptoki*), 110
 - CK_WTLS_RANDOM_DATA (class in *pycryptoki.cryptoki*), 102
 - CK_WTLS_RANDOM_DATA_PTR (in module *pycryptoki.cryptoki*), 85
 - CK_X9_42_DH1_DERIVE_PARAMS (class in *pycryptoki.cryptoki*), 126
 - CK_X9_42_DH1_DERIVE_PARAMS_PTR (in module *pycryptoki.cryptoki*), 103
 - CK_X9_42_DH2_DERIVE_PARAMS (class in *pycryptoki.cryptoki*), 87
 - CK_X9_42_DH2_DERIVE_PARAMS_PTR (in module *pycryptoki.cryptoki*), 108
 - CK_X9_42_DH_KDF_TYPE (in module *pycryptoki.cryptoki*), 66
 - CK_X9_42_DH_KDF_TYPE_PTR (in module *pycryptoki.cryptoki*), 125
 - CK_X9_42_MQV_DERIVE_PARAMS (class in *pycryptoki.cryptoki*), 67
 - CK_X9_42_MQV_DERIVE_PARAMS_PTR (in module *pycryptoki.cryptoki*), 125
 - CK_XOR_BASE_DATA_KDF_PARAMS (class in *pycryptoki.cryptoki*), 133
 - CK_XOR_BASE_DATA_KDF_PARAMS_PTR (in module *pycryptoki.cryptoki*), 75
 - CKA_SIM_AUTH_FORM (in module *pycryptoki.cryptoki*), 121
 - CKCA_MODULE_ID (in module *pycryptoki.cryptoki*), 115
 - CKCA_MODULE_ID_PTR (in module *pycryptoki.cryptoki*), 112
 - CKCA_MODULE_INFO (class in *pycryptoki.cryptoki*), 135
 - CKCA_MODULE_INFO_PTR (in module *pycryptoki.cryptoki*), 106
 - CKM_DH_PKCS_PARAMETER_GEN_TEMP (in module *pycryptoki.default_templates*), 61
 - CKM_SSL3_PRE_MASTER_KEY_GEN_TEMP (in module *pycryptoki.default_templates*), 62
 - clear_keys() (in module *pycryptoki.key_generator*), 23
 - ConcatenationDeriveMechanism (class in *pycryptoki.mechanism.generic*), 39
 - convert_c_ubyte_array_to_string() (in module *pycryptoki.attributes*), 32
 - CreateMutex (in module *pycryptoki.cryptoki.CK_C_INITIALIZE_ARGS* attribute), 131
 - CryptokiConfigException, 61
 - CryptokiDllException, 61
 - CryptokiDllSingleton (class in *pycryptoki.cryptoki_helpers*), 61
 - cryptokiVersion (in module *pycryptoki.cryptoki.CK_INFO* attribute), 66
 - CT_Token (class in *pycryptoki.cryptoki*), 114
 - CT_TokenHandle (in module *pycryptoki.cryptoki*), 107
 - ctxID (in module *pycryptoki.cryptoki.CK_AES_CBC_PAD_EXTRACT_PARAMS* attribute), 113
 - ctxID (in module *pycryptoki.cryptoki.CK_AES_CBC_PAD_INSERT_PARAMS* attribute), 142
- ## D
- day (in module *pycryptoki.cryptoki.CK_DATE* attribute), 63

DestroyMutex (pycryptoki.cryptoki.CK_C_INITIALIZE_ARGS attribute), 131

developerName (pycryptoki.cryptoki.CKCA_MODULE_INFO attribute), 135

dhPrimitive (pycryptoki.cryptoki.CK_ECIES_PARAMS attribute), 107

DigestMechanism (pycryptoki.cryptoki.CK_WTLS_KEY_MAT_PARAMS attribute), 76

DigestMechanism (pycryptoki.cryptoki.CK_WTLS_MASTER_KEY_DERIVE_PARAMS attribute), 146

DigestMechanism (pycryptoki.cryptoki.CK_WTLS_PRF_PARAMS attribute), 64

do_multipart_operation() (in module pycryptoki.encryption), 29

do_multipart_verify() (in module pycryptoki.sign_verify), 31

E

eInitMsgs (in module pycryptoki.cryptoki), 74

encScheme (pycryptoki.cryptoki.CK_ECIES_PARAMS attribute), 108

F

firmwareVersion (pycryptoki.cryptoki.CK_SLOT_INFO attribute), 81

firmwareVersion (pycryptoki.cryptoki.CK_TOKEN_INFO attribute), 129

flags (pycryptoki.cryptoki.CA_ROLE_STATE attribute), 121

flags (pycryptoki.cryptoki.CK_C_INITIALIZE_ARGS attribute), 131

flags (pycryptoki.cryptoki.CK_INFO attribute), 66

flags (pycryptoki.cryptoki.CK_MECHANISM_INFO attribute), 65

flags (pycryptoki.cryptoki.CK_SESSION_INFO attribute), 75

flags (pycryptoki.cryptoki.CK_SLOT_INFO attribute), 81

flags (pycryptoki.cryptoki.CK_TOKEN_INFO attribute), 129

Float (in module pycryptoki.cryptoki), 72

Float32 (in module pycryptoki.cryptoki), 151

Float64 (in module pycryptoki.cryptoki), 66

from_bin() (in module pycryptoki.conversions), 34

from_bytestring() (in module pycryptoki.conversions), 35

from_c_struct() (pycryptoki.attributes.Attributes static method), 32

from_hex() (in module pycryptoki.conversions), 35

from_path() (pycryptoki.cryptoki_helpers.CryptokiDLLSingleton class method), 61

fwResultCode (in module pycryptoki.cryptoki), 103

G

get_c_struct() (pycryptoki.attributes.Attributes method), 32

get_c_struct_from_mechanism() (in module pycryptoki.mechanism.helpers), 36

get_default_key_pair_template() (in module pycryptoki.default_templates), 62

get_default_key_template() (in module pycryptoki.default_templates), 62

get_dll() (pycryptoki.cryptoki_helpers.CryptokiDLLSingleton method), 61

get_firmware_version() (in module pycryptoki.session_management), 16

get_python_dict_from_c_mechanism() (in module pycryptoki.mechanism.helpers), 37

get_slot_dict() (in module pycryptoki.session_management), 16

get_slot_dict_ex() (in module pycryptoki.session_management), 16

get_token_by_label() (in module pycryptoki.token_management), 19

get_token_by_label_ex() (in module pycryptoki.token_management), 19

groupSerial (pycryptoki.cryptoki.CK_HA_STATUS attribute), 118

H

HalfWord (in module pycryptoki.cryptoki), 106

HANDLE (in module pycryptoki.cryptoki), 117

hardwareVersion (pycryptoki.cryptoki.CK_SLOT_INFO attribute), 81

hardwareVersion (pycryptoki.cryptoki.CK_TOKEN_INFO attribute), 129

hashAlg (pycryptoki.cryptoki.CK_RSA_PKCS_OAEP_PARAMS attribute), 130

hashAlg (pycryptoki.cryptoki.CK_RSA_PKCS_PSS_PARAMS attribute), 84

hClientKey (pycryptoki.cryptoki.CK_SSL3_KEY_MAT_OUT attribute), 105

hClientMacSecret (pycryptoki.cryptoki.CK_SSL3_KEY_MAT_OUT attribute), 105

- `hKey` (`pycryptoki.cryptoki.CK_KIP_PARAMS` attribute), 66
- `hKey` (`pycryptoki.cryptoki.CK_WTLS_KEY_MAT_OUT` attribute), 75
- `hMacSecret` (`pycryptoki.cryptoki.CK_WTLS_KEY_MAT_OUT` attribute), 75
- `hPrivateKey` (`pycryptoki.cryptoki.CK_ECDH2_DERIVE_PARAMS` attribute), 127
- `hPrivateKey` (`pycryptoki.cryptoki.CK_ECMQV_DERIVE_PARAMS` attribute), 69
- `hPrivateKey` (`pycryptoki.cryptoki.CK_X9_42_DH2_DERIVE_PARAMS` attribute), 87
- `hPrivateKey` (`pycryptoki.cryptoki.CK_X9_42_MQV_DERIVE_PARAMS` attribute), 67
- `hServerKey` (`pycryptoki.cryptoki.CK_SSL3_KEY_MAT_OUT` attribute), 105
- `hServerMacSecret` (`pycryptoki.cryptoki.CK_SSL3_KEY_MAT_OUT` attribute), 105
- `HSM_STATS_PARAMS` (class in `pycryptoki.cryptoki`), 128
- `hTweakKey` (`pycryptoki.cryptoki.CK_AES_XTS_PARAMS` attribute), 132
- I**
- `id` (`pycryptoki.cryptoki.CK_LKM_TOKEN_ID_S` attribute), 108
- `Int` (in module `pycryptoki.cryptoki`), 112
- `Int16` (in module `pycryptoki.cryptoki`), 139
- `Int32` (in module `pycryptoki.cryptoki`), 62
- `Int64` (in module `pycryptoki.cryptoki`), 87
- `Int8` (in module `pycryptoki.cryptoki`), 150
- `isSender` (`pycryptoki.cryptoki.CK_KEA_DERIVE_PARAMS` attribute), 111
- `iterations` (`pycryptoki.cryptoki.CK_PKCS5_PBKD2_PARAMS` attribute), 86
- `iv` (`pycryptoki.cryptoki.CK_AES_CBC_ENCRYPT_DATA_PARAMS` attribute), 123
- `iv` (`pycryptoki.cryptoki.CK_ARIA_CBC_ENCRYPT_DATA_PARAMS` attribute), 117
- `iv` (`pycryptoki.cryptoki.CK_CAMELLIA_CBC_ENCRYPT_DATA_PARAMS` attribute), 120
- `iv` (`pycryptoki.cryptoki.CK_DES_CBC_ENCRYPT_DATA_PARAMS` attribute), 120
- `iv` (`pycryptoki.cryptoki.CK_RC2_CBC_PARAMS` attribute), 85
- `Iv16Mechanism` (class in `pycryptoki.mechanism.aes`), 39
- `IvMechanism` (class in `pycryptoki.mechanism.aes`), 39
- K**
- `kdf` (`pycryptoki.cryptoki.CK_ECDH1_DERIVE_PARAMS` attribute), 109
- `kdf` (`pycryptoki.cryptoki.CK_ECDH2_DERIVE_PARAMS` attribute), 127
- `kdf` (`pycryptoki.cryptoki.CK_ECIES_PARAMS` attribute), 108
- `kdf` (`pycryptoki.cryptoki.CK_ECMQV_DERIVE_PARAMS` attribute), 69
- `kdf` (`pycryptoki.cryptoki.CK_X9_42_DH1_DERIVE_PARAMS` attribute), 126
- `kdf` (`pycryptoki.cryptoki.CK_X9_42_DH2_DERIVE_PARAMS` attribute), 87
- `kdf` (`pycryptoki.cryptoki.CK_X9_42_MQV_DERIVE_PARAMS` attribute), 67
- `kdf` (`pycryptoki.cryptoki.CK_XOR_BASE_DATA_KDF_PARAMS` attribute), 133
- `KEY_PAIR_GENERATOR_TEMPLATES` (in module `pycryptoki.default_templates`), 62
- L**
- `label` (`pycryptoki.cryptoki.CK_TOKEN_INFO` attribute), 129
- `length` (`pycryptoki.cryptoki.CK_AES_CBC_ENCRYPT_DATA_PARAMS` attribute), 123
- `length` (`pycryptoki.cryptoki.CK_ARIA_CBC_ENCRYPT_DATA_PARAMS` attribute), 117
- `length` (`pycryptoki.cryptoki.CK_CAMELLIA_CBC_ENCRYPT_DATA_PARAMS` attribute), 121
- `length` (`pycryptoki.cryptoki.CK_DES_CBC_ENCRYPT_DATA_PARAMS` attribute), 120
- `libraryDescription` (`pycryptoki.cryptoki.CK_INFO` attribute), 66
- `libraryVersion` (`pycryptoki.cryptoki.CK_INFO` attribute), 66
- `listSize` (`pycryptoki.cryptoki.CK_HA_STATUS` attribute), 118
- `loaded_dll_library` (`pycryptoki.cryptoki_helpers.CryptokiDLLSingleton` attribute), 61
- `LockMutex` (`pycryptoki.cryptoki.CK_C_INITIALIZE_ARGS` attribute), 131
- `log_args()` (in module `pycryptoki.cryptoki_helpers`), 131
- `login()` (in module `pycryptoki.session_management`), 17
- `login_ex()` (in module `pycryptoki.session_management`), 17
- `loginAttemptsLeft` (`pycryptoki.cryptoki.CA_ROLE_STATE` attribute), 17

121

M

macScheme (pycryptoki.cryptoki.CK_ECIES_PARAMS attribute), 108

major (pycryptoki.cryptoki.CK_VERSION attribute), 64

make_late_binding_function() (in module pycryptoki.cryptoki_helpers), 61

manufacturerID (pycryptoki.cryptoki.CK_INFO attribute), 66

manufacturerID (pycryptoki.cryptoki.CK_SLOT_INFO attribute), 81

manufacturerID (pycryptoki.cryptoki.CK_TOKEN_INFO attribute), 129

Mechanism (class in pycryptoki.mechanism.helpers), 36

mechanism (pycryptoki.cryptoki.CK_MECHANISM attribute), 104

MechanismException, 36

memberList (pycryptoki.cryptoki.CK_HA_STATUS attribute), 118

memberSerial (pycryptoki.cryptoki.CK_HA_MEMBER attribute), 134

memberStatus (pycryptoki.cryptoki.CK_HA_MEMBER attribute), 134

mgf (pycryptoki.cryptoki.CK_RSA_PKCS_OAEP_PARAMS attribute), 130

mgf (pycryptoki.cryptoki.CK_RSA_PKCS_PSS_PARAMS attribute), 84

minor (pycryptoki.cryptoki.CK_VERSION attribute), 65

model (pycryptoki.cryptoki.CK_TOKEN_INFO attribute), 129

moduleDescription (pycryptoki.cryptoki.CKCA_MODULE_INFO attribute), 135

moduleVersion (pycryptoki.cryptoki.CKCA_MODULE_INFO attribute), 135

month (pycryptoki.cryptoki.CK_DATE attribute), 63

N

NullMech (class in pycryptoki.mechanism.generic), 40

P

pAAD (pycryptoki.cryptoki.CK_AES_GCM_PARAMS attribute), 132

parse_chrystoki_conf() (in module pycryptoki.cryptoki_helpers), 61

parse_mechanism() (in module pycryptoki.mechanism.helpers), 37

pBaseG (pycryptoki.cryptoki.CK_SKIPJACK_PRIVATE_WRAP_PARAMS attribute), 115

pbFileName (pycryptoki.cryptoki.CK_AES_CBC_PAD_EXTRACT_PARAMS attribute), 113

pbFileName (pycryptoki.cryptoki.CK_AES_CBC_PAD_INSERT_PARAMS attribute), 142

pBuffer (pycryptoki.cryptoki.CK_AES_CBC_PAD_EXTRACT_PARAMS attribute), 113

pBuffer (pycryptoki.cryptoki.CK_AES_CBC_PAD_INSERT_PARAMS attribute), 142

pClientRandom (pycryptoki.cryptoki.CK_SSL3_RANDOM_DATA attribute), 101

pClientRandom (pycryptoki.cryptoki.CK_WTLS_RANDOM_DATA attribute), 102

pContentType (pycryptoki.cryptoki.CK_CMS_SIG_PARAMS attribute), 68

pContext (pycryptoki.cryptoki.CK_KDF_PRF_PARAMS attribute), 72

pData (pycryptoki.cryptoki.CK_AES_CBC_ENCRYPT_DATA_PARAMS attribute), 123

pData (pycryptoki.cryptoki.CK_ARIA_CBC_ENCRYPT_DATA_PARAMS attribute), 117

pData (pycryptoki.cryptoki.CK_CAMELLIA_CBC_ENCRYPT_DATA_PARAMS attribute), 121

pData (pycryptoki.cryptoki.CK_DES_CBC_ENCRYPT_DATA_PARAMS attribute), 120

pData (pycryptoki.cryptoki.CK_KEY_DERIVATION_STRING_DATA attribute), 142

pDigestMechanism (pycryptoki.cryptoki.CK_CMS_SIG_PARAMS attribute), 68

pedId (pycryptoki.cryptoki.CK_AES_CBC_PAD_EXTRACT_PARAMS attribute), 113

pedId (pycryptoki.cryptoki.CK_AES_CBC_PAD_INSERT_PARAMS attribute), 142

pInitVector (pycryptoki.cryptoki.CK_PBE_PARAMS attribute), 136

pIv (pycryptoki.cryptoki.CK_AES_GCM_PARAMS attribute), 133

pIv (pycryptoki.cryptoki.CK_RC5_CBC_PARAMS attribute), 135

pIV (pycryptoki.cryptoki.CK_WTLS_KEY_MAT_OUT attribute), 75

pIVClient (pycryptoki.cryptoki.CK_SSL3_KEY_MAT_OUT attribute), 105

pIVServer (pycryptoki.cryptoki.CK_SSL3_KEY_MAT_OUT attribute), 105

pLabel (pycryptoki.cryptoki.CK_KDF_PRF_PARAMS attribute), 72

pLabel (pycryptoki.cryptoki.CK_TLS_PRF_PARAMS attribute), 113	pPrfData (pycryptoki.cryptoki.CK_PKCS5_PBKD2_PARAMS attribute), 86
pLabel (pycryptoki.cryptoki.CK_WTLS_PRF_PARAMS attribute), 64	pPrimeP (pycryptoki.cryptoki.CK_SKIPJACK_PRIVATE_WRAP_PARAMS attribute), 115
pMechanism (pycryptoki.cryptoki.CK_KIP_PARAMS attribute), 66	pPublicData (pycryptoki.cryptoki.CK_ECDH1_DERIVE_PARAMS attribute), 109
pNewPassword (pycryptoki.cryptoki.CK_SKIPJACK_RELAYX_PARAMS attribute), 77	pPublicData (pycryptoki.cryptoki.CK_ECDH2_DERIVE_PARAMS attribute), 127
pNewPublicData (pycryptoki.cryptoki.CK_SKIPJACK_RELAYX_PARAMS attribute), 77	pPublicData (pycryptoki.cryptoki.CK_ECMQV_DERIVE_PARAMS attribute), 69
pNewRandomA (pycryptoki.cryptoki.CK_SKIPJACK_RELAYX_PARAMS attribute), 77	pPublicData (pycryptoki.cryptoki.CK_KEA_DERIVE_PARAMS attribute), 111
PointerDifference (in module pycryptoki.cryptoki), 81	pPublicData (pycryptoki.cryptoki.CK_SKIPJACK_PRIVATE_WRAP_PARAMS attribute), 115
pOldPassword (pycryptoki.cryptoki.CK_SKIPJACK_RELAYX_PARAMS attribute), 77	pPublicData (pycryptoki.cryptoki.CK_X9_42_DH1_DERIVE_PARAMS attribute), 126
pOldPublicData (pycryptoki.cryptoki.CK_SKIPJACK_RELAYX_PARAMS attribute), 77	pPublicData (pycryptoki.cryptoki.CK_X9_42_DH2_DERIVE_PARAMS attribute), 87
pOldRandomA (pycryptoki.cryptoki.CK_SKIPJACK_RELAYX_PARAMS attribute), 77	pPublicData (pycryptoki.cryptoki.CK_X9_42_MQV_DERIVE_PARAMS attribute), 67
pOldWrappedX (pycryptoki.cryptoki.CK_SKIPJACK_RELAYX_PARAMS attribute), 77	pPublicData2 (pycryptoki.cryptoki.CK_ECDH2_DERIVE_PARAMS attribute), 127
pOtherInfo (pycryptoki.cryptoki.CK_X9_42_DH1_DERIVE_PARAMS attribute), 126	pPublicData2 (pycryptoki.cryptoki.CK_ECMQV_DERIVE_PARAMS attribute), 69
pOtherInfo (pycryptoki.cryptoki.CK_X9_42_DH2_DERIVE_PARAMS attribute), 87	pPublicData2 (pycryptoki.cryptoki.CK_X9_42_DH2_DERIVE_PARAMS attribute), 87
pOtherInfo (pycryptoki.cryptoki.CK_X9_42_MQV_DERIVE_PARAMS attribute), 67	pPublicData2 (pycryptoki.cryptoki.CK_X9_42_MQV_DERIVE_PARAMS attribute), 67
pOutput (pycryptoki.cryptoki.CK_TLS_PRF_PARAMS attribute), 113	pRandomA (pycryptoki.cryptoki.CK_KEA_DERIVE_PARAMS attribute), 111
pOutput (pycryptoki.cryptoki.CK_WTLS_PRF_PARAMS attribute), 64	pRandomA (pycryptoki.cryptoki.CK_SKIPJACK_PRIVATE_WRAP_PARAMS attribute), 115
pParameter (pycryptoki.cryptoki.CK_MECHANISM attribute), 104	pRandomB (pycryptoki.cryptoki.CK_KEA_DERIVE_PARAMS attribute), 111
pParams (pycryptoki.cryptoki.CK_OTP_PARAMS attribute), 147	RequestedAttributes (pycryptoki.cryptoki.CK_CMS_SIG_PARAMS attribute), 68
pParams (pycryptoki.cryptoki.CK_OTP_SIGNATURE_INFO attribute), 62	pRequiredAttributes (pycryptoki.cryptoki.CK_CMS_SIG_PARAMS attribute), 68
pPassword (pycryptoki.cryptoki.CK_PBE_PARAMS attribute), 136	RequestedAttributes (pycryptoki.cryptoki.CK_C_INITIALIZE_ARGS attribute), 131
pPassword (pycryptoki.cryptoki.CK_PKCS5_PBKD2_PARAMS attribute), 86	
pPassword (pycryptoki.cryptoki.CK_SKIPJACK_PRIVATE_WRAP_PARAMS attribute), 115	

<code>pReturnedKeyMaterial</code> (<code>pycryptoki.cryptoki.CK_SSL3_KEY_MAT_PARAMS</code> attribute), 102	<code>pSubprimeQ</code> (<code>pycryptoki.cryptoki.CK_SKIPJACK_PRIVATE_WRAP_PARAMS</code> attribute), 115
<code>pReturnedKeyMaterial</code> (<code>pycryptoki.cryptoki.CK_WTLS_KEY_MAT_PARAMS</code> attribute), 76	<code>publicKey</code> (<code>pycryptoki.cryptoki.CK_ECMQV_DERIVE_PARAMS</code> attribute), 69
<code>prf</code> (<code>pycryptoki.cryptoki.CK_PKCS5_PBKD2_PARAMS</code> attribute), 87	<code>publicKey</code> (<code>pycryptoki.cryptoki.CK_X9_42_MQV_DERIVE_PARAMS</code> attribute), 67
<code>prfType</code> (<code>pycryptoki.cryptoki.CK_KDF_PRF_PARAMS</code> attribute), 72	<code>pulBufferLen</code> (<code>pycryptoki.cryptoki.CK_AES_CBC_PAD_EXTRACT_PARAMS</code> attribute), 113
<code>primaryAuthMech</code> (<code>pycryptoki.cryptoki.CA_ROLE_STATE</code> attribute), 121	<code>pulHandle</code> (<code>pycryptoki.cryptoki.CK_AES_CBC_PAD_INSERT_PARAMS</code> attribute), 142
<code>pSalt</code> (<code>pycryptoki.cryptoki.CK_PBE_PARAMS</code> attribute), 136	<code>pulOutputLen</code> (<code>pycryptoki.cryptoki.CK_TLS_PRF_PARAMS</code> attribute), 113
<code>pSaltSourceData</code> (<code>pycryptoki.cryptoki.CK_PKCS5_PBKD2_PARAMS</code> attribute), 87	<code>pulOutputLen</code> (<code>pycryptoki.cryptoki.CK_WTLS_PRF_PARAMS</code> attribute), 64
<code>pSeed</code> (<code>pycryptoki.cryptoki.CK_KIP_PARAMS</code> attribute), 67	<code>pulType</code> (<code>pycryptoki.cryptoki.CK_AES_CBC_PAD_INSERT_PARAMS</code> attribute), 142
<code>pSeed</code> (<code>pycryptoki.cryptoki.CK_TLS_PRF_PARAMS</code> attribute), 113	<code>pValue</code> (<code>pycryptoki.cryptoki.CK_ATTRIBUTE</code> attribute), 104
<code>pSeed</code> (<code>pycryptoki.cryptoki.CK_WTLS_PRF_PARAMS</code> attribute), 64	<code>pValue</code> (<code>pycryptoki.cryptoki.CK_OTP_PARAM</code> attribute), 128
<code>pServerRandom</code> (<code>pycryptoki.cryptoki.CK_SSL3_RANDOM_DATA</code> attribute), 101	<code>pVector</code> (<code>pycryptoki.cryptoki.CA_MOFN_ACTIVATION</code> attribute), 70
<code>pServerRandom</code> (<code>pycryptoki.cryptoki.CK_WTLS_RANDOM_DATA</code> attribute), 102	<code>pVector</code> (<code>pycryptoki.cryptoki.CA_MOFN_GENERATION</code> attribute), 112
<code>pSharedData</code> (<code>pycryptoki.cryptoki.CK_ECDH1_DERIVE_PARAMS</code> attribute), 109	<code>pVersion</code> (<code>pycryptoki.cryptoki.CK_SSL3_MASTER_KEY_DERIVE_PARAMS</code> attribute), 138
<code>pSharedData</code> (<code>pycryptoki.cryptoki.CK_ECDH2_DERIVE_PARAMS</code> attribute), 127	<code>pVersion</code> (<code>pycryptoki.cryptoki.CK_WTLS_MASTER_KEY_DERIVE_PARAMS</code> attribute), 146
<code>pSharedData</code> (<code>pycryptoki.cryptoki.CK_ECMQV_DERIVE_PARAMS</code> attribute), 69	<code>pX</code> (<code>pycryptoki.cryptoki.CK_KEY_WRAP_SET_OAEP_PARAMS</code> attribute), 79
<code>pSharedData</code> (<code>pycryptoki.cryptoki.CK_XOR_BASE_DATA_KDF_PARAMS</code> attribute), 133	<code>pycryptoki.attributes</code> (module), 32
<code>pSharedData1</code> (<code>pycryptoki.cryptoki.CK_ECIES_PARAMS</code> attribute), 108	<code>pycryptoki.attributes.KEY_TRANSFORMS</code> (in module <code>pycryptoki.attributes</code>), 32
<code>pSharedData2</code> (<code>pycryptoki.cryptoki.CK_ECIES_PARAMS</code> attribute), 108	<code>pycryptoki.audit_handling</code> (module), 55
<code>pSigningMechanism</code> (<code>pycryptoki.cryptoki.CK_CMS_SIG_PARAMS</code> attribute), 68	<code>pycryptoki.backup</code> (module), 57
<code>pSourceData</code> (<code>pycryptoki.cryptoki.CK_RSA_PKCS_OAEP_PARAMS</code> attribute), 130	<code>pycryptoki.conversions</code> (module), 34
	<code>pycryptoki.cryptoki</code> (module), 62
	<code>pycryptoki.cryptoki_helpers</code> (module), 61
	<code>pycryptoki.default_templates</code> (module), 61
	<code>pycryptoki.defaults</code> (module), 62
	<code>pycryptoki.hsm_management</code> (module), 47
	<code>pycryptoki.key_generator</code> (module), 20
	<code>pycryptoki.key_management</code> (module), 23
	<code>pycryptoki.key_usage</code> (module), 24
	<code>pycryptoki.lookup_dicts</code> (module), 61
	<code>pycryptoki.mechanism</code> (module), 35
	<code>pycryptoki.mechanism.aes</code> (module), 38
	<code>pycryptoki.mechanism.generic</code> (module), 39
	<code>pycryptoki.mechanism.helpers</code> (module), 36
	<code>pycryptoki.mechanism.rc</code> (module), 40

pycryptoki.mechanism.rsa (module), 41
 pycryptoki.misc (module), 41
 pycryptoki.object_attr_lookup (module), 45
 pycryptoki.session_management (module), 8
 pycryptoki.token_management (module), 17

R

RandomInfo (pycryptoki.cryptoki.CK_SSL3_KEY_MAT_PARAMS attribute), 101
 RandomInfo (pycryptoki.cryptoki.CK_SSL3_MASTER_KEY_DERIVE_PARAMS attribute), 138
 RandomInfo (pycryptoki.cryptoki.CK_WTLS_KEY_MAT_PARAMS attribute), 76
 RandomInfo (pycryptoki.cryptoki.CK_WTLS_MASTER_KEY_DERIVE_PARAMS attribute), 146
 RC2CBCMechanism (class in pycryptoki.mechanism.rc), 40
 RC2Mechanism (class in pycryptoki.mechanism.rc), 40
 RC5CBCMechanism (class in pycryptoki.mechanism.rc), 41
 RC5Mechanism (class in pycryptoki.mechanism.rc), 41
 REQUIRED_PARAMS (pycryptoki.mechanism.aes.AESCBCEncryptDataMechanism attribute), 38
 REQUIRED_PARAMS (pycryptoki.mechanism.aes.AESCTRMechanism attribute), 38
 REQUIRED_PARAMS (pycryptoki.mechanism.aes.AESECBEncryptDataMechanism attribute), 38
 REQUIRED_PARAMS (pycryptoki.mechanism.aes.AESGCMMechanism attribute), 38
 REQUIRED_PARAMS (pycryptoki.mechanism.aes.AESXTSMechanism attribute), 39
 REQUIRED_PARAMS (pycryptoki.mechanism.generic.ConcatenationDeriveMechanism attribute), 40
 REQUIRED_PARAMS (pycryptoki.mechanism.generic.StringDataDerivationMechanism attribute), 40
 REQUIRED_PARAMS (pycryptoki.mechanism.helpers.Mechanism attribute), 36
 REQUIRED_PARAMS (pycryptoki.mechanism.rc.RC2CBCMechanism attribute), 40
 REQUIRED_PARAMS (pycryptoki.mechanism.rc.RC2Mechanism attribute),

40
 REQUIRED_PARAMS (pycryptoki.mechanism.rc.RC5CBCMechanism attribute), 41
 REQUIRED_PARAMS (pycryptoki.mechanism.rc.RC5Mechanism attribute), 41
 REQUIRED_PARAMS (pycryptoki.mechanism.rsa.RSAPKCSOAEPMechanism attribute), 41
 REQUIRED_PARAMS (pycryptoki.mechanism.rsa.RSAPKCSPSSMechanism attribute), 41
 ResultCodeValue (in module pycryptoki.cryptoki), 109
 ret_type () (in module pycryptoki.attributes), 32
 ret_vals_dictionary (in module pycryptoki.lookup_dicts), 61
 RSAPKCSOAEPMechanism (class in pycryptoki.mechanism.rsa), 41
 RSAPKCSPSSMechanism (class in pycryptoki.mechanism.rsa), 41
 S
 saltSource (pycryptoki.cryptoki.CK_PKCS5_PBKD2_PARAMS attribute), 87
 secondaryAuthMech (pycryptoki.cryptoki.CA_ROLE_STATE attribute), 121
 serialNumber (pycryptoki.cryptoki.CK_TOKEN_INFO attribute), 129
 SInt (in module pycryptoki.cryptoki), 140
 SInt16 (in module pycryptoki.cryptoki), 138
 SInt32 (in module pycryptoki.cryptoki), 122
 SInt64 (in module pycryptoki.cryptoki), 140
 SInt8 (in module pycryptoki.cryptoki), 73
 SizeType (in module pycryptoki.cryptoki), 131
 slotDescription (pycryptoki.cryptoki.CK_SLOT_INFO attribute), 81
 slotID (pycryptoki.cryptoki.CK_SESSION_INFO attribute), 75
 state (pycryptoki.cryptoki.CK_RSA_PKCS_OAEP_PARAMS attribute), 130
 state (pycryptoki.cryptoki.CK_SESSION_INFO attribute), 75
 StringDataDerivationMechanism (class in pycryptoki.mechanism.generic), 40
 swapper (class in pycryptoki.cryptoki), 138

T

to_bin () (in module pycryptoki.conversions), 35

[to_bool\(\)](#) (in module `pycryptoki.attributes`), 33
[to_byte_array\(\)](#) (in module `pycryptoki.attributes`), 33
[to_bytestring\(\)](#) (in module `pycryptoki.conversions`), 35
[to_c_mech\(\)](#) (pycryptoki.mechanism.aes.AESCBCEncryptDataMechanism method), 38
[to_c_mech\(\)](#) (pycryptoki.mechanism.aes.AESCTRMechanism method), 38
[to_c_mech\(\)](#) (pycryptoki.mechanism.aes.AESECBEncryptDataMechanism method), 38
[to_c_mech\(\)](#) (pycryptoki.mechanism.aes.AESGCMMechanism method), 38
[to_c_mech\(\)](#) (pycryptoki.mechanism.aes.AESXTSMechanism method), 39
[to_c_mech\(\)](#) (pycryptoki.mechanism.aes.Iv16Mechanism method), 39
[to_c_mech\(\)](#) (pycryptoki.mechanism.aes.IvMechanism method), 39
[to_c_mech\(\)](#) (pycryptoki.mechanism.generic.AutoMechanism method), 39
[to_c_mech\(\)](#) (pycryptoki.mechanism.generic.ConcatenationDeriveMechanism method), 40
[to_c_mech\(\)](#) (pycryptoki.mechanism.generic.NullMechanism method), 40
[to_c_mech\(\)](#) (pycryptoki.mechanism.generic.StringDataDerivationMechanism method), 40
[to_c_mech\(\)](#) (pycryptoki.mechanism.helpers.Mechanism method), 36
[to_c_mech\(\)](#) (pycryptoki.mechanism.rc.RC2CBCEncryptDataMechanism method), 40
[to_c_mech\(\)](#) (pycryptoki.mechanism.rc.RC2Mechanism method), 40
[to_c_mech\(\)](#) (pycryptoki.mechanism.rc.RC5CBCEncryptDataMechanism method), 41
[to_c_mech\(\)](#) (pycryptoki.mechanism.rc.RC5Mechanism method), 41
[to_c_mech\(\)](#) (pycryptoki.mechanism.rsa.RSAPKCSOAEPMechanism method), 41
[to_c_mech\(\)](#) (pycryptoki.mechanism.rsa.RSAPKCSPPSSMechanism method), 41
[to_char_array\(\)](#) (in module `pycryptoki.attributes`), 33
[to_ck_date\(\)](#) (in module `pycryptoki.attributes`), 33
[to_hex\(\)](#) (in module `pycryptoki.conversions`), 35
[to_long\(\)](#) (in module `pycryptoki.attributes`), 33
[to_pka_key_status\(\)](#) (in module `pycryptoki.attributes`), 34
[to_sub_attributes\(\)](#) (in module `pycryptoki.attributes`), 34
[type](#) (pycryptoki.cryptoki.CK_ATTRIBUTE attribute), 104
[type](#) (pycryptoki.cryptoki.CK_OTP_PARAM attribute), 128

U

[UInt](#) (in module `pycryptoki.cryptoki`), 103
[UInt16](#) (in module `pycryptoki.cryptoki`), 76
[UInt32](#) (in module `pycryptoki.cryptoki`), 85
[UInt64](#) (in module `pycryptoki.cryptoki`), 112
[UInt8](#) (in module `pycryptoki.cryptoki`), 140
[ulAADLen](#) (pycryptoki.cryptoki.CK_AES_GCM_PARAMS attribute), 133
[ulBufferLen](#) (pycryptoki.cryptoki.CK_AES_CBC_PAD_INSERT_PARAMS attribute), 142
[ulCryptokiRandomLen](#) (pycryptoki.cryptoki.CK_SSL3_RANDOM_DATA attribute), 101
[ulClientRandomLen](#) (pycryptoki.cryptoki.CK_WTLS_RANDOM_DATA attribute), 102
[ulCryptokiState](#) (pycryptoki.cryptoki.CK_AES_CBC_PAD_INSERT_PARAMS attribute), 142
[ulContextLen](#) (pycryptoki.cryptoki.CK_KDF_PRF_PARAMS attribute), 72
[ulCount](#) (pycryptoki.cryptoki.CK_OTP_PARAMS attribute), 147
[ulCount](#) (pycryptoki.cryptoki.CK_OTP_SIGNATURE_INFO attribute), 62
[ulCounter](#) (pycryptoki.cryptoki.CK_KDF_PRF_PARAMS attribute), 72
[ulCounterBits](#) (pycryptoki.cryptoki.CK_AES_CTR_PARAMS attribute), 108
[ulCounterBits](#) (pycryptoki.cryptoki.CK_CAMELLIA_CTR_PARAMS attribute), 86

<code>ulCounterBits</code>	(<code>pycryptoki.CK_DES_CTR_PARAMS</code> attribute), 70	<code>toki.cryptoki.CK_WTLS_PRF_PARAMS</code> attribute), 64
<code>ulDeleteAfterExtract</code>	(<code>pycryptoki.CK_AES_CBC_PAD_EXTRACT_PARAMS</code> attribute), 113	<code>ulLen</code> (<code>pycryptoki.cryptoki.CK_KEY_DERIVATION_STRING_DATA</code> attribute), 142
<code>ulEncKeyLenInBits</code>	(<code>pycryptoki.CK_ECIES_PARAMS</code> attribute), 108	<code>ulSwValue</code> (<code>pycryptoki.HSM_STATS_PARAMS</code> attribute), 128
<code>ulEncodingScheme</code>	(<code>pycryptoki.CK_KDF_PRF_PARAMS</code> attribute), 72	<code>ulM</code> (<code>pycryptoki.cryptoki.CA_M_OF_N_STATUS</code> attribute), 116
<code>ulFlag</code> (<code>pycryptoki.cryptoki.CA_M_OF_N_STATUS</code> attribute), 116		<code>ulMacKeyLenInBits</code> (<code>pycryptoki.CK_ECIES_PARAMS</code> attribute), 108
<code>ulFreePrivateMemory</code>	(<code>pycryptoki.CK_TOKEN_INFO</code> attribute), 129	<code>ulMacLength</code> (<code>pycryptoki.CK_RC2_MAC_GENERAL_PARAMS</code> attribute), 81
<code>ulFreePublicMemory</code>	(<code>pycryptoki.CK_TOKEN_INFO</code> attribute), 129	<code>ulMacLength</code> (<code>pycryptoki.CK_RC5_MAC_GENERAL_PARAMS</code> attribute), 73
<code>ulHandle</code> (<code>pycryptoki.cryptoki.CK_AES_CBC_PAD_EXTRACT_PARAMS</code> attribute), 113		<code>ulMacLenInBits</code> (<code>pycryptoki.CK_ECIES_PARAMS</code> attribute), 108
<code>ulHighValue</code>	(<code>pycryptoki.HSM_STATS_PARAMS</code> attribute), 128	<code>ulMacSizeInBits</code> (<code>pycryptoki.CK_SSL3_KEY_MAT_PARAMS</code> attribute), 102
<code>ulID</code> (<code>pycryptoki.cryptoki.CA_M_OF_N_STATUS</code> attribute), 116		<code>ulMacSizeInBits</code> (<code>pycryptoki.CK_WTLS_KEY_MAT_PARAMS</code> attribute), 76
<code>ulId</code> (<code>pycryptoki.cryptoki.HSM_STATS_PARAMS</code> attribute), 128		<code>ulMaxKeySize</code> (<code>pycryptoki.CK_MECHANISM_INFO</code> attribute), 65
<code>ulIvBits</code> (<code>pycryptoki.cryptoki.CK_AES_GCM_PARAMS</code> attribute), 133		<code>ulMemberStatus</code> (<code>pycryptoki.CK_CLUSTER_STATE</code> attribute), 74
<code>ulIvLen</code> (<code>pycryptoki.cryptoki.CK_AES_GCM_PARAMS</code> attribute), 133		<code>ulMinKeySize</code> (<code>pycryptoki.CK_MECHANISM_INFO</code> attribute), 65
<code>ulIvLen</code> (<code>pycryptoki.cryptoki.CK_RC5_CBC_PARAMS</code> attribute), 135		<code>ulModuleSize</code> (<code>pycryptoki.CKCA_MODULE_INFO</code> attribute), 135
<code>ulIVSizeInBits</code> (<code>pycryptoki.CK_SSL3_KEY_MAT_PARAMS</code> attribute), 102		<code>ulN</code> (<code>pycryptoki.cryptoki.CA_M_OF_N_STATUS</code> attribute), 116
<code>ulIVSizeInBits</code> (<code>pycryptoki.CK_WTLS_KEY_MAT_PARAMS</code> attribute), 76		<code>ulNewPasswordLen</code> (<code>pycryptoki.CK_SKIPJACK_RELAYX_PARAMS</code> attribute), 77
<code>ulKeySizeInBits</code> (<code>pycryptoki.CK_SSL3_KEY_MAT_PARAMS</code> attribute), 102		<code>ulNewPublicDataLen</code> (<code>pycryptoki.CK_SKIPJACK_RELAYX_PARAMS</code> attribute), 77
<code>ulKeySizeInBits</code> (<code>pycryptoki.CK_WTLS_KEY_MAT_PARAMS</code> attribute), 76		<code>ulNewRandomLen</code> (<code>pycryptoki.CK_SKIPJACK_RELAYX_PARAMS</code> attribute), 77
<code>ulLabelLen</code> (<code>pycryptoki.CK_KDF_PRF_PARAMS</code> attribute), 72		<code>ulOldPasswordLen</code> (<code>pycryptoki.CK_SKIPJACK_RELAYX_PARAMS</code> attribute), 77
<code>ulLabelLen</code> (<code>pycryptoki.CK_TLS_PRF_PARAMS</code> attribute), 114		<code>ulOldPublicDataLen</code> (<code>pycryptoki.CK_SKIPJACK_RELAYX_PARAMS</code> attribute), 77
<code>ulLabelLen</code>	(<code>pycryptoki.CK_TLS_PRF_PARAMS</code> attribute), 114	

<i>toki.cryptoki.CK_SKIPJACK_RELAYX_PARAMS</i>	<i>ulPublicDataLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_X9_42_DH2_DERIVE_PARAMS</i> <i>attribute</i>), 77
<i>ulOldRandomLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_SKIPJACK_RELAYX_PARAMS</i> <i>attribute</i>), 77	<i>ulPublicDataLen</i>
<i>ulOldWrappedXLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_SKIPJACK_RELAYX_PARAMS</i> <i>attribute</i>), 77	<i>ulPublicDataLen2</i>
<i>ULong</i> (in module <i>pycryptoki.cryptoki</i>), 124		<i>toki.cryptoki.CK_ECDH2_DERIVE_PARAMS</i> <i>attribute</i>), 127
<i>ulOtherInfoLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_X9_42_DH1_DERIVE_PARAMS</i> <i>attribute</i>), 126	<i>ulPublicDataLen2</i>
<i>ulOtherInfoLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_X9_42_DH2_DERIVE_PARAMS</i> <i>attribute</i>), 87	<i>toki.cryptoki.CK_ECMQV_DERIVE_PARAMS</i> <i>attribute</i>), 69
<i>ulOtherInfoLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_X9_42_MQV_DERIVE_PARAMS</i> <i>attribute</i>), 67	<i>ulPublicDataLen2</i>
<i>ulPAndGLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_SKIPJACK_PRIVATE_WRAP_PARAMS</i> <i>attribute</i>), 115	<i>toki.cryptoki.CK_X9_42_DH2_DERIVE_PARAMS</i> <i>attribute</i>), 88
<i>ulPrfDataLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_PKCS5_PBKD2_PARAMS</i> <i>attribute</i>), 87	<i>ulQLen</i> (<i>pycryptoki.cryptoki.CK_SKIPJACK_PRIVATE_WRAP_PARAMS</i> <i>attribute</i>), 115
<i>ulPrivateDataLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_ECDH2_DERIVE_PARAMS</i> <i>attribute</i>), 127	<i>ulRandomLen</i>
<i>ulPrivateDataLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_ECMQV_DERIVE_PARAMS</i> <i>attribute</i>), 69	<i>toki.cryptoki.CK_KEA_DERIVE_PARAMS</i> <i>attribute</i>), 111
<i>ulPrivateDataLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_X9_42_DH2_DERIVE_PARAMS</i> <i>attribute</i>), 87	<i>ulRandomLen</i>
<i>ulPrivateDataLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_X9_42_MQV_DERIVE_PARAMS</i> <i>attribute</i>), 67	<i>toki.cryptoki.CK_SKIPJACK_PRIVATE_WRAP_PARAMS</i> <i>attribute</i>), 115
<i>ulPublicDataLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_ECDH1_DERIVE_PARAMS</i> <i>attribute</i>), 110	<i>ulRequestedAttributesLen</i>
<i>ulPublicDataLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_ECDH2_DERIVE_PARAMS</i> <i>attribute</i>), 127	<i>toki.cryptoki.CK_CMS_SIG_PARAMS</i> <i>attribute</i>), 68
<i>ulPublicDataLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_ECMQV_DERIVE_PARAMS</i> <i>attribute</i>), 69	<i>ulRequiredAttributesLen</i>
<i>ulPublicDataLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_KEA_DERIVE_PARAMS</i> <i>attribute</i>), 111	<i>toki.cryptoki.CK_CMS_SIG_PARAMS</i> <i>attribute</i>), 68
<i>ulPublicDataLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_SKIPJACK_PRIVATE_WRAP_PARAMS</i> <i>attribute</i>), 115	<i>ulRounds</i> (<i>pycryptoki.cryptoki.CK_RC5_CBC_PARAMS</i> <i>attribute</i>), 135
<i>ulPublicDataLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_X9_42_DH1_DERIVE_PARAMS</i> <i>attribute</i>), 126	<i>ulRounds</i> (<i>pycryptoki.cryptoki.CK_RC5_MAC_GENERAL_PARAMS</i> <i>attribute</i>), 73
<i>ulPublicDataLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_X9_42_DH2_DERIVE_PARAMS</i> <i>attribute</i>), 87	<i>ulRounds</i> (<i>pycryptoki.cryptoki.CK_RC5_PARAMS</i> <i>attribute</i>), 119
<i>ulPublicDataLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_X9_42_MQV_DERIVE_PARAMS</i> <i>attribute</i>), 67	<i>ulSaltSourceDataLen</i>
<i>ulPublicDataLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_ECDH2_DERIVE_PARAMS</i> <i>attribute</i>), 127	<i>toki.cryptoki.CK_PKCS5_PBKD2_PARAMS</i> <i>attribute</i>), 87
<i>ulPublicDataLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_ECMQV_DERIVE_PARAMS</i> <i>attribute</i>), 69	<i>ulSecretSize</i>
<i>ulPublicDataLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_KEA_DERIVE_PARAMS</i> <i>attribute</i>), 111	<i>toki.cryptoki.CA_M_OF_N_STATUS</i> <i>attribute</i>), 116
<i>ulPublicDataLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_SKIPJACK_PRIVATE_WRAP_PARAMS</i> <i>attribute</i>), 115	<i>ulSeedLen</i> (<i>pycryptoki.cryptoki.CK_KIP_PARAMS</i> <i>attribute</i>), 67
<i>ulPublicDataLen</i>	(<i>pycrypt-</i> <i>toki.cryptoki.CK_X9_42_DH1_DERIVE_PARAMS</i> <i>attribute</i>), 126	<i>ulSeedLen</i> (<i>pycryptoki.cryptoki.CK_TLS_PRF_PARAMS</i> <i>attribute</i>), 114
		<i>ulSeedLen</i> (<i>pycryptoki.cryptoki.CK_WTLS_PRF_PARAMS</i> <i>attribute</i>), 64
		<i>ulSequenceNumber</i>
		<i>toki.cryptoki.CK_WTLS_KEY_MAT_PARAMS</i> <i>attribute</i>), 76
		<i>ulServerRandomLen</i>

<code>toki.cryptoki.CK_SSL3_RANDOM_DATA</code> (<code>attribute</code>), 101	<code>ulWordsize</code> (<code>pycryptoki.cryptoki.CK_RC5_MAC_GENERAL_PARAMS</code> <code>attribute</code>), 73
<code>ulServerRandomLen</code> (<code>pycryptoki.cryptoki.CK_WTLS_RANDOM_DATA</code> <code>attribute</code>), 102	<code>ulWordsize</code> (<code>pycryptoki.cryptoki.CK_RC5_PARAMS</code> <code>attribute</code>), 119
<code>ulSharedDataLen</code> (<code>pycryptoki.cryptoki.CK_ECDH1_DERIVE_PARAMS</code> <code>attribute</code>), 110	<code>ulXLen</code> (<code>pycryptoki.cryptoki.CK_KEY_WRAP_SET_OAEP_PARAMS</code> <code>attribute</code>), 79
<code>ulSharedDataLen</code> (<code>pycryptoki.cryptoki.CK_ECDH2_DERIVE_PARAMS</code> <code>attribute</code>), 127	<code>UnlockMutex</code> (<code>pycryptoki.cryptoki.CK_C_INITIALIZE_ARGS</code> <code>attribute</code>), 131
<code>ulSharedDataLen</code> (<code>pycryptoki.cryptoki.CK_ECMQV_DERIVE_PARAMS</code> <code>attribute</code>), 69	<code>usDeviceError</code> (<code>pycryptoki.cryptoki.CK_SESSION_INFO</code> <code>attribute</code>), 75
<code>ulSharedDataLen</code> (<code>pycryptoki.cryptoki.CK_XOR_BASE_DATA_KDF_PARAMS</code> <code>attribute</code>), 133	<code>usEffectiveBits</code> (<code>pycryptoki.cryptoki.CK_RC2_CBC_PARAMS</code> <code>attribute</code>), 85
<code>ulSharedDataLen1</code> (<code>pycryptoki.cryptoki.CK_ECIES_PARAMS</code> <code>attribute</code>), 108	<code>usEffectiveBits</code> (<code>pycryptoki.cryptoki.CK_RC2_MAC_GENERAL_PARAMS</code> <code>attribute</code>), 81
<code>ulSharedDataLen2</code> (<code>pycryptoki.cryptoki.CK_ECIES_PARAMS</code> <code>attribute</code>), 108	<code>usIteration</code> (<code>pycryptoki.cryptoki.CK_PBE_PARAMS</code> <code>attribute</code>), 136
<code>ulSourceDataLen</code> (<code>pycryptoki.cryptoki.CK_RSA_PKCS_OAEP_PARAMS</code> <code>attribute</code>), 130	<code>usMaxPinLen</code> (<code>pycryptoki.cryptoki.CK_TOKEN_INFO</code> <code>attribute</code>), 129
<code>ulStorage</code> (<code>pycryptoki.cryptoki.CK_AES_CBC_PAD_EXTRACT_PARAMS</code> <code>attribute</code>), 113	<code>usMaxRwSessionCount</code> (<code>pycryptoki.cryptoki.CK_TOKEN_INFO</code> <code>attribute</code>), 129
<code>ulStorage</code> (<code>pycryptoki.cryptoki.CK_AES_CBC_PAD_INSERT_PARAMS</code> <code>attribute</code>), 142	<code>usMaxSessionCount</code> (<code>pycryptoki.cryptoki.CK_TOKEN_INFO</code> <code>attribute</code>), 129
<code>ulStorageType</code> (<code>pycryptoki.cryptoki.CK_AES_CBC_PAD_INSERT_PARAMS</code> <code>attribute</code>), 142	<code>usMinPinLen</code> (<code>pycryptoki.cryptoki.CK_TOKEN_INFO</code> <code>attribute</code>), 129
<code>ulTagBits</code> (<code>pycryptoki.cryptoki.CK_AES_GCM_PARAMS</code> <code>attribute</code>), 133	<code>usParameterLen</code> (<code>pycryptoki.cryptoki.CK_MECHANISM</code> <code>attribute</code>), 104
<code>ulTotalPrivateMemory</code> (<code>pycryptoki.cryptoki.CK_TOKEN_INFO</code> <code>attribute</code>), 129	<code>usPasswordLen</code> (<code>pycryptoki.cryptoki.CK_PBE_PARAMS</code> <code>attribute</code>), 136
<code>ulTotalPublicMemory</code> (<code>pycryptoki.cryptoki.CK_TOKEN_INFO</code> <code>attribute</code>), 129	<code>usPasswordLen</code> (<code>pycryptoki.cryptoki.CK_PKCS5_PBKD2_PARAMS</code> <code>attribute</code>), 87
<code>ulType</code> (<code>pycryptoki.cryptoki.CK_AES_CBC_PAD_EXTRACT_PARAMS</code> <code>attribute</code>), 113	<code>usPasswordLen</code> (<code>pycryptoki.cryptoki.CK_SKIPJACK_PRIVATE_WRAP_PARAMS</code> <code>attribute</code>), 115
<code>ulVectorLen</code> (<code>pycryptoki.cryptoki.CA_MOFN_ACTIVATION</code> <code>attribute</code>), 70	<code>usRwSessionCount</code> (<code>pycryptoki.cryptoki.CK_TOKEN_INFO</code> <code>attribute</code>), 130
<code>ulVectorLen</code> (<code>pycryptoki.cryptoki.CA_MOFN_GENERATION</code> <code>attribute</code>), 112	<code>usSaltLen</code> (<code>pycryptoki.cryptoki.CK_PBE_PARAMS</code> <code>attribute</code>), 136
<code>ulWeight</code> (<code>pycryptoki.cryptoki.CA_MOFN_GENERATION</code> <code>attribute</code>), 112	<code>usSaltLen</code> (<code>pycryptoki.cryptoki.CK_RSA_PKCS_PSS_PARAMS</code> <code>attribute</code>), 85
<code>ulWordsize</code> (<code>pycryptoki.cryptoki.CK_RC5_CBC_PARAMS</code> <code>attribute</code>), 135	<code>usSessionCount</code> (<code>pycryptoki.cryptoki.CK_TOKEN_INFO</code> <code>attribute</code>), 129

130

`usValueLen` (`pycryptoki.cryptoki.CK_ATTRIBUTE` attribute), 104

`usValueLen` (`pycryptoki.cryptoki.CK_OTP_PARAM` attribute), 129

`utcTime` (`pycryptoki.cryptoki.CK_TOKEN_INFO` attribute), 130

V

`version` (`pycryptoki.cryptoki.CK_FUNCTION_LIST` attribute), 146

`version` (`pycryptoki.cryptoki.CK_SFNT_CA_FUNCTION_LIST` attribute), 101

W

`Word` (in module `pycryptoki.cryptoki`), 70

`words` (`pycryptoki.cryptoki.swapper` attribute), 138

Y

`year` (`pycryptoki.cryptoki.CK_DATE` attribute), 63