è una società del **Gruppo Maggioli**

# DIGITAL SIGNATURE LIBRARY

AUTORE: FABIO PARISE

VERSIONE LIBRERIA: 1.2.4

VERSIONE DOCUMENTO: 1.1

DATA: 12/07/2011

# 1. INTODUZIONE

Questa libreria mette a disposizione le funzionalità per la firma digitale di documenti e la verifica di un documento firmato digitalmente.

La libreria fornisce tali funzionalità utilizzando la libreria di terze parti di Ellipsis Java Toolkit fornita da Actalis S.p.A.. La versione di Ellipsis Java Toolkit attualmente in distribuzione è la 3.1.2.

## 1.1 Configurazione

La configurazione della libreria per il suo completo funzionamento all'interno di SicraWeb viene eseguita interamente dall'applicativo di setup e pertanto non è necessaria nessuna operazione di configurazione manuale per poter utilizzare le funzionalità della libreria.

Le informazioni di configurazione sono pertanto riportate per sola finalità conoscitiva.

La cartella home dell'application server usato verrà indicata con `%J2EE_HOME%`
La cartella locale di SicraWeb verrà indicata con `%SICRAWEB_HOME%`

***Librerie esterne necessarie:***

Assicurarsi che nella cartella `%SICRAWEB_HOME%\client\lib` sia presente il seguente jar:

- *siacapi-full-10-strip14.jar*
- *jpcsc.jar*
- *bc-mail.jar*
- *bc-prov.jar*

Assicurarsi che nella cartella `%SICRAWEB_HOME%\client\lib\native` sia presente il seguente jar:

- *it.saga.extern.siacapi.native.jar*

  Elenco DLL contenute:
  *actalisjpkcs11.dll, pcscutil.dll, pcscutiljni2.dll, pkcs11wrapper.dll*

***Configurazione JNLP:***

- Nel file %J2EE_HOME%\applications\SicraWeb\client\sicraweb-resurces.jnlp controllare che siano presenti i seguenti tags:

```
<package name="it.saga.library.digitalsignature.*" part="DSG"  recursive="true"/>

<nativelib href="/client/signed-jars/lib/native/it.saga.extern.siacapi.native.jar"/>

<!-- DigitalSignature jars -->
<jar href="/client/signed-jars/it.saga.library.digitalsignature.client.jar" part="DSG"
download="lazy"/>
```

# 2. RISOLUZIONE DEI PROBLEMI

## 2.1 Mancato riconoscimento delle smartcard

Quando si firma un documento, il software chiede all'utente di inserire la carta nel lettore ed esegue una procedura per il riconoscimento automatico del dispositivo di firma.
Può succedere che alcuni dispositivi (spesso quelli di nuova produzione) non siano riconosciuti dal sistema e che la libreria ritorni un messaggio di errore in fase di riconoscimento/inizializzazione del dispositivo. Ad esempio un messaggio di errore potrebbe essere il seguente:

*Si è verificato il seguente errore: Token not initalized.*

Per la soluzione del problema, si consiglia di eseguire nell'ordine le seguenti operazioni:

1) Identificare con il massimo dettaglio il dispositivo di firma in base alle informazioni fornite dal cliente e altri elementi identificativi: Produttore, tipo di carta (CNS, CRS), serie o numero di serie, dispositivo hw: smartcard o chiavetta USB.
   Esempi:
   a. *Smartcard Infocamere CNS serie 1204,*
   b. *Infocamere Business Key USB,*
   c. *CRS Regione Lombardia (prodotta da Siemens)*

2) verificare di aver installato i driver corretti per il lettore e la scheda oggetto del problema. Si faccia riferimento al sito web del produttore o alla tabella dei driver in questo stesso documento [vedi Elenco dei Driver].

3) Risalire al codice ATR del dispositivo.
   Lo si può fare analizzando il log (nella console di JAVA WEB START) del processo di riconoscimento del dispositivo. Vediamo come fare:

   - abilitare la console di JAVA WEB START
   - eseguire la procedura di firma che va in errore
   - copiare il contenuto della console ed inviarlo all'assistenza (segue esempio)

```
CONFIG FILE: C:\Documents and Settings\administrator.SICCCM\sicraweb-dsg.properties... not
found.
ATR HEX STRING=3B:FF:18:00:FF:C1:0A:31:FE:55:00:6B:05:08:C8:05:01:11:01:43:4E:53:10:31:80:0C
codice ATR (per sicraweb-dsg.properties): 3bff1800ffc10a31fe55006b0508c805011101434e531031800c
24-dic-2010 12.50.09 it.saga.library.digitalsignature.DsgActalis initToken
GRAVE: SMARTCART NON RICONOSCIUTA
24-dic-2010 12.50.10 it.saga.library.digitalsignature.DsgActalis handleCapiException
GRAVE: CAPI ERROR CODE: 10003 - Token not initalized
it.actalis.ellips.capi.CapiException: Token not initalized
    at it.actalis.ellips.capi.Token.getInfo(Unknown Source)
    at it.saga.library.digitalsignature.DsgActalis.initToken(DsgActalis.java:780)
    at it.saga.library.digitalsignature.DsgActalis.initToken(DsgActalis.java:556)
    at it.saga.library.digitalsignature.DsgProviderActalis.initToken(DsgProviderActalis.java:91)
    at it.saga.library.digitalsignature.DsgFRMFirma.initToken(DsgFRMFirma.java:350)
    at it.saga.library.digitalsignature.DsgFRMFirma.rilevaLettoreSmartCard(DsgFRMFirma.java:337)
```

   il codice ATR è quello riportato nella riga "`ATR HEX STRING=`" e nell'esempio è il seguente:

```
3B:FF:18:00:FF:C1:0A:31:FE:55:00:6B:05:08:C8:05:01:11:01:43:4E:53:10:31:80:0C
```

4) Identificare la libreria PKCS11
   In base al modello di carta ed al codice ATR cercare di identificare la libreria PKCS11 che pilota il dispositivo non riconosciuto (è una DLL).
   (Opzioni: cercare sul sito web del produttore, cercare su Internet, cercare nel file ATR.ini di Dike)

5) Verificare che la libreria PKCS11 sia presente nel sistema (altrimenti installarla) e verificare che la DLL sia in path del sistema operativo (di solito queste DLL vengono copiate nella WINDOWS\system32).

6) Suggerire alla libreria quale libreria PKCS11 utilizzare.
   Si può suggerire al programma quale libreria PKCS11 utilizzare per inizializzare il sistema. Per farlo è sufficiente creare il file *sicraweb-dsg.properties* nella user home dell'utente (Ad es. sulla mia macchina la user home è in C:\Documents and Settings\fabio\) e la si può evincere dal log (vedi sopra).

   Il file *sicraweb-dsg.properties* è un semplice file di testo e può essere aperto con un qualsiasi editor di testo. Per indicare quale libreria PKCS11 (DLL) utilizzare con la nostra carta è sufficiente inserire una riga in cui si riporta il codice ATR e separato dall' = il nome della libreria, come nell'esempio:

```
3bff1800ffc10a31fe55006b0508c805011101434e531031800c=cnsPKCS11.dll
```

## 2.2 File di configurazione: *sicraweb-dsg.properties*

Si tratta di un file di testo con righe di tipo proprietà=valore. Per inserire commenti utilizzare il carattere # ad inizio riga.

### 2.2.1 Forzare l'utilizzo della libreria PKCS11 per i dispositivi con il codice ATR specificato

Per forzare l'utilizzo della libreria PKCS11 per i dispositivi con un determinato codice ATR è sufficiente inserire nel file una proprietà pari al codice ATR (a destra dell'uguale) e come valore (a sinistra dell'uguale) il nome della libreria PKCS11 → `codiceATR=libPKCS11`
Esempio:

```
# Forzo la libreria cnsPKCS11.dll per i dispositivi con il codice ATR:
# 3B:FF:18:00:FF:C1:0A:31:FE:55:00:6B:05:08:C8:05:01:11:01:43:4E:53:10:31:80:0C
3bff1800ffc10a31fe55006b0508c805011101434e531031800c=cnsPKCS11.dll
```

### 2.2.2 Forzare l'utilizzo di una smartcard conosciuta (sconsigliato)

E' possibile inserire una voce *smartcard=NN* dove *NN* è il numero della smartcard che si vuole suggerire (forzare). Per l'elenco delle smartcard supportate fare riferimento al paragrafo 5.1 Modelli di smartcard conosciuti.

Ad es. per forzare l'uso della smartcard *"#5: Datakey 330"* è sufficiente scrivere nel file

```
# Forzo la seguente smattcard: Datakey 330
smartcard=5.
```

### 2.2.3 Forzare l'utilizzo di una libreria conosciuto (sconsigliato)

E' possibile anche forzare l'uso di una certa libreria nota usando la dicitura: *`token=NN`*
Per l'elenco delle librerie note fare riferimento al paragrafo 5.2 Librerie PKCS11 conosciute (token).

### 2.2.4 Forzare l'utilizzo di una libreria PKCS11 esplicita (sconsigliato)

```
# Forza l'utilizzo della libreria PKCS11 per TUTTI I DISPOSITIVI (da utilizzare solo nei casi
disperati)
pkcs11.library=libPKCS11
Esempio:
pkcs11.library=cnsPKCS11.dll
```

## 2.3 Aruba Key USB - Configurazione dei dispositivi di firma

Per questo tipo di dispositivi (USB) potrebbero esserci problemi di riconoscimento da parte della libreria di firma. Un esempio di errore potrebbe essere il seguente:

```
it.actalis.ellips.util.ATRException: SCardListReaders: 0x8010002e, General error.
       at it.actalis.ellips.util.ATR.getATRusingJPCSC(Unknown Source)
       at it.actalis.ellips.util.ATR.getATRusingJPCSC(Unknown Source)
       at it.actalis.ellips.capi.Token.getATR(Unknown Source)
```

Per la gestione degli errori segnalati (vedi allegato) è sufficiente che l'Arubakey venga switchata in modalità **CCID** attivando la funzione di **Import Certificato** (vedi sotto per dettagli). Segue una breve descrizione delle due modalità con cui può essere utilizzato questo dispositivo:

### 2.3.1 Modalità di funzionamento del dispositivo Aruba Key

In ambiente WINDOWS l'Arubakey prevede un interfacciamento basato su specifica PKCS#11/CSP. Le configurazioni del dispositivo sono sostanzialmente due:

ARUBA KEY HID (l'utente NON ha attivato la funzione di Import Certificato all'interno delle Utilities – vedi figura sotto)

- **INTERFACCIAMENTO BASATO SU PKCS#11**
- Smart card INCARD o Smart card OBERTHUR:
- Percorso del modulo Cryptoki: `Let_unità/Main/akxloader.dll`
- Questa modalità consente di utilizzare l'AK anche da parte delle applicazioni installate sul PC host.

L'unica limitazione è che l'applicativo che s'intende utilizzare può dialogare con il dispositivo solo attraverso interfaccia PKCS#11.
In questa modalità non è necessario leggere l'ATR della carta poiché il modulo `akxloader.dll` è automaticamente in grado di utilizzare il Cryptoki corretto per la particolare carta inserita.

ARUBA KEY CCID (l'utente HA attivato la funzione di Import Certificato all'interno delle Utilities – vedi figura sotto)

- **INTERFACCIAMENTO BASATO SU PKCS#11 e/o CSP**
- Smart card INCARD o Smart card OBERTHUR:
  Percorso del modulo Cryptoki: `Let_unità/Main/akxloader.dll`
  in alternativa possono essere utilizzate le seguenti librerie che vengono installate nel PC host dopo aver eseguito l'Import Certificato.
  Smart card INCARD (seriale della carta riportato sul retro della SIM):
  Percorso: `C:\WINDOWS\system32\bit4ipki.dll`
  Smart card OBERTHUR (seriale della carta NON riportato sul retro della SIM):
  Percorso: `C:\WINDOWS\system32\bit4opki.dll`

Questa modalità consente di utilizzare AK come una normale smartcard, quindi generalmente "visibile" e sfruttabile da una qualsiasi applicazione residente sul PC. In particolare non esistono le limitazioni della modalità HID e gli applicativi possono dialogare con AK sia attraverso interfaccia PKCS#11 che CSP. Per la lettura dell'ATR della carta è sufficiente basarsi sul servizio PC/SC di WINDOWS.

### 2.3.2 Configurazione Aruba Key in modalità CCID

1) inserire la Aruba Key
2) far partire l'autorun, nel menù della key andare sul menù "Utilities-Import certificato".
Questa procedura installa alcune dll nella system32 e scarica i certificati sul pc
3) riavviare il pc

4) nella cartella dell'utente corrente collegato al s.o. (C:\Documents and Settings\utente o similari) creare un file denominato "sicraweb-dsg.properties"
5) editare il file con "notepad" e aggiungere la seguente stringa:

```
3bdb960080b1fe451f830031c064c30801000f90009b=bit4ipki.dll
```

dove "bit4pki.dll" è la libreria che viene installata dalla procedura di Import certificati e dove quel codice lunghissimo è il "codice ATR".
Tale codice sono riuscito a ricavarlo dalla console di Java del client, come potrete vedere nel log allegato.
Nel log di java vedrete la prima operazione fatta senza settare "sicraweb-dsg.properties" e di seguito l'operazione andata a buon fine.

6) verificare che la versione di JRE sia 1.6.0_23

## 2.4 Carte CRS (Siemens) - Regione Lombardia

1) installare i driver della carta in oggetto: **Download CNS API 1.0.3 Build 5**
https://www.firma.infocert.it/software/CNS%20API%201%5B1%5D.0.3%20build%205%20-%20Setup.zip

2) inserire la seguente riga nel file *sicraweb-dsg.properties*, in modo da forzare l'utilizzo della libreria PKCS#11: cnsPKCS11.dll

```
3bff1800ffc10a31fe55006b0508c805011101434e531031800c=cnsPKCS11.dll
```

# 3. FUNZIONI DISPONIBILI DA RIGA DI COMANDO (BOZZA)

E' possibile utilizzare alcune delle funzionalità della libreria di firma al di fuori di SicraWeb direttamente da riga di comando, per finalità di test. Questo permette di testare le funzionalità di nuove versioni della libreria senza obbligare il cliente ad aggiornare l'applicativo.

Queste sono le funzionalità disponibili:

```
SicraWeb - Digital Signature Tool - ver.1.00 (standalone)
java version "1.6.0_23"
Java(TM) SE Runtime Environment (build 1.6.0_23-b05)
Java HotSpot(TM) Client VM (build 19.0-b09, mixed mode, sharing)

Argomenti non validi
Utilizzo: it.saga.library.digitalsignature.DsgMain -command [args]
 -firma: firma un doucmento
  DsgMain -firma <<PIN>> <<file da firmare>>

 -firmaPDF: firma un doucmento
  DsgMain -firmaPDF <<PIN>> <<file da firmare>>

 -verifica: verifica la firma di un documento
  DsgMain -verifica <<file da verificare>>

 -probe: verifica la presenza e il tipo della carta di firma
  DsgMain -probe

 -signers: lista i firmatari di un documento
  DsgMain -signers <<file_firmato.p7m>>

 -certificates: elenca i certificati di un documento
  DsgMain -certificates <<file_firmato.p7m>>

 -extract: estrae il documento firmato
  DsgMain -extract <<file_firmato.p7m>>

 -importCert: importa un certificato nel db dei certificati
  DsgMain -importCert <<certificate.cer>> <<certdb.cdb>>

 -cards: elenca l'elenco delle carte di firma supportate dalla libreria
  DsgMain -cards
```

## 3.1 Firma di un documento

E' possibile eseguire la firma di un documento direttamente da linea di comando lanciando la classe *it.saga.library.digitalsignature.DsgMain* con il parametro *–firma* con questa sintassi:

```
it.saga.library.digitalsignature.DsgMain -firma <<PIN>> <<file da firmare>>
```

o in alternativa è sufficiente lanciare il comando firma.bat persente nella directory utils del package con la stessa sintassi:

```
firma.bat <<PIN>> <<file da firmare>>
```

## 3.2 Funzioni di diagnostica

Dalla versione 1.0.2 è possibile invocare la funzione di diagnostica dei dispositivi di firma direttamente da linea di comando: è sufficiente lanciare la classe *it.saga.library.digitalsignature.DsgMain* con il parametro *-probe*

in alternativa è sufficiente lanciare il file probe.bat persente nella directory utils del package

Segue un esempio di chiamata: probe.bat

```
java.exe -client -cp
C:\SicraWeb\client\it.saga.library.digitalsignature.client.jar;C:\SicraWeb\client\lib\siacapi-full-10-
strip14.jar;C:\SicraWeb\client\lib\jdev-
rt.jar;C:\SicraWeb\Client\it.saga.library.authentication.client.jar;C:\SicraWeb\Client\it.saga.library.common.clie
nt.jar;C:\SicraWeb\client\it.saga.library.commonDataTypes.client.jar;C:\SicraWeb
\client\it.saga.library.baseForms.client.jar;C:\SicraWeb\client\it.saga.library.controls.client.jar;C:\SicraWeb\cl
ient\it.saga.library.logging.client.jar;C:\SicraWeb\client\it.saga.pubblici.protocollo.client.jar;C:\SicraWeb\clie
nt\it.saga.pubblici.utentiruoli.client.jar;C:\SicraWeb\client\lib\ejb.jar;C:\SicraWeb\client\it.saga.library.messa
ges.client.jar it.saga.library.digitalsignature.DsgMain –probe
```

# 4. ELENCO DEI DRIVER

| SMARTCARD | DRIVER |
|---|---|
| Actalis (serie HB053) | **Chipdrive driver v.2.14.41** (fornito con il cd di installazione del kit actalis) |
| Actalis (serie HB055) | **Chipdrive driver v.2.14.41** (fornito con il cd di installazione del kit actalis) + **Siemens CardOS API 2.2.1.7**(contenuto nell'ultimo cd di actalis E:\Smart Card Tools\Siemens CardOS\2.2.1.7) |
| Infocamere (serie 1202) | **Chipdrive driver v.2.14.41** (fornito con il cd di installazione del kit actalis) |
| Infocamere (serie 1203) | **(viene riconosciuta ma non funziona)**<br>**Chipdrive driver v.2.14.41** (fornito con il cd di installazione del kit actalis) |
| Infocamere (serie 1401) | **Chipdrive driver v.2.14.41** (fornito con il cd di installazione del kit actalis) + **Siemens CardOS API 2.2.1.7** (contenuto nell'ultimo cd di actalis E:\Smart Card Tools\Siemens CardOS\2.2.1.7) |
| Driver Infocamere | **https://www.firma.infocert.it/installazione/installazione_DiKe.php** |
| | |

Attenzione: nel caso di sistema operativo Windows XP SP2 è necessario installare anche il driver **Chipdrive Driver v.3.0** disponibile sul sito http://www.txsystems.com/download.html

**Chipdrive Driver Downloads**
http://www.txsystems.com/chipdrive_downloads.html

# 5. ELENCO SMARTCARD E LIBRERIE PKCS11 SUPPORTATI

## 5.1 Modelli di smartcard conosciuti

```
smartcard #0: Siemens CardOS/M4.01a, nuovo file system [ATR=3b:f2:98:00:ff:c1:10:31:fe:55:c8:04:12,
pkcs#11 lib ID=SIEMENS_NEW, backup ID=null]
smartcard #1: Siemens CardOS/M4.01, vecchio file system [ATR=3b:f2:98:00:ff:c1:10:31:fe:55:c8:03:15,
pkcs#11 lib ID=SIEMENS, backup ID=SIEMENS_NEW;ETOKEN]
smartcard #2: Setec 4.3.1  [ATR=3b:9f:94:40:1e:00:67:11:43:46:49:53:45:10:52:66:ff:81:90:00, pkcs#11
lib ID=SETEC, backup ID=null]
smartcard #3: Setec 4.3.0  [ATR=3b:1f:95:00:67:16:43:46:49:53:45:11:52:66:ff:81:90:00, pkcs#11 lib
ID=SETEC, backup ID=null]
smartcard #4: Schlumberger CryptoFlex [ATR=3b:95:15:40:ff:68:01:02:01:01, pkcs#11 lib ID=SLB, backup
ID=null]
smartcard #5: Datakey 330
[ATR=3b:ff:11:00:00:81:31:fe:4d:80:25:a0:00:00:00:56:57:44:4b:33:33:30:06:00:d0, pkcs#11 lib
ID=DATAKEY, backup ID=null]
smartcard #6: Rainbow iKey 2032 [ATR=null, pkcs#11 lib ID=IKEY, backup ID=null]
smartcard #7: Eutron CryptoIdentity ITSEC [ATR=null, pkcs#11 lib ID=CRYPTOIDITSEC, backup ID=null]
smartcard #8: Eutron CryptoIdentity [ATR=null, pkcs#11 lib ID=CRYPTOIDENTITY, backup ID=null]
smartcard #9: gemGATE 32k Std
[ATR=3b:fb:98:00:ff:c1:10:31:fe:55:00:64:05:20:47:03:31:80:00:90:00:f3, pkcs#11 lib ID=GEMPLUS,
backup ID=null]
smartcard #10: Gemplus GPK cardos [ATR=3b:a7:00:40:18:80:65:a2:09:01:01:52, pkcs#11 lib ID=GEMPLUS,
backup ID=null]
smartcard #11: Gemplus GPK gempkcs [ATR=3b:e2:00:ff:c1:10:31:fe:55:c8:02:9c, pkcs#11 lib ID=GEMPLUS,
backup ID=null]
smartcard #12: ORGA Micardo
[ATR=3b:ff:94:00:ff:80:b1:fe:45:1f:03:00:68:d2:76:00:00:28:ff:05:1e:31:80:00:90:00:23, pkcs#11 lib
ID=MICARDO, backup ID=null]
smartcard #13: Oberthur Ellips [ATR=3b:6f:00:ff:90:53:53:42:2d:50:4b:43:53:23:31:31:04:90:00,
pkcs#11 lib ID=ELLIPS, backup ID=null]
smartcard #14: Oberthur Identrus [ATR=3b:7f:18:00:00:00:31:c0:53:1d:e2:12:64:52:d9:04:00:82:90:00,
pkcs#11 lib ID=IDENTRUS, backup ID=null]
smartcard #15: IPM SysGillo [ATR=3b:9f:94:40:1e:00:67:16:43:46:49:53:45:10:52:66:ff:81:90:00,
pkcs#11 lib ID=IPM, backup ID=null]
smartcard #16: Incrypto34 v2
[ATR=3b:ff:18:00:ff:81:31:fe:55:00:6b:02:09:02:00:01:01:01:43:4e:53:10:31:80:9f, pkcs#11 lib
ID=STINCARD, backup ID=null]
smartcard #17: G&D StarCOS SPK 2.3 [ATR=3b:b7:94:00:81:31:fe:65:53:50:4b:32:33:90:00:d1, pkcs#11 lib
ID=SAFESIGN, backup ID=INCARD]
smartcard #18: ASE CARD CRYPTO SDK [ATR=3b:d6:18:00:81:b1:80:7d:1f:03:80:51:00:61:10:30:8f, pkcs#11
lib ID=ASECARD, backup ID=null]
smartcard #19: RSA Security [ATR=3b:76:11:00:00:00:9c:11:01:02:03, pkcs#11 lib ID=RSACARDEN, backup
ID=RSACARDEC]
smartcard #20: Oberthur Identrus [ATR=3b:7f:18:00:00:00:31:c0:53:1d:e2:12:64:52:d9:03:00:81:90:00,
pkcs#11 lib ID=IDENTRUS, backup ID=null]
smartcard #21: Oberthur CNS
[ATR=3b:ff:18:00:00:81:31:fe:45:00:6b:04:05:01:00:01:11:01:43:4e:53:10:31:80:69, pkcs#11 lib
ID=OBERTHURCNS_NEW, backup ID=OBERTHURCNS]
smartcard #22: Oberthur CNS
[ATR=3b:ff:18:00:00:81:31:fe:45:00:6b:04:05:01:00:01:21:01:43:4e:53:10:31:80:59, pkcs#11 lib
ID=OBERTHURCNS_NEW, backup ID=OBERTHURCNS]
smartcard #23: G&D StarCOS SPK 2.4 [ATR=3b:b7:18:00:81:31:fe:65:53:50:4b:32:34:90:00:5a, pkcs#11 lib
ID=SAFESIGN, backup ID=null]
smartcard #24: Gemplus GPK [ATR=3b:a7:00:40:18:80:65:a2:09:01:02:52, pkcs#11 lib ID=GEMPLUS, backup
ID=null]
smartcard #25: G&D StarCOS SPK 2.4 [ATR=3b:b7:18:00:c0:3e:31:fe:65:53:50:4b:32:34:90:00:25, pkcs#11
lib ID=SAFESIGN, backup ID=null]
smartcard #26: G&D StarCOS SPK 3.0
[ATR=3b:bb:18:00:c0:10:31:fe:45:80:67:04:12:b0:03:03:00:00:81:05:3c, pkcs#11 lib ID=SAFESIGN, backup
ID=null]
```

```
smartcard #27: Siemens CardOS/M4.01a, nuovo file system
[ATR=3b:fc:98:00:ff:c1:10:31:fe:55:c8:03:49:6e:66:6f:63:61:6d:65:72:65:28, pkcs#11 lib
ID=SIEMENS_NEW, backup ID=null]
smartcard #28: Microelectronica Espanola TEMD
[ATR=3b:7f:94:00:00:80:31:80:71:90:67:54:45:4d:44:31:2e:30:90:00, pkcs#11 lib ID=MSYSTEM, backup
ID=null]
smartcard #29: Oberthur CNS
[ATR=3b:ff:18:00:00:81:31:fe:45:00:6b:04:05:01:00:01:12:02:48:50:43:ff:31:80:83, pkcs#11 lib
ID=OBERTHURCNS_NEW, backup ID=OBERTHURCNS]
smartcard #30: Actalis cryptoflash 32K per charismathic
[ATR=3b:f4:18:00:02:c1:0a:31:fe:58:56:34:63:76:c5, pkcs#11 lib ID=ACTALISCRYPTOFLASH, backup
ID=null]
smartcard #31: Gemplus GPK cardos [ATR=3b:a7:00:40:18:80:65:a2:08:01:01:52, pkcs#11 lib ID=GEMPLUS,
backup ID=null]
smartcard #32: Oberthur CNS
[ATR=3b:ff:18:00:00:81:31:fe:45:00:6b:04:05:01:00:01:12:02:48:50:43:10:31:80:6c, pkcs#11 lib
ID=OBERTHURCNS_NEW, backup ID=OBERTHURCNS]
smartcard #33: Siemens CardOS 4.3B [ATR=3b:f2:18:00:02:c1:0a:31:fe:58:c8:08:74, pkcs#11 lib
ID=SIEMENS_4_3B, backup ID=null]
smartcard #34: Incard CNS
[ATR=3b:ff:18:00:ff:81:31:fe:55:00:6b:02:09:02:00:01:..:01:43:4e:53:..:31:80:.., pkcs#11 lib
ID=STINCARD, backup ID=null]
smartcard #35: Siemens CardOS 4.2B 64K [ATR=3b:f2:18:00:02:c1:0a:31:fe:58:c8:09:75, pkcs#11 lib
ID=SIEMENS64K, backup ID=null]
smartcard #36: Athena CNS
[ATR=3b:df:..:00:81:31:fe:..:00:6b:..:0c:..:..:01:..:01:43:4e:53:10:31:80:.., pkcs#11 lib
ID=ATHENA_CNS, backup ID=null]
```

## 5.2 Librerie PKCS11 conosciute (token)

```
token #0: Setec SetTokI [SETEC]
token #1: Schlumberger CryptoFlex [SLB]
token #2: Datakey CIP 330 [DATAKEY]
token #3: Rainbow IKey [IKEY]
token #4: Eutron CryptoIdentity ITSEC [CRYPTOIDITSEC]
token #5: Eutron CryptoIdentity [CRYPTOIDENTITY]
token #6: Gemplus [GEMPLUS]
token #7: AET SafeSign [SAFESIGN]
token #8: Siemens CardOS/M4.01 [SIEMENS]
token #9: Siemens CardOS/M4.01a [SIEMENS_NEW]
token #10: Micardo [MICARDO]
token #11: Oberthur Ellips [ELLIPS]
token #12: Oberthur Identrus [IDENTRUS]
token #13: Incard SysGillo [INCARD]
token #14: ST-Incard [STINCARD]
token #15: Aladdin eToken [ETOKEN]
token #16: IPM SysGillo [IPM]
token #17: nCipher nFast Pkcs#11 SUN Solaris [NFAST_SUN]
token #18: IBM 4758-2 [IBM4758]
token #19: Eracom CSA8000 [ERACOM]
token #20: ASECard Crypto [ASECARD]
token #21: RSA Security card EN [RSACARDEN]
token #22: RSA Security card EC [RSACARDEC]
token #23: Oberthur CNS [OBERTHURCNS]
token #24: Microelectronica Espanola TEMD [MSYSTEM]
token #25: Actalis cryptoflash per charismathic 32K [ACTALISCRYPTOFLASH]
token #26: Oberthur CNS Bit4id [OBERTHURCNS_NEW]
token #27: Siemens CardOS 4.3B [SIEMENS_4_3B]
token #28: Actalis One HID [ACTALISONEHID]
token #29: Siemens CardOS 4.2B 64K [SIEMENS64K]
token #30: Athena CNS [ATHENA_CNS]
```