

Software a corredo delle Smart Card per sistemi di emissione dei titoli di accesso agli spettacoli pubblici

Author: Giuseppe Amato

Version: 1.0

Indice

Questa è un'esenzione della libreria libSIAEdll. Essa implementa oltre a tutte le funzioni già implementate dalla libSIAEdll anche le funzioni PKCS7SignML e PKCS7SignML qui descritte.

Queste due funzioni rendono possibile la firma nel formato PKCS#7 e SMIME. Le funzioni ritornano un codice che in caso di successo corrisponde al valore intero 0, gli altri casi sono elencati nell'ultimo capitolo di questa documentazione.

Per usare la libreria da programmi C/C++ è necessario includere i seguenti file .h, distribuiti insieme alle librerie: scardhal.h libsiaeacard.h libsiaeap7.h

1 PKCS7SignML

Prototipo:

```
int PKCS7SignML(
    const char *pin,
    unsigned long slot,
    const char* szInputFileName,
    const char* szOutputFileName,
    int bInitialize);
```

| Parametro | Descrizione |
|------------------|--|
| pin | pin smartcard |
| slot | slot da utilizzare, zero based |
| szInputFileName | nome del file di input |
| szOutputFileName | nome del file di output |
| bInitialize | 1 = Inizializza e finalizza la libreria automaticamente. |

Note

bInitialize va impostato normalmente ad 1, l'unico caso in cui va impostata a 0 è quello in cui vengono utilizzate le funzioni della libSIAEdll.

Esempio d'uso

nel seguente esempio viene firmato il file "test.txt", usando la smartcard nel primo slot (slot 0); il file pkcs#7 firmato è salvato in "test.txt.p7m"

```
int res;
char pin[] = "123456";
res = pPKCS7SignML(pin, 0, "test.txt", "test.txt.p7m", TRUE);
```

2 SMIMESignML

Prototipo:

```
int SMIMESignML(
    const char* pin,
    unsigned long slot,
    const char* szOutputFilePath,
    const char* szFrom,
    const char* szTo,
    const char* szSubject,
    const char* szOtherHeaders,
    const char* szBody,
    const char* szAttachments,
    unsigned long dwFlags,
    int bInitialize);
```

| Parametro | Descrizione |
|-----------|---------------|
| pin | pin smartcard |

... continued on next page



| Parametro | Descrizione |
|----------------|---|
| slot | slot da utilizzare, zero based |
| szFrom | Campo 'From:' dello header rfc822. Es.1: "Giuseppe Verdi" < gverdi@xcom.it > Es.2: gverdi@xcom.it . |
| szTo | Opzionale. Campo 'To:' dello header rfc822. Block quote ends without a blank line; unexpected unindent. Es: vedere parametro szFrom |
| szSubject | Opzionale. Subject del messaggio. Dovrebbe contenere solo testo ASCII a 7 bit. (alcuni server ammettono testo ASCII-8bit). |
| szOtherHeaders | Header rfc822/MIME aggiuntivi. Es.1: X-Priority: 3. Es.2: References: 198325897234@xcom.it ; sdiof24323432423@email.it . |
| szBody | Opzionale. Contenuto del BODY del messaggio. Deve contenere solo testo ASCII-7bit alcuni server ammettono testo ASCII-8bit). |
| szAttachments | files da allegare separati dal carattere ';' es: c:file1.txt;c:temptmpfile.pdf è possibile specificare il nome dell'allegato, in tal caso ogni allegato dovrà essere specificato come segue: nome_allegato1.txt c:percorsoallegato1.txt;... |
| dwFlags | Riservato. Specificare come valore 0 |
| bInitialize | 1 = Inizializza e finalizza la libreria automaticamente. |

Note

bInitialize va impostato normalmente ad 1, l'unico caso in cui va impostata a 0 è quello in cui vengono utilizzate le funzioni della libSIAEdll.

Esempio d'uso

nel seguente esempio viene creata una mail SMIME firmata, usando la smartcard nel primo slot (slot 0); la mail viene salvata nel file "prova.eml"; alla mail viene allegato il file "test.txt":



```
int res;
char pin[] = "123456";
    res = pSMIMESignML("12345678",
        0,
        "prova.eml",
        "Mario Rossi <mariorossi@prova.it>",
        "Giuseppe Verdi <mariorossi@prova.it>",
        "Prova",
        NULL,
        "Email firmata di prova",
        "test.txt|c:\\\\test.txt",
        0, TRUE);
```

3 Codici di ritorno

- C_OK 0x0000
- C_CONTEXT_ERROR 0x0001
- C_NOT_INITIALIZED 0x0002
- C_ALREADY_INITIALIZED 0x0003
- C_NO_CARD 0x0004
- C_UNKNOWN_CARD 0x0005
- C_WRONG_LENGTH 0x6282
- C_WRONG_TYPE 0x6981
- C_NOT_AUTHORIZED 0x6982
- C_PIN_BLOCKED 0x6983
- C_WRONG_DATA 0x6A80
- C_FILE_NOT_FOUND 0x6A82
- C_RECORD_NOT_FOUND 0x6A83
- C_WRONG_LEN 0x6A85
- C_UNKNOWN_OBJECT 0x6A88
- C_ALREADY_EXISTS 0x6A89
- C_GENERIC_ERROR 0xFFFF

