

# Middleware Universale

## Per smart card Incard

### Indice

DEFINIZIONI, ACRONIMI, ABBREVIAZIONI.....	3
MIDDLEWARE UNIVERSALE.....	5
CARATTERISTICHE DEL MODULO CSP.....	7
CARATTERISTICHE DEL MODULO PKCS#11.....	8
CARATTERISTICHE DEL MODULO DI GESTIONE DELLA SMART CARD.....	10
FILE DI CONFIGURAZIONE DEL MU.....	12

Revisione	Autori	Note
A (20/08/2007)	Giuseppe Amato Vincenzo Palazzo	
B (04/09/2007)	Giuseppe Amato	Aggiunto Windows Vista alla lista dei sistemi operativi supportati
C (17/10/2007)	Giuseppe Amato	Aggiornato paragrafo “Peculiarità del MU e non conformità con la specifica PKCS#11”. Aggiornato paragrafo “Creazione dei diversi tipi di oggetti: CNS, Firma Forte, FullP11”.
D (26/10/2007)	Giuseppe Amato	Aggiornato paragrafo “Peculiarità del MU e non conformità con la specifica PKCS#11”. Aggiornato paragrafo “Creazione dei diversi tipi di oggetti: CNS, Firma Forte, FullP11”. Aggiornata lista delle “Smart Card supportate in maniera nativa”. Aggiornata la lista dei “Sistemi operativi supportati”
E (29/11/2007)	Giuseppe Amato	Aggiunto Windows Vista ai sistemi operativi usati per il collaudo prima del rilascio.
F (05/12/2007)	Giuseppe Amato	Aggiornata documentazione del Modulo di gestione del PIN Aggiornata documentazione del criterio di selezione del Container di default
G (21/01/2008)	Giuseppe Amato	Aggiornata documentazione del Modulo di gestione del PIN Aggiornata documentazione della gestione del PIN di firma forte
H (04/06/2008)	Giuseppe Amato	Aggiornata elenco delle smartcard/filesystem supportati: aggiunta FS-DS-v2.0 2048bit Aggiornata sezione “Creazione dei diversi tipi di oggetti: CNS, Firma Forte, FullP11”
I (14/07/2008)	Giuseppe Amato	Aggiunta sezione: “Creazione dei diversi tipi di oggetti (CNS, Firma Forte, etc.)” Aggiunta sezione: “Peculiarità del MU e non conformità con la specifica CSP/CryptoAPI”
J (14/10/2008)	Giuseppe Amato	Aggiornata documentazione del Modulo di gestione Smart Card Aggiornata la lista dei “Sistemi operativi supportati”

## Definizioni, Acronimi, abbreviazioni

<b>PKCS#11</b>	interfaccia di programmazione (API) standard, multi piattaforma, per l'accesso a generici token crittografici, quali le smart card, sviluppata da RSA
<b>Libreria o modulo PKCS#11</b>	Modulo software che implementa della API PKCS#11 specifica per uno o più token crittografici di un determinato produttore.
<b>CSP</b>	Interfaccia di programmazione (API) proprietaria Microsoft che permette di aggiungere funzioni crittografiche, anche fornite da hardware come le smart card, nei sistemi operativi Windows; un CSP è un modulo software che può essere utilizzato esclusivamente tramite API crittografiche del sistema operativo (CryptoAPI)
<b>CryptoSPI</b>	Acronimo che sta per Crypto Service Provider Interface, indica in modo specifico la API che un modulo CSP deve implementare.
<b>API</b>	Acronimo che sta per Application Programming Interface, indica ogni insieme di procedure disponibili al programmatore, di solito raggruppate a formare un set di strumenti specifici per un determinato compito
<b>CryptoAPI</b>	Acronimo che sta per Cryptographic Application Programming Interface; rappresenta l'interfaccia di programmazione che i sistemi operativi Windows mettono a disposizione delle applicazioni per l'uso della crittografia.
<b>Tray-bar</b>	Nei sistemi operativi Microsoft Windows rappresenta l'area localizzata tra la barra delle applicazioni e l'orologio, in cui le applicazioni possono installare un'icona che le rappresenti quando non sono in primo piano.
<b>PIN</b>	Acronimo di Personal Identification Number; nell'ambito delle smart card rappresenta un codice che permette di accedere alle funzioni il cui uso è riservato esclusivamente al possessore della carta; generalmente il PIN si blocca dopo un numero predefinito di tentativi con valori errati, bloccando dunque l'accesso alla smart card
<b>PUK</b>	Acronimo che sta per Pin Unblocking Key; nell'ambito delle smart card è un codice del tutto simile al PIN, il cui scopo generalmente è esclusivamente quello di sbloccare un PIN bloccato dai troppi tentativi con valori non corretti.
<b>CNS</b>	Carta Nazionale dei Servizi; in questo documento può indicare la specifica CNS rilasciata dal CNIPA oppure le sole specifiche del file system.
<b>MU</b>	Middleware Universale, il software in oggetto
<b>file system</b>	Nell'ambito delle smart card indica la struttura ed il formato dei file e dei dati presenti su una smart card e che servono a implementare una determinata funzionalità/applicazione.
<b>ATR</b>	Acronimo che sta per <b>Answere To Reset</b> ; è un codice restituito da una smart card quando viene inserita nel lettore o resettata che viene spesso utilizzato per identificare il tipo di smart card in maniera univoca.
<b>Store di certificati</b>	Rappresenta il punto in cui il sistema operativo Windows memorizza i certificati di sicurezza, in modo che possano essere utilizzati dalle applicazioni che fanno uso delle CryptoAPI
<b>SSL</b>	Acronimo che sta per <b>Secure Socket Layer</b> : protocollo standard di comunicazione cifrata che permette anche la mutua autenticazione tra le parti comunicanti (SSL Server authentication e Client Authentication)
<b>TLS</b>	Acronimo che sta per <b>Transport Layer Security</b> : è il successore del protocollo SSL.
<b>FS</b>	Acronimo che sta per File System
<b>DS</b>	Acronimo che sta per Digital Signature: indica il filesystem di firma digitale (in questo documento può indicare anche un file system non CNS)
<b>DS-v1.0, DS-v2.0</b>	Rispettivamente le implementazioni 1.0 e 2.0 del filesystem DS; la versione 2.0 supporta le chiavi a 2048bit
<b>FullP11</b>	Indica il filesystem che supporta la maggior parte degli attributi definiti dalla specifica PKCS#11



## Middleware Universale

Il "Middleware Universale" (in seguito MU) consiste in:

- Modulo di libreria che espone una API compatibile con specifica di standard PKCS#11 v2.11.
- Modulo di sistema che espone una API compatibile con specifica di standard Microsoft Cryptographic Service Provider (CryptoSPI/2006).
- Modulo di sistema "certificate store" che implementa un meccanismo di importazione automatica dei certificati nello store utente windows.
- Modulo utente per la gestione PIN/PUK (cambio PIN, sblocco PIN).

Tali moduli offrono ai software che ne utilizzano le relative interfacce di programmazione la possibilità di utilizzare le smart card supportate come token crittografici.

Smart Card supportate in maniera nativa:

- Incard Incrypto34 V2 con filesystem: CNS + Firma Digitale + FullP11:
  - Filesystem CNS conforme alla specifica CNS del CNIPA (v.1.1.3) (FS CNS)
  - FileSystem di firma digitale a validità legale (FS DS-v1.0)
  - FileSystem full compliant PKCS#11 (FS FullP11)
- Incard Touch&Sign2048:
  - Tutti i filesystem supportati sulla Incrypto34 V2 ed in più:
    - FileSystem full compliant PKCS#11 (FS FullP11), con oggetti a 2048 bit
    - FileSystem di firma digitale a validità legale (FS DS-v2.0), con tre coppie di chiavi a 2048bit oppure con tre coppie di chiavi a 1024bit più tre coppie di chiavi a 2048bit

Smart Card supportate tramite l'uso di librerie PKCS#11 esterne:

- Incard CryptoSmartCard16 (SetecOS): Infocamere N/S 1201..
- Incard CryptoSmartCardE4H (Starcos): Infocamere N/S 1202..
- Incard M4.01a FS1111 (Siemens CardOS/M4): Infocamere N/S 1203..

### Le funzionalità garantite dall'interfaccia PKCS#11 sono le seguenti:

Per il filesystem CNS:

- Firma digitale e decifra tramite la chiave RSA di autenticazione CNS,
- Lettura e scrittura del certificato di autenticazione CNS,
- Lettura e scrittura del file dei dati personali CNS (PDATA)
- Generazione di una coppia di chiavi RSA di autenticazione CNS
- Cancellazione degli oggetti

Per il filesystem di Firma digitale:

- Firma digitale tramite la chiave RSA di firma forte,
- Generazione di una coppia di chiavi RSA di Firma forte
- Lettura e scrittura di certificati di firma forte,
- Cancellazione degli oggetti

Per il filesystem Full P11:

- Firma digitale e decifra tramite le chiavi RSA,
- Generazione di una coppia di chiavi RSA
- Importazione di coppie di chiavi RSA
- Lettura e scrittura di certificati,
- Cancellazione degli oggetti

Per le smart card gestite tramite moduli PKCS#11 esterni (\*):

- Firma digitale e decifra tramite le chiavi RSA,
- Generazione di una coppia di chiavi RSA
- Importazione di coppie di chiavi RSA
- Lettura e scrittura di certificati,
- Cancellazione degli oggetti

(\*) Le funzionalità dipendono dal modulo PKCS#11 esterno utilizzato.

**Le funzionalità garantite dall'interfaccia CSP sono le seguenti:**

Per il filesystem CNS:

- Firma digitale e decifra tramite la chiave RSA di autenticazione CNS,
- Lettura e del certificato di autenticazione CNS,
- Generazione di coppia di chiavi RSA di autenticazione CNS
- Importazione del certificato di autenticazione CNS

Per il filesystem di Firma digitale:

- Firma digitale e decifra tramite le chiavi RSA di firma forte
- Lettura dei certificati di firma forte,
- Generazione di coppia di chiavi RSA di firma forte
- Associazione di un certificato ad una coppia di chiavi di firma forte

Per il filesystem Full P11:

- Firma digitale e decifra tramite le chiavi RSA
- Lettura dei certificati,
- Generazione di coppia di chiavi RSA
- Associazione di un certificato ad una coppia di chiavi

Per le smart card gestite tramite moduli PKCS#11 esterni (\*):

- Firma digitale e decifra tramite le chiavi RSA
- Lettura dei certificati di firma forte,
- Generazione di coppia di chiavi RSA
- Associazione di un certificato ad una coppia di chiavi

(\*) Le funzionalità dipendono dal modulo PKCS#11 esterno utilizzato.

Il modulo utente di gestione PIN-PUK è una applicazione attivabile mediante una icona presente nella tray-bar. Consente il cambio del PIN, lo sblocco del PIN mediante PUK.

Il MU comunica con le smart card attraverso un lettore di smart card controllato dallo strato PC/SC implementato nel sistema operativo.

I moduli che espongono le due interfacce PKCS#11 e CSP si appoggiano su di un unico "motore" che gestisce gli oggetti di sicurezza presenti sulle smart card CNS. Tale motore ha la possibilità di essere esteso attraverso un meccanismo di "plug-in".

Il modulo che espone l'interfaccia PKCS#11 viene interfacciato direttamente dalle applicazioni che utilizzano tale API per utilizzare i servizi offerti dalle smart card. Attraverso l'interfaccia PKCS#11 è possibile leggere i certificati ed utilizzare la chiave RSA associata.

Il modulo che espone l'interfaccia CryptoSPI viene interfacciato dal sistema operativo che integra le funzioni esposte con quelle di più alto livello del CSP che poi saranno messe a disposizione delle applicazioni.

Il modulo "certificate store" viene interfacciato dal sistema operativo che estende in questo modo lo store di certificati "logico" dell'utente (o store "My") estendendolo utilizzando uno store di certificati "fisico" afferente alla smart card. Tale modulo consente l'uso dei certificati presenti sulla smart card da parte delle applicazioni che fanno uso delle CryptoAPI in maniera del tutto automatica e trasparente. Alla rimozione della smart card i certificati verranno automaticamente rimossi. Il modulo "certificate store" è opzionale e nel caso venga disattivato sarà il sistema operativo, all'inserimento della carta, ad importare automaticamente i certificati nello store dei certificati personali; in tal caso i certificati saranno visibili nel sistema anche dopo la rimozione della smart card.

Il modulo "gestione PIN/PUK" non espone una interfaccia di programmazione (API) ma ha una interfaccia utente (GUI) che consente all'utente di svolgere le minimali attività per il cambio e lo sblocco del PIN. L'interfaccia utente può essere attivata intervenendo su di una icona presente nella tray-bar che fornisce inoltre anche informazioni sull'attuale stato di attività della smart card.

**Sistemi operativi supportati**

- Windows 2000 SP4
- Windows XP SP2
- Windows Vista
- Windows Server 2003
- MacOS X 10.4.x e 10.5.x (Intel, PPC)
- Le seguenti distribuzioni Linux:

- Debian 3.x o 4.x
- Ubuntu 7.x
- Mepis

## Caratteristiche del modulo CSP

1. Libreria compatibile con la specifica CSP di Microsoft
2. Funzionamento del CSP limitato all'utilizzo nei seguenti contesti applicativi:
  1. autenticazione SSL V3 con Microsoft IE,
  2. funzione di "Firma leggera per Attestazione" per applicazioni web
  3. funzione di "Firma Forte" per applicazioni di firma digitale
3. Sistemi operativi sui quali Bit4id certifica il pieno funzionamento ed il superamento dei propri test prima del rilascio:
  1. Windows 2000 SP4, WindowsXP SP2, Windows Server 2003, Vista Business Premium
4. Rilascio libreria in formato binario sotto forma di libreria a "link dinamico".
5. Applicazione con la quale Bit4id certifica il pieno funzionamento con il superamento dei propri test prima del rilascio: Microsoft IE vers. 6
6. Caricamento dei certificati presenti sulla smart card in maniera trasparente per l'utente all'inserimento nel lettore.

L'interfaccia CSP implementa le seguenti funzioni:

- CryptGetProvParam (PP\_NAME, PP\_CONTAINER, PP\_UNIQUE\_CONTAINER)
- CryptSetProvParam (PP\_SIGNATURE\_PIN)
- CryptAcquireContext
- CryptReleaseContext
- CryptCreateHash
- CryptSetHashParam
- CryptGetHashParam (HP\_ALGID, HP\_HASHSIZE, HP\_HASHVAL)
- CryptHashData,
- CryptDestroyHash
- CryptSignHash
- CryptGetUserKey
- CryptDestroyKey
- CryptGetKeyParam (KP\_CERTIFICATE)
- CryptExportKey (PUBLICKEYBLOB)
- CryptSetKeyParam (KP\_CERTIFICATE)
- CryptGenKey (ALGID: RSA )
- CryptImportKey (SIMPLEBLOB)

## Container di default del CSP

Per l'uso con applicazioni di logon (ad esempio l'accesso alla workstation con smart card) il sistema operativo richiede la presenza sulla smart card di un container di default.

Il modulo CSP considera il primo container che viene trovato sulla carta come quello di default.

Tuttavia ogni volta che tramite l'interfaccia CSP viene importato un certificato che contiene le estensioni X509 specifiche per smart card logon, il container corrispondente viene impostato automaticamente come default.

Internamente la libreria crea un oggetto dati PKCS#11 (CKA\_CLASS=CKO\_DATA) con attributo CKA\_LABEL "CSPSmartLogon" ed il cui contenuto (CKA\_VALUE) coincide con l'identificativo del container di default. E' possibile aggiornare o creare manualmente questo oggetto per modificare il container di default, usando l'interfaccia PKCS#11 del middleware universale. Quando tale oggetto è assente il comportamento è il seguente: viene cercato un certificato con CKA\_ID pari al valore "SmartLogon"; se è assente viene selezionato il primo certificato che possiede le estensioni per lo smart card logon. Se non esiste un certificato del genere il primo certificato trovato diventa quello di default.

## Creazione dei diversi tipi di oggetti (CNS, Firma Forte, etc.)

Vedere nel prossimo paragrafo la sezione "**Creazione dei diversi tipi di oggetti: CNS, Firma Forte, FullP11**", con la differenza che quando si parla di CKA\_ID/CKA\_LABEL, per il CSP si intende il **Container Name**.

## Peculiarità del MU e non conformità con la specifica CSP/CryptoAPI

Il MU ha un limite di **39** caratteri per il **Container Name**. Nel caso si richiami la funzione CryptAcquireContext() specificando un container name di lunghezza maggiore di 39 caratteri verrà restituito l'errore NTE\_BAD\_KEYSET.

## Caratteristiche del modulo PKCS#11

1. Libreria compatibile con la specifica PKCS#11 di RSA (v. 2.11)
2. Funzionamento del PKCS#11 limitato all'utilizzo nei seguenti contesti applicativi:
  1. autenticazione SSL V3 con il browser Mozilla FireFox 2.0,
  2. funzione di "Firma leggera per Attestazione" per applicazioni web
  3. funzione di "Firma Forte" per applicazioni di firma digitale
  4. funzione di enrollment di credenziali CNS, di firma forte, di autenticazione tramite Software CA di Infocamere
3. Sistemi operativi sui quali Bit4id certifica il pieno funzionamento ed il superamento dei propri test prima del rilascio:
  1. Windows 2000 SP4, Windows XP SP2, Windows Server 2003, Vista Business Premium
4. Rilascio del modulo PKCS#11 in formato binario come libreria a "link dinamico".
5. Applicazione con la quale Bit4id certifica il pieno funzionamento con il superamento dei propri test prima del rilascio: Browser Mozilla Firefox vers. 2.0

L'interfaccia PKCS#11 implementa le seguenti funzioni:

- C\_Initialize
- C\_Finalize
- C\_GetInfo
- C\_GetSlotList
- C\_GetSlotInfo
- C\_GetTokenInfo
- C\_GetMechanismList
- C\_GetMechanismInfo
- C\_OpenSession
- C\_CloseSession
- C\_CloseAllSessions
- C\_GetSessionInfo
- C\_Login (CKU\_USER, CKU\_SO)
- C\_Logout
- C\_SetPIN
- C\_FindObjectsInit
- C\_FindObjects
- C\_FindObjectsFinal
- C\_GetAttributeValue
- C\_GetObjectSize
- C\_SignInit (meccanismi RSA\_PKCS e RSA\_SHA1\_PKCS)
- C\_Sign
- C\_DecryptInit (meccanismo RSA\_PKCS)
- C\_Decrypt
- C\_DigestInit (meccanismo SHA\_1)
- C\_Digest
- C\_DigestUpdate
- C\_DigestFinal
- C\_CreateObject
- C\_GenerateKeyPair (meccanismo RSA\_PKCS\_KEY\_PAIR\_GEN)
- C\_SetAttributeValue
- C\_DestroyObject

### Mechanisms supportati:

CKM\_RSA\_PKCS\_KEY\_PAIR\_GEN

CKM\_RSA\_PKCS (firma, decifra)

CKM\_RSA\_SHA1\_PKCS (firma)

CKM\_SHA\_1 (digest)

### Creazione dei diversi tipi di oggetti: CNS, Firma Forte, FullP11

Il Middleware Universale riserva i valori di alcuni attributi per stabilire il filesystem sul quale creare gli oggetti:

- Quando CKA\_ID oppure CKA\_LABEL di una chiave RSA o un certificato ha come valore '**CNS0**' viene creato un oggetto nel filesystem CNS. Nel caso in l'oggetto da creare esista già la libreria restituisce l'errore CKR\_DEVICE\_MEMORY.
- Quando si crea un oggetto dati con CKA\_LABEL oppure CKA\_APPLICATION pari a '**PDATA**' viene scritto il file dei dati personali CNS. Nel caso l'oggetto esista già la libreria restituisce l'errore CKR\_DEVICE\_MEMORY.



- Quando CKA\_ID oppure CKA\_LABEL di una chiave RSA o un certificato è 'DS0', 'DS1', 'DS2' viene selezionato il Filesystem di firma forte a **1024** bit, rispettivamente usando lo "slot" **0, 1 o 2**. Nel caso l'oggetto esista già la libreria restituisce l'errore CKR\_DEVICE\_MEMORY. Non è possibile importare chiavi RSA usando tali valori riservati.
- Quando CKA\_ID oppure CKA\_LABEL di una chiave RSA o un certificato è 'DS3', 'DS4', 'DS5' viene selezionato il Filesystem di firma forte a **2048** bit, rispettivamente usando lo "slot" **3, 4 o 5**. Nel caso l'oggetto esista già la libreria restituisce l'errore CKR\_DEVICE\_MEMORY. Non è possibile importare chiavi RSA usando tali valori riservati.
- Quando CKA\_ID oppure CKA\_LABEL di una chiave RSA o un certificato è 'DS' viene selezionato il Filesystem di firma forte e la libreria usa il primo slot disponibile per quel tipo di oggetto. Per gli oggetti a **1024** bit vengono usati gli slot **0, 1 e 2**; per gli oggetti a **2048** bit vengono usati gli slot **3, 4 e 5**. Nel caso non esistano slot disponibili la libreria restituisce l'errore CKR\_DEVICE\_MEMORY. Non è possibile importare chiavi RSA usando tale valore riservato, ma solo generare una coppia di chiavi.
- In tutti gli altri casi gli oggetti sono creati nel FileSystem FullP11.
- Se il filesystem CNS è assente, gli oggetti corrispondenti verranno creati nel filesystem FullP11.
- Se il filesystem di firma forte è assente, gli oggetti corrispondenti verranno creati nel filesystem FullP11.
- Il filesystem PKCS#11 supporta le chiavi RSA a **2048** bit. Se si tenta di generare una chiave 2048 bit nel filesystem CNS la libreria restituirà un errore.
- Il filesystem DS-v2.0 supporta le chiavi RSA a 2048 bit. Se si tenta di generare una chiave 2048 bit nel filesystem DS-v1.0 la libreria restituirà un errore.

### Peculiarità del MU e non conformità con la specifica PKCS#11

La funzione C\_Initialize() restituisce sempre CKR\_OK anche se la libreria è stata già inizializzata, per motivi di compatibilità con applicazioni non aderenti al 100% allo standard. Per capire se la libreria sia già stata inizializzata è possibile usare la funzione C\_GetInfo() che restituisce CKR\_CRYPTOKI\_NOT\_INITIALIZED.

I filesystem CNS e di firma forte non permettono di memorizzare su smart card tutti gli attributi PKCS#11. Alcuni attributi, tra cui CKA\_ID e CKA\_LABEL, torneranno ai loro valori di default una volta che la carta sia stata sfilata dal lettore (ovvero la libreria sia stata scaricata dalla memoria).

Quando gli attributi CKA\_SUBJECT, CKA\_ISSUER, CKA\_SERIAL\_NUMBER di un certificato non sono memorizzati (perché non specificati durante la creazione o per limiti del filesystem) i rispettivi valori verranno estratti dal certificato stesso.

La libreria associa il PIN principale all'utente CKU\_USER ed il PUK all'utente CKU\_SO. Poiché la specifica PKCS#11 non prevede la possibilità di cambiare o sbloccare PIN aggiuntivi, il PIN ed il PUK di firma forte vengono gestiti sempre parallelamente a PIN e PUK principali. A seconda della configurazione della libreria, ogni volta che si si cambia PIN o PUK o si sblocca il PIN tramite PUK potrebbero essere chiesti all'utente tramite interfaccia grafica i valori di PIN e/o PUK per la firma forte.

Per cambiare i PIN è necessario loggarsi con l'utente CKU\_USER ed usare la funzione C\_SetPIN().

Per cambiare i PUK è necessario loggarsi con l'utente CKU\_SO ed usare la funzione C\_SetPIN().

Per attivare il PIN di firma forte è necessario loggarsi con l'utente CKU\_SO ed usare la funzione C\_InitPIN().

Per sbloccare i PIN, o semplicemente reimpostarli ad un nuovo valore, è necessario loggarsi con l'utente CKU\_SO ed usare la funzione C\_InitPIN().

Se il PIN di firma forte è già stato inizializzato, la funzione C\_InitPIN() sbloccherà i PIN, principale e di firma forte. A seconda della configurazione della libreria il PUK ed il nuovo PIN di firma forte potrebbero essere richiesti all'utente tramite interfaccia grafica.

L'assenza del FLAG CKF\_USER\_PIN\_INITIALIZED nella struttura CK\_TOKEN\_INFO restituita dalla funzione C\_GetTokenInfo() indica che è necessario richiamare la funzione C\_InitPIN() per attivare il PIN di firma forte.

C\_Sign() supporta la firma RSA RAW: se viene passato un blocco di lunghezza pari alla lunghezza del modulo della chiave, non viene effettuato il padding PKCS#1 per la firma. La funzionalità è disponibile solamente se la chiave selezionata supporta la firma RAW (le chiavi di firma forte ad esempio non la supportano). In tal caso la libreria potrebbe restituire gli errori CKR\_FUNCTION\_FAILED, CKR\_DEVICE\_ERROR o CKR\_GENERAL\_ERROR.

Per limiti del Filesystem di firma forte, su tale filesystem è possibile solamente generare coppie di chiavi RSA e non importarle. Nel caso in cui si tenta di importare chiavi RSA di firma forte su filesystem DS verrà restituito dalla funzione C\_CreateObject() l'errore CKR\_TEMPLATE\_INCONSISTENT.

La funzione C\_GetTokenInfo() non restituisce l'informazione sullo spazio libero residuo per un limite della smart card: il valore restituito è CK\_UNAVAILABLE\_INFORMATION.

Per motivi di compatibilità con software preesistente è possibile loggarsi come SO anche su sessioni ReadOnly. La sessione verrà trasformata internamente in Read/Write. Una volta effettuato il logoff la sessione resterà Read/Write.

La libreria supporta oggetti RSA 2048 bit nel filesystem FullP11, quando la carta è una touch&Sign2048. Pertanto il mechanism\_info relativo al mechanism CKM\_RSA\_PKCS\_KEY\_PAIR\_GEN riporterà il valore 2048 nel campo

“ulMaxKeySize”. Tuttavia nel caso in cui si tenti di generare chiavi RSA 2048 bit in filesystem diverso (CNS, DS) la libreria restituirà l'errore CKR\_ATTRIBUTE\_VALUE\_INVALID.

## Caratteristiche del modulo di gestione della Smart Card

Il modulo di gestione Smart Card è un'applicazione dotata di interfaccia utente (GUI) che permette di gestire alcune funzionalità del Middleware Universale.

L'applicazione permette di eseguire le seguenti operazioni:

- Ottenere informazioni sulla smart card inserita, quali il numero di serie, il modello e il lettore in uso.
- Abilitare o disabilitare il modulo CSP per la smart card inserita
- Importare eventuali certificati di CA dalla smart card allo store di sistema di Windows
- Selezionare il certificato per il logon con smart card

Tutte le operazioni sono eseguite utilizzando l'interfaccia PKCS#11 messa a disposizione dal MU.

L'applicazione, quando la finestra principale non è visibile, è nascosta e mostrerà esclusivamente un'icona nella barra delle applicazioni, accanto all'orologio (tray bar). Facendo doppio click su tale icona si attiva/disattiva la finestra principale dell'applicazione, l'applicazione rimarrà attiva in background e sarà riattivabile in due modi: lanciando nuovamente l'applicazione oppure facendo doppio click sull'icona che l'applicazione installa nella tray bar; per terminare completamente l'applicazione si può usare la voce “Uscita” presente nel menu contestuale che appare cliccando sull'icona che l'applicazione visualizza nella tray-bar.



Per ogni sessione utente può esistere una singola istanza dell'applicazione; se essa viene eseguita più volte verrà attivata la finestra dell'istanza precedente.

L'applicazione è caratterizzata da un'interfaccia a schede (o tab), ognuna delle quali fornisce una funzionalità; la prima scheda è la principale e fornisce le informazioni sulla smart card.



Quando non è inserita nessuna smart card oppure se l'unica smart card inserita non è riconosciuta la scheda principale sarà l'unica visibile.



Nel caso in cui non siano installato un lettore di smart card l'applicazione mostrerà la sola scheda delle informazioni indicando che non è stato rilevato alcun lettore.



## Impostazioni avanzate



Nella scheda “Avanzate” è possibile associare la smart card inserita al CSP del Middleware Universale: è infatti necessario che l'ATR della carta inserita sia correttamente associato al CSP perché possa essere riconosciuta da applicazioni come Internet Explorer e Outlook Express.

Dalla scheda avanzate è possibile importare i certificati di ROOT CA presenti sulla smart card nello store dei certificati di Windows “Autorità di certificazione attendibili”, usando il pulsante “Importa” presente nell'area “Certificati di CA”.

E' infine possibile selezionare un certificato da usare per lo il logon con smart card, usando il pulsante “Esegui” presente nell'area “Smart Card Logon”.



Cliccando sul pulsante “Seleziona” il programma chiederà il PIN e configurerà la smart card in modo che il certificato selezionato sia usato per il logon.



## File di configurazione del MU

Il MU ha un file di configurazione che permette di variarne il comportamento.

Il file di configurazione, che si chiama **bit4ipki.dll.conf** DEVE sempre trovarsi nella stessa cartella in cui si trova il modulo principale bit4ipki.dll.

### Formato del file di configurazione

Il file di configurazione è composto da una serie di righe ed ogni riga contiene una stringa del tipo NomeValore=Valore; sono ammesse righe vuote.

### Contenuto del file di configurazione

Le impostazioni utilizzabili nel file di configurazione sono le seguenti:

Nome	Possibili Valori	Descrizione
DSPinIsCnsPin	true o false	Se impostato a "true" viene indicato forzata l'uguaglianza del pin primario con il pin di firma. Valore di default: false
DSPinUseGui	true o false	Se impostato a "false" la libreria PKCS#11 gestisce un eventuale PIN secondario che protegge gli oggetti di firma in maniera compliant col la specifica PKCS#11, restituendo dunque l'errore CKR_USER_NOT_LOGGED_IN in seguito ad una chiamata alla funzione C_Sign su una chiave protetta da quel PIN, aspettandosi immediatamente dopo una chiamata a C_Login(CKU_CONTEXT_SPECIFIC) col PIN di firma. Se impostato a "true" verrà utilizzato il plugin per la richiesta del PIN secondario tramite interfaccia grafica (GUI). Il parametro è ignorato quando DSPinIsCnsPin è impostato a "true". Valore di default: true
HideCacheDsPinCheck	true o false	Se impostato a true viene nascosto dall'interfaccia utente che richiede il PIN di firma forte il checkbox che permette di utilizzare lo stesso PIN per più operazioni di firma in sequenza, fino al logout dalla carta. Valore di default: false