draft RISC-V Internal TEE API

National Institute of Advanced Industrial Science and Technology (AIST)

Akira Tsukamoto, 2019/10/02

Contents

1 Main Page

2 asymmetric-key-varification

3 Asymmetric Key Verification Functions

Crypto, Sing and Verify with Asymmetric Key Verification Functions.
Pseudo code.
Not sure these crypto features should be prepared as APIs for Enclave or just using openssl variants.

The library used in keystone.
https://github.com/orlp/ed25519/
(zlib License)
https://github.com/mjosaarinen/tiny_sha3/
(MIT license)
In keystone.
https://github.com/keystone-enclave/keystone-sdk/tree/master/lib/verifier/ed25519
https://github.com/keystone-enclave/keystone-sdk/tree/master/lib/verifier/sha3

```
1 --- Asymmetric Key sign start ---
2     TEE_OperationHandle ho;
3     TEE_OperationHandle ao;
4     uint8_t hash[HASH_LENGTH];
5     char data[DSIZE];
6     char sig[64];
7     uint8_t sig_len = 64;
8
9     /* Calculate hash */
10     /* sha3_init() in sha3.c */
11     TEE_AllocateOperation(&ho, ALG_SHA256, SHA_LENGTH);
12
13     /* sha3_update() in sha3.c */
14     TEE_DigestUpdate(ho, data, CSIZE);
15
16     /* sha3_final() in sha3.c */
17     TEE_DigestDoFinal(ho, hash, data + CSIZE, DSIZE - CSIZE);
18
19     /* set ed25519 key */
20     TEE_AllocateOperation(&ao, TEE_ALG_ED25519, TEE_MODE_SIGN, BITS));
21     TEE_SetOperationKey(&ao, rsa_keypair);
22
23     /* Keystone has ed25519_sign()
24      * Equivalent in openssl is EVP_DigestSign() */
25     TEE_AsymmetricSignDigest(ho, hash, HASH_LENGTH, sig, &sig_len);
26
27     /* free up */
28     TEE_FreeOperation(ho);
29     TEE_FreeOperation(ao);
30
31     /* Get the signature */
32
33 --- Asymmetric Key sign end ---
```

```
1 --- Asymmetric Key verify start ---
2     TEE_OperationHandle ho;
3     TEE_OperationHandle ao;
4     uint8_t hash[HASH_LENGTH];
5     char data[DSIZE];
6     char sig[64];
7     uint8_t sig_len = 64;
8
9     /* set ed25519 key */
10     TEE_AllocateOperation(&ao, TEE_ALG_ED25519, TEE_MODE_VERIFY, BITS));
11     TEE_SetOperationKey(&ao, rsa_keypair);
12
13     /* Keystone has ed25519_verify()
14      * Equivalent in openssl is EVP_DigestVerify() */
15     verify_ok = TEE_AsymmetricVerifyDigest(ao, data, HASH_LENGTH, sig, sig_len);
16
17     /* Check verify_ok for success of verification */
18
19 --- Asymmetric Key verify end ---
```

4 message-digest

5 Message Digest Functions

Pseudo code of how to use Message Digest Functions.
Keystone uses sha3.c which is almost identical.
Ultimate question is whether this should be done in Enclave (U-Mode) or Runtime (S-Mode).

The library used in keystone.
https://github.com/mjosaarinen/tiny_sha3/
(MIT license)
In keystone.
https://github.com/keystone-enclave/keystone-sdk/tree/master/lib/verifier/sha3

```
1 --- start digest ---
2 #define SHA_LENGTH 256 / 8
3 #define CSIZE 8
4 #define DSIZE 16
5     TEE_OperationHandle ho;
6     uint8_t hash[SHA_LENGTH];
7     char data[DSIZE];
8     char *pdata;
9
10     /* sha3_init() in sha3.c */
11     TEE_AllocateOperation(&ho, ALG_SHA256, SHA_LENGTH);
12
13     /* sha3_update() in sha3.c */
14     TEE_DigestUpdate(ho, data, CSIZE);
15
16     /* sha3_final() in sha3.c */
17     TEE_DigestDoFinal(ho, hash, data + CSIZE, DSIZE - CSIZE);
18
19     /* hash value is ready */
20
21     TEE_FreeOperation(ho);
22 --- end digest ---
```

6 Generate PDF doc

```
1 $ make doc
```

Build

```
1 $ make
```

7 secure-storage

8 Secure Storage Functions

Core Functions, Secure Storage Functions.
Pseudo code of how to use Secure Storage.
These could be implemented using ocall on Keystone.
I prefer this feature is implemented in runtime in S-Mode for less overhead rather than switching to host os every time.

Almost identical to open(), clone(), read(), write() in POSIX API.

```
1 --- write file start ---
2      TEE_ObjectHandle o;
3      char buf[bufsize];
4
5      TEE_CreatePersistantObject(&o, filename, namelen, WO);
6
7      /* fill the date in buffer */
8
9      TEE_WriteObjectData(o, buf, bufsize);
10
11      TEE_CloseObject(o);
12 --- write file end ---
```

```
1 --- read file start ---
2      TEE_ObjectHandle o;
3      char buf[bufsize];
4
5      TEE_OpenPersistantObject(&o, filename, namelen, RO);
6
7      TEE_ReadObjectData(o, buf, bufsize);
8
9      /* use the date in buffer */
10
11      TEE_CloseObject(o);
12 --- read file end ---
```

9 symmetric-key-varification

10 Symmetric Key Verification Functions

Crypto, Authenticated Encryption with Symmetric Key Verification Functions.
Pseudo code.
Not sure these crypto features should be prepared as APIs for Enclave or just using openssl variants.

The library used in keystone.
https://github.com/kokke/tiny-AES-c
(The Unlicense, public domain)

In keystone.
https://github.com/keystone-enclave/keystone-sdk/tree/ef484d36db1c40a0e0a4367f31c95b90d6
AES-c/app

```
1 --- AE encryption start ---
2     TEE_OperationHandle ho;
3
4     /* set the AES key, skipping in this pseudo code */
5
6     /* Equivalent in openssl is EVP_EncryptInit_ex() */
7     TEE_AEInit(ho, nonce, nonce_len, AES_256_GCM_BITS);
8
9     /* Equivalent in openssl is EVP_EncryptUpdate() */
10     TEE_AEUpdate(ho, plain, plain_len, cipher, &cipher_len);
11
12     /* Equivalent in openssl is EVP_EncryptFinal() */
13     TEE_AEEncryptFinal(ho, plain, plain_len, cipher, &cipher_len, tag, &tag_len);
14
15     /* Get the auth_tag */
16
17 --- AE encryption end ---
```

```
1 --- AE decrypt and verify start ---
2     TEE_OperationHandle ho;
3
4     /* set the AES key, skipping in this pseudo code */
5
6     /* Equivalent in openssl is EVP_DecryptInit_ex() */
7     TEE_AEInit(ho, nonce, nonce_len, AES_256_GCM_BITS);
8
```

```
9     /* Equivalent in openssl is EVP_DecryptUpdate() */
10     TEE_AEUpdate(ho, plain, plain_len, cipher, cipher_len);
11
12     /* Equivalent in openssl require two functions
13     EVP_CIPHER_CTX_ctrl(tag) and EVP_DecryptFinal(others) */
14     verify_ok = TEE_AEDecryptFinal(ho, plain, plain_len, cipher, &cipher_len, tag, tag_len);
15
16     /* Check verify_ok for success of decrypting and authentication */
17
18 --- AE decrypt and verify end ---
```

11 Class Index

11.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

12 File Index

12.1 File List

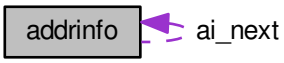
Here is a list of all files with brief descriptions:

13 Class Documentation

13.1 addrinfo Struct Reference

`#include <tee_api_types.h>`

Collaboration diagram for addrinfo:



Public Attributes

- int ai_flags
- int ai_family
- int ai_socktype
- int ai_protocol
- socklen_t ai_addrlen
- struct sockaddr * ai_addr
- char * ai_canonname
- struct addrinfo * ai_next

13.1.1 Member Data Documentation

13.1.1.1 struct sockaddr* addrinfo::ai_addr

13.1.1.2 socklen_t addrinfo::ai_addrlen

13.1.1.3 char* addrinfo::ai_canonname

13.1.1.4 int addrinfo::ai_family

13.1.1.5 int addrinfo::ai_flags

13.1.1.6 struct addrinfo* addrinfo::ai_next

13.1.1.7 int addrinfo::ai_protocol

13.1.1.8 int addrinfo::ai_socktype

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

- include/tee_api_types.h

13.2 pollfd Struct Reference

`#include <tee_api_types.h>`

Public Attributes

- int fd
- short int events
- short int revents

13.2.1 Member Data Documentation

13.2.1.1 short int pollfd::events

13.2.1.2 int pollfd::fd

13.2.1.3 short int pollfd::revents

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

- include/tee_api_types.h

13.3 TEE_Attribute Struct Reference

`#include <tee_api_types.h>`

Public Attributes

- uint32_t attributeID
- union {
  struct {
  void * buffer
  uint32_t length
  } ref
  struct {
  uint32_t a
  uint32_t b
  } value
  } content

13.3.1 Member Data Documentation

13.3.1.1 uint32_t TEE_Attribute::a

13.3.1.2 uint32_t TEE_Attribute::attributeID

13.3.1.3 uint32_t TEE_Attribute::b

13.3.1.4 void* TEE_Attribute::buffer

13.3.1.5 union { ... } TEE_Attribute::content

13.3.1.6 uint32_t TEE_Attribute::length

13.3.1.7 struct { ... } TEE_Attribute::ref

13.3.1.8 struct { ... } TEE_Attribute::value

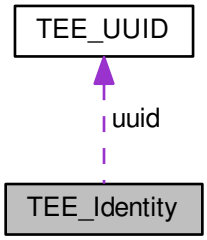
The documentation for this struct was generated from the following file:

- include/tee_api_types.h

13.4 TEE_Identity Struct Reference

`#include <tee_api_types.h>`

Collaboration diagram for TEE_Identity:



Public Attributes

- uint32_t login
- TEE_UUID uuid

13.4.1 Member Data Documentation

13.4.1.1 uint32_t TEE_Identity::login

13.4.1.2 TEE_UUID TEE_Identity::uuid

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

- include/tee_api_types.h

13.5 TEE_ObjectInfo Struct Reference

`#include <tee_api_types.h>`

Public Attributes

- uint32_t objectType
- union {
  uint32_t keySize
  uint32_t objectSize
  };
- union {
  uint32_t maxKeySize
  uint32_t maxObjectSize
  };
- uint32_t objectUsage
- uint32_t dataSize
- uint32_t dataPosition
- uint32_t handleFlags

13.5.1 Member Data Documentation

13.5.1.1 __extension__ { ... }

13.5.1.2 __extension__ { ... }

13.5.1.3 uint32_t TEE_ObjectInfo::dataPosition

13.5.1.4 uint32_t TEE_ObjectInfo::dataSize

13.5.1.5 uint32_t TEE_ObjectInfo::handleFlags

13.5.1.6 uint32_t TEE_ObjectInfo::keySize

13.5.1.7 uint32_t TEE_ObjectInfo::maxKeySize

13.5.1.8 uint32_t TEE_ObjectInfo::maxObjectSize

13.5.1.9 uint32_t TEE_ObjectInfo::objectSize

13.5.1.10 uint32_t TEE_ObjectInfo::objectType

13.5.1.11 uint32_t TEE_ObjectInfo::objectUsage

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

- include/tee_api_types.h

13.6 TEE_OperationInfo Struct Reference

`#include <tee_api_types.h>`

Public Attributes

- uint32_t algorithm
- uint32_t operationClass
- uint32_t mode
- uint32_t digestLength
- uint32_t maxKeySize
- uint32_t keySize
- uint32_t requiredKeyUsage
- uint32_t handleState

13.6.1 Member Data Documentation

13.6.1.1 uint32_t TEE_OperationInfo::algorithm

13.6.1.2 uint32_t TEE_OperationInfo::digestLength

13.6.1.3 uint32_t TEE_OperationInfo::handleState

13.6.1.4 uint32_t TEE_OperationInfo::keySize

13.6.1.5 uint32_t TEE_OperationInfo::maxKeySize

13.6.1.6 uint32_t TEE_OperationInfo::mode

13.6.1.7 uint32_t TEE_OperationInfo::operationClass

13.6.1.8 uint32_t TEE_OperationInfo::requiredKeyUsage

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

- include/tee_api_types.h

13.7 TEE_OperationInfoKey Struct Reference

```
#include <tee_api_types.h>
```

Public Attributes

- uint32_t keySize
- uint32_t requiredKeyUsage

13.7.1 Member Data Documentation

13.7.1.1 uint32_t TEE_OperationInfoKey::keySize

13.7.1.2 uint32_t TEE_OperationInfoKey::requiredKeyUsage

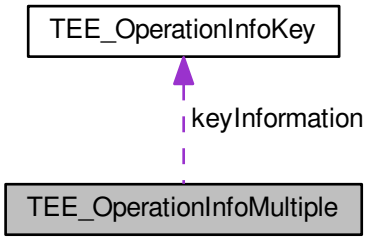
The documentation for this struct was generated from the following file:

- include/tee_api_types.h

13.8 TEE_OperationInfoMultiple Struct Reference

`#include <tee_api_types.h>`

Collaboration diagram for TEE_OperationInfoMultiple:



Public Attributes

- uint32_t algorithm
- uint32_t operationClass
- uint32_t mode
- uint32_t digestLength
- uint32_t maxKeySize
- uint32_t handleState
- uint32_t operationState
- uint32_t numberOfKeys
- TEE_OperationInfoKey keyInformation [ ]

13.8.1 Member Data Documentation

13.8.1.1 uint32_t TEE_OperationInfoMultiple::algorithm

13.8.1.2 uint32_t TEE_OperationInfoMultiple::digestLength

13.8.1.3 uint32_t TEE_OperationInfoMultiple::handleState

13.8.1.4 TEE_OperationInfoKey TEE_OperationInfoMultiple::keyInformation[ ]

13.8.1.5 uint32_t TEE_OperationInfoMultiple::maxKeySize

13.8.1.6 uint32_t TEE_OperationInfoMultiple::mode

13.8.1.7 uint32_t TEE_OperationInfoMultiple::numberOfKeys

13.8.1.8 uint32_t TEE_OperationInfoMultiple::operationClass

13.8.1.9 uint32_t TEE_OperationInfoMultiple::operationState

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

- include/tee_api_types.h

13.9 TEE_Param Union Reference

`#include <tee_api_types.h>`

Public Attributes

- struct {
  void * buffer
  uint32_t size
  } memref
- struct {
  uint32_t a
  uint32_t b
  } value

13.9.1 Member Data Documentation

13.9.1.1 uint32_t TEE_Param::a

13.9.1.2 uint32_t TEE_Param::b

13.9.1.3 void* TEE_Param::buffer

13.9.1.4 struct { ... } TEE_Param::memref

13.9.1.5 uint32_t TEE_Param::size

13.9.1.6 struct { ... } TEE_Param::value

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

- include/tee_api_types.h

13.10 TEE_SEAID Struct Reference

`#include <tee_api_types.h>`

Public Attributes

- uint8_t * buffer
- size_t bufferLen

13.10.1 Member Data Documentation

13.10.1.1 uint8_t* TEE_SEAID::buffer

13.10.1.2 size_t TEE_SEAID::bufferLen

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

- include/tee_api_types.h

13.11 TEE_SEReaderProperties Struct Reference

```
#include <tee_api_types.h>
```

Public Attributes

- bool sePresent
- bool teeOnly
- bool selectResponseEnable

13.11.1 Member Data Documentation

13.11.1.1 bool TEE_SEReaderProperties::selectResponseEnable

13.11.1.2 bool TEE_SEReaderProperties::sePresent

13.11.1.3 bool TEE_SEReaderProperties::teeOnly

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

- include/tee_api_types.h

13.12 TEE_Time Struct Reference

```
#include <tee_api_types.h>
```

Public Attributes

- uint32_t seconds
- uint32_t millis

13.12.1 Member Data Documentation

13.12.1.1 uint32_t TEE_Time::millis

13.12.1.2 uint32_t TEE_Time::seconds

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

- include/tee_api_types.h

13.13 TEE_UUID Struct Reference

```
#include <tee_api_types.h>
```

Public Attributes

- uint32_t timeLow
- uint16_t timeMid
- uint16_t timeHiAndVersion
- uint8_t clockSeqAndNode [8]

13.13.1 Member Data Documentation

13.13.1.1 uint8_t TEE_UUID::clockSeqAndNode[8]

13.13.1.2 uint16_t TEE_UUID::timeHiAndVersion

13.13.1.3 uint32_t TEE_UUID::timeLow

13.13.1.4 uint16_t TEE_UUID::timeMid

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

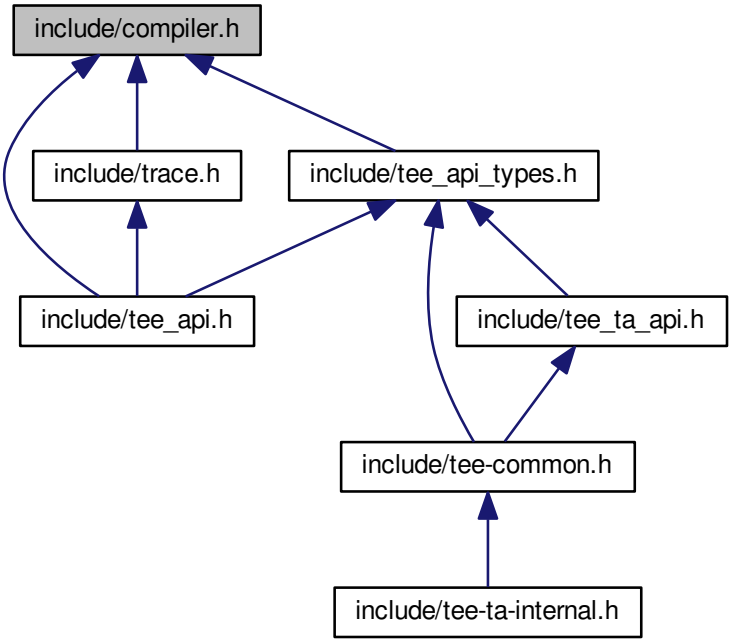
- include/tee_api_types.h

14 File Documentation

14.1 asymmetric-key-varification.md File Reference

14.2 include/compiler.h File Reference

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



Macros

- #define __deprecated __attribute__((deprecated))
- #define __packed __attribute__((packed))
- #define __weak __attribute__((weak))
- #define __noreturn __attribute__((noreturn))
- #define __pure __attribute__((pure))
- #define __aligned(x) __attribute__((aligned(x)))
- #define __printf(a, b) __attribute__((format(printf, a, b)))
- #define __noinline __attribute__((noinline))
- #define __attr_const __attribute__((__const__))
- #define __unused __attribute__((unused))
- #define __maybe_unused __attribute__((unused))
- #define __used __attribute__((__used__))
- #define __must_check __attribute__((warn_unused_result))
- #define __cold __attribute__((__cold__))
- #define __section(x) __attribute__((section(x)))

- #define __data __section(".data")
- #define __bss __section(".bss")
- #define __rodata __section(".rodata")
- #define __rodata_unpaged __section(".rodata.__unpaged")
- #define __early_ta __section(".rodata.early_ta")
- #define __noprof __attribute__((no_instrument_function))
- #define __compiler_bswap64(x) __builtin_bswap64((x))
- #define __compiler_bswap32(x) __builtin_bswap32((x))
- #define __compiler_bswap16(x) __builtin_bswap16((x))
- #define __GCC_VERSION
- #define __INTOF_HALF_MAX_SIGNED(type) ((type)1 << (sizeof(type)*8-2))
- #define __INTOF_MAX_SIGNED(type)
- #define __INTOF_MIN_SIGNED(type) (-1 - __INTOF_MAX_SIGNED(type))
- #define __INTOF_MIN(type) ((type)-1 < 1?__INTOF_MIN_SIGNED(type):(type)0)
- #define __INTOF_MAX(type) ((type)~__INTOF_MIN(type))
- #define __INTOF_ASSIGN(dest, src)
- #define __INTOF_ADD(c, a, b)
- #define __INTOF_SUB(c, a, b)
- #define __intof_mul_negate ((__intof_oa < 1) != (__intof_ob < 1))
- #define __intof_mul_hshift (sizeof(uintmax_t) * 8 / 2)
- #define __intof_mul_hmask (UINTMAX_MAX >> __intof_mul_hshift)
- #define __intof_mul_a0 ((uintmax_t)(__intof_a) >> __intof_mul_hshift)
- #define __intof_mul_b0 ((uintmax_t)(__intof_b) >> __intof_mul_hshift)
- #define __intof_mul_a1 ((uintmax_t)(__intof_a) & __intof_mul_hmask)
- #define __intof_mul_b1 ((uintmax_t)(__intof_b) & __intof_mul_hmask)
- #define __intof_mul_t
- #define __INTOF_MUL(c, a, b)
- #define __compiler_add_overflow(a, b, res) __INTOF_ADD(*(res), (a), (b))
- #define __compiler_sub_overflow(a, b, res) __INTOF_SUB(*(res), (a), (b))
- #define __compiler_mul_overflow(a, b, res) __INTOF_MUL(*(res), (a), (b))
- #define __compiler_compare_and_swap(p, oval, nval)
- #define __compiler_atomic_load(p) __atomic_load_n((p), __ATOMIC_RELAXED)
- #define __compiler_atomic_store(p, val) __atomic_store_n((p), (val), __ATOMIC_RELAXED)

14.2.1 Macro Definition Documentation

14.2.1.1 #define __aligned(x) __attribute__((aligned(x)))

14.2.1.2 #define __attr_const __attribute__((__const__))

14.2.1.3 #define __bss __section(".bss")

14.2.1.4 #define __cold __attribute__((__cold__))

14.2.1.5 #define __compiler_add_overflow(a, b, res) __INTOF_ADD(*(res), (a), (b))

14.2.1.6 #define __compiler_atomic_load(p) __atomic_load_n((p), __ATOMIC_RELAXED)

14.2.1.7 #define __compiler_atomic_store(p, val) __atomic_store_n((p), (val), __ATOMIC_RELAXED)

14.2.1.8 #define __compiler_bswap16(x) __builtin_bswap16((x))

14.2.1.9 #define __compiler_bswap32(x) __builtin_bswap32((x))

14.2.1.10 #define __compiler_bswap64(x) __builtin_bswap64((x))

14.2.1.11 #define __compiler_compare_and_swap(p, oval, nval)

Value:

```
__atomic_compare_exchange_n((p), (oval), (nval), true, \
                  __ATOMIC_ACQUIRE, __ATOMIC_RELAXED) \
```

__HAVE_BUILTIN_OVERFLOW

14.2.1.12 #define __compiler_mul_overflow(a, b, res) __INTOF_MUL(*(res), (a), (b))

14.2.1.13 #define __compiler_sub_overflow(a, b, res) __INTOF_SUB(*(res), (a), (b))

14.2.1.14 #define __data __section(".data")

14.2.1.15 #define __deprecated __attribute__((deprecated))

14.2.1.16 #define __early_ta __section(".rodata.early_ta")

14.2.1.17 #define __GCC_VERSION

Value:

```
(__GNUC__ * 10000 + __GNUC_MINOR__ * 100 + \
               __GNUC_PATCHLEVEL__)
```

14.2.1.18 #define __INTOF_ADD(c, a, b)

Value:

```
(__extension__({ \
    typeof(a) __intofa_a = (a); \
    typeof(b) __intofa_b = (b); \
    \
    __intofa_b < 1 ? \
        ((__INTOF_MIN(typeof(c)) - __intofa_b <= __intofa_a) ? \
            __INTOF_ASSIGN((c), __intofa_a + __intofa_b) : 1) : \
        ((__INTOF_MAX(typeof(c)) - __intofa_b >= __intofa_a) ? \
            __INTOF_ASSIGN((c), __intofa_a + __intofa_b) : 1); \
}))
```

14.2.1.19 #define __INTOF_ASSIGN(dest, src)

Value:

```
(__extension__({ \
    typeof(src) __intof_x = (src); \
    typeof(dest) __intof_y = __intof_x; \
    (((uintmax_t)__intof_x == (uintmax_t)__intof_y) && \
     ((__intof_x < 1) == (__intof_y < 1)) ? \
        (void)((dest) = __intof_y) , 0 : 1); \
}))
```

14.2.1.20 #define __INTOF_HALF_MAX_SIGNED(type) ((type)1 << (sizeof(type)*8-2))

__HAVE_BUILTIN_OVERFLOW

14.2.1.21 #define __INTOF_MAX(type) ((type)~__INTOF_MIN(type))

14.2.1.22 #define __INTOF_MAX_SIGNED(type)

Value:

```
(__INTOF_HALF_MAX_SIGNED(type) - 1 + \
                __INTOF_HALF_MAX_SIGNED(type))
```

14.2.1.23 #define __INTOF_MIN(type) ((type)-1 < 1?__INTOF_MIN_SIGNED(type):(type)0)

14.2.1.24 #define __INTOF_MIN_SIGNED(type) (-1 - __INTOF_MAX_SIGNED(type))

14.2.1.25 #define __INTOF_MUL(c, a, b)

Value:

```
(__extension__({ \
    typeof(a) __intof_oa = (a); \
    typeof(a) __intof_a = __intof_oa < 1 ? -__intof_oa : __intof_oa; \
    typeof(b) __intof_ob = (b); \
    typeof(b) __intof_b = __intof_ob < 1 ? -__intof_ob : __intof_ob; \
    typeof(c) __intof_c; \
    \
    __intof_oa == 0 || __intof_ob == 0 || \
    __intof_oa == 1 || __intof_ob == 1 ? \
        __INTOF_ASSIGN((c), __intof_oa * __intof_ob) : \
     (__intof_mul_a0 && __intof_mul_b0) || \
      __intof_mul_t > __intof_mul_hmask ?  1 : \
    __INTOF_ADD((__intof_c), __intof_mul_t <<
      __intof_mul_hshift, \
                __intof_mul_a1 * __intof_mul_b1) ? 1 : \
    __intof_mul_negate ? __INTOF_ASSIGN((c), -__intof_c) : \
                __INTOF_ASSIGN((c), __intof_c); \
}))
```

14.2.1.26 #define __intof_mul_a0 ((uintmax_t)(__intof_a) >> __intof_mul_hshift)

14.2.1.27 #define __intof_mul_a1 ((uintmax_t)(__intof_a) & __intof_mul_hmask)

14.2.1.28 #define __intof_mul_b0 ((uintmax_t)(__intof_b) >> __intof_mul_hshift)

14.2.1.29 #define __intof_mul_b1 ((uintmax_t)(__intof_b) & __intof_mul_hmask)

14.2.1.30 #define __intof_mul_hmask (UINTMAX_MAX >> __intof_mul_hshift)

14.2.1.31 #define __intof_mul_hshift (sizeof(uintmax_t) * 8 / 2)

14.2.1.32 #define __intof_mul_negate ((__intof_oa < 1) != (__intof_ob < 1))

14.2.1.33 #define __intof_mul_t

Value:

```
(__intof_mul_a1 * __intof_mul_b0 + \
                __intof_mul_a0 * __intof_mul_b1)
```

14.2.1.34 #define __INTOF_SUB(c, a, b)

Value:

```
(__extension__({ \
    typeof(a) __intofs_a = a; \
    typeof(b) __intofs_b = b; \
    \
    __intofs_b < 1 ? \
        ((__INTOF_MAX(typeof(c)) + __intofs_b >= __intofs_a) ? \
            __INTOF_ASSIGN((c), __intofs_a - __intofs_b) : 1) : \
        ((__INTOF_MIN(typeof(c)) + __intofs_b <= __intofs_a) ? \
            __INTOF_ASSIGN((c), __intofs_a - __intofs_b) : 1); \
}))
```

14.2.1.35 #define __maybe_unused __attribute__((unused))

14.2.1.36 #define __must_check __attribute__((warn_unused_result))

14.2.1.37 #define __noinline __attribute__((noinline))

14.2.1.38 #define __noprof __attribute__((no_instrument_function))

14.2.1.39 #define __noreturn __attribute__((noreturn))

14.2.1.40 #define __packed __attribute__((packed))

14.2.1.41 #define __printf(a, b) __attribute__((format(printf, a, b)))

14.2.1.42 #define __pure __attribute__((pure))

14.2.1.43 #define __rodata __section(".rodata")

14.2.1.44 #define __rodata_unpaged __section(".rodata.__unpaged")

14.2.1.45 #define __section(x) __attribute__((section(x)))

14.2.1.46 #define __unused __attribute__((unused))

14.2.1.47 #define __used __attribute__((__used__))

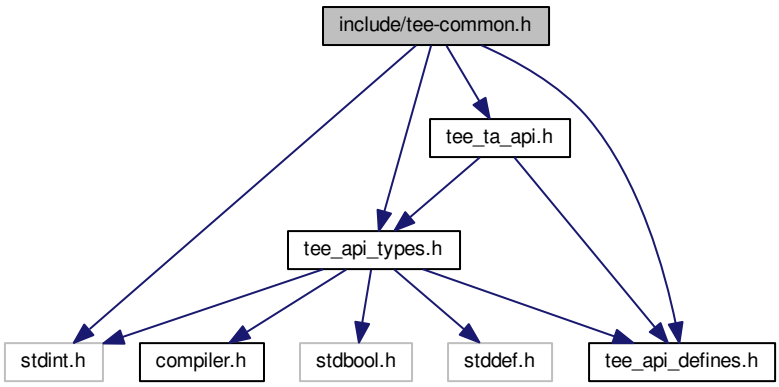
14.2.1.48 #define __weak __attribute__((weak))

14.3 include/tee-common.h File Reference

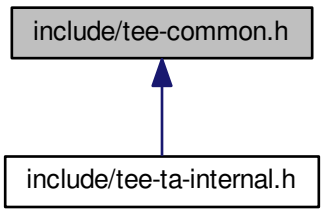
Common type and definitions of RISC-V TEE.

```
#include <stdint.h>
#include <tee_api_defines.h>
#include <tee_api_types.h>
#include <tee_ta_api.h>
```

Include dependency graph for tee-common.h:



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



Macros

- #define pr_deb(...) do { } while (0)

14.3.1 Detailed Description

Common type and definitions of RISC-V TEE.

draft RISC-V Internal TEE API

Author

Akira Tsukamoto, AIST

Date

2019/09/25

14.3.2 Macro Definition Documentation

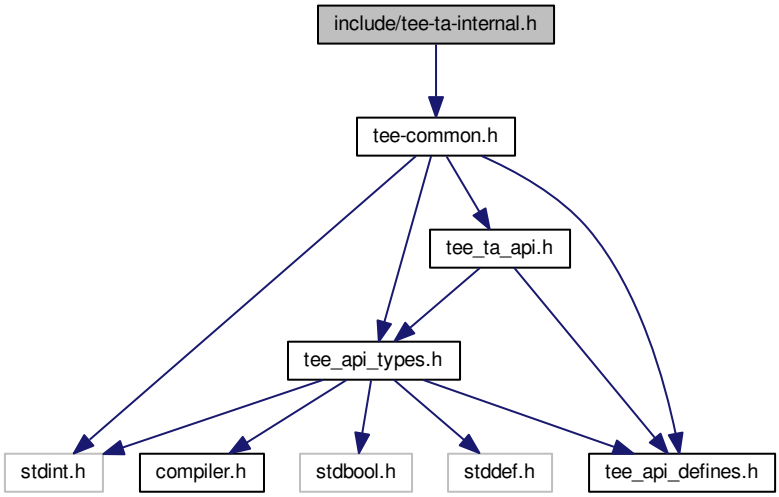
14.3.2.1 #define pr_deb(...) do { } while (0)

14.4 include/tee-ta-internal.h File Reference

Candidate API list for Global Platform like RISC-V TEE.

```
#include "tee-common.h"
```

Include dependency graph for tee-ta-internal.h:



Functions

- void TEE_GetREETime (TEE_Time *time)

  Core Functions, Time Functions.
- void TEE_GetSystemTime (TEE_Time *time)

  Core Functions, Time Functions.
- TEE_Result GetRelTimeStart (uint64_t start)

  Core Functions, Time Functions.
- TEE_Result GetRelTimeEnd (uint64_t end)

  Core Functions, Time Functions.
- TEE_Result TEE_CreatePersistentObject (uint32_t storageID, const void *objectID, uint32_t objectIDLen, uint32_t flags, TEE_ObjectHandle attributes, const void *initialData, uint32_t initialDataLen, TEE_Object↩Handle *object)

  Core Functions, Secure Storage Functions (data is isolated for each TA)
- TEE_Result TEE_OpenPersistentObject (uint32_t storageID, const void *objectID, uint32_t objectIDLen, uint32_t flags, TEE_ObjectHandle *object)

  Core Functions, Secure Storage Functions (data is isolated for each TA)
- TEE_Result TEE_GetObjectInfo1 (TEE_ObjectHandle object, TEE_ObjectInfo *objectInfo)

  Core Functions, Secure Storage Functions (data is isolated for each TA)
- TEE_Result TEE_WriteObjectData (TEE_ObjectHandle object, const void *buffer, uint32_t size)

Core Functions, Secure Storage Functions (data is isolated for each TA)

- TEE_Result TEE_ReadObjectData (TEE_ObjectHandle object, void *buffer, uint32_t size, uint32_t *count)

  Core Functions, Secure Storage Functions (data is isolated for each TA)
- void TEE_CloseObject (TEE_ObjectHandle object)

  Core Functions, Secure Storage Functions (data is isolated for each TA)
- void TEE_GenerateRandom (void *randomBuffer, uint32_t randomBufferLen)

  Crypto, common.
- TEE_Result TEE_AllocateOperation (TEE_OperationHandle *operation, uint32_t algorithm, uint32_t mode, uint32_t maxKeySize)

  Crypto, for all Crypto Functions.
- void TEE_FreeOperation (TEE_OperationHandle operation)

  Crypto, for all Crypto Functions.
- void TEE_DigestUpdate (TEE_OperationHandle operation, const void *chunk, uint32_t chunkSize)

  Crypto, Message Digest Functions.
- TEE_Result TEE_DigestDoFinal (TEE_OperationHandle operation, const void *chunk, uint32_t chunkLen, void *hash, uint32_t *hashLen)
- TEE_Result TEE_AEInit (TEE_OperationHandle operation, const void *nonce, uint32_t nonceLen, uint32_t tagLen, uint32_t AADLen, uint32_t payloadLen)

  Crypto, Authenticated Encryption with Symmetric key Verification Functions.
- TEE_Result TEE_AEUpdate (TEE_OperationHandle operation, const void *srcData, uint32_t srcLen, void *destData, uint32_t *destLen)

  Crypto, Authenticated Encryption with Symmetric key Verification Functions.
- TEE_Result TEE_AEEncryptFinal (TEE_OperationHandle operation, const void *srcData, uint32_t srcLen, void *destData, uint32_t *destLen, void *tag, uint32_t *tagLen)

  Crypto, Authenticated Encryption with Symmetric key Verification Functions.
- TEE_Result TEE_AEDecryptFinal (TEE_OperationHandle operation, const void *srcData, uint32_t srcLen, void *destData, uint32_t *destLen, void *tag, uint32_t tagLen)

  Crypto, Authenticated Encryption with Symmetric key Verification Functions.
- TEE_Result TEE_AllocateTransientObject (TEE_ObjectType objectType, uint32_t maxKeySize, TEE_↩ ObjectHandle *object)

  Crypto, Asymmetric key Verification Functions.
- void TEE_InitRefAttribute (TEE_Attribute *attr, uint32_t attributeID, const void *buffer, uint32_t length)

  Crypto, Asymmetric key Verification Functions.
- void TEE_FreeTransientObject (TEE_ObjectHandle object)

  Crypto, Asymmetric key Verification Functions.
- TEE_Result TEE_AsymmetricSignDigest (TEE_OperationHandle operation, const TEE_Attribute *params, uint32_t paramCount, const void *digest, uint32_t digestLen, void *signature, uint32_t *signatureLen)

  Crypto, Asymmetric key Verification Functions.
- TEE_Result TEE_AsymmetricVerifyDigest (TEE_OperationHandle operation, const TEE_Attribute *params, uint32_t paramCount, const void *digest, uint32_t digestLen, const void *signature, uint32_t signatureLen)

  Crypto, Asymmetric key Verification Functions.

14.4.1 Detailed Description

Candidate API list for Global Platform like RISC-V TEE.

draft RISC-V Internal TEE API

Author

Akira Tsukamoto, AIST

Date

2019/09/25

14.4.2 Function Documentation

14.4.2.1 TEE_Result GetRelTimeEnd (uint64_t *end*)

Core Functions, Time Functions.

Return the elapsed.

14.4.2.2 TEE_Result GetRelTimeStart (uint64_t *start*)

Core Functions, Time Functions.

Fast relative Time function which guarantees no hart switch or context switch between Trusted and Untrusted sides.

Most of the time ending up writing similar functions when only measuring the relative time in usec resolution which do not require the quality of the time itself but the distance of the two points.
For the usage above, the function does not have to return wall clock time.
Not prepared in both Keystone and GP.

14.4.2.3 TEE_Result TEE_AEDecryptFinal (TEE_OperationHandle *operation,* const void * *srcData,* uint32_t *srcLen,* void * *destData,* uint32_t * *destLen,* void * *tag,* uint32_t *tagLen*)

Crypto, Authenticated Encryption with Symmetric key Verification Functions.

Supports TEE_ALG_AES_CCM, TEE_ALG_AES_GCM.

14.4.2.4 TEE_Result TEE_AEEncryptFinal (TEE_OperationHandle *operation,* const void * *srcData,* uint32_t *srcLen,* void * *destData,* uint32_t * *destLen,* void * *tag,* uint32_t * *tagLen*)

Crypto, Authenticated Encryption with Symmetric key Verification Functions.

Supports TEE_ALG_AES_CCM, TEE_ALG_AES_GCM.

14.4.2.5 TEE_Result TEE_AEInit (TEE_OperationHandle *operation,* const void * *nonce,* uint32_t *nonceLen,* uint32_t *tagLen,* uint32_t *AADLen,* uint32_t *payloadLen*)

Crypto, Authenticated Encryption with Symmetric key Verification Functions.

Supports TEE_ALG_AES_CCM, TEE_ALG_AES_GCM.

14.4.2.6 TEE_Result TEE_AEUpdate (TEE_OperationHandle *operation,* const void * *srcData,* uint32_t *srcLen,* void * *destData,* uint32_t * *destLen*)

Crypto, Authenticated Encryption with Symmetric key Verification Functions.

Supports TEE_ALG_AES_CCM, TEE_ALG_AES_GCM.

14.4.2.7 TEE_Result TEE_AllocateOperation (TEE_OperationHandle * *operation,* uint32_t *algorithm,* uint32_t *mode,* uint32_t *maxKeySize*)

Crypto, for all Crypto Functions.

All Crypto Functions use TEE_OperationHandle* operation instances.
Create Crypto instance.

14.4.2.8 TEE_Result TEE_AllocateTransientObject (TEE_ObjectType *objectType*, uint32_t *maxKeySize*, TEE_ObjectHandle * *object*)

Crypto, Asymmetric key Verification Functions.

Create object storing asymmetric key.

14.4.2.9 TEE_Result TEE_AsymmetricSignDigest (TEE_OperationHandle *operation*, const TEE_Attribute * *params*, uint32_t *paramCount*, const void * *digest*, uint32_t *digestLen*, void * *signature*, uint32_t * *signatureLen*)

Crypto, Asymmetric key Verification Functions.

Sign a message digest within an asymmetric key operation.
Keystone has ed25519_sign().
Equivalent in openssl is EVP_DigestSign().

14.4.2.10 TEE_Result TEE_AsymmetricVerifyDigest (TEE_OperationHandle *operation*, const TEE_Attribute * *params*, uint32_t *paramCount*, const void * *digest*, uint32_t *digestLen*, const void * *signature*, uint32_t *signatureLen*)

Crypto, Asymmetric key Verification Functions.

Verifies a message digest signature within an asymmetric key operation.
Keystone has ed25519_verify().
Equivalent in openssl is EVP_DigestVerify().

14.4.2.11 void TEE_CloseObject (TEE_ObjectHandle *object*)

Core Functions, Secure Storage Functions (data is isolated for each TA)

Destroy object (key, key-pair or Data).

14.4.2.12 TEE_Result TEE_CreatePersistentObject (uint32_t *storageID*, const void * *objectID*, uint32_t *objectIDLen*, uint32_t *flags*, TEE_ObjectHandle *attributes*, const void * *initialData*, uint32_t *initialDataLen*, TEE_ObjectHandle * *object*)

Core Functions, Secure Storage Functions (data is isolated for each TA)

Create persistent object (key, key-pair or Data).
For the people who have not written code on GP then probably do not need to care the meaning of what is Persistent Object is, since the following are enough to use secure storage feature.

14.4.2.13 TEE_Result TEE_DigestDoFinal (TEE_OperationHandle *operation*, const void * *chunk*, uint32_t *chunkLen*, void * *hash*, uint32_t * *hashLen*)

Function accumulates message data for hashing.

14.4.2.14 void TEE_DigestUpdate (TEE_OperationHandle *operation*, const void * *chunk*, uint32_t *chunkSize*)

Crypto, Message Digest Functions.

Function accumulates message data for hashing.

14.4.2.15 void TEE_FreeOperation (TEE_OperationHandle *operation*)

Crypto, for all Crypto Functions.

All Crypto Functions use TEE_OperationHandle* operation instances.
Destroy Crypto instance.

14.4.2.16 void TEE_FreeTransientObject (TEE_ObjectHandle *object*)

Crypto, Asymmetric key Verification Functions.

Destroy object storing asymmetric key.

14.4.2.17 void TEE_GenerateRandom (void * *randomBuffer,* uint32_t *randomBufferLen*)

Crypto, common.

Random Data Generation Function. The quality of the random is implementation dependent.
I am not sure this should be in Keystone or not, but it is very handy.
Good to have adding a way to check the quality of the random implementation.

14.4.2.18 TEE_Result TEE_GetObjectInfo1 (TEE_ObjectHandle *object,* TEE_ObjectInfo * *objectInfo*)

Core Functions, Secure Storage Functions (data is isolated for each TA)

Get length of object required before reading the object.

14.4.2.19 void TEE_GetREETime (TEE_Time * *time*)

Core Functions, Time Functions.

Wall clock time of host OS, expressed in the number of seconds since 1970-01-01 UTC.
This could be implemented on Keystone using ocall.

14.4.2.20 void TEE_GetSystemTime (TEE_Time * *time*)

Core Functions, Time Functions.

Time of TEE-controlled secure timer or Host OS time, implementation dependent.

14.4.2.21 void TEE_InitRefAttribute (TEE_Attribute * *attr,* uint32_t *attributeID,* const void * *buffer,* uint32_t *length*)

Crypto, Asymmetric key Verification Functions.

Storing asymmetric key.

14.4.2.22 TEE_Result TEE_OpenPersistentObject (uint32_t *storageID,* const void * *objectID,* uint32_t *objectIDLen,* uint32_t *flags,* TEE_ObjectHandle * *object*)

Core Functions, Secure Storage Functions (data is isolated for each TA)

Open persistent object.

14.4.2.23 TEE_Result TEE_ReadObjectData (TEE_ObjectHandle *object*, void * *buffer*, uint32_t *size*, uint32_t * *count*)

Core Functions, Secure Storage Functions (data is isolated for each TA)

Read object.

14.4.2.24 TEE_Result TEE_WriteObjectData (TEE_ObjectHandle *object*, const void * *buffer*, uint32_t *size*)

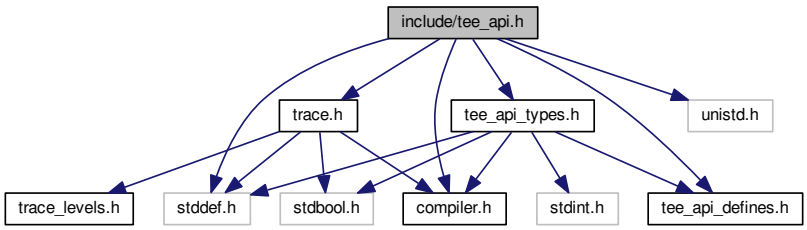
Core Functions, Secure Storage Functions (data is isolated for each TA)

Write object.

14.5 include/tee_api.h File Reference

```
#include <stddef.h>
#include <compiler.h>
#include <tee_api_defines.h>
#include <tee_api_types.h>
#include <trace.h>
#include <unistd.h>
```

Include dependency graph for tee_api.h:



Functions

- TEE_Result TEE_GetPropertyAsString (TEE_PropSetHandle propsetOrEnumerator, const char *name, char *valueBuffer, uint32_t *valueBufferLen)
- TEE_Result TEE_GetPropertyAsBool (TEE_PropSetHandle propsetOrEnumerator, const char *name, bool *value)
- TEE_Result TEE_GetPropertyAsU32 (TEE_PropSetHandle propsetOrEnumerator, const char *name, uint32_t *value)
- TEE_Result TEE_GetPropertyAsBinaryBlock (TEE_PropSetHandle propsetOrEnumerator, const char *name, void *valueBuffer, uint32_t *valueBufferLen)
- TEE_Result TEE_GetPropertyAsUUID (TEE_PropSetHandle propsetOrEnumerator, const char *name, TEE_UUID *value)
- TEE_Result TEE_GetPropertyAsIdentity (TEE_PropSetHandle propsetOrEnumerator, const char *name, TEE_Identity *value)
- TEE_Result TEE_AllocatePropertyEnumerator (TEE_PropSetHandle *enumerator)
- void TEE_FreePropertyEnumerator (TEE_PropSetHandle enumerator)
- void TEE_StartPropertyEnumerator (TEE_PropSetHandle enumerator, TEE_PropSetHandle propSet)
- void TEE_ResetPropertyEnumerator (TEE_PropSetHandle enumerator)

- void TEE_ResetPersistentObjectEnumerator (TEE_ObjectEnumHandle objectEnumerator)
- TEE_Result TEE_StartPersistentObjectEnumerator (TEE_ObjectEnumHandle objectEnumerator, uint32_t storageID)
- TEE_Result TEE_GetNextPersistentObject (TEE_ObjectEnumHandle objectEnumerator, TEE_ObjectInfo *objectInfo, void *objectID, uint32_t *objectIDLen)
- TEE_Result TEE_ReadObjectData (TEE_ObjectHandle object, void *buffer, uint32_t size, uint32_t *count)
- TEE_Result TEE_WriteObjectData (TEE_ObjectHandle object, const void *buffer, uint32_t size)
- TEE_Result TEE_TruncateObjectData (TEE_ObjectHandle object, uint32_t size)
- TEE_Result TEE_SeekObjectData (TEE_ObjectHandle object, int32_t offset, TEE_Whence whence)
- TEE_Result OPTEE_DataMountReq (uint32_t req)
- TEE_Result TEE_AllocateOperation (TEE_OperationHandle *operation, uint32_t algorithm, uint32_t mode, uint32_t maxKeySize)
- void TEE_FreeOperation (TEE_OperationHandle operation)
- void TEE_GetOperationInfo (TEE_OperationHandle operation, TEE_OperationInfo *operationInfo)
- TEE_Result TEE_GetOperationInfoMultiple (TEE_OperationHandle operation, TEE_OperationInfoMultiple *operationInfoMultiple, uint32_t *operationSize)
- void TEE_ResetOperation (TEE_OperationHandle operation)
- TEE_Result TEE_SetOperationKey (TEE_OperationHandle operation, TEE_ObjectHandle key)
- TEE_Result TEE_SetOperationKey2 (TEE_OperationHandle operation, TEE_ObjectHandle key1, TEE_ObjectHandle key2)
- void TEE_CopyOperation (TEE_OperationHandle dstOperation, TEE_OperationHandle srcOperation)
- void TEE_DigestUpdate (TEE_OperationHandle operation, const void *chunk, uint32_t chunkSize)
- TEE_Result TEE_DigestDoFinal (TEE_OperationHandle operation, const void *chunk, uint32_t chunkLen, void *hash, uint32_t *hashLen)
- void TEE_CipherInit (TEE_OperationHandle operation, const void *IV, uint32_t IVLen)
- TEE_Result TEE_CipherUpdate (TEE_OperationHandle operation, const void *srcData, uint32_t srcLen, void *destData, uint32_t *destLen)
- TEE_Result TEE_CipherDoFinal (TEE_OperationHandle operation, const void *srcData, uint32_t srcLen, void *destData, uint32_t *destLen)
- void TEE_MACInit (TEE_OperationHandle operation, const void *IV, uint32_t IVLen)
- void TEE_MACUpdate (TEE_OperationHandle operation, const void *chunk, uint32_t chunkSize)
- TEE_Result TEE_MACComputeFinal (TEE_OperationHandle operation, const void *message, uint32_t messageLen, void *mac, uint32_t *macLen)
- TEE_Result TEE_MACCompareFinal (TEE_OperationHandle operation, const void *message, uint32_t messageLen, const void *mac, uint32_t macLen)
- TEE_Result TEE_AEInit (TEE_OperationHandle operation, const void *nonce, uint32_t nonceLen, uint32_t tagLen, uint32_t AADLen, uint32_t payloadLen)
- void TEE_AEUpdateAAD (TEE_OperationHandle operation, const void *AADdata, uint32_t AADdataLen)
- TEE_Result TEE_AEUpdate (TEE_OperationHandle operation, const void *srcData, uint32_t srcLen, void *destData, uint32_t *destLen)
- TEE_Result TEE_AEEncryptFinal (TEE_OperationHandle operation, const void *srcData, uint32_t srcLen, void *destData, uint32_t *destLen, void *tag, uint32_t *tagLen)
- TEE_Result TEE_AEDecryptFinal (TEE_OperationHandle operation, const void *srcData, uint32_t srcLen, void *destData, uint32_t *destLen, void *tag, uint32_t tagLen)
- TEE_Result TEE_AsymmetricEncrypt (TEE_OperationHandle operation, const TEE_Attribute *params, uint32_t paramCount, const void *srcData, uint32_t srcLen, void *destData, uint32_t *destLen)
- TEE_Result TEE_AsymmetricDecrypt (TEE_OperationHandle operation, const TEE_Attribute *params, uint32_t paramCount, const void *srcData, uint32_t srcLen, void *destData, uint32_t *destLen)
- TEE_Result TEE_AsymmetricSignDigest (TEE_OperationHandle operation, const TEE_Attribute *params, uint32_t paramCount, const void *digest, uint32_t digestLen, void *signature, uint32_t *signatureLen)
- TEE_Result TEE_AsymmetricVerifyDigest (TEE_OperationHandle operation, const TEE_Attribute *params, uint32_t paramCount, const void *digest, uint32_t digestLen, const void *signature, uint32_t signatureLen)
- void TEE_DeriveKey (TEE_OperationHandle operation, const TEE_Attribute *params, uint32_t paramCount, TEE_ObjectHandle derivedKey)
- void TEE_GenerateRandom (void *randomBuffer, uint32_t randomBufferLen)
- void TEE_GetSystemTime (TEE_Time *time)

- TEE_Result TEE_AtbRpcRecv (int sockfd, void ∗buf, size_t len, int flags, ssize_t ∗retcode)
- TEE_Result TEE_AtbRpcGetsockopt (int sockfd, int level, int optname, void ∗optval, socklen_t ∗optlen, int ∗retcode)
- TEE_Result TEE_AtbRpcSetsockopt (int sockfd, int level, int optname, const void ∗optval, socklen_t optlen, int ∗retcode)
- TEE_Result TEE_AtbRpcPoll (struct pollfd ∗fds, nfds_t nfds, int timeout, int ∗retcode)
- TEE_Result TEE_AtbRpcImagedata (void ∗buf, int len, int ∗retcode)
- TEE_Result TEE_AtbRpcClose (int sockfd, int ∗retcode)
- TEE_Result TEE_Oneway (unsigned long mask_on, unsigned long ∗result)
- TEE_Result TEE_AtbRpcPhysmem (uint64_t pa, void ∗buf, size_t len)
- TEE_Result TEE_AtbRpcSpinlock (uint64_t pa, char acquire)
- TEE_Result TEE_AtbRpcRwlock (uint64_t pa, char acquire)
- TEE_Result TEE_AtbRpcWatchdog (uint64_t ms)
- TEE_Result TEE_AtbRpcResetReason (uint64_t job, uint8_t ∗buf, int ∗plen)

14.5.1 Function Documentation

14.5.1.1 TEE_Result OPTEE_DataMountReq (uint32_t *req*)

14.5.1.2 TEE_Result TEE_AEDecryptFinal (TEE_OperationHandle *operation*, const void ∗ *srcData*, uint32_t *srcLen*, void ∗ *destData*, uint32_t ∗ *destLen*, void ∗ *tag*, uint32_t *tagLen*)

14.5.1.3 TEE_Result TEE_AEEncryptFinal (TEE_OperationHandle *operation*, const void ∗ *srcData*, uint32_t *srcLen*, void ∗ *destData*, uint32_t ∗ *destLen*, void ∗ *tag*, uint32_t ∗ *tagLen*)

14.5.1.4 TEE_Result TEE_AEInit (TEE_OperationHandle *operation*, const void ∗ *nonce*, uint32_t *nonceLen*, uint32_t *tagLen*, uint32_t *AADLen*, uint32_t *payloadLen*)

14.5.1.5 TEE_Result TEE_AEUpdate (TEE_OperationHandle *operation*, const void ∗ *srcData*, uint32_t *srcLen*, void ∗ *destData*, uint32_t ∗ *destLen*)

14.5.1.6 void TEE_AEUpdateAAD (TEE_OperationHandle *operation*, const void ∗ *AADdata*, uint32_t *AADdataLen*)

14.5.1.7 TEE_Result TEE_AllocateOperation (TEE_OperationHandle ∗ *operation*, uint32_t *algorithm*, uint32_t *mode*, uint32_t *maxKeySize*)

14.5.1.8 TEE_Result TEE_AllocatePersistentObjectEnumerator (TEE_ObjectEnumHandle ∗ *objectEnumerator*)

14.5.1.9 TEE_Result TEE_AllocatePropertyEnumerator (TEE_PropSetHandle ∗ *enumerator*)

14.5.1.10 TEE_Result TEE_AllocateTransientObject (TEE_ObjectType *objectType*, uint32_t *maxKeySize*, TEE_ObjectHandle ∗ *object*)

14.5.1.11 TEE_Result TEE_AsymmetricDecrypt (TEE_OperationHandle *operation*, const TEE_Attribute ∗ *params*, uint32_t *paramCount*, const void ∗ *srcData*, uint32_t *srcLen*, void ∗ *destData*, uint32_t ∗ *destLen*)

14.5.1.12 TEE_Result TEE_AsymmetricEncrypt (TEE_OperationHandle *operation*, const TEE_Attribute ∗ *params*, uint32_t *paramCount*, const void ∗ *srcData*, uint32_t *srcLen*, void ∗ *destData*, uint32_t ∗ *destLen*)

14.5.1.13 TEE_Result TEE_AsymmetricSignDigest (TEE_OperationHandle *operation*, const TEE_Attribute ∗ *params*, uint32_t *paramCount*, const void ∗ *digest*, uint32_t *digestLen*, void ∗ *signature*, uint32_t ∗ *signatureLen*)

14.5.1.14 **TEE_Result** TEE_AsymmetricVerifyDigest (**TEE_OperationHandle** *operation,* const **TEE_Attribute** ∗ *params,* uint32_t *paramCount,* const void ∗ *digest,* uint32_t *digestLen,* const void ∗ *signature,* uint32_t *signatureLen*)

14.5.1.15 **TEE_Result** TEE_AtbRpcBind (int *sockfd,* const struct sockaddr ∗ *addr,* **socklen_t** *addrlen,* int ∗ *retcode*)

14.5.1.16 **TEE_Result** TEE_AtbRpcClose (int *sockfd,* int ∗ *retcode*)

14.5.1.17 **TEE_Result** TEE_AtbRpcConnect (int *sockfd,* const struct sockaddr ∗ *addr,* **socklen_t** *addrlen,* int ∗ *retcode*)

14.5.1.18 **TEE_Result** TEE_AtbRpcFreeaddrinfo (struct **addrinfo** ∗ *res*)

14.5.1.19 **TEE_Result** TEE_AtbRpcGetaddrinfo (const char ∗ *node,* const char ∗ *service,* const struct **addrinfo** ∗ *hints,* struct **addrinfo** ∗∗ *res,* int ∗ *retcode*)

14.5.1.20 **TEE_Result** TEE_AtbRpcGetsockopt (int *sockfd,* int *level,* int *optname,* void ∗ *optval,* **socklen_t** ∗ *optlen,* int ∗ *retcode*)

14.5.1.21 **TEE_Result** TEE_AtbRpcImagedata (void ∗ *buf,* int *len,* int ∗ *retcode*)

14.5.1.22 **TEE_Result** TEE_AtbRpcPhysmem (uint64_t *pa,* void ∗ *buf,* size_t *len*)

14.5.1.23 **TEE_Result** TEE_AtbRpcPoll (struct **pollfd** ∗ *fds,* **nfds_t** *nfds,* int *timeout,* int ∗ *retcode*)

14.5.1.24 **TEE_Result** TEE_AtbRpcRecv (int *sockfd,* void ∗ *buf,* size_t *len,* int *flags,* ssize_t ∗ *retcode*)

14.5.1.25 **TEE_Result** TEE_AtbRpcResetReason (uint64_t *job,* uint8_t ∗ *buf,* int ∗ *plen*)

14.5.1.26 **TEE_Result** TEE_AtbRpcRwlock (uint64_t *pa,* char *acquire*)

14.5.1.27 **TEE_Result** TEE_AtbRpcSend (int *sockfd,* const void ∗ *buf,* size_t *len,* int *flags,* ssize_t ∗ *retcode*)

14.5.1.28 **TEE_Result** TEE_AtbRpcSetsockopt (int *sockfd,* int *level,* int *optname,* const void ∗ *optval,* **socklen_t** *optlen,* int ∗ *retcode*)

14.5.1.29 **TEE_Result** TEE_AtbRpcSocket (int ∗ *sockfd,* unsigned int *args*)

14.5.1.30 **TEE_Result** TEE_AtbRpcSpinlock (uint64_t *pa,* char *acquire*)

14.5.1.31 **TEE_Result** TEE_AtbRpcWatchdog (uint64_t *ms*)

14.5.1.32 void TEE_BigIntAdd (**TEE_BigInt** ∗ *dest,* const **TEE_BigInt** ∗ *op1,* const **TEE_BigInt** ∗ *op2*)

14.5.1.33 void TEE_BigIntAddMod (**TEE_BigInt** ∗ *dest,* const **TEE_BigInt** ∗ *op1,* const **TEE_BigInt** ∗ *op2,* const **TEE_BigInt** ∗ *n*)

14.5.1.34 int32_t TEE_BigIntCmp (const **TEE_BigInt** ∗ *op1,* const **TEE_BigInt** ∗ *op2*)

14.5.1.35 int32_t TEE_BigIntCmpS32 (const **TEE_BigInt** ∗ *op,* int32_t *shortVal*)

14.5.1.36 void TEE_BigIntComputeExtendedGcd (TEE_BigInt * *gcd,* TEE_BigInt * *u,* TEE_BigInt * *v,* const TEE_BigInt * *op1,* const TEE_BigInt * *op2*)

14.5.1.37 void TEE_BigIntComputeFMM (TEE_BigIntFMM * *dest,* const TEE_BigIntFMM * *op1,* const TEE_BigIntFMM * *op2,* const TEE_BigInt * *n,* const TEE_BigIntFMMContext * *context*)

14.5.1.38 void TEE_BigIntConvertFromFMM (TEE_BigInt * *dest,* const TEE_BigIntFMM * *src,* const TEE_BigInt * *n,* const TEE_BigIntFMMContext * *context*)

14.5.1.39 TEE_Result TEE_BigIntConvertFromOctetString (TEE_BigInt * *dest,* const uint8_t * *buffer,* uint32_t *bufferLen,* int32_t *sign*)

14.5.1.40 void TEE_BigIntConvertFromS32 (TEE_BigInt * *dest,* int32_t *shortVal*)

14.5.1.41 void TEE_BigIntConvertToFMM (TEE_BigIntFMM * *dest,* const TEE_BigInt * *src,* const TEE_BigInt * *n,* const TEE_BigIntFMMContext * *context*)

14.5.1.42 TEE_Result TEE_BigIntConvertToOctetString (uint8_t * *buffer,* uint32_t * *bufferLen,* const TEE_BigInt * *bigInt*)

14.5.1.43 TEE_Result TEE_BigIntConvertToS32 (int32_t * *dest,* const TEE_BigInt * *src*)

14.5.1.44 void TEE_BigIntDiv (TEE_BigInt * *dest_q,* TEE_BigInt * *dest_r,* const TEE_BigInt * *op1,* const TEE_BigInt * *op2*)

14.5.1.45 uint32_t TEE_BigIntFMMContextSizeInU32 (uint32_t *modulusSizeInBits*)

14.5.1.46 void TEE_BigIntFMMConvertToBigInt (TEE_BigInt * *dest,* const TEE_BigIntFMM * *src,* const TEE_BigInt * *n,* const TEE_BigIntFMMContext * *context*)

14.5.1.47 uint32_t TEE_BigIntFMMSizeInU32 (uint32_t *modulusSizeInBits*)

14.5.1.48 bool TEE_BigIntGetBit (const TEE_BigInt * *src,* uint32_t *bitIndex*)

14.5.1.49 uint32_t TEE_BigIntGetBitCount (const TEE_BigInt * *src*)

14.5.1.50 void TEE_BigIntInit (TEE_BigInt * *bigInt,* uint32_t *len*)

14.5.1.51 void TEE_BigIntInitFMM (TEE_BigIntFMM * *bigIntFMM,* uint32_t *len*)

14.5.1.52 void TEE_BigIntInitFMMContext (TEE_BigIntFMMContext * *context,* uint32_t *len,* const TEE_BigInt * *modulus*)

14.5.1.53 void TEE_BigIntInvMod (TEE_BigInt * *dest,* const TEE_BigInt * *op,* const TEE_BigInt * *n*)

14.5.1.54 int32_t TEE_BigIntIsProbablePrime (const TEE_BigInt * *op,* uint32_t *confidenceLevel*)

14.5.1.55 void TEE_BigIntMod (TEE_BigInt * *dest,* const TEE_BigInt * *op,* const TEE_BigInt * *n*)

14.5.1.56 void TEE_BigIntMul (TEE_BigInt * *dest,* const TEE_BigInt * *op1,* const TEE_BigInt * *op2*)

14.5.1.57 void TEE_BigIntMulMod (TEE_BigInt * *dest,* const TEE_BigInt * *op1,* const TEE_BigInt * *op2,* const TEE_BigInt * *n*)

14.5.1.58 void TEE_BigIntNeg (TEE_BigInt * *dest,* const TEE_BigInt * *op*)

14.5.1.59 bool TEE_BigIntRelativePrime (const TEE_BigInt * *op1,* const TEE_BigInt * *op2*)

14.5.1.60 void TEE_BigIntShiftRight (TEE_BigInt * *dest,* const TEE_BigInt * *op,* size_t *bits*)

14.5.1.61 void TEE_BigIntSquare (TEE_BigInt * *dest,* const TEE_BigInt * *op*)

14.5.1.62 void TEE_BigIntSquareMod (TEE_BigInt * *dest,* const TEE_BigInt * *op,* const TEE_BigInt * *n*)

14.5.1.63 void TEE_BigIntSub (TEE_BigInt * *dest,* const TEE_BigInt * *op1,* const TEE_BigInt * *op2*)

14.5.1.64 void TEE_BigIntSubMod (TEE_BigInt * *dest,* const TEE_BigInt * *op1,* const TEE_BigInt * *op2,* const TEE_BigInt * *n*)

14.5.1.65 TEE_Result TEE_CheckMemoryAccessRights (uint32_t *accessFlags,* void * *buffer,* uint32_t *size*)

14.5.1.66 TEE_Result TEE_CipherDoFinal (TEE_OperationHandle *operation,* const void * *srcData,* uint32_t *srcLen,* void * *destData,* uint32_t * *destLen*)

14.5.1.67 void TEE_CipherInit (TEE_OperationHandle *operation,* const void * *IV,* uint32_t *IVLen*)

14.5.1.68 TEE_Result TEE_CipherUpdate (TEE_OperationHandle *operation,* const void * *srcData,* uint32_t *srcLen,* void * *destData,* uint32_t * *destLen*)

14.5.1.69 void TEE_CloseAndDeletePersistentObject (TEE_ObjectHandle *object*)

14.5.1.70 TEE_Result TEE_CloseAndDeletePersistentObject1 (TEE_ObjectHandle *object*)

14.5.1.71 void TEE_CloseObject (TEE_ObjectHandle *object*)

14.5.1.72 void TEE_CloseTASession (TEE_TASessionHandle *session*)

14.5.1.73 void TEE_CopyObjectAttributes (TEE_ObjectHandle *destObject,* TEE_ObjectHandle *srcObject*)

14.5.1.74 TEE_Result TEE_CopyObjectAttributes1 (TEE_ObjectHandle *destObject,* TEE_ObjectHandle *srcObject*)

14.5.1.75 void TEE_CopyOperation (TEE_OperationHandle *dstOperation,* TEE_OperationHandle *srcOperation*)

14.5.1.76 TEE_Result TEE_CreatePersistentObject (uint32_t *storageID,* const void * *objectID,* uint32_t *objectIDLen,* uint32_t *flags,* TEE_ObjectHandle *attributes,* const void * *initialData,* uint32_t *initialDataLen,* TEE_ObjectHandle * *object*)

14.5.1.77 void TEE_DeriveKey (TEE_OperationHandle *operation,* const TEE_Attribute * *params,* uint32_t *paramCount,* TEE_ObjectHandle *derivedKey*)

14.5.1.78 **TEE_Result** TEE_DigestDoFinal (**TEE_OperationHandle** *operation,* const void * *chunk,* uint32_t *chunkLen,* void * *hash,* uint32_t * *hashLen*)

14.5.1.79 void TEE_DigestUpdate (**TEE_OperationHandle** *operation,* const void * *chunk,* uint32_t *chunkSize*)

14.5.1.80 void TEE_Free (void * *buffer*)

14.5.1.81 void TEE_FreeOperation (**TEE_OperationHandle** *operation*)

14.5.1.82 void TEE_FreePersistentObjectEnumerator (**TEE_ObjectEnumHandle** *objectEnumerator*)

14.5.1.83 void TEE_FreePropertyEnumerator (**TEE_PropSetHandle** *enumerator*)

14.5.1.84 void TEE_FreeTransientObject (**TEE_ObjectHandle** *object*)

14.5.1.85 **TEE_Result** TEE_GenerateKey (**TEE_ObjectHandle** *object,* uint32_t *keySize,* const **TEE_Attribute** * *params,* uint32_t *paramCount*)

14.5.1.86 void TEE_GenerateRandom (void * *randomBuffer,* uint32_t *randomBufferLen*)

14.5.1.87 bool TEE_GetCancellationFlag (void)

14.5.1.88 const void* TEE_GetInstanceData (void)

14.5.1.89 **TEE_Result** TEE_GetNextPersistentObject (**TEE_ObjectEnumHandle** *objectEnumerator,* **TEE_ObjectInfo** * *objectInfo,* void * *objectID,* uint32_t * *objectIDLen*)

14.5.1.90 **TEE_Result** TEE_GetNextProperty (**TEE_PropSetHandle** *enumerator*)

14.5.1.91 **TEE_Result** TEE_GetObjectBufferAttribute (**TEE_ObjectHandle** *object,* uint32_t *attributeID,* void * *buffer,* uint32_t * *size*)

14.5.1.92 void TEE_GetObjectInfo (**TEE_ObjectHandle** *object,* **TEE_ObjectInfo** * *objectInfo*)

14.5.1.93 **TEE_Result** TEE_GetObjectInfo1 (**TEE_ObjectHandle** *object,* **TEE_ObjectInfo** * *objectInfo*)

14.5.1.94 **TEE_Result** TEE_GetObjectValueAttribute (**TEE_ObjectHandle** *object,* uint32_t *attributeID,* uint32_t * *a,* uint32_t * *b*)

14.5.1.95 void TEE_GetOperationInfo (**TEE_OperationHandle** *operation,* **TEE_OperationInfo** * *operationInfo*)

14.5.1.96 **TEE_Result** TEE_GetOperationInfoMultiple (**TEE_OperationHandle** *operation,* **TEE_OperationInfoMultiple** * *operationInfoMultiple,* uint32_t * *operationSize*)

14.5.1.97 **TEE_Result** TEE_GetPropertyAsBinaryBlock (**TEE_PropSetHandle** *propsetOrEnumerator,* const char * *name,* void * *valueBuffer,* uint32_t * *valueBufferLen*)

14.5.1.98 **TEE_Result** TEE_GetPropertyAsBool (**TEE_PropSetHandle** *propsetOrEnumerator,* const char * *name,* bool * *value*)

14.5.1.99 **TEE_Result** TEE_GetPropertyAsIdentity (**TEE_PropSetHandle** *propsetOrEnumerator,* const char * *name,* **TEE_Identity** * *value*)

14.5.1.100 **TEE_Result** TEE_GetPropertyAsString (**TEE_PropSetHandle** *propsetOrEnumerator,* const char * *name,* char * *valueBuffer,* uint32_t * *valueBufferLen*)

14.5.1.101 **TEE_Result** TEE_GetPropertyAsU32 (**TEE_PropSetHandle** *propsetOrEnumerator,* const char * *name,* uint32_t * *value*)

14.5.1.102 **TEE_Result** TEE_GetPropertyAsUUID (**TEE_PropSetHandle** *propsetOrEnumerator,* const char * *name,* **TEE_UUID** * *value*)

14.5.1.103 **TEE_Result** TEE_GetPropertyName (**TEE_PropSetHandle** *enumerator,* void * *nameBuffer,* uint32_t * *nameBufferLen*)

14.5.1.104 void TEE_GetREETime (**TEE_Time** * *time*)

14.5.1.105 void TEE_GetSystemTime (**TEE_Time** * *time*)

14.5.1.106 **TEE_Result** TEE_GetTAPersistentTime (**TEE_Time** * *time*)

14.5.1.107 void TEE_InitRefAttribute (**TEE_Attribute** * *attr,* uint32_t *attributeID,* const void * *buffer,* uint32_t *length*)

14.5.1.108 void TEE_InitValueAttribute (**TEE_Attribute** * *attr,* uint32_t *attributeID,* uint32_t *a,* uint32_t *b*)

14.5.1.109 **TEE_Result** TEE_InvokeTACommand (**TEE_TASessionHandle** *session,* uint32_t *cancellationRequest*↩ *Timeout,* uint32_t *commandID,* uint32_t *paramTypes,* **TEE_Param** *params[TEE_NUM_PARAMS],* uint32_t * *returnOrigin*)

14.5.1.110 **TEE_Result** TEE_MACCompareFinal (**TEE_OperationHandle** *operation,* const void * *message,* uint32_t *messageLen,* const void * *mac,* uint32_t *macLen*)

14.5.1.111 **TEE_Result** TEE_MACComputeFinal (**TEE_OperationHandle** *operation,* const void * *message,* uint32_t *messageLen,* void * *mac,* uint32_t * *macLen*)

14.5.1.112 void TEE_MACInit (**TEE_OperationHandle** *operation,* const void * *IV,* uint32_t *IVLen*)

14.5.1.113 void TEE_MACUpdate (**TEE_OperationHandle** *operation,* const void * *chunk,* uint32_t *chunkSize*)

14.5.1.114 void* TEE_Malloc (uint32_t *size,* uint32_t *hint*)

14.5.1.115 bool TEE_MaskCancellation (void)

14.5.1.116 int32_t TEE_MemCompare (const void * *buffer1,* const void * *buffer2,* uint32_t *size*)

14.5.1.117 void* TEE_MemFill (void * *buff,* uint32_t *x,* uint32_t *size*)

14.5.1.118 void* TEE_MemMove (void * *dest,* const void * *src,* uint32_t *size*)

14.5.1.119 **TEE_Result** TEE_Oneway (unsigned long *mask_on,* unsigned long * *result*)

14.5.1.120 **TEE_Result** TEE_OpenPersistentObject (uint32_t *storageID,* const void * *objectID,* uint32_t *objectIDLen,* uint32_t *flags,* **TEE_ObjectHandle** * *object*)

14.5.1.121 **TEE_Result** TEE_OpenTASession (const **TEE_UUID** * *destination,* uint32_t *cancellationRequestTimeout,* uint32_t *paramTypes,* **TEE_Param** *params[TEE_NUM_PARAMS],* **TEE_TASessionHandle** * *session,* uint32_t * *returnOrigin*)

14.5.1.122 void TEE_Panic (**TEE_Result** *panicCode*)

14.5.1.123 **TEE_Result** TEE_PopulateTransientObject (**TEE_ObjectHandle** *object,* const **TEE_Attribute** * *attrs,* uint32_t *attrCount*)

14.5.1.124 **TEE_Result** TEE_ReadObjectData (**TEE_ObjectHandle** *object,* void * *buffer,* uint32_t *size,* uint32_t * *count*)

14.5.1.125 void* TEE_Realloc (const void * *buffer,* uint32_t *newSize*)

14.5.1.126 **TEE_Result** TEE_RenamePersistentObject (**TEE_ObjectHandle** *object,* const void * *newObjectID,* uint32_t *newObjectIDLen*)

14.5.1.127 void TEE_ResetOperation (**TEE_OperationHandle** *operation*)

14.5.1.128 void TEE_ResetPersistentObjectEnumerator (**TEE_ObjectEnumHandle** *objectEnumerator*)

14.5.1.129 void TEE_ResetPropertyEnumerator (**TEE_PropSetHandle** *enumerator*)

14.5.1.130 void TEE_ResetTransientObject (**TEE_ObjectHandle** *object*)

14.5.1.131 void TEE_RestrictObjectUsage (**TEE_ObjectHandle** *object,* uint32_t *objectUsage*)

14.5.1.132 **TEE_Result** TEE_RestrictObjectUsage1 (**TEE_ObjectHandle** *object,* uint32_t *objectUsage*)

14.5.1.133 **TEE_Result** TEE_SeekObjectData (**TEE_ObjectHandle** *object,* int32_t *offset,* **TEE_Whence** *whence*)

14.5.1.134 void TEE_SetInstanceData (const void * *instanceData*)

14.5.1.135 **TEE_Result** TEE_SetOperationKey (**TEE_OperationHandle** *operation,* **TEE_ObjectHandle** *key*)

14.5.1.136 **TEE_Result** TEE_SetOperationKey2 (**TEE_OperationHandle** *operation,* **TEE_ObjectHandle** *key1,* **TEE_ObjectHandle** *key2*)

14.5.1.137 **TEE_Result** TEE_SetTAPersistentTime (const **TEE_Time** * *time*)

14.5.1.138 **TEE_Result** TEE_StartPersistentObjectEnumerator (**TEE_ObjectEnumHandle** *objectEnumerator,* uint32_t *storageID*)

14.5.1.139 void TEE_StartPropertyEnumerator (**TEE_PropSetHandle** *enumerator,* **TEE_PropSetHandle** *propSet*)

14.5.1.140 **TEE_Result** TEE_TruncateObjectData (**TEE_ObjectHandle** *object,* uint32_t *size*)

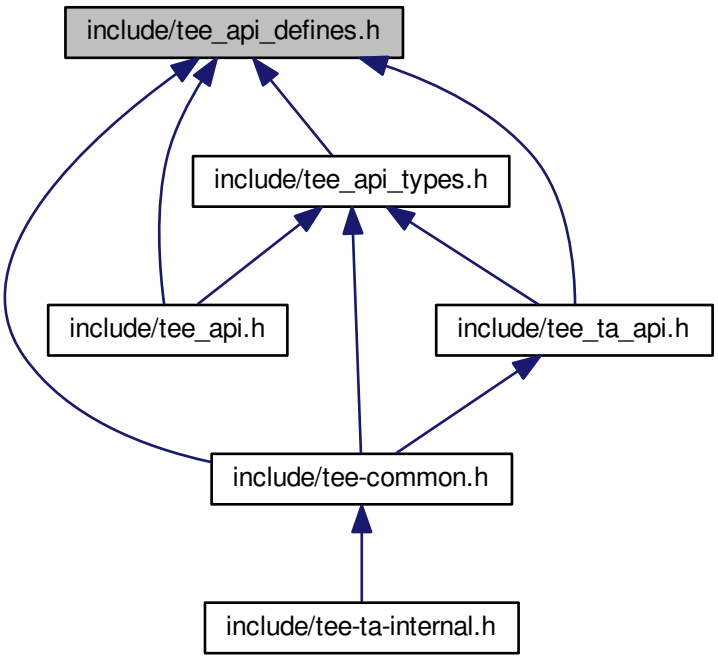
14.5.1.141 bool TEE_UnmaskCancellation (void)

14.5.1.142 TEE_Result TEE_Wait (uint32_t *timeout*)

14.5.1.143 TEE_Result TEE_WriteObjectData (TEE_ObjectHandle *object*, const void * *buffer*, uint32_t *size*)

14.6 include/tee_api_defines.h File Reference

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



Macros

- #define TEE_INT_CORE_API_SPEC_VERSION 0x0000000A
- #define TEE_HANDLE_NULL 0
- #define TEE_TIMEOUT_INFINITE 0xFFFFFFFF
- #define TEE_SUCCESS 0x00000000
- #define TEE_ERROR_CORRUPT_OBJECT 0xF0100001
- #define TEE_ERROR_CORRUPT_OBJECT_2 0xF0100002
- #define TEE_ERROR_STORAGE_NOT_AVAILABLE 0xF0100003
- #define TEE_ERROR_STORAGE_NOT_AVAILABLE_2 0xF0100004
- #define TEE_ERROR_GENERIC 0xFFFF0000
- #define TEE_ERROR_ACCESS_DENIED 0xFFFF0001
- #define TEE_ERROR_CANCEL 0xFFFF0002
- #define TEE_ERROR_ACCESS_CONFLICT 0xFFFF0003
- #define TEE_ERROR_EXCESS_DATA 0xFFFF0004
- #define TEE_ERROR_BAD_FORMAT 0xFFFF0005
- #define TEE_ERROR_BAD_PARAMETERS 0xFFFF0006
- #define TEE_ERROR_BAD_STATE 0xFFFF0007

- #define TEE_ERROR_ITEM_NOT_FOUND 0xFFFF0008
- #define TEE_ERROR_NOT_IMPLEMENTED 0xFFFF0009
- #define TEE_ERROR_NOT_SUPPORTED 0xFFFF000A
- #define TEE_ERROR_NO_DATA 0xFFFF000B
- #define TEE_ERROR_OUT_OF_MEMORY 0xFFFF000C
- #define TEE_ERROR_BUSY 0xFFFF000D
- #define TEE_ERROR_COMMUNICATION 0xFFFF000E
- #define TEE_ERROR_SECURITY 0xFFFF000F
- #define TEE_ERROR_SHORT_BUFFER 0xFFFF0010
- #define TEE_ERROR_EXTERNAL_CANCEL 0xFFFF0011
- #define TEE_ERROR_OVERFLOW 0xFFFF300F
- #define TEE_ERROR_TARGET_DEAD 0xFFFF3024
- #define TEE_ERROR_STORAGE_NO_SPACE 0xFFFF3041
- #define TEE_ERROR_MAC_INVALID 0xFFFF3071
- #define TEE_ERROR_SIGNATURE_INVALID 0xFFFF3072
- #define TEE_ERROR_TIME_NOT_SET 0xFFFF5000
- #define TEE_ERROR_TIME_NEEDS_RESET 0xFFFF5001
- #define TEE_PARAM_TYPE_NONE 0
- #define TEE_PARAM_TYPE_VALUE_INPUT 1
- #define TEE_PARAM_TYPE_VALUE_OUTPUT 2
- #define TEE_PARAM_TYPE_VALUE_INOUT 3
- #define TEE_PARAM_TYPE_MEMREF_INPUT 5
- #define TEE_PARAM_TYPE_MEMREF_OUTPUT 6
- #define TEE_PARAM_TYPE_MEMREF_INOUT 7
- #define TEE_LOGIN_PUBLIC 0x00000000
- #define TEE_LOGIN_USER 0x00000001
- #define TEE_LOGIN_GROUP 0x00000002
- #define TEE_LOGIN_APPLICATION 0x00000004
- #define TEE_LOGIN_APPLICATION_USER 0x00000005
- #define TEE_LOGIN_APPLICATION_GROUP 0x00000006
- #define TEE_LOGIN_TRUSTED_APP 0xF0000000
- #define TEE_ORIGIN_API 0x00000001
- #define TEE_ORIGIN_COMMS 0x00000002
- #define TEE_ORIGIN_TEE 0x00000003
- #define TEE_ORIGIN_TRUSTED_APP 0x00000004
- #define TEE_PROPSET_TEE_IMPLEMENTATION (TEE_PropSetHandle)0xFFFFFFFD
- #define TEE_PROPSET_CURRENT_CLIENT (TEE_PropSetHandle)0xFFFFFFFE
- #define TEE_PROPSET_CURRENT_TA (TEE_PropSetHandle)0xFFFFFFFF
- #define TEE_MEMORY_ACCESS_READ 0x00000001
- #define TEE_MEMORY_ACCESS_WRITE 0x00000002
- #define TEE_MEMORY_ACCESS_ANY_OWNER 0x00000004
- #define TEE_MALLOC_FILL_ZERO 0x00000000
- #define TEE_STORAGE_PRIVATE 0x00000001
- #define TEE_DATA_FLAG_ACCESS_READ 0x00000001
- #define TEE_DATA_FLAG_ACCESS_WRITE 0x00000002
- #define TEE_DATA_FLAG_ACCESS_WRITE_META 0x00000004
- #define TEE_DATA_FLAG_SHARE_READ 0x00000010
- #define TEE_DATA_FLAG_SHARE_WRITE 0x00000020
- #define TEE_DATA_FLAG_OVERWRITE 0x00000400
- #define TEE_DATA_MAX_POSITION 0xFFFFFFFF
- #define TEE_OBJECT_ID_MAX_LEN 64
- #define TEE_USAGE_EXTRACTABLE 0x00000001
- #define TEE_USAGE_ENCRYPT 0x00000002
- #define TEE_USAGE_DECRYPT 0x00000004
- #define TEE_USAGE_MAC 0x00000008

- #define TEE_USAGE_SIGN 0x00000010
- #define TEE_USAGE_VERIFY 0x00000020
- #define TEE_USAGE_DERIVE 0x00000040
- #define TEE_HANDLE_FLAG_PERSISTENT 0x00010000
- #define TEE_HANDLE_FLAG_INITIALIZED 0x00020000
- #define TEE_HANDLE_FLAG_KEY_SET 0x00040000
- #define TEE_HANDLE_FLAG_EXPECT_TWO_KEYS 0x00080000
- #define TEE_OPERATION_CIPHER 1
- #define TEE_OPERATION_MAC 3
- #define TEE_OPERATION_AE 4
- #define TEE_OPERATION_DIGEST 5
- #define TEE_OPERATION_ASYMMETRIC_CIPHER 6
- #define TEE_OPERATION_ASYMMETRIC_SIGNATURE 7
- #define TEE_OPERATION_KEY_DERIVATION 8
- #define TEE_OPERATION_STATE_INITIAL 0x00000000
- #define TEE_OPERATION_STATE_ACTIVE 0x00000001
- #define TEE_ALG_AES_ECB_NOPAD 0x10000010
- #define TEE_ALG_AES_CBC_NOPAD 0x10000110
- #define TEE_ALG_AES_CTR 0x10000210
- #define TEE_ALG_AES_CTS 0x10000310
- #define TEE_ALG_AES_XTS 0x10000410
- #define TEE_ALG_AES_CBC_MAC_NOPAD 0x30000110
- #define TEE_ALG_AES_CBC_MAC_PKCS5 0x30000510
- #define TEE_ALG_AES_CMAC 0x30000610
- #define TEE_ALG_AES_CCM 0x40000710
- #define TEE_ALG_AES_GCM 0x40000810
- #define TEE_ALG_DES_ECB_NOPAD 0x10000011
- #define TEE_ALG_DES_CBC_NOPAD 0x10000111
- #define TEE_ALG_DES_CBC_MAC_NOPAD 0x30000111
- #define TEE_ALG_DES_CBC_MAC_PKCS5 0x30000511
- #define TEE_ALG_DES3_ECB_NOPAD 0x10000013
- #define TEE_ALG_DES3_CBC_NOPAD 0x10000113
- #define TEE_ALG_DES3_CBC_MAC_NOPAD 0x30000113
- #define TEE_ALG_DES3_CBC_MAC_PKCS5 0x30000513
- #define TEE_ALG_RSASSA_PKCS1_V1_5_MD5 0x70001830
- #define TEE_ALG_RSASSA_PKCS1_V1_5_SHA1 0x70002830
- #define TEE_ALG_RSASSA_PKCS1_V1_5_SHA224 0x70003830
- #define TEE_ALG_RSASSA_PKCS1_V1_5_SHA256 0x70004830
- #define TEE_ALG_RSASSA_PKCS1_V1_5_SHA384 0x70005830
- #define TEE_ALG_RSASSA_PKCS1_V1_5_SHA512 0x70006830
- #define TEE_ALG_RSASSA_PKCS1_V1_5_MD5SHA1 0x7000F830
- #define TEE_ALG_RSASSA_PKCS1_PSS_MGF1_SHA1 0x70212930
- #define TEE_ALG_RSASSA_PKCS1_PSS_MGF1_SHA224 0x70313930
- #define TEE_ALG_RSASSA_PKCS1_PSS_MGF1_SHA256 0x70414930
- #define TEE_ALG_RSASSA_PKCS1_PSS_MGF1_SHA384 0x70515930
- #define TEE_ALG_RSASSA_PKCS1_PSS_MGF1_SHA512 0x70616930
- #define TEE_ALG_RSAES_PKCS1_V1_5 0x60000130
- #define TEE_ALG_RSAES_PKCS1_OAEP_MGF1_SHA1 0x60210230
- #define TEE_ALG_RSAES_PKCS1_OAEP_MGF1_SHA224 0x60310230
- #define TEE_ALG_RSAES_PKCS1_OAEP_MGF1_SHA256 0x60410230
- #define TEE_ALG_RSAES_PKCS1_OAEP_MGF1_SHA384 0x60510230
- #define TEE_ALG_RSAES_PKCS1_OAEP_MGF1_SHA512 0x60610230
- #define TEE_ALG_RSA_NOPAD 0x60000030
- #define TEE_ALG_DSA_SHA1 0x70002131
- #define TEE_ALG_DSA_SHA224 0x70003131

- #define TEE_ALG_DSA_SHA256 0x70004131
- #define TEE_ALG_DH_DERIVE_SHARED_SECRET 0x80000032
- #define TEE_ALG_MD5 0x50000001
- #define TEE_ALG_SHA1 0x50000002
- #define TEE_ALG_SHA224 0x50000003
- #define TEE_ALG_SHA256 0x50000004
- #define TEE_ALG_SHA384 0x50000005
- #define TEE_ALG_SHA512 0x50000006
- #define TEE_ALG_MD5SHA1 0x5000000F
- #define TEE_ALG_HMAC_MD5 0x30000001
- #define TEE_ALG_HMAC_SHA1 0x30000002
- #define TEE_ALG_HMAC_SHA224 0x30000003
- #define TEE_ALG_HMAC_SHA256 0x30000004
- #define TEE_ALG_HMAC_SHA384 0x30000005
- #define TEE_ALG_HMAC_SHA512 0x30000006
- #define TEE_ALG_ECDSA_P192 0x70001041
- #define TEE_ALG_ECDSA_P224 0x70002041
- #define TEE_ALG_ECDSA_P256 0x70003041
- #define TEE_ALG_ECDSA_P384 0x70004041
- #define TEE_ALG_ECDSA_P521 0x70005041
- #define TEE_ALG_ECDH_P192 0x80001042
- #define TEE_ALG_ECDH_P224 0x80002042
- #define TEE_ALG_ECDH_P256 0x80003042
- #define TEE_ALG_ECDH_P384 0x80004042
- #define TEE_ALG_ECDH_P521 0x80005042
- #define TEE_TYPE_AES 0xA0000010
- #define TEE_TYPE_DES 0xA0000011
- #define TEE_TYPE_DES3 0xA0000013
- #define TEE_TYPE_HMAC_MD5 0xA0000001
- #define TEE_TYPE_HMAC_SHA1 0xA0000002
- #define TEE_TYPE_HMAC_SHA224 0xA0000003
- #define TEE_TYPE_HMAC_SHA256 0xA0000004
- #define TEE_TYPE_HMAC_SHA384 0xA0000005
- #define TEE_TYPE_HMAC_SHA512 0xA0000006
- #define TEE_TYPE_RSA_PUBLIC_KEY 0xA0000030
- #define TEE_TYPE_RSA_KEYPAIR 0xA1000030
- #define TEE_TYPE_DSA_PUBLIC_KEY 0xA0000031
- #define TEE_TYPE_DSA_KEYPAIR 0xA1000031
- #define TEE_TYPE_DH_KEYPAIR 0xA1000032
- #define TEE_TYPE_ECDSA_PUBLIC_KEY 0xA0000041
- #define TEE_TYPE_ECDSA_KEYPAIR 0xA1000041
- #define TEE_TYPE_ECDH_PUBLIC_KEY 0xA0000042
- #define TEE_TYPE_ECDH_KEYPAIR 0xA1000042
- #define TEE_TYPE_GENERIC_SECRET 0xA0000000
- #define TEE_TYPE_CORRUPTED_OBJECT 0xA00000BE
- #define TEE_TYPE_DATA 0xA00000BF
- #define TEE_ATTR_SECRET_VALUE 0xC0000000
- #define TEE_ATTR_RSA_MODULUS 0xD0000130
- #define TEE_ATTR_RSA_PUBLIC_EXPONENT 0xD0000230
- #define TEE_ATTR_RSA_PRIVATE_EXPONENT 0xC0000330
- #define TEE_ATTR_RSA_PRIME1 0xC0000430
- #define TEE_ATTR_RSA_PRIME2 0xC0000530
- #define TEE_ATTR_RSA_EXPONENT1 0xC0000630
- #define TEE_ATTR_RSA_EXPONENT2 0xC0000730
- #define TEE_ATTR_RSA_COEFFICIENT 0xC0000830

- #define TEE_ATTR_DSA_PRIME 0xD0001031
- #define TEE_ATTR_DSA_SUBPRIME 0xD0001131
- #define TEE_ATTR_DSA_BASE 0xD0001231
- #define TEE_ATTR_DSA_PUBLIC_VALUE 0xD0000131
- #define TEE_ATTR_DSA_PRIVATE_VALUE 0xC0000231
- #define TEE_ATTR_DH_PRIME 0xD0001032
- #define TEE_ATTR_DH_SUBPRIME 0xD0001132
- #define TEE_ATTR_DH_BASE 0xD0001232
- #define TEE_ATTR_DH_X_BITS 0xF0001332
- #define TEE_ATTR_DH_PUBLIC_VALUE 0xD0000132
- #define TEE_ATTR_DH_PRIVATE_VALUE 0xC0000232
- #define TEE_ATTR_RSA_OAEP_LABEL 0xD0000930
- #define TEE_ATTR_RSA_PSS_SALT_LENGTH 0xF0000A30
- #define TEE_ATTR_ECC_PUBLIC_VALUE_X 0xD0000141
- #define TEE_ATTR_ECC_PUBLIC_VALUE_Y 0xD0000241
- #define TEE_ATTR_ECC_PRIVATE_VALUE 0xC0000341
- #define TEE_ATTR_ECC_CURVE 0xF0000441
- #define TEE_ATTR_BIT_PROTECTED (1 << 28)
- #define TEE_ATTR_BIT_VALUE (1 << 29)
- #define TEE_ECC_CURVE_NIST_P192 0x00000001
- #define TEE_ECC_CURVE_NIST_P224 0x00000002
- #define TEE_ECC_CURVE_NIST_P256 0x00000003
- #define TEE_ECC_CURVE_NIST_P384 0x00000004
- #define TEE_ECC_CURVE_NIST_P521 0x00000005
- #define TEE_PANIC_ID_TA_CLOSESESSIONENTRYPOINT 0x00000101
- #define TEE_PANIC_ID_TA_CREATEENTRYPOINT 0x00000102
- #define TEE_PANIC_ID_TA_DESTROYENTRYPOINT 0x00000103
- #define TEE_PANIC_ID_TA_INVOKECOMMANDENTRYPOINT 0x00000104
- #define TEE_PANIC_ID_TA_OPENSESSIONENTRYPOINT 0x00000105
- #define TEE_PANIC_ID_TEE_ALLOCATEPROPERTYENUMERATOR 0x00000201
- #define TEE_PANIC_ID_TEE_FREEPROPERTYENUMERATOR 0x00000202
- #define TEE_PANIC_ID_TEE_GETNEXTPROPERTY 0x00000203
- #define TEE_PANIC_ID_TEE_GETPROPERTYASBINARYBLOCK 0x00000204
- #define TEE_PANIC_ID_TEE_GETPROPERTYASBOOL 0x00000205
- #define TEE_PANIC_ID_TEE_GETPROPERTYASIDENTITY 0x00000206
- #define TEE_PANIC_ID_TEE_GETPROPERTYASSTRING 0x00000207
- #define TEE_PANIC_ID_TEE_GETPROPERTYASU32 0x00000208
- #define TEE_PANIC_ID_TEE_GETPROPERTYASUUID 0x00000209
- #define TEE_PANIC_ID_TEE_GETPROPERTYNAME 0x0000020A
- #define TEE_PANIC_ID_TEE_RESETPROPERTYENUMERATOR 0x0000020B
- #define TEE_PANIC_ID_TEE_STARTPROPERTYENUMERATOR 0x0000020C
- #define TEE_PANIC_ID_TEE_PANIC 0x00000301
- #define TEE_PANIC_ID_TEE_CLOSETASESSION 0x00000401
- #define TEE_PANIC_ID_TEE_INVOKETACOMMAND 0x00000402
- #define TEE_PANIC_ID_TEE_OPENTASESSION 0x00000403
- #define TEE_PANIC_ID_TEE_GETCANCELLATIONFLAG 0x00000501
- #define TEE_PANIC_ID_TEE_MASKCANCELLATION 0x00000502
- #define TEE_PANIC_ID_TEE_UNMASKCANCELLATION 0x00000503
- #define TEE_PANIC_ID_TEE_CHECKMEMORYACCESSRIGHTS 0x00000601
- #define TEE_PANIC_ID_TEE_FREE 0x00000602
- #define TEE_PANIC_ID_TEE_GETINSTANCEDATA 0x00000603
- #define TEE_PANIC_ID_TEE_MALLOC 0x00000604
- #define TEE_PANIC_ID_TEE_MEMCOMPARE 0x00000605
- #define TEE_PANIC_ID_TEE_MEMFILL 0x00000606
- #define TEE_PANIC_ID_TEE_MEMMOVE 0x00000607

- #define TEE_PANIC_ID_TEE_REALLOC 0x00000608
- #define TEE_PANIC_ID_TEE_SETINSTANCEDATA 0x00000609
- #define TEE_PANIC_ID_TEE_CLOSEOBJECT 0x00000701
- #define TEE_PANIC_ID_TEE_GETOBJECTBUFFERATTRIBUTE 0x00000702
- #define TEE_PANIC_ID_TEE_GETOBJECTINFO 0x00000703
- #define TEE_PANIC_ID_TEE_GETOBJECTVALUEATTRIBUTE 0x00000704
- #define TEE_PANIC_ID_TEE_RESTRICTOBJECTUSAGE 0x00000705
- #define TEE_PANIC_ID_TEE_GETOBJECTINFO1 0x00000706
- #define TEE_PANIC_ID_TEE_RESTRICTOBJECTUSAGE1 0x00000707
- #define TEE_PANIC_ID_TEE_ALLOCATETRANSIENTOBJECT 0x00000801
- #define TEE_PANIC_ID_TEE_COPYOBJECTATTRIBUTES 0x00000802
- #define TEE_PANIC_ID_TEE_FREETRANSIENTOBJECT 0x00000803
- #define TEE_PANIC_ID_TEE_GENERATEKEY 0x00000804
- #define TEE_PANIC_ID_TEE_INITREFATTRIBUTE 0x00000805
- #define TEE_PANIC_ID_TEE_INITVALUEATTRIBUTE 0x00000806
- #define TEE_PANIC_ID_TEE_POPULATETRANSIENTOBJECT 0x00000807
- #define TEE_PANIC_ID_TEE_RESETTRANSIENTOBJECT 0x00000808
- #define TEE_PANIC_ID_TEE_COPYOBJECTATTRIBUTES1 0x00000809
- #define TEE_PANIC_ID_TEE_CLOSEANDDELETEPERSISTENTOBJECT 0x00000901
- #define TEE_PANIC_ID_TEE_CREATEPERSISTENTOBJECT 0x00000902
- #define TEE_PANIC_ID_TEE_OPENPERSISTENTOBJECT 0x00000903
- #define TEE_PANIC_ID_TEE_RENAMEPERSISTENTOBJECT 0x00000904
- #define TEE_PANIC_ID_TEE_CLOSEANDDELETEPERSISTENTOBJECT1 0x00000905
- #define TEE_PANIC_ID_TEE_ALLOCATEPERSISTENTOBJECTENUMERATOR 0x00000A01
- #define TEE_PANIC_ID_TEE_FREEPERSISTENTOBJECTENUMERATOR 0x00000A02
- #define TEE_PANIC_ID_TEE_GETNEXTPERSISTENTOBJECT 0x00000A03
- #define TEE_PANIC_ID_TEE_RESETPERSISTENTOBJECTENUMERATOR 0x00000A04
- #define TEE_PANIC_ID_TEE_STARTPERSISTENTOBJECTENUMERATOR 0x00000A05
- #define TEE_PANIC_ID_TEE_READOBJECTDATA 0x00000B01
- #define TEE_PANIC_ID_TEE_SEEKOBJECTDATA 0x00000B02
- #define TEE_PANIC_ID_TEE_TRUNCATEOBJECTDATA 0x00000B03
- #define TEE_PANIC_ID_TEE_WRITEOBJECTDATA 0x00000B04
- #define TEE_PANIC_ID_TEE_ALLOCATEOPERATION 0x00000C01
- #define TEE_PANIC_ID_TEE_COPYOPERATION 0x00000C02
- #define TEE_PANIC_ID_TEE_FREEOPERATION 0x00000C03
- #define TEE_PANIC_ID_TEE_GETOPERATIONINFO 0x00000C04
- #define TEE_PANIC_ID_TEE_RESETOPERATION 0x00000C05
- #define TEE_PANIC_ID_TEE_SETOPERATIONKEY 0x00000C06
- #define TEE_PANIC_ID_TEE_SETOPERATIONKEY2 0x00000C07
- #define TEE_PANIC_ID_TEE_GETOPERATIONINFOMULTIPLE 0x00000C08
- #define TEE_PANIC_ID_TEE_DIGESTDOFINAL 0x00000D01
- #define TEE_PANIC_ID_TEE_DIGESTUPDATE 0x00000D02
- #define TEE_PANIC_ID_TEE_CIPHERDOFINAL 0x00000E01
- #define TEE_PANIC_ID_TEE_CIPHERINIT 0x00000E02
- #define TEE_PANIC_ID_TEE_CIPHERUPDATE 0x00000E03
- #define TEE_PANIC_ID_TEE_MACCOMPAREFINAL 0x00000F01
- #define TEE_PANIC_ID_TEE_MACCOMPUTEFINAL 0x00000F02
- #define TEE_PANIC_ID_TEE_MACINIT 0x00000F03
- #define TEE_PANIC_ID_TEE_MACUPDATE 0x00000F04
- #define TEE_PANIC_ID_TEE_AEDECRYPTFINAL 0x00001001
- #define TEE_PANIC_ID_TEE_AEENCRYPTFINAL 0x00001002
- #define TEE_PANIC_ID_TEE_AEINIT 0x00001003
- #define TEE_PANIC_ID_TEE_AEUPDATE 0x00001004
- #define TEE_PANIC_ID_TEE_AEUPDATEAAD 0x00001005
- #define TEE_PANIC_ID_TEE_ASYMMETRICDECRYPT 0x00001101

- #define TEE_PANIC_ID_TEE_ASYMMETRICENCRYPT 0x00001102
- #define TEE_PANIC_ID_TEE_ASYMMETRICSIGNDIGEST 0x00001103
- #define TEE_PANIC_ID_TEE_ASYMMETRICVERIFYDIGEST 0x00001104
- #define TEE_PANIC_ID_TEE_DERIVEKEY 0x00001201
- #define TEE_PANIC_ID_TEE_GENERATERANDOM 0x00001301
- #define TEE_PANIC_ID_TEE_GETREETIME 0x00001401
- #define TEE_PANIC_ID_TEE_GETSYSTEMTIME 0x00001402
- #define TEE_PANIC_ID_TEE_GETTAPERSISTENTTIME 0x00001403
- #define TEE_PANIC_ID_TEE_SETTAPERSISTENTTIME 0x00001404
- #define TEE_PANIC_ID_TEE_WAIT 0x00001405
- #define TEE_PANIC_ID_TEE_BIGINTFMMCONTEXTSIZEINU32 0x00001501
- #define TEE_PANIC_ID_TEE_BIGINTFMMSIZEINU32 0x00001502
- #define TEE_PANIC_ID_TEE_BIGINTINIT 0x00001601
- #define TEE_PANIC_ID_TEE_BIGINTINITFMM 0x00001602
- #define TEE_PANIC_ID_TEE_BIGINTINITFMMCONTEXT 0x00001603
- #define TEE_PANIC_ID_TEE_BIGINTCONVERTFROMOCTETSTRING 0x00001701
- #define TEE_PANIC_ID_TEE_BIGINTCONVERTFROMS32 0x00001702
- #define TEE_PANIC_ID_TEE_BIGINTCONVERTTOOCTETSTRING 0x00001703
- #define TEE_PANIC_ID_TEE_BIGINTCONVERTTOS32 0x00001704
- #define TEE_PANIC_ID_TEE_BIGINTCMP 0x00001801
- #define TEE_PANIC_ID_TEE_BIGINTCMPS32 0x00001802
- #define TEE_PANIC_ID_TEE_BIGINTGETBIT 0x00001803
- #define TEE_PANIC_ID_TEE_BIGINTGETBITCOUNT 0x00001804
- #define TEE_PANIC_ID_TEE_BIGINTSHIFTRIGHT 0x00001805
- #define TEE_PANIC_ID_TEE_BIGINTADD 0x00001901
- #define TEE_PANIC_ID_TEE_BIGINTDIV 0x00001902
- #define TEE_PANIC_ID_TEE_BIGINTMUL 0x00001903
- #define TEE_PANIC_ID_TEE_BIGINTNEG 0x00001904
- #define TEE_PANIC_ID_TEE_BIGINTSQUARE 0x00001905
- #define TEE_PANIC_ID_TEE_BIGINTSUB 0x00001906
- #define TEE_PANIC_ID_TEE_BIGINTADDMOD 0x00001A01
- #define TEE_PANIC_ID_TEE_BIGINTINVMOD 0x00001A02
- #define TEE_PANIC_ID_TEE_BIGINTMOD 0x00001A03
- #define TEE_PANIC_ID_TEE_BIGINTMULMOD 0x00001A04
- #define TEE_PANIC_ID_TEE_BIGINTSQUAREMOD 0x00001A05
- #define TEE_PANIC_ID_TEE_BIGINTSUBMOD 0x00001A06
- #define TEE_PANIC_ID_TEE_BIGINTCOMPUTEEXTENDEDGCD 0x00001B01
- #define TEE_PANIC_ID_TEE_BIGINTISPROBABLEPRIME 0x00001B02
- #define TEE_PANIC_ID_TEE_BIGINTRELATIVEPRIME 0x00001B03
- #define TEE_PANIC_ID_TEE_BIGINTCOMPUTEFMM 0x00001C01
- #define TEE_PANIC_ID_TEE_BIGINTCONVERTFROMFMM 0x00001C02
- #define TEE_PANIC_ID_TEE_BIGINTCONVERTTOFMM 0x00001C03
- #define TEE_PARAM_TYPES(t0, t1, t2, t3) ((t0) | ((t1) << 4) | ((t2) << 8) | ((t3) << 12))
- #define TEE_PARAM_TYPE_GET(t, i) ((((uint32_t)t) >> ((i)*4)) & 0xF)
- #define TEE_PARAM_TYPE_SET(t, i) (((uint32_t)(t) & 0xF) << ((i)*4))
- #define TEE_NUM_PARAMS 4
- #define TEE_BigIntSizeInU32(n) ((((n)+31)/32)+2)

14.6.1 Macro Definition Documentation

14.6.1.1 #define TEE_ALG_AES_CBC_MAC_NOPAD 0x30000110

14.6.1.2 #define TEE_ALG_AES_CBC_MAC_PKCS5 0x30000510

14.6.1.3 #define TEE_ALG_AES_CBC_NOPAD 0x10000110

14.6.1.4 #define TEE_ALG_AES_CCM 0x40000710

14.6.1.5 #define TEE_ALG_AES_CMAC 0x30000610

14.6.1.6 #define TEE_ALG_AES_CTR 0x10000210

14.6.1.7 #define TEE_ALG_AES_CTS 0x10000310

14.6.1.8 #define TEE_ALG_AES_ECB_NOPAD 0x10000010

14.6.1.9 #define TEE_ALG_AES_GCM 0x40000810

14.6.1.10 #define TEE_ALG_AES_XTS 0x10000410

14.6.1.11 #define TEE_ALG_DES3_CBC_MAC_NOPAD 0x30000113

14.6.1.12 #define TEE_ALG_DES3_CBC_MAC_PKCS5 0x30000513

14.6.1.13 #define TEE_ALG_DES3_CBC_NOPAD 0x10000113

14.6.1.14 #define TEE_ALG_DES3_ECB_NOPAD 0x10000013

14.6.1.15 #define TEE_ALG_DES_CBC_MAC_NOPAD 0x30000111

14.6.1.16 #define TEE_ALG_DES_CBC_MAC_PKCS5 0x30000511

14.6.1.17 #define TEE_ALG_DES_CBC_NOPAD 0x10000111

14.6.1.18 #define TEE_ALG_DES_ECB_NOPAD 0x10000011

14.6.1.19 #define TEE_ALG_DH_DERIVE_SHARED_SECRET 0x80000032

14.6.1.20 #define TEE_ALG_DSA_SHA1 0x70002131

14.6.1.21 #define TEE_ALG_DSA_SHA224 0x70003131

14.6.1.22 #define TEE_ALG_DSA_SHA256 0x70004131

14.6.1.23 #define TEE_ALG_ECDH_P192 0x80001042

14.6.1.24 #define TEE_ALG_ECDH_P224 0x80002042

14.6.1.25 #define TEE_ALG_ECDH_P256 0x80003042

14.6.1.26 #define TEE_ALG_ECDH_P384 0x80004042

14.6.1.27 #define TEE_ALG_ECDH_P521 0x80005042

14.6.1.28 #define TEE_ALG_ECDSA_P192 0x70001041

14.6.1.29 #define TEE_ALG_ECDSA_P224 0x70002041

14.6.1.30 #define TEE_ALG_ECDSA_P256 0x70003041

14.6.1.31 #define TEE_ALG_ECDSA_P384 0x70004041

14.6.1.32 #define TEE_ALG_ECDSA_P521 0x70005041

14.6.1.33 #define TEE_ALG_HMAC_MD5 0x30000001

14.6.1.34 #define TEE_ALG_HMAC_SHA1 0x30000002

14.6.1.35 #define TEE_ALG_HMAC_SHA224 0x30000003

14.6.1.36 #define TEE_ALG_HMAC_SHA256 0x30000004

14.6.1.37 #define TEE_ALG_HMAC_SHA384 0x30000005

14.6.1.38 #define TEE_ALG_HMAC_SHA512 0x30000006

14.6.1.39 #define TEE_ALG_MD5 0x50000001

14.6.1.40 #define TEE_ALG_MD5SHA1 0x5000000F

14.6.1.41 #define TEE_ALG_RSA_NOPAD 0x60000030

14.6.1.42 #define TEE_ALG_RSAES_PKCS1_OAEP_MGF1_SHA1 0x60210230

14.6.1.43 #define TEE_ALG_RSAES_PKCS1_OAEP_MGF1_SHA224 0x60310230

14.6.1.44 #define TEE_ALG_RSAES_PKCS1_OAEP_MGF1_SHA256 0x60410230

14.6.1.45 #define TEE_ALG_RSAES_PKCS1_OAEP_MGF1_SHA384 0x60510230

14.6.1.46 #define TEE_ALG_RSAES_PKCS1_OAEP_MGF1_SHA512 0x60610230

14.6.1.47 #define TEE_ALG_RSAES_PKCS1_V1_5 0x60000130

14.6.1.48 #define TEE_ALG_RSASSA_PKCS1_PSS_MGF1_SHA1 0x70212930

14.6.1.49 #define TEE_ALG_RSASSA_PKCS1_PSS_MGF1_SHA224 0x70313930

14.6.1.50 #define TEE_ALG_RSASSA_PKCS1_PSS_MGF1_SHA256 0x70414930

14.6.1.51 #define TEE_ALG_RSASSA_PKCS1_PSS_MGF1_SHA384 0x70515930

14.6.1.52 #define TEE_ALG_RSASSA_PKCS1_PSS_MGF1_SHA512 0x70616930

14.6.1.53 #define TEE_ALG_RSASSA_PKCS1_V1_5_MD5 0x70001830

14.6.1.54 #define TEE_ALG_RSASSA_PKCS1_V1_5_MD5SHA1 0x7000F830

14.6.1.55 #define TEE_ALG_RSASSA_PKCS1_V1_5_SHA1 0x70002830

14.6.1.56 #define TEE_ALG_RSASSA_PKCS1_V1_5_SHA224 0x70003830

14.6.1.57 #define TEE_ALG_RSASSA_PKCS1_V1_5_SHA256 0x70004830

14.6.1.58 #define TEE_ALG_RSASSA_PKCS1_V1_5_SHA384 0x70005830

14.6.1.59 #define TEE_ALG_RSASSA_PKCS1_V1_5_SHA512 0x70006830

14.6.1.60 #define TEE_ALG_SHA1 0x50000002

14.6.1.61 #define TEE_ALG_SHA224 0x50000003

14.6.1.62 #define TEE_ALG_SHA256 0x50000004

14.6.1.63 #define TEE_ALG_SHA384 0x50000005

14.6.1.64 #define TEE_ALG_SHA512 0x50000006

14.6.1.65 #define TEE_ATTR_BIT_PROTECTED (1 << 28)

14.6.1.66 #define TEE_ATTR_BIT_VALUE (1 << 29)

14.6.1.67 #define TEE_ATTR_DH_BASE 0xD0001232

14.6.1.68 #define TEE_ATTR_DH_PRIME 0xD0001032

14.6.1.69 #define TEE_ATTR_DH_PRIVATE_VALUE 0xC0000232

14.6.1.70 #define TEE_ATTR_DH_PUBLIC_VALUE 0xD0000132

14.6.1.71 #define TEE_ATTR_DH_SUBPRIME 0xD0001132

14.6.1.72 #define TEE_ATTR_DH_X_BITS 0xF0001332

14.6.1.73 #define TEE_ATTR_DSA_BASE 0xD0001231

14.6.1.74 #define TEE_ATTR_DSA_PRIME 0xD0001031

14.6.1.75 #define TEE_ATTR_DSA_PRIVATE_VALUE 0xC0000231

14.6.1.76 #define TEE_ATTR_DSA_PUBLIC_VALUE 0xD0000131

14.6.1.77 #define TEE_ATTR_DSA_SUBPRIME 0xD0001131

14.6.1.78 #define TEE_ATTR_ECC_CURVE 0xF0000441

14.6.1.79 #define TEE_ATTR_ECC_PRIVATE_VALUE 0xC0000341

14.6.1.80 #define TEE_ATTR_ECC_PUBLIC_VALUE_X 0xD0000141

14.6.1.81 #define TEE_ATTR_ECC_PUBLIC_VALUE_Y 0xD0000241

14.6.1.82 #define TEE_ATTR_RSA_COEFFICIENT 0xC0000830

14.6.1.83 #define TEE_ATTR_RSA_EXPONENT1 0xC0000630

14.6.1.84 #define TEE_ATTR_RSA_EXPONENT2 0xC0000730

14.6.1.85 #define TEE_ATTR_RSA_MODULUS 0xD0000130

14.6.1.86 #define TEE_ATTR_RSA_OAEP_LABEL 0xD0000930

14.6.1.87 #define TEE_ATTR_RSA_PRIME1 0xC0000430

14.6.1.88 #define TEE_ATTR_RSA_PRIME2 0xC0000530

14.6.1.89 #define TEE_ATTR_RSA_PRIVATE_EXPONENT 0xC0000330

14.6.1.90 #define TEE_ATTR_RSA_PSS_SALT_LENGTH 0xF0000A30

14.6.1.91 #define TEE_ATTR_RSA_PUBLIC_EXPONENT 0xD0000230

14.6.1.92 #define TEE_ATTR_SECRET_VALUE 0xC0000000

14.6.1.93 #define TEE_BigIntSizeInU32(*n*) ((((n)+31)/32)+2)

14.6.1.94 #define TEE_DATA_FLAG_ACCESS_READ 0x00000001

14.6.1.95 #define TEE_DATA_FLAG_ACCESS_WRITE 0x00000002

14.6.1.96 #define TEE_DATA_FLAG_ACCESS_WRITE_META 0x00000004

14.6.1.97 #define TEE_DATA_FLAG_OVERWRITE 0x00000400

14.6.1.98 #define TEE_DATA_FLAG_SHARE_READ 0x00000010

14.6.1.99 #define TEE_DATA_FLAG_SHARE_WRITE 0x00000020

14.6.1.100 #define TEE_DATA_MAX_POSITION 0xFFFFFFFF

14.6.1.101 #define TEE_ECC_CURVE_NIST_P192 0x00000001

14.6.1.102 #define TEE_ECC_CURVE_NIST_P224 0x00000002

14.6.1.103 #define TEE_ECC_CURVE_NIST_P256 0x00000003

14.6.1.104 #define TEE_ECC_CURVE_NIST_P384 0x00000004

14.6.1.105 #define TEE_ECC_CURVE_NIST_P521 0x00000005

14.6.1.106 #define TEE_ERROR_ACCESS_CONFLICT 0xFFFF0003

14.6.1.107 #define TEE_ERROR_ACCESS_DENIED 0xFFFF0001

14.6.1.108 #define TEE_ERROR_BAD_FORMAT 0xFFFF0005

14.6.1.109 #define TEE_ERROR_BAD_PARAMETERS 0xFFFF0006

14.6.1.110 #define TEE_ERROR_BAD_STATE 0xFFFF0007

14.6.1.111 #define TEE_ERROR_BUSY 0xFFFF000D

14.6.1.112 #define TEE_ERROR_CANCEL 0xFFFF0002

14.6.1.113 #define TEE_ERROR_COMMUNICATION 0xFFFF000E

14.6.1.114 #define TEE_ERROR_CORRUPT_OBJECT 0xF0100001

14.6.1.115 #define TEE_ERROR_CORRUPT_OBJECT_2 0xF0100002

14.6.1.116 #define TEE_ERROR_EXCESS_DATA 0xFFFF0004

14.6.1.117 #define TEE_ERROR_EXTERNAL_CANCEL 0xFFFF0011

14.6.1.118 #define TEE_ERROR_GENERIC 0xFFFF0000

14.6.1.119 #define TEE_ERROR_ITEM_NOT_FOUND 0xFFFF0008

14.6.1.120 #define TEE_ERROR_MAC_INVALID 0xFFFF3071

14.6.1.121 #define TEE_ERROR_NO_DATA 0xFFFF000B

14.6.1.122 #define TEE_ERROR_NOT_IMPLEMENTED 0xFFFF0009

14.6.1.123 #define TEE_ERROR_NOT_SUPPORTED 0xFFFF000A

14.6.1.124 #define TEE_ERROR_OUT_OF_MEMORY 0xFFFF000C

14.6.1.125 #define TEE_ERROR_OVERFLOW 0xFFFF300F

14.6.1.126 #define TEE_ERROR_SECURITY 0xFFFF000F

14.6.1.127 #define TEE_ERROR_SHORT_BUFFER 0xFFFF0010

14.6.1.128 #define TEE_ERROR_SIGNATURE_INVALID 0xFFFF3072

14.6.1.129 #define TEE_ERROR_STORAGE_NO_SPACE 0xFFFF3041

14.6.1.130 #define TEE_ERROR_STORAGE_NOT_AVAILABLE 0xF0100003

14.6.1.131 #define TEE_ERROR_STORAGE_NOT_AVAILABLE_2 0xF0100004

14.6.1.132 #define TEE_ERROR_TARGET_DEAD 0xFFFF3024

14.6.1.133 #define TEE_ERROR_TIME_NEEDS_RESET 0xFFFF5001

14.6.1.134 #define TEE_ERROR_TIME_NOT_SET 0xFFFF5000

14.6.1.135 #define TEE_HANDLE_FLAG_EXPECT_TWO_KEYS 0x00080000

14.6.1.136 #define TEE_HANDLE_FLAG_INITIALIZED 0x00020000

14.6.1.137 #define TEE_HANDLE_FLAG_KEY_SET 0x00040000

14.6.1.138 #define TEE_HANDLE_FLAG_PERSISTENT 0x00010000

14.6.1.139 #define TEE_HANDLE_NULL 0

14.6.1.140 #define TEE_INT_CORE_API_SPEC_VERSION 0x0000000A

14.6.1.141 #define TEE_LOGIN_APPLICATION 0x00000004

14.6.1.142 #define TEE_LOGIN_APPLICATION_GROUP 0x00000006

14.6.1.143 #define TEE_LOGIN_APPLICATION_USER 0x00000005

14.6.1.144 #define TEE_LOGIN_GROUP 0x00000002

14.6.1.145 #define TEE_LOGIN_PUBLIC 0x00000000

14.6.1.146 #define TEE_LOGIN_TRUSTED_APP 0xF0000000

14.6.1.147 #define TEE_LOGIN_USER 0x00000001

14.6.1.148 #define TEE_MALLOC_FILL_ZERO 0x00000000

14.6.1.149 #define TEE_MEMORY_ACCESS_ANY_OWNER 0x00000004

14.6.1.150 #define TEE_MEMORY_ACCESS_READ 0x00000001

14.6.1.151 #define TEE_MEMORY_ACCESS_WRITE 0x00000002

14.6.1.152 #define TEE_NUM_PARAMS 4

14.6.1.153 #define TEE_OBJECT_ID_MAX_LEN 64

14.6.1.154 #define TEE_OPERATION_AE 4

14.6.1.155 #define TEE_OPERATION_ASYMMETRIC_CIPHER 6

14.6.1.156 #define TEE_OPERATION_ASYMMETRIC_SIGNATURE 7

14.6.1.157 #define TEE_OPERATION_CIPHER 1

14.6.1.158 #define TEE_OPERATION_DIGEST 5

14.6.1.159 #define TEE_OPERATION_KEY_DERIVATION 8

14.6.1.160 #define TEE_OPERATION_MAC 3

14.6.1.161 #define TEE_OPERATION_STATE_ACTIVE 0x00000001

14.6.1.162 #define TEE_OPERATION_STATE_INITIAL 0x00000000

14.6.1.163 #define TEE_ORIGIN_API 0x00000001

14.6.1.164 #define TEE_ORIGIN_COMMS 0x00000002

14.6.1.165 #define TEE_ORIGIN_TEE 0x00000003

14.6.1.166 #define TEE_ORIGIN_TRUSTED_APP 0x00000004

14.6.1.167 #define TEE_PANIC_ID_TA_CLOSESESSIONENTRYPOINT 0x00000101

14.6.1.168 #define TEE_PANIC_ID_TA_CREATEENTRYPOINT 0x00000102

14.6.1.169 #define TEE_PANIC_ID_TA_DESTROYENTRYPOINT 0x00000103

14.6.1.170 #define TEE_PANIC_ID_TA_INVOKECOMMANDENTRYPOINT 0x00000104

14.6.1.171 #define TEE_PANIC_ID_TA_OPENSESSIONENTRYPOINT 0x00000105

14.6.1.172 #define TEE_PANIC_ID_TEE_AEDECRYPTFINAL 0x00001001

14.6.1.173 #define TEE_PANIC_ID_TEE_AEENCRYPTFINAL 0x00001002

14.6.1.174 #define TEE_PANIC_ID_TEE_AEINIT 0x00001003

14.6.1.175 #define TEE_PANIC_ID_TEE_AEUPDATE 0x00001004

14.6.1.176 #define TEE_PANIC_ID_TEE_AEUPDATEAAD 0x00001005

14.6.1.177 #define TEE_PANIC_ID_TEE_ALLOCATEOPERATION 0x00000C01

14.6.1.178 #define TEE_PANIC_ID_TEE_ALLOCATEPERSISTENTOBJECTENUMERATOR 0x00000A01

14.6.1.179 #define TEE_PANIC_ID_TEE_ALLOCATEPROPERTYENUMERATOR 0x00000201

14.6.1.180 #define TEE_PANIC_ID_TEE_ALLOCATETRANSIENTOBJECT 0x00000801

14.6.1.181 #define TEE_PANIC_ID_TEE_ASYMMETRICDECRYPT 0x00001101

14.6.1.182 #define TEE_PANIC_ID_TEE_ASYMMETRICENCRYPT 0x00001102

14.6.1.183 #define TEE_PANIC_ID_TEE_ASYMMETRICSIGNDIGEST 0x00001103

14.6.1.184 #define TEE_PANIC_ID_TEE_ASYMMETRICVERIFYDIGEST 0x00001104

14.6.1.185 #define TEE_PANIC_ID_TEE_BIGINTADD 0x00001901

14.6.1.186 #define TEE_PANIC_ID_TEE_BIGINTADDMOD 0x00001A01

14.6.1.187 #define TEE_PANIC_ID_TEE_BIGINTCMP 0x00001801

14.6.1.188 #define TEE_PANIC_ID_TEE_BIGINTCMPS32 0x00001802

14.6.1.189 #define TEE_PANIC_ID_TEE_BIGINTCOMPUTEEXTENDEDGCD 0x00001B01

14.6.1.190 #define TEE_PANIC_ID_TEE_BIGINTCOMPUTEFMM 0x00001C01

14.6.1.191 #define TEE_PANIC_ID_TEE_BIGINTCONVERTFROMFMM 0x00001C02

14.6.1.192 #define TEE_PANIC_ID_TEE_BIGINTCONVERTFROMOCTETSTRING 0x00001701

14.6.1.193 #define TEE_PANIC_ID_TEE_BIGINTCONVERTFROMS32 0x00001702

14.6.1.194 #define TEE_PANIC_ID_TEE_BIGINTCONVERTTOFMM 0x00001C03

14.6.1.195 #define TEE_PANIC_ID_TEE_BIGINTCONVERTTOOCTETSTRING 0x00001703

14.6.1.196 #define TEE_PANIC_ID_TEE_BIGINTCONVERTTOS32 0x00001704

14.6.1.197 #define TEE_PANIC_ID_TEE_BIGINTDIV 0x00001902

14.6.1.198 #define TEE_PANIC_ID_TEE_BIGINTFMMCONTEXTSIZEINU32 0x00001501

14.6.1.199 #define TEE_PANIC_ID_TEE_BIGINTFMMSIZEINU32 0x00001502

14.6.1.200 #define TEE_PANIC_ID_TEE_BIGINTGETBIT 0x00001803

14.6.1.201 #define TEE_PANIC_ID_TEE_BIGINTGETBITCOUNT 0x00001804

14.6.1.202 #define TEE_PANIC_ID_TEE_BIGINTINIT 0x00001601

14.6.1.203 #define TEE_PANIC_ID_TEE_BIGINTINITFMM 0x00001602

14.6.1.204 #define TEE_PANIC_ID_TEE_BIGINTINITFMMCONTEXT 0x00001603

14.6.1.205 #define TEE_PANIC_ID_TEE_BIGINTINVMOD 0x00001A02

14.6.1.206 #define TEE_PANIC_ID_TEE_BIGINTISPROBABLEPRIME 0x00001B02

14.6.1.207 #define TEE_PANIC_ID_TEE_BIGINTMOD 0x00001A03

14.6.1.208 #define TEE_PANIC_ID_TEE_BIGINTMUL 0x00001903

14.6.1.209 #define TEE_PANIC_ID_TEE_BIGINTMULMOD 0x00001A04

14.6.1.210 #define TEE_PANIC_ID_TEE_BIGINTNEG 0x00001904

14.6.1.211 #define TEE_PANIC_ID_TEE_BIGINTRELATIVEPRIME 0x00001B03

14.6.1.212 #define TEE_PANIC_ID_TEE_BIGINTSHIFTRIGHT 0x00001805

14.6.1.213 #define TEE_PANIC_ID_TEE_BIGINTSQUARE 0x00001905

14.6.1.214 #define TEE_PANIC_ID_TEE_BIGINTSQUAREMOD 0x00001A05

14.6.1.215 #define TEE_PANIC_ID_TEE_BIGINTSUB 0x00001906

14.6.1.216 #define TEE_PANIC_ID_TEE_BIGINTSUBMOD 0x00001A06

14.6.1.217 #define TEE_PANIC_ID_TEE_CHECKMEMORYACCESSRIGHTS 0x00000601

14.6.1.218 #define TEE_PANIC_ID_TEE_CIPHERDOFINAL 0x00000E01

14.6.1.219 #define TEE_PANIC_ID_TEE_CIPHERINIT 0x00000E02

14.6.1.220 #define TEE_PANIC_ID_TEE_CIPHERUPDATE 0x00000E03

14.6.1.221 #define TEE_PANIC_ID_TEE_CLOSEANDDELETEPERSISTENTOBJECT 0x00000901

14.6.1.222 #define TEE_PANIC_ID_TEE_CLOSEANDDELETEPERSISTENTOBJECT1 0x00000905

14.6.1.223 #define TEE_PANIC_ID_TEE_CLOSEOBJECT 0x00000701

14.6.1.224 #define TEE_PANIC_ID_TEE_CLOSETASESSION 0x00000401

14.6.1.225 #define TEE_PANIC_ID_TEE_COPYOBJECTATTRIBUTES 0x00000802

14.6.1.226 #define TEE_PANIC_ID_TEE_COPYOBJECTATTRIBUTES1 0x00000809

14.6.1.227 #define TEE_PANIC_ID_TEE_COPYOPERATION 0x00000C02

14.6.1.228 #define TEE_PANIC_ID_TEE_CREATEPERSISTENTOBJECT 0x00000902

14.6.1.229 #define TEE_PANIC_ID_TEE_DERIVEKEY 0x00001201

14.6.1.230 #define TEE_PANIC_ID_TEE_DIGESTDOFINAL 0x00000D01

14.6.1.231 #define TEE_PANIC_ID_TEE_DIGESTUPDATE 0x00000D02

14.6.1.232 #define TEE_PANIC_ID_TEE_FREE 0x00000602

14.6.1.233 #define TEE_PANIC_ID_TEE_FREEOPERATION 0x00000C03

14.6.1.234 #define TEE_PANIC_ID_TEE_FREEPERSISTENTOBJECTENUMERATOR 0x00000A02

14.6.1.235 #define TEE_PANIC_ID_TEE_FREEPROPERTYENUMERATOR 0x00000202

14.6.1.236 #define TEE_PANIC_ID_TEE_FREETRANSIENTOBJECT 0x00000803

14.6.1.237 #define TEE_PANIC_ID_TEE_GENERATEKEY 0x00000804

14.6.1.238 #define TEE_PANIC_ID_TEE_GENERATERANDOM 0x00001301

14.6.1.239 #define TEE_PANIC_ID_TEE_GETCANCELLATIONFLAG 0x00000501

14.6.1.240 #define TEE_PANIC_ID_TEE_GETINSTANCEDATA 0x00000603

14.6.1.241 #define TEE_PANIC_ID_TEE_GETNEXTPERSISTENTOBJECT 0x00000A03

14.6.1.242 #define TEE_PANIC_ID_TEE_GETNEXTPROPERTY 0x00000203

14.6.1.243 #define TEE_PANIC_ID_TEE_GETOBJECTBUFFERATTRIBUTE 0x00000702

14.6.1.244 #define TEE_PANIC_ID_TEE_GETOBJECTINFO 0x00000703

14.6.1.245 #define TEE_PANIC_ID_TEE_GETOBJECTINFO1 0x00000706

14.6.1.246 #define TEE_PANIC_ID_TEE_GETOBJECTVALUEATTRIBUTE 0x00000704

14.6.1.247 #define TEE_PANIC_ID_TEE_GETOPERATIONINFO 0x00000C04

14.6.1.248 #define TEE_PANIC_ID_TEE_GETOPERATIONINFOMULTIPLE 0x00000C08

14.6.1.249 #define TEE_PANIC_ID_TEE_GETPROPERTYASBINARYBLOCK 0x00000204

14.6.1.250 #define TEE_PANIC_ID_TEE_GETPROPERTYASBOOL 0x00000205

14.6.1.251 #define TEE_PANIC_ID_TEE_GETPROPERTYASIDENTITY 0x00000206

14.6.1.252 #define TEE_PANIC_ID_TEE_GETPROPERTYASSTRING 0x00000207

14.6.1.253 #define TEE_PANIC_ID_TEE_GETPROPERTYASU32 0x00000208

14.6.1.254 #define TEE_PANIC_ID_TEE_GETPROPERTYASUUID 0x00000209

14.6.1.255 #define TEE_PANIC_ID_TEE_GETPROPERTYNAME 0x0000020A

14.6.1.256 #define TEE_PANIC_ID_TEE_GETREETIME 0x00001401

14.6.1.257 #define TEE_PANIC_ID_TEE_GETSYSTEMTIME 0x00001402

14.6.1.258 #define TEE_PANIC_ID_TEE_GETTAPERSISTENTTIME 0x00001403

14.6.1.259 #define TEE_PANIC_ID_TEE_INITREFATTRIBUTE 0x00000805

14.6.1.260 #define TEE_PANIC_ID_TEE_INITVALUEATTRIBUTE 0x00000806

14.6.1.261 #define TEE_PANIC_ID_TEE_INVOKETACOMMAND 0x00000402

14.6.1.262 #define TEE_PANIC_ID_TEE_MACCOMPAREFINAL 0x00000F01

14.6.1.263 #define TEE_PANIC_ID_TEE_MACCOMPUTEFINAL 0x00000F02

14.6.1.264 #define TEE_PANIC_ID_TEE_MACINIT 0x00000F03

14.6.1.265 #define TEE_PANIC_ID_TEE_MACUPDATE 0x00000F04

14.6.1.266 #define TEE_PANIC_ID_TEE_MALLOC 0x00000604

14.6.1.267 #define TEE_PANIC_ID_TEE_MASKCANCELLATION 0x00000502

14.6.1.268 #define TEE_PANIC_ID_TEE_MEMCOMPARE 0x00000605

14.6.1.269 #define TEE_PANIC_ID_TEE_MEMFILL 0x00000606

14.6.1.270 #define TEE_PANIC_ID_TEE_MEMMOVE 0x00000607

14.6.1.271 #define TEE_PANIC_ID_TEE_OPENPERSISTENTOBJECT 0x00000903

14.6.1.272 #define TEE_PANIC_ID_TEE_OPENTASESSION 0x00000403

14.6.1.273 #define TEE_PANIC_ID_TEE_PANIC 0x00000301

14.6.1.274 #define TEE_PANIC_ID_TEE_POPULATETRANSIENTOBJECT 0x00000807

14.6.1.275 #define TEE_PANIC_ID_TEE_READOBJECTDATA 0x00000B01

14.6.1.276 #define TEE_PANIC_ID_TEE_REALLOC 0x00000608

14.6.1.277 #define TEE_PANIC_ID_TEE_RENAMEPERSISTENTOBJECT 0x00000904

14.6.1.278 #define TEE_PANIC_ID_TEE_RESETOPERATION 0x00000C05

14.6.1.279 #define TEE_PANIC_ID_TEE_RESETPERSISTENTOBJECTENUMERATOR 0x00000A04

14.6.1.280 #define TEE_PANIC_ID_TEE_RESETPROPERTYENUMERATOR 0x0000020B

14.6.1.281 #define TEE_PANIC_ID_TEE_RESETTRANSIENTOBJECT 0x00000808

14.6.1.282 #define TEE_PANIC_ID_TEE_RESTRICTOBJECTUSAGE 0x00000705

14.6.1.283 #define TEE_PANIC_ID_TEE_RESTRICTOBJECTUSAGE1 0x00000707

14.6.1.284 #define TEE_PANIC_ID_TEE_SEEKOBJECTDATA 0x00000B02

14.6.1.285 #define TEE_PANIC_ID_TEE_SETINSTANCEDATA 0x00000609

14.6.1.286 #define TEE_PANIC_ID_TEE_SETOPERATIONKEY 0x00000C06

14.6.1.287 #define TEE_PANIC_ID_TEE_SETOPERATIONKEY2 0x00000C07

14.6.1.288 #define TEE_PANIC_ID_TEE_SETTAPERSISTENTTIME 0x00001404

14.6.1.289 #define TEE_PANIC_ID_TEE_STARTPERSISTENTOBJECTENUMERATOR 0x00000A05

14.6.1.290 #define TEE_PANIC_ID_TEE_STARTPROPERTYENUMERATOR 0x0000020C

14.6.1.291 #define TEE_PANIC_ID_TEE_TRUNCATEOBJECTDATA 0x00000B03

14.6.1.292 #define TEE_PANIC_ID_TEE_UNMASKCANCELLATION 0x00000503

14.6.1.293 #define TEE_PANIC_ID_TEE_WAIT 0x00001405

14.6.1.294 #define TEE_PANIC_ID_TEE_WRITEOBJECTDATA 0x00000B04

14.6.1.295 #define TEE_PARAM_TYPE_GET(*t*, *i*) ((((uint32_t)t) >> ((i)*4)) & 0xF)

14.6.1.296 #define TEE_PARAM_TYPE_MEMREF_INOUT 7

14.6.1.297 #define TEE_PARAM_TYPE_MEMREF_INPUT 5

14.6.1.298 #define TEE_PARAM_TYPE_MEMREF_OUTPUT 6

14.6.1.299 #define TEE_PARAM_TYPE_NONE 0

14.6.1.300 #define TEE_PARAM_TYPE_SET(*t, i*) (((uint32_t)(t) & 0xF) << ((i)*4))

14.6.1.301 #define TEE_PARAM_TYPE_VALUE_INOUT 3

14.6.1.302 #define TEE_PARAM_TYPE_VALUE_INPUT 1

14.6.1.303 #define TEE_PARAM_TYPE_VALUE_OUTPUT 2

14.6.1.304 #define TEE_PARAM_TYPES(*t0, t1, t2, t3*) ((t0) | ((t1) << 4) | ((t2) << 8) | ((t3) << 12))

14.6.1.305 #define TEE_PROPSET_CURRENT_CLIENT (TEE_PropSetHandle)0xFFFFFFFE

14.6.1.306 #define TEE_PROPSET_CURRENT_TA (TEE_PropSetHandle)0xFFFFFFFF

14.6.1.307 #define TEE_PROPSET_TEE_IMPLEMENTATION (TEE_PropSetHandle)0xFFFFFFFD

14.6.1.308 #define TEE_STORAGE_PRIVATE 0x00000001

14.6.1.309 #define TEE_SUCCESS 0x00000000

14.6.1.310 #define TEE_TIMEOUT_INFINITE 0xFFFFFFFF

14.6.1.311 #define TEE_TYPE_AES 0xA0000010

14.6.1.312 #define TEE_TYPE_CORRUPTED_OBJECT 0xA00000BE

14.6.1.313 #define TEE_TYPE_DATA 0xA00000BF

14.6.1.314 #define TEE_TYPE_DES 0xA0000011

14.6.1.315 #define TEE_TYPE_DES3 0xA0000013

14.6.1.316 #define TEE_TYPE_DH_KEYPAIR 0xA1000032

14.6.1.317 #define TEE_TYPE_DSA_KEYPAIR 0xA1000031

14.6.1.318 #define TEE_TYPE_DSA_PUBLIC_KEY 0xA0000031

14.6.1.319 #define TEE_TYPE_ECDH_KEYPAIR 0xA1000042

14.6.1.320 #define TEE_TYPE_ECDH_PUBLIC_KEY 0xA0000042

14.6.1.321 #define TEE_TYPE_ECDSA_KEYPAIR 0xA1000041

14.6.1.322 #define TEE_TYPE_ECDSA_PUBLIC_KEY 0xA0000041

14.6.1.323 #define TEE_TYPE_GENERIC_SECRET 0xA0000000

14.6.1.324 #define TEE_TYPE_HMAC_MD5 0xA0000001

14.6.1.325 #define TEE_TYPE_HMAC_SHA1 0xA0000002

14.6.1.326 #define TEE_TYPE_HMAC_SHA224 0xA0000003

14.6.1.327 #define TEE_TYPE_HMAC_SHA256 0xA0000004

14.6.1.328 #define TEE_TYPE_HMAC_SHA384 0xA0000005

14.6.1.329 #define TEE_TYPE_HMAC_SHA512 0xA0000006

14.6.1.330 #define TEE_TYPE_RSA_KEYPAIR 0xA1000030

14.6.1.331 #define TEE_TYPE_RSA_PUBLIC_KEY 0xA0000030

14.6.1.332 #define TEE_USAGE_DECRYPT 0x00000004

14.6.1.333 #define TEE_USAGE_DERIVE 0x00000040

14.6.1.334 #define TEE_USAGE_ENCRYPT 0x00000002

14.6.1.335 #define TEE_USAGE_EXTRACTABLE 0x00000001

14.6.1.336 #define TEE_USAGE_MAC 0x00000008

14.6.1.337 #define TEE_USAGE_SIGN 0x00000010

14.6.1.338 #define TEE_USAGE_VERIFY 0x00000020

14.7 include/tee_api_defines_extensions.h File Reference

Macros

- #define TEE_ALG_HKDF_MD5_DERIVE_KEY 0x800010C0
- #define TEE_ALG_HKDF_SHA1_DERIVE_KEY 0x800020C0
- #define TEE_ALG_HKDF_SHA224_DERIVE_KEY 0x800030C0
- #define TEE_ALG_HKDF_SHA256_DERIVE_KEY 0x800040C0
- #define TEE_ALG_HKDF_SHA384_DERIVE_KEY 0x800050C0
- #define TEE_ALG_HKDF_SHA512_DERIVE_KEY 0x800060C0
- #define TEE_TYPE_HKDF_IKM 0xA10000C0
- #define TEE_ATTR_HKDF_IKM 0xC00001C0
- #define TEE_ATTR_HKDF_SALT 0xD00002C0
- #define TEE_ATTR_HKDF_INFO 0xD00003C0
- #define TEE_ATTR_HKDF_OKM_LENGTH 0xF00004C0
- #define TEE_ALG_CONCAT_KDF_SHA1_DERIVE_KEY 0x800020C1
- #define TEE_ALG_CONCAT_KDF_SHA224_DERIVE_KEY 0x800030C1
- #define TEE_ALG_CONCAT_KDF_SHA256_DERIVE_KEY 0x800040C1
- #define TEE_ALG_CONCAT_KDF_SHA384_DERIVE_KEY 0x800050C1
- #define TEE_ALG_CONCAT_KDF_SHA512_DERIVE_KEY 0x800060C1
- #define TEE_TYPE_CONCAT_KDF_Z 0xA10000C1
- #define TEE_ATTR_CONCAT_KDF_Z 0xC00001C1
- #define TEE_ATTR_CONCAT_KDF_OTHER_INFO 0xD00002C1

- #define TEE_ATTR_CONCAT_KDF_DKM_LENGTH 0xF00003C1
- #define TEE_ALG_PBKDF2_HMAC_SHA1_DERIVE_KEY 0x800020C2
- #define TEE_TYPE_PBKDF2_PASSWORD 0xA10000C2
- #define TEE_ATTR_PBKDF2_PASSWORD 0xC00001C2
- #define TEE_ATTR_PBKDF2_SALT 0xD00002C2
- #define TEE_ATTR_PBKDF2_ITERATION_COUNT 0xF00003C2
- #define TEE_ATTR_PBKDF2_DKM_LENGTH 0xF00004C2
- #define TEE_STORAGE_PRIVATE_REE 0x80000000
- #define TEE_STORAGE_PRIVATE_RPMB 0x80000100
- #define TEE_STORAGE_PRIVATE_SQL_RESERVED 0x80000200
- #define TEE_MEMORY_ACCESS_NONSECURE 0x10000000
- #define TEE_MEMORY_ACCESS_SECURE 0x20000000

14.7.1 Macro Definition Documentation

14.7.1.1 #define TEE_ALG_CONCAT_KDF_SHA1_DERIVE_KEY 0x800020C1

14.7.1.2 #define TEE_ALG_CONCAT_KDF_SHA224_DERIVE_KEY 0x800030C1

14.7.1.3 #define TEE_ALG_CONCAT_KDF_SHA256_DERIVE_KEY 0x800040C1

14.7.1.4 #define TEE_ALG_CONCAT_KDF_SHA384_DERIVE_KEY 0x800050C1

14.7.1.5 #define TEE_ALG_CONCAT_KDF_SHA512_DERIVE_KEY 0x800060C1

14.7.1.6 #define TEE_ALG_HKDF_MD5_DERIVE_KEY 0x800010C0

14.7.1.7 #define TEE_ALG_HKDF_SHA1_DERIVE_KEY 0x800020C0

14.7.1.8 #define TEE_ALG_HKDF_SHA224_DERIVE_KEY 0x800030C0

14.7.1.9 #define TEE_ALG_HKDF_SHA256_DERIVE_KEY 0x800040C0

14.7.1.10 #define TEE_ALG_HKDF_SHA384_DERIVE_KEY 0x800050C0

14.7.1.11 #define TEE_ALG_HKDF_SHA512_DERIVE_KEY 0x800060C0

14.7.1.12 #define TEE_ALG_PBKDF2_HMAC_SHA1_DERIVE_KEY 0x800020C2

14.7.1.13 #define TEE_ATTR_CONCAT_KDF_DKM_LENGTH 0xF00003C1

14.7.1.14 #define TEE_ATTR_CONCAT_KDF_OTHER_INFO 0xD00002C1

14.7.1.15 #define TEE_ATTR_CONCAT_KDF_Z 0xC00001C1

14.7.1.16 #define TEE_ATTR_HKDF_IKM 0xC00001C0

14.7.1.17 #define TEE_ATTR_HKDF_INFO 0xD00003C0

14.7.1.18 #define TEE_ATTR_HKDF_OKM_LENGTH 0xF00004C0

14.7.1.19 #define TEE_ATTR_HKDF_SALT 0xD00002C0

14.7.1.20 #define TEE_ATTR_PBKDF2_DKM_LENGTH 0xF00004C2

14.7.1.21 #define TEE_ATTR_PBKDF2_ITERATION_COUNT 0xF00003C2

14.7.1.22 #define TEE_ATTR_PBKDF2_PASSWORD 0xC00001C2

14.7.1.23 #define TEE_ATTR_PBKDF2_SALT 0xD00002C2

14.7.1.24 #define TEE_MEMORY_ACCESS_NONSECURE 0x10000000

14.7.1.25 #define TEE_MEMORY_ACCESS_SECURE 0x20000000

14.7.1.26 #define TEE_STORAGE_PRIVATE_REE 0x80000000

14.7.1.27 #define TEE_STORAGE_PRIVATE_RPMB 0x80000100

14.7.1.28 #define TEE_STORAGE_PRIVATE_SQL_RESERVED 0x80000200

14.7.1.29 #define TEE_TYPE_CONCAT_KDF_Z 0xA10000C1

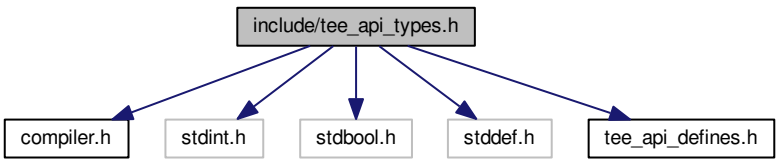
14.7.1.30 #define TEE_TYPE_HKDF_IKM 0xA10000C0

14.7.1.31 #define TEE_TYPE_PBKDF2_PASSWORD 0xA10000C2

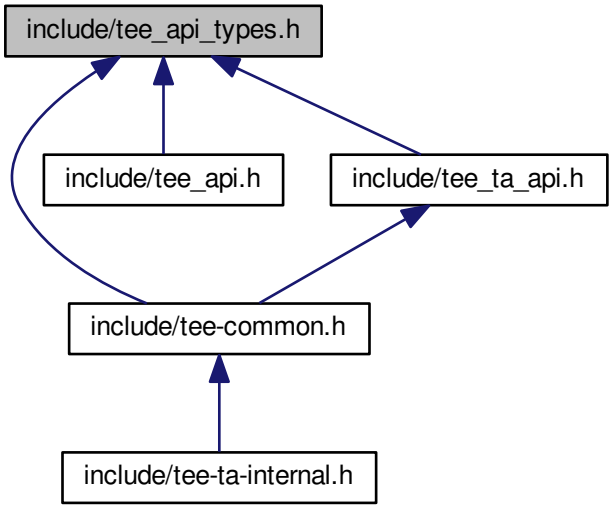
14.8 include/tee_api_types.h File Reference

```
#include <compiler.h>
#include <stdint.h>
#include <stdbool.h>
#include <stddef.h>
#include <tee_api_defines.h>
```

Include dependency graph for tee_api_types.h:



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



Classes

- struct TEE_UUID
- struct TEE_Identity
- union TEE_Param
- struct TEE_ObjectInfo
- struct TEE_Attribute
- struct TEE_OperationInfo
- struct TEE_OperationInfoKey
- struct TEE_OperationInfoMultiple
- struct TEE_Time
- struct TEE_SEReaderProperties
- struct TEE_SEAID
- struct pollfd
- struct addrinfo

Macros

- #define DMREQ_FINISH 0
- #define DMREQ_WRITE 1
- #define TEE_MEM_INPUT 0x00000001
- #define TEE_MEM_OUTPUT 0x00000002
- #define TEE_MEMREF_0_USED 0x00000001
- #define TEE_MEMREF_1_USED 0x00000002
- #define TEE_MEMREF_2_USED 0x00000004
- #define TEE_MEMREF_3_USED 0x00000008
- #define TEE_SE_READER_NAME_MAX 20
- #define socklen_t unsigned int

Typedefs

- typedef uint32_t TEE_Result
- typedef struct __TEE_TASessionHandle * TEE_TASessionHandle
- typedef struct __TEE_PropSetHandle * TEE_PropSetHandle
- typedef struct __TEE_ObjectHandle * TEE_ObjectHandle
- typedef struct __TEE_ObjectEnumHandle * TEE_ObjectEnumHandle
- typedef struct __TEE_OperationHandle * TEE_OperationHandle
- typedef uint32_t TEE_ObjectType
- typedef uint32_t TEE_BigInt
- typedef uint32_t TEE_BigIntFMM
- typedef uint32_t TEE_BigIntFMMContext __aligned(__alignof__(void *))
- typedef struct __TEE_SEServiceHandle * TEE_SEServiceHandle
- typedef struct __TEE_SEReaderHandle * TEE_SEReaderHandle
- typedef struct __TEE_SESessionHandle * TEE_SESessionHandle
- typedef struct __TEE_SEChannelHandle * TEE_SEChannelHandle
- typedef uint32_t TEE_ErrorOrigin
- typedef void * TEE_Session
- typedef unsigned long int nfds_t

Enumerations

- enum TEE_Whence { TEE_DATA_SEEK_SET = 0, TEE_DATA_SEEK_CUR = 1, TEE_DATA_SEEK_END = 2 }
- enum TEE_OperationMode {
  TEE_MODE_ENCRYPT = 0, TEE_MODE_DECRYPT = 1, TEE_MODE_SIGN = 2, TEE_MODE_VERIFY = 3,
  TEE_MODE_MAC = 4, TEE_MODE_DIGEST = 5, TEE_MODE_DERIVE = 6 }

14.8.1 Macro Definition Documentation

14.8.1.1 #define DMREQ_FINISH 0

14.8.1.2 #define DMREQ_WRITE 1

14.8.1.3 #define socklen_t unsigned int

14.8.1.4 #define TEE_MEM_INPUT 0x00000001

14.8.1.5 #define TEE_MEM_OUTPUT 0x00000002

14.8.1.6 #define TEE_MEMREF_0_USED 0x00000001

14.8.1.7 #define TEE_MEMREF_1_USED 0x00000002

14.8.1.8 #define TEE_MEMREF_2_USED 0x00000004

14.8.1.9 #define TEE_MEMREF_3_USED 0x00000008

14.8.1.10 #define TEE_SE_READER_NAME_MAX 20

14.8.2 Typedef Documentation

14.8.2.1 typedef uint32_t TEE_BigIntFMMContext __aligned(__alignof__(void *))

14.8.2.2 typedef unsigned long int nfds_t

14.8.2.3 typedef uint32_t TEE_BigInt

14.8.2.4 typedef uint32_t TEE_BigIntFMM

14.8.2.5 typedef uint32_t TEE_ErrorOrigin

14.8.2.6 typedef struct __TEE_ObjectEnumHandle* TEE_ObjectEnumHandle

14.8.2.7 typedef struct __TEE_ObjectHandle* TEE_ObjectHandle

14.8.2.8 typedef uint32_t TEE_ObjectType

14.8.2.9 typedef struct __TEE_OperationHandle* TEE_OperationHandle

14.8.2.10 typedef struct __TEE_PropSetHandle* TEE_PropSetHandle

14.8.2.11 typedef uint32_t TEE_Result

14.8.2.12 typedef struct __TEE_SEChannelHandle* TEE_SEChannelHandle

14.8.2.13 typedef struct __TEE_SEReaderHandle* TEE_SEReaderHandle

14.8.2.14 typedef struct __TEE_SEServiceHandle* TEE_SEServiceHandle

14.8.2.15 typedef struct __TEE_SESessionHandle* TEE_SESessionHandle

14.8.2.16 typedef void* TEE_Session

14.8.2.17 typedef struct __TEE_TASessionHandle* TEE_TASessionHandle

14.8.3 Enumeration Type Documentation

14.8.3.1 enum TEE_OperationMode

Enumerator

TEE_MODE_ENCRYPT

TEE_MODE_DECRYPT

TEE_MODE_SIGN

TEE_MODE_VERIFY

TEE_MODE_MAC

TEE_MODE_DIGEST

TEE_MODE_DERIVE

14.8.3.2 enum TEE_Whence

Enumerator

TEE_DATA_SEEK_SET

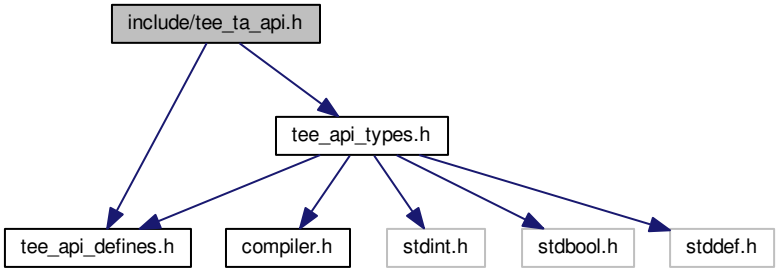
TEE_DATA_SEEK_CUR

TEE_DATA_SEEK_END

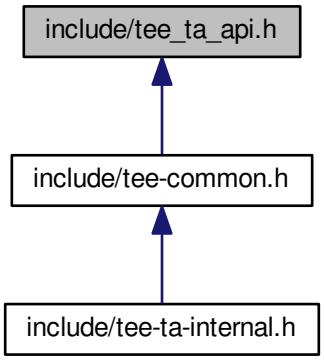
14.9 include/tee_ta_api.h File Reference

```
#include <tee_api_defines.h>
#include <tee_api_types.h>
```

Include dependency graph for tee_ta_api.h:



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



Macros

- #define TA_EXPORT

Functions

- TEE_Result TA_EXPORT TA_CreateEntryPoint (void)
- void TA_EXPORT TA_DestroyEntryPoint (void)
- TEE_Result TA_EXPORT TA_OpenSessionEntryPoint (uint32_t paramTypes, TEE_Param params[TEE_↩ NUM_PARAMS], void ∗∗sessionContext)
- void TA_EXPORT TA_CloseSessionEntryPoint (void ∗sessionContext)
- TEE_Result TA_EXPORT TA_InvokeCommandEntryPoint (void ∗sessionContext, uint32_t commandID, uint32_t paramTypes, TEE_Param params[TEE_NUM_PARAMS])

14.9.1 Macro Definition Documentation

14.9.1.1 #define TA_EXPORT

14.9.2 Function Documentation

14.9.2.1 void TA_EXPORT TA_CloseSessionEntryPoint (void ∗ *sessionContext*)

14.9.2.2 TEE_Result TA_EXPORT TA_CreateEntryPoint (void)

14.9.2.3 void TA_EXPORT TA_DestroyEntryPoint (void)

14.9.2.4 TEE_Result TA_EXPORT TA_InvokeCommandEntryPoint (void ∗ *sessionContext,* uint32_t *commandID,* uint32_t *paramTypes,* TEE_Param *params[TEE_NUM_PARAMS]*)

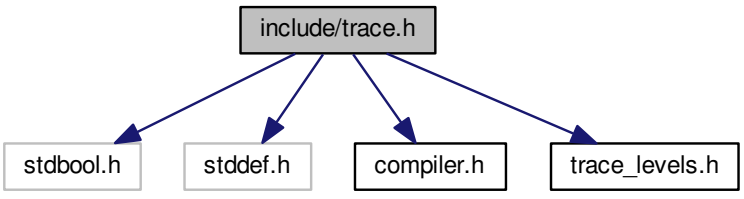
14.9.2.5 TEE_Result TA_EXPORT TA_OpenSessionEntryPoint (uint32_t *paramTypes,* TEE_Param *params[TEE_NUM_PARAMS],* void ∗∗ *sessionContext*)

14.10 include/test_dev_key.h File Reference

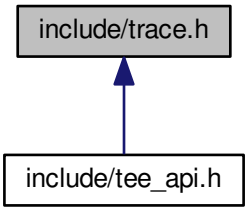
14.11 include/trace.h File Reference

```
#include <stdbool.h>
#include <stddef.h>
#include <compiler.h>
#include <trace_levels.h>
```

Include dependency graph for trace.h:



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



Macros

- #define MAX_PRINT_SIZE 256
- #define MAX_FUNC_PRINT_SIZE 32
- #define TRACE_LEVEL TRACE_MAX
- #define trace_printf_helper(level, level_ok, ...)
- #define MSG(...) (void)0
- #define EMSG(...) trace_printf_helper(TRACE_ERROR, true, __VA_ARGS__)
- #define IMSG(...) trace_printf_helper(TRACE_INFO, true, __VA_ARGS__)
- #define DMSG(...) trace_printf_helper(TRACE_DEBUG, true, __VA_ARGS__)
- #define FMSG(...) trace_printf_helper(TRACE_FLOW, true, __VA_ARGS__)
- #define INMSG(...) FMSG("> " __VA_ARGS__)
- #define OUTMSG(...) FMSG("< " __VA_ARGS__)
- #define OUTRMSG(r)
- #define DHEXDUMP(buf, len)
- #define trace_printf_helper_raw(level, level_ok, ...) trace_printf(NULL, 0, (level), (level_ok), __VA_ARGS__)
- #define MSG_RAW(...) (void)0
- #define EMSG_RAW(...) trace_printf_helper_raw(TRACE_ERROR, true, __VA_ARGS__)
- #define IMSG_RAW(...) trace_printf_helper_raw(TRACE_INFO, true, __VA_ARGS__)
- #define DMSG_RAW(...) trace_printf_helper_raw(TRACE_DEBUG, true, __VA_ARGS__)
- #define FMSG_RAW(...) trace_printf_helper_raw(TRACE_FLOW, true, __VA_ARGS__)
- #define SMSG(...) (void)0
- #define EPRINT_STACK() (void)0
- #define IPRINT_STACK() (void)0
- #define DPRINT_STACK() (void)0
- #define FPRINT_STACK() (void)0

Functions

- void trace_ext_puts (const char *str)
- int trace_ext_get_thread_id (void)
- void trace_set_level (int level)
- int trace_get_level (void)
- void trace_printf (const char *func, int line, int level, bool level_ok, const char *fmt,...) __printf(5
- void dhex_dump (const char *function, int line, int level, const void *buf, int len)

Variables

- int trace_level
- const char trace_ext_prefix [ ]

14.11.1 Macro Definition Documentation

14.11.1.1 #define DHEXDUMP(*buf, len*)

Value:

```
dhex_dump(__func__, __LINE__, TRACE_DEBUG, \
                  buf, len)
```

14.11.1.2 #define DMSG(...) trace_printf_helper(TRACE_DEBUG, true, __VA_ARGS__)

14.11.1.3 #define DMSG_RAW(...) trace_printf_helper_raw(TRACE_DEBUG, true, __VA_ARGS__)

14.11.1.4 #define DPRINT_STACK( ) (void)0

14.11.1.5 #define EMSG(...) trace_printf_helper(TRACE_ERROR, true, __VA_ARGS__)

14.11.1.6 #define EMSG_RAW(...) trace_printf_helper_raw(TRACE_ERROR, true, __VA_ARGS__)

14.11.1.7 #define EPRINT_STACK( ) (void)0

14.11.1.8 #define FMSG(...) trace_printf_helper(TRACE_FLOW, true, __VA_ARGS__)

14.11.1.9 #define FMSG_RAW(...) trace_printf_helper_raw(TRACE_FLOW, true, __VA_ARGS__)

14.11.1.10 #define FPRINT_STACK( ) (void)0

14.11.1.11 #define IMSG(...) trace_printf_helper(TRACE_INFO, true, __VA_ARGS__)

14.11.1.12 #define IMSG_RAW(...) trace_printf_helper_raw(TRACE_INFO, true, __VA_ARGS__)

14.11.1.13 #define INMSG(...) FMSG("> " __VA_ARGS__)

14.11.1.14 #define IPRINT_STACK( ) (void)0

14.11.1.15 #define MAX_FUNC_PRINT_SIZE 32

14.11.1.16 #define MAX_PRINT_SIZE 256

14.11.1.17 #define MSG(...) (void)0

14.11.1.18 #define MSG_RAW(...) (void)0

14.11.1.19 #define OUTMSG(...) FMSG("< " __VA_ARGS__)

14.11.1.20 #define OUTRMSG(*r*)

Value:

```
do {                \
        OUTMSG("r=[%x]", r);    \
        return r;       \
    } while (0)
```

14.11.1.21 #define SMSG(...) (void)0

14.11.1.22 #define TRACE_LEVEL TRACE_MAX

14.11.1.23 #define trace_printf_helper(level, level_ok, ...)

Value:

```
trace_printf(__func__, __LINE__, (level), (level_ok), \
             __VA_ARGS__)
```

14.11.1.24 #define trace_printf_helper_raw(level, level_ok, ...) trace_printf(NULL, 0, (level), (level_ok), __VA_ARGS__)

14.11.2 Function Documentation

14.11.2.1 void dhex_dump (const char * function, int line, int level, const void * buf, int len)

14.11.2.2 int trace_ext_get_thread_id (void)

14.11.2.3 void trace_ext_puts (const char * str)

14.11.2.4 int trace_get_level (void)

14.11.2.5 void trace_printf (const char * func, int line, int level, bool level_ok, const char * fmt, ...)

14.11.2.6 void trace_set_level (int level)

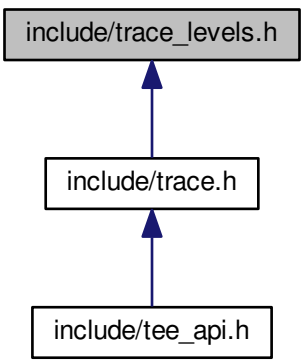
14.11.3 Variable Documentation

14.11.3.1 const char trace_ext_prefix[ ]

14.11.3.2 int trace_level

14.12 include/trace_levels.h File Reference

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



Macros

- #define TRACE_MIN 1
- #define TRACE_ERROR TRACE_MIN
- #define TRACE_INFO 2
- #define TRACE_DEBUG 3
- #define TRACE_FLOW 4
- #define TRACE_MAX TRACE_FLOW
- #define TRACE_PRINTF_LEVEL TRACE_ERROR

14.12.1 Macro Definition Documentation

14.12.1.1 #define TRACE_DEBUG 3

14.12.1.2 #define TRACE_ERROR TRACE_MIN

14.12.1.3 #define TRACE_FLOW 4

14.12.1.4 #define TRACE_INFO 2

14.12.1.5 #define TRACE_MAX TRACE_FLOW

14.12.1.6 #define TRACE_MIN 1

14.12.1.7 #define TRACE_PRINTF_LEVEL TRACE_ERROR

14.13 mainpage.md File Reference

14.14 message-digest.md File Reference

14.15 README.md File Reference

14.16 secure-storage.md File Reference

14.17 symmetric-key-varification.md File Reference

Index