25 Agosto 2001

CONTRATTO LOTTO "F"

OBIETTIVO 43 DI MANUTENZIONE EVOLUTIVA DELL'AREA 08/SPESE – FUNZIONI DI SICUREZZA SU HOST

DISEGNO E MODALITA' D'USO DELLA ROUTINE DI VERIFICA FIRMA

Versione 2

25 Agosto 2001

CONTRATTO LOTTO "F"

OBIETTIVO 43 DI MANUTENZIONE EVOLUTIVA DELL'AREA 08/SPESE – FUNZIONI DI SICUREZZA SU HOST

DISEGNO E MODALITA' D'USO DELLA ROUTINE DI VERIFICA FIRMA

Versione 2

1. INTRODUZIONE

Il presente documento illustra le funzioni di sicurezza relative alla verifica della firma realizzate su Host nell'ambito dell'intervento di manutenzione evolutiva '43' previsto dal Lotto "F" – area 08 –Spese.

Il documento è articolato nelle seguenti sezioni:

1. introduzione,
2. generalità e nuove caratteristiche architetturali,
3. descrizione delle funzionalità,
4. disegno, articolato in:
   4.1. INPUT e OUTPUT,
   4.2. utilizzo MBM,
   4.3. pseudocodice,
5. modalità di rilascio della routine,
6. modalità di richiamo della routine da programma COBOL, articolata in:
   6.1. invocazione,
   6.2. JCL di esecuzione,
   6.3. codici di errore,
7. progettazione dei casi di test.

2. GENERALITA' E NUOVE CARATTERISTICHE ARCHITETTURALI

L'obiettivo in oggetto nasce dalla variazione architetturale, ipotizzata da Consip, dell'applicazione Mandato Informatico della Ragioneria Generale dello Stato.
Tale variazione si espleta nel ridisegno dei processi di gestione della firma e delle funzioni di identificazione ed autenticazione dell'utente (per una descrizione delle problematiche che si sono evidenziate nello studio svolto in merito e delle soluzioni tecniche proposte, si rimanda al documento R02-A04-K43-0001 – "Area 08/43 – Ridisegno dei processi di gestione della firma e delle funzionalità di identificazione ed autenticazione dell'utente per l'adozione della firma digitale a validità legale").

Nel presente documento verranno descritte le caratteristiche implementative e di utilizzo dei nuovi servizi di verifica della firma.

Gli elementi salienti della nuova architettura che comportano il ridisegno del processo di verifica firma sono:

- i dati sono firmati utilizzando smart-card fornite, già inizializzate, da Postecom per il Centro Tecnico RUPA (che, nel nuovo contesto, svolge il ruolo di Certification Authority),
- il dato, la firma ed il certificato utente sono inviati, dalla postazione client, su Host in una 'busta' PKCS#7,
- il processo di verifica della firma deve essere effettuato su sistema centrale,
- gli autocertificati di CA (Certification Authority) e le Certification Revocation List (CRL) sono presenti su sistema centrale. Per l'alimentazione di queste ultime, inoltre, è previsto un processo di scarico a tempo dal sito del Centro Tecnico RUPA.

Il nuovo flusso si articolerà come segue:

- Nella fase TP:
  1. i dati, provenienti da Host, vengono inviati sulla postazione Client,
  2. la firma è prodotta su Client,
  3. i dati firmati sono 'imbustati' in un file di formato PKCS#7,
  4. la stringa PKCS#7 viene inviata su sistema Host attraverso emulatore 3270 AVIVA,
  5. tale stringa è memorizzata su base dati DB2 su campi di tipo BYTE.

- Nella successiva fase batch, su Host, sono effettuati i seguenti trattamenti e controlli:
  a. è verificata la validità delle CRL disponibili su Host,
  b. i dati sono 'sbustati' e sottoposti a verifica della firma (il certificato dell'utente firmatario è ricavabile dallo stesso PKCS#7),

c. il certificato dell'utente è verificato tramite il certificato di CA corrispondente (memorizzato su Host su dataset sequenziale),
d. viene controllata l'eventuale revoca del certificato utente usando le CRL contenute in dataset sequenziali su Host,
e. i dati che abbiano superato i controlli dei tre punti precedenti sono inviati mediante i servizi APIEDI-SGCE su Front End Unix e, da qui, con modalità invariate rispetto al flusso attualmente operativo, in Banca d'Italia.

Da quanto detto nasce l'esigenza di rendere disponibili dei servizi su sistema centrale che realizzino i trattamenti corrispondenti ai punti a, b, c, d.
Tali servizi devono essere realizzati, in Cobol, usando le funzioni del prodotto MBM della Telvox.

Nell'ambito dell'integrazione con le altre Amministrazioni (applicazione 'Ordini di pagare'), è richiesta la disponibilità di un insieme di servizi di verifica firma che prevedano un numero ridotto di controlli e trattamenti (corrispondenti, sostanzialmente, ai punti b e c) rispetto a quanto stabilito per l'applicazione 'Mandato Informatico'.

Nel seguito del documento verrà data una descrizione dettagliata dei vari aspetti di interesse riguardanti la routine.

3. DESCRIZIONE DEI SERVIZI

Nel presente paragrafo viene data una descrizione dettagliata dei servizi di verifica firma da rendere disponibili alle applicazioni 'Ordini di Pagare' e 'Mandato Informatico'.

Tali servizi verranno forniti, in modo integrato, da una routine richiamabile da programma Cobol.

Nel caso di impiego nell'ambito di 'Ordini di Pagare', la routine COBOL da implementare deve consentire di:

- estrarre il dato firmato da una 'busta' di tipo PKCS#7 SignedData e convertirlo da ASCII ad EBCDIC,
- verificare la firma utilizzando il certificato estratto dalla 'busta',
- effettuare controlli aggiuntivi sul certificato, quali la sua validità, la sua emissione da parte di una CA fidata.

Per poter svolgere i controlli dell'ultimo punto, la routine dovrà gestire i certificati delle CA emittenti nel seguente modo:

- verificare la loro consistenza (verificandone, cioè, la firma - impiegando come chiave RSA quella presente nella componente pubblica del certificato - e la validità),
- estrarne la chiave RSA pubblica,
- estrarne il valore del campo 'subjectKeyIdentifier' (che permette di identificare univocamente la chiave pubblica estratta).

Nel caso di 'Mandato Informatico', la routine dovrà fornire, oltre a quelli suesposti, i seguenti servizi:

- verifica di validità delle CRL, ovvero: emissione da parte di una CA fidata e validità temporale a meno di un *delta* (pari a 3 ore),
- verifica della non revoca del certificato utente,
- controllo sulla disponibilità di tutte le CRL necessarie alla suddetta verifica,
- estrazione dal certificato utente e restituzione del valore del codice fiscale dell'utente.

E', quindi, previsto che la routine da implementare (SXPK07) realizzi le seguenti cinque macrofunzioni (sulla base di un parametro fornito in input dal programma chiamante).

Inizializzazione (INIZIA)

Questo servizio, che deve essere richiamato prima di qualsiasi altro, effettua una serie di operazioni propedeutiche all'esecuzione dei servizi VERIRID e VERIEST.

Innanzitutto, permette l'acquisizione ed il caricamento in memoria (in un'area comune con il programma chiamante) delle chiavi pubbliche delle CA e dei valori dei campi 'subjectKeyIdentifier' (SKI) relativi, estraendoli dai certificati stessi delle CA dopo averne verificato la consistenza

Controlli aggiuntivi sui certificati di CA (ad esempio, il controllo che non esista duplicazione sugli SKI dei certificati di CA gestite) sono considerati propedeutici a questo servizio, s'intende, cioè, che siano stati svolti all'atto del caricamento del data set contenente tali certificati.

Il numero massimo di certificati di CA gestiti dalla routine è 30. Tale valore può essere modificato effettuando un intervento di manutenzione di lieve entità sulla routine.

Il servizio, inoltre, acquisisce e rende disponibili all'ambiente di lavoro MBM le CRL, se presenti, da utilizzare per la verifica della non-revoca dei certificati e carica in memoria (in un'area comune con il programma chiamante) il puntamento (*handle*) restituito dall'ambiente MBM relativo ad una singola CRL insieme all'identificativo della chiave di firma della CRL stessa (AuthorityKeyIdentifier).

Il servizio gestisce fino ad un massimo di 10 CRL. Tale valore puo' essere modificato effettuando un intervento di manutenzione sulla routine.

Oltre a ciò, viene caricata in memoria una tabella, a partire da un file (eventualmente fornito in input), contenente la lista degli URL da cui viene effettuato lo scarico periodico delle CRL.

Verifica ridotta (VERIRID)

Questo servizio (da richiamare dopo il servizio INIZIA) permette l'estrazione del dato firmato da un PKCS#7 fornito in input. Il dato viene convertito dalla codifica ASCII alla codifica EBCDIC (secondo il code-page 037) tramite una tabella di conversione fornita in input.
Inoltre, il servizio verifica la firma, utilizzando il certificato presente nel PKCS#7.

Dal certificato viene estratto il valore del campo 'authorityKeyIdentifier' e, in base a questo, viene effettuata la verifica del certificato mediante la chiave pubblica (caricata in memoria nella fase di inizializzazione) avente tale valore di SKI.

Verifica estesa (VERIEST)

Il servizio, da richiamare dopo il servizio INIZIA, esegue le stesse operazioni del servizio VERIRID ed, in aggiunta, verifica la non-revoca del certificato rispetto alle CRL reperite nel servizio INIZIA. Prima di questo, estrae dal certificato l'URL della CRL corrispondente (cioè quella destinata a contenere l'eventuale revoca del certificato) e controlla che sia tra quelli da cui viene effettuato lo scarico periodico. Restituisce, infine, al chiamante il CODICE FISCALE presente nel Common Name del Subject del certificato estratto dal PKCS#7 ricevuto.

Terminazione (TERMINA)

Tale servizio provvede a liberare dall'ambiente MBM le aree dati relative alle CRL caricate nel servizio INIZIA

Verifica CRL (VERICRL)

Esegue le operazioni di verifica di una o più CRL ricevute in ingresso. In particolare:

- verifica la firma della CRL.
- verifica il certificato della CA relativo alla firma della CRL.
- controlla che la validità della CRL non sia scaduta da più di 3 ore (tale valore, impostato in un parametro interno al software, è di facile modificabilità).

Questo servizio è (l'unico) richiamabile dal chiamante senza che questi abbia preventivamente invocato il servizio INIZIA.

4. DISEGNO

4.1. INPUT E OUTPUT

In questo paragrafo sono descritti l'input e l'output di ogni servizio. Alcuni valori di output per un servizio costituiscono l'input per un altro. E', quindi, stata utilizzata un'area di memoria comune ai vari servizi e condivisa con il programma chiamante ('mappata' da una COPY Cobol).

4.1.1. INIZIA

Input

	TIPO	DESCRIZIONE
SXPK071-COD-FUNZ	Campo COPY	Assume il valore 'INIZIA'
MBMCA	archivio	E' l'archivio sequenziale (allocato a JCL ma 'trasparente' al chiamante) contenente i certificati di CA (uno per ogni record). Ha RECFM=VB ed LRECL= 30004[1]
LISTAURL	archivio	E' l'archivio sequenziale (allocato a JCL ma 'trasparente' al chiamante) contenente gli URL delle CRL (uno per ogni record).
MBMCRLi (i=01,...10)	archivio	E' l'archivio sequenziale (allocato a JCL ma 'trasparente' al chiamante) contenente una CRL. Ha RECFM=VB e LRECL=30004.

Output

1 Il valore della massima lunghezza record è sicuramente sufficiente a contenere un certificato. Infatti, pur non avendo a priori un limite massimo stabilito, un certificato non supera mai tale ampiezza.

	TIPO	DESCRIZIONE
SXPK071-RET-COD	Campo COPY	E' impostato con il valore del return code[2]
SXPK071-CA	Campo COPY	E' l'area di formato tabellare contenente le informazioni sui certificati di CA (SKI e chiave pubblica) da utilizzare nei servizi VERIRID e VERIEST.
SXPK071-CRL	Campo COPY	E' l'area di formato tabellare contenente le informazioni sulle CRL (AKI e *handle* MBM) da utilizzare nel servizio VERIEST

4.1.2. VERIRID

Input

	TIPO	DESCRIZIONE
SXPK071-COD-FUNZ	Campo COPY	Assume il valore 'VERIRID'
SXPK071-CA	Campo COPY	E' l'area di formato tabellare contenente le informazioni sui certificati di CA (SKI e chiave pubblica) prodotti nella fase di inizializzazione.
SXPK071-RID-DD-PKCS#7	Campo COPY	Deve essere impostato con il DDname (*mbmpkcs7)* dell'archivio

2 Si veda la sezione relativa alle modalità di richiamo della routine per i valori che tale campo può assumere

		contenente il file PKCS#7.
SXPK071-RID-DD-DATO	Campo COPY	Deve essere impostato con il DDname (*mbmdato)* dell'archivio sequenziale (RECFM=VB) destinato a contenere il dato 'sbustato'.
Mbmpkcs7	archivio	Archivio sequenziale (RECFM=VB) contenente il file PKCS#7.
MBMTCONV	archivio	E' l'archivio sequenziale (allocato a JCL ma 'trasparente' al chiamante') contenente la tabella di conversione ASCII-EBCDC

Output

	TIPO	**DESCRIZIONE**
SXPK071-RET-COD	Campo COPY	E' impostato con il valore del return code[3]
Mbmdato	archivio	Archivio sequenziale (RECFM=VB) contenente il dato 'sbustato'.

4.1.3. VERIEST

3 Si veda la sezione relativa alle modalità di richiamo della routine per i valori che tale campo può assumere

Input

	TIPO	**DESCRIZIONE**
SXPK071-COD-FUNZ	Campo COPY	Assume il valore 'VERIEST'
SXPK071-CA	Campo COPY	E' l'area di formato tabellare contenente le informazioni sui certificati di CA (SKI e chiave pubblica) prodotti nella fase di inizializzazione.
SXPK071-CRL	Campo COPY	E' l'area di formato tabellare contenente le informazioni sulle CRL (AKI e *handle*) prodotti nella fase di inizializzazione.
SXPK071-RID-DD-PKCS#7	Campo COPY	Deve essere impostato con il Ddname (*mbmpkcs7)* dell'archivio contenente il file PKCS#7.
SXPK071-RID-DD-DATO	Campo COPY	Deve essere impostato con il Ddname (*mbmdato)* dell'archivio sequenziale (RECFM=VB) destinato a contenere il dato 'sbustato'.
Mbmpkcs7	archivio	Archivio sequenziale (RECFM=VB) contenente il file PKCS#7.
MBMTCONV	archivio	E' l'archivio sequenziale (allocato a JCL ma 'trasparente' al chiamante) contenente la tabella di conversione ASCII-EBCDIC

Output

	TIPO	DESCRIZIONE
SXPK071-RET-COD	Campo COPY	E' impostato con il valore del return code[4]
Mbmdato	archivio	Archivio sequenziale (RECFM=VB) contenente il dato 'sbustato'.
SXPK071-EST-COD-FISC	Campo COPY	Contiene il codice fiscale estratto dal Common Name del Subject del certificato relativo alla firma del PKCS#7.

4.1.4. TERMINA

Input

	TIPO	DESCRIZIONE
SXPK071-COD-FUNZ	Campo COPY	Assume il valore 'TERMINA'
SXPK071-CRL	Campo COPY	E' l'area di formato tabellare contenente le informazioni sulle CRL (AKI e *handle*) prodotti nella fase di inizializzazione.

Output

	TIPO	DESCRIZIONE
SXPK071-RET-COD	Campo COPY	E' impostato con il valore del return-code.

4.1.5. VERICRL

4 Si veda la sezione relativa alle modalità di richiamo della routine per i valori che tale campo può assumere

Input

	TIPO	DESCRIZIONE
SXPK071-COD-FUNZ	Campo COPY	Assume il valore 'VERICRL'
MBMCA	archivio	E' l'archivio sequenziale (allocato a JCL ma 'trasparente' al chiamante) contenente i certificati di CA che hanno firmato le CRL (uno per ogni record). Ha RECFM=VB ed LRECL= 30004[5]
MBMCRLi (i=01,...10)	archivio	E' l'archivio sequenziale (allocato a JCL ma 'trasparente' al chiamante) contenente una CRL. Ha RECFM= VB e LRECL= 30004.
SXPK071-VER-DATAORA	Campo COPY	Data e ora di riferimento[6] per stabilire se la CRL è ancora "valida"

Output

	TIPO	DESCRIZIONE
SXPK071-RET-COD	Campo COPY	E' impostato con il valore del return-code.

4.1. UTILIZZO MBM

MBM fornisce due modalità di accesso ai suoi servizi: tramite MACRO e tramite FRAMEWORK. Nel caso di verifica di certificati il FRAMEWORK

5 Il valore della massima lunghezza record è sicuramente sufficiente a contenere un certificato. Infatti, pur non avendo a priori un limite massimo stabilito, un certificato non supera mai tale ampiezza.

6 Possono essere la data e l'ora attuali o quelle di riferimento valide per tutto il batch serale.

offre funzionalità più potenti e pertanto sarà quest'ultimo a venire utilizzato. Nel caso di verifica firma sono disponibili solo servizi MACRO.

N:B: le chiamate alle funzioni del Framework(Fi) ed alle macro(Mi) sono esplicitate nel seguito.

Il servizio INIZIA:

Per ogni record esistente nel sequenziale dei certificati CA, invoca i seguenti servizi del FRAMEWORK:
(F1) **YCHKCRT** per verificare la consistenza del certificato presente nel record.
(F2) **YGETCRTP** per estrarre la PublicKey dal certificato
(F3) **YLOADCRT** per caricare nell'ambiente MBM il certificato
(F4) **YGETCRT** per reperire la SubjectKeyIdentifier e le date di inizio e fine validità del certificato.
(F5) **YFREECRT** per liberare dall'ambiente MBM il certificato
(F6) **YASCEBCD** per convertire le date di inizio e fine validità del certificato da ASCII ad EBCDIC.Per ogni CRL presente nei DataSet convenuti, invoca i seguenti servizi del framework:
(F7) **YLOADCRL** per caricare nell'ambiente MBM la CRL.
(F8) **YGETCRL** per reperire la AuthorityKeyIdentifier della CRL.

Il servizio VERIRID invoca 3 volte la MACRO YP7SIGND (M1):
- la prima invocazione estrae il certificato dal PKCS#7 ricevuto,
- la seconda verifica il PKCS#7 utilizzando il certificato appena ricavato,
- la terza estrae il dato dal PKCS#7 e lo converte in EBCDIC

ed inoltre invoca le funzioni del FRAMEWORK:
(F3) **YLOADCRT** per caricare nell'ambiente MBM il certificato,
(F4) **YGETCRT** per reperire la AuthorityKeyIdentifier e le date di inizio e fine validità del certificato,
(F1) **YCHKCRT** per verificare il certificato utilizzando la PublicKey reperita nel passo precedente,
(F5) **YFREECRT** per liberare dall'ambiente MBM il certificato,
(F6) **YASCEBCD** per convertire le date di inizio e fine validità del certificato da ASCII ad EBCDIC.

Il servizio VERIEST invoca:
-la MACRO YP7SIGND (M1) 3 volte:
- la prima invocazione estrae il certificato dal PKCS#7 ricevuto,
- la seconda verifica il PKCS#7 utilizzando il certificato appena ricavato,
- la terza estrae il dato dal PKCS#7 e lo converte in EBCDIC

ed inoltre invoca le funzioni del FRAMEWORK:

(F3) **YLOADCRT** per caricare nell'ambiente MBM il certificato,
(F4) **YGETCRT** per reperire dal Certificato la AuthorityKeyIdentifier, le date di inizio e fine validità ed il CommonName del Subject.
(F1) **YCHKCRT** per verificare il certificato utilizzando la PublicKey dell'Autorità di Certificazione,
(F5) **YFREECRT** per liberare dall'ambiente MBM il certificato,
(F6) **YASCEBCD** per convertire le date di inizio e fine validità del certificato da ASCII ad EBCDIC.
(F9)**FINDCRL** per verificare la revoca del certificato.

Il servizio TERMINA invoca, per ogni CRL caricata nell'ambiente MBM
(FA) **YFREECRL** per liberare dall'ambiente MBM la CRL.

Il servizio VERICRL invoca le funzioni del FRAMEWORK:
Per ogni record esistente nel sequenziale dei certificati CA, invoca i seguenti servizi del FRAMEWORK:
(F1) **YCHKCRT** per verificare la consistenza del certificato presente nel record.
(F2) **YGETCRTP** per estrarre la PublicKey dal certificato
(F3) **YLOADCRT** per caricare nell'ambiente MBM il certificato
(F4) **YGETCRT** per reperire la SubjectKeyIdentifier e le date di inizio e fine validità del certificato.
(F5) **YFREECRT** per liberare dall'ambiente MBM il certificato
(F6) **YASCEBCD** per convertire le date di inizio e fine validità del certificato da ASCII ad EBCDIC.
Per ogni CRL presente nei DataSet convenuti, invoca i seguenti servizi del framework:
(F7) **YLOADCRL** per caricare nell'ambiente MBM la CRL.
(F8) **YGETCRL** per reperire la AuthorityKeyIdentifier della CRL e la data della successiva emissione della CRL.
(F6) **YASCEBCD** per convertire le date di emissione e l'AKI della CRL da ASCII ad EBCDIC.
(F1) **YCHKCRL** per verificare la CRL utilizzando la PublicKey dell'Autorità di Certificazione
(F5) **YFREECRL** per liberare dall'ambiente MBM la CRL

Le aree dati utilizzate dalle macro MBM fanno parte della WORKING-STORAGE della routine e sono:

01 WS-PARAM PIC X(80) VALUE SPACES.
contiene i parametri passati alla macro e avrà valori diversi in base alle tre chiamate:

prima invocazione (estrazione certificato):
'DEV -FIN **DD:mbmpkcs7** -FOUT **DD:MBMCERT** -CODCRT 1 -O'

seconda invocazione (verifica firma):
'CHK -FIN DD:mbmpkcs7 -FCERTM DD:MBMCERT '

terza invocazione (estrazione dato):
'DEV -FIN DD:mbmpkcs7 -FOUT DD:mbmdato -DEVTBL DD:MBMTCONV -O'

01 WS-LEN PIC S9(09) COMP VALUE +80.
avrà sempre il valore +80 prima di ogni chiamata della macro, indica la lunghezza massima della stringa WS-PARAM.

01 WS-RC PIC S9(04) COMP VALUE +0,
conterrà al ritorno della CALL alla macro il valore di ritorno MBM.

NB: Nella terza chiamata la stringa "-DEVTBL DD:MBMTCONV" nel parametro è necessaria solo se la firma digitale viene apposta a dati in codifica ASCII e la conversione viene effettuata verso una code-page non standard (diversa da 280).

Le aree dati utilizzate dalla funzioni del FRAMEWORK MBM fanno parte della WORKING-STORAGE della routine e sono:

```
01 W-MBM.
05 MBM-RC                          PIC S9(4) COMP.
    05 MBM-BUFIN.
          15 MBM-BUFIN-VAL         PIC X(1000000).
          15 MBM-BUFIN-LEN         PIC S9(9) COMP.
    05 MBM-FIELD.
          15 MBM-FIELD-VAL         PIC X(256).
          15 MBM-FIELD-LEN         PIC S9(9) COMP.
    05 MBM-KEY.
          15 MBM-KEY-N             PIC S9(4) COMP.
          15 MBM-KEY-E.
             20 MBM-KEY-E-EL OCCURS 3 PIC S9(4) COMP.
          15 MBM-KEY-M.
             20 MBM-KEY-M-EL OCCURS 1025 PIC S9(4) COMP.
    05 MBM-LABEL.
          15 MBM-LABEL-VAL         PIC X(107).
          15 MBM-LABEL-LEN         PIC S9(9) COMP.
```

05 MBM-CRITICAL	PIC S9(9) COMP.
05 MBM-FIELD-ASCII	PIC X(256).
05 MBM-HANDLE	PIC S9(9) COMP.
05 MBM-CRLHANDLE	PIC S9(9) COMP..

Per le aree della COPY COBOL si veda l'Appendice:

M1:
Le tre chiamate alla macro MBM **YP7SIGND** da COBOL sono sempre
CALL 'YP7SIGND' USING WS-PARAM WS-LEN WS_RC.

Le invocazioni delle funzioni del Framework sono:

F1: - verifica firma certificato
CALL YCHKCRT USING MBM-BUFIN-VAL, MBM-BUFIN-LEN, MBM-KEY-N, MBM-KEY-E, MBM-KEY-M, MBM-RC.
il codice di ritorno è in MBM-RC.

F2: -estrazione chiave da certificato
CALL YGETCRTP USING MBM-BUFIN-VAL, MBM-BUFIN-LEN, MBM-KEY-N, MBM-KEY-E, MBM-KEY-M, MBM-RC.
il codice di ritorno è in MBM-RC.
la chiave pubblica della CA è in MBM-KEY.

F3-caricamento certificato in ambiente MBM
CALL YLOADCRT USING MBM-BUFIN-VAL, MBM-BUFIN-LEN, MBM-HANDLE, MBM.RC
in MBM-HANDLE avremo il descrittore (handle) del certificato.
il codice di ritorno è in MBM-RC.

F4 - estrazione proprietà da certificato
CALL YGETCRT USING MBM-HANDLE, MBM-LABEL-VAL, MBM-LABEL-LEN MBM-FIELD-VAL, MBM-FIELD-LEN, MBM-CRITICAL, MBM-RC
in MBM-FIELD avremo la proprietà richiesta.
MBM-CRITICAL non è utilizzato.
il codice di ritorno è in MBM-RC.

F5- liberazione certificato da ambiente MBM
CALL YFREECRT USING MBM-HANDLE MBM-RC
il codice di ritorno è in MBM-RC

F6- conversione da ASCII ad EBCDIC

CALL YASCEBCD USING MBM-FIELD-ASCII MBM-FIELD-LEN MBM-FIELD-VAL MBM-RC
il codice di ritorno è in MBM-RC

F7-caricamento CRL in ambiente MBM
CALL YLOADCRL USING MBM-BUFIN-VAL, MBM-BUFIN-LEN, MBM-HANDLE, MBM.RC
in MBM-HANDLE avremo il descrittore (handle) della CRL.
il codice di ritorno è in MBM-RC.

F8 - estrazione proprietà da CRL
CALL YGETCRL USING MBM-HANDLE, MBM-LABEL-VAL, MBM-LABEL-LEN MBM-FIELD-VAL, MBM-FIELD-LEN, MBM-CRITICAL, MBM-RC
in MBM-FIELD avremo la proprietà richiesta.
MBM-CRITICAL non è utilizzato.
il codice di ritorno è in MBM-RC.

F9 - verifica presenza estremi certificato in CRL
CALL YFINDCRL USING MBM-HANDLE, MBM-BUFIN-VAL, MBM-BUFIN-LEN, MBM-POS, MBM-RC
in MBM-POS avremo la posizione del certificato nella CRL, se presente.
il codice di ritorno è in MBM-RC: N.B: il codice *37* indica che il certificato non è presente nella lista di revoca, il codice 0 indica che il certificato è presente nella lista di revoca, altri codici indicano errore nell'esecuzione del servizio.

FA- liberazione CRL da ambiente MBM
CALL YFREECRL USING MBM-HANDLE MBM-RC
il codice di ritorno è in MBM-RC

FB- verifica firma CRL
CALL YCHKCRL USING MBM-BUFIN-VAL, MBM-BUFIN-LEN, MBM-KEY-N, MBM-KEY-E, MBM-KEY-M, MBM-RC.
il codice di ritorno è in MBM-RC.

4.3. PSEUDO CODICE

SE il campo SXPK071-COD-FUNZ = 'INIZIA'
VAI a FUN-INIZIA
SE il campo SXPK071-COD-FUNZ = 'TERMINA'
VAI a FUN-TERMINA

SE il campo SXPK071-COD-FUNZ = 'VERICRL'
VAI a FUN-VERICRL
SE il campo SXPK071-COD-FUNZ = 'VERIEST'
VAI a FUN-VERIEST
SE il campo SXPK071-COD-FUNZ = 'VERIRID'
VAI a FUN-VERIRID
ALTRIMENTI
assegna 8 a SXPK071-RET-COD.
ritorna al chiamante.

FUN-INIZIA:
Reperisci la data di sistema e ponila in *data-attuale*.
assegna 0 ad IPOS, indicatore di prima posizione libera nella tabella SXPK071-CA-DATI
Per ogni record presente nel sequenziale delle CA:
incrementa IPOS controllando che non "sfondi tabella"
se Tabella Piena :
assegna codice errore routine a SXPK071-RET-COD
ritorna al chiamante.
leggi il certificato presente nel record
se ERRORE in lettura
assegna codice errore routine a SXPK071-RET-COD
ritorna al chiamante.
poni il certificato letto in MBM-BUFIN-VAL
poni la lunghezza del certificato in MBM-BUFIN-LEN
poni 0 in MBM-KEY
chiama **F1 – verifica firma certificato**
se ERRORE in chiamata MBM:
assegna codice errore MBM a SXPK071-RET-COD
ritorna al chiamante.
chiama **F2- estrazione chiave da certificato**
se ERRORE in chiamata MBM :
assegna codice errore MBM a SXPK071-RET-COD
ritorna al chiamante.
Copia MBM-KEY in SXPK071-CA-KEY[IPOS]
chiama **F3- caricamento certificato in ambiente MBM**
se ERRORE in chiamata MBM :
assegna codice errore MBM a SXPK071-RET-COD
ritorna al chiamante.
poni la label "**extensions.subjectKeyIdentifier**" in MBM-LABEL-VAL.
poni la lunghezza della label (31) in MBM-LABEL-LEN.
chiama **F4 – estrazione proprietà da certificato**
se ERRORE in chiamata MBM :

assegna codice errore MBM a SXPK071-RET-COD
ritorna al chiamante.
Copia MBM-FIELD in SXPK071-CA-SKI[IPOS]
poni la label "**validity.notBefore**" in MBM-LABEL-VAL.
poni la lunghezza della label (18) in MBM-LABEL-LEN.
chiama **F4 – estrazione proprietà da certificato**
se ERRORE in chiamata MBM :
assegna codice errore MBM a SXPK071-RET-COD
ritorna al chiamante.
poni MBM-FIELD-VAL in MBM-FIELD-ASCII
chiama **F6 – conversione da ASCII ad EBCDIC**
se ERRORE in chiamata MBM :
assegna codice errore MBM a SXPK071-RET-COD
ritorna al chiamante.
poni MBM-FIELD_VAL in *data-inizio-validità*.
poni la label "**validity.notAfter**" in MBM-LABEL-VAL.
poni la lunghezza della label (17) in MBM-LABEL-LEN.
chiama **F4 – estrazione proprietà da certificato**
se ERRORE in chiamata MBM :
assegna codice errore MBM a SXPK071-RET-COD
ritorna al chiamante.
poni MBM-FIELD-VAL in MBM-FIELD-ASCII
chiama **F6 – conversione da ASCII ad EBCDIC**
se ERRORE in chiamata MBM :
assegna codice errore MBM a SXPK071-RET-COD
ritorna al chiamante.
poni MBM-FIELD_VAL in data-fine-validità.
Se data-attuale > data-fine-validità o data-attuale < data-inizio-validità
assegna codice routine a SXPK071-RET-COD
ritorna al chiamante.
chiama **F5 liberazione certificato da ambiente MBM**
se ERRORE in chiamata MBM :
assegna codice errore MBM a SXPK071-RET-COD
ritorna al chiamante.

assegna 0 ad IPOS, indicatore di prima posizione libera nella tabella SXPK071-CRL-DATI
Per ogni ddname relativo alle CRL presente nella configurazione della routine:
incrementa IPOS controllando che non "sfondi tabella"
se Tabella Piena :
assegna codice errore routine a SXPK071-RET-COD
ritorna al chiamante.
leggi la CRL

se ERRORE in lettura
assegna codice errore routine a SXPK071-RET-COD
ritorna al chiamante.
poni la CRL letta in MBM-BUFIN-VAL
poni la lunghezza della CRL in MBM-BUFIN-LEN
chiama **F7- caricamento CRL in ambiente MBM**
se ERRORE in chiamata MBM :
assegna codice errore MBM a SXPK071-RET-COD
ritorna al chiamante
assegna MBM-HANDLE a SXPK071-CRL-HANDLE[iPOS]
poni la label "crlExtensions.authorityKeyIdentifier.keyIdentifier" in MBM-LABEL-VAL.
poni la lunghezza della label (50) in MBM-LABEL-LEN.
chiama **F8 - estrazione proprietà da CRL**
se ERRORE in chiamata MBM :
assegna codice errore MBM a SXPK071-RET-COD
ritorna al chiamante.
assegna MBM-FIELD a SXPK071-CRL-AKI[iPOS]

assegna 0 ad IPOS, indicatore di prima posizione libera nella tabella SXPK071-URL
Per ogni record presente nel sequenziale degli URL - LISTAURL:
incrementa IPOS controllando che non "sfondi tabella"
se Tabella Piena :
assegna codice errore routine a SXPK071-RET-COD
ritorna al chiamante.
leggi il record in SXPK071-URL-VAL[IPOS].
se ERRORE in lettura
assegna codice errore routine a SXPK071-RET-COD
ritorna al chiamante.
ritorna al chiamante.

FUN-TERMINA:
Per ogni CRL caricata:
copia SXPK071-CRL[iPos].HANDLE in MBM-HANDLE.
FA– liberazione CRL da ambiente MBM.
se ERRORE in chiamata MBM :
assegna codice errore MBM a SXPK071-RET-COD
ritorna al chiamante.
pulisci le aree SXPK071-CRL e SXPK071-CA.
assegna SXPK071-RET-SUCCESS a SXPK071-RET-COD.
ritorna al chiamante.

FUN-VERIRID:
chiama la macro YP7SIGND per estrarre il certificato nel PKCS#7 e porlo nel file MBMCERT.
se ERRORE :
    assegna codice di ritorno di YP7SIGND a SXPK071-RET-COD
    ritorna al chiamante.
chiama la macro YP7SIGND per verificare la firma del PKCS#7 utilizzando il certificato contenuto in MBMCERT.
se ERRORE :
assegna codice di ritorno di YP7SIGND a SXPK071-RET-COD
ritorna al chiamante.
leggi il certificato presente in MBMCERT e ponilo in MBM-BUFIN-VAL, la sua lunghezza in MBM-BUFIN-LEN.
chiama **F3 caricamento certificato in ambiente MBM**
se ERRORE in chiamata MBM :
    assegna codice di ritorno a SXPK071-RET-COD
    ritorna al chiamante.
poni la label "**validity.notBefore**" in MBM-LABEL-VAL.
poni la lunghezza della label (18) in MBM-LABEL-LEN.
chiama **F4 – estrazione proprietà da certificato**
se ERRORE in chiamata MBM :
    assegna codice errore MBM a SXPK071-RET-COD
    ritorna al chiamante.
poni MBM-FIELD-VAL in MBM-FIELD-ASCII
chiama **F6 – conversione da ASCII ad EBCDIC**
se ERRORE in chiamata MBM :
    assegna codice errore MBM a SXPK071-RET-COD
    ritorna al chiamante.
poni MBM-FIELD_VAL in *data-inizio-validità*.
poni la label "**validity.notAfter**" in MBM-LABEL-VAL.
poni la lunghezza della label (17) in MBM-LABEL-LEN.
chiama **F4 – estrazione proprietà da certificato**
se ERRORE in chiamata MBM :
    assegna codice errore MBM a SXPK071-RET-COD
    ritorna al chiamante.
poni MBM-FIELD-VAL in MBM-FIELD-ASCII
chiama **F6 – conversione da ASCII ad EBCDIC**
se ERRORE in chiamata MBM :
    assegna codice errore MBM a SXPK071-RET-COD
    ritorna al chiamante.
poni MBM-FIELD_VAL in *data-fine-validità*.
Se *data-attuale* > *data-fine-validità* o *data-attuale* < *data-inizio-validità*

assegna codice routine a SXPK071-RET-COD
ritorna al chiamante.
poni la label "**extensions.authorityKeyIdentifier.keyIdentifier**" in MBM-LABEL-VAL
poni la lunghezza della label (**47**) in MBM-LABEL-LEN.
chiama **F4 – estrazione proprietà da certificato**
se ERRORE in chiamata MBM :
assegna codice di ritorno a SXPK071-RET-COD
ritorna al chiamante.
Individua, dal confronto fra la **AuthorityKeyIdentifier** reperita, presente in MBM-FIELD e le **SubjectKeyIdentifier** trovate in precedenza e presenti in SXPK071-CA-SKI, la PublicKey della CA da usare per la verifica del certificato e ponila in MBM-KEY.
Se chiave CA non trovata in tabella
assegna codice di ritorno della routine a SXPK071-RET-COD
ritorna al chiamante.
chiama **F1 -verifica firma certificato**
se ERRORE in chiamata MBM :
assegna codice di ritorno a SXPK071-RET-COD
ritorna al chiamante.
chiama **F5 liberazione certificato da ambiente MBM**
se ERRORE in chiamata MBM :
assegna codice di ritorno a SXPK071-RET-COD
ritorna al chiamante.
chiama la macro YP7SIGND per estrarre il dato firmato nel PKCS#7 e porlo nel file MBMDATO utilizzando la tabella di conversione.
se ERRORE :
assegna codice di ritorno di P7SIGND a SXPK071-RET-COD
ritorna al chiamante.
assegna SXPK071-RET-SUCCESS a SXPK071-RET-COD
ritorna al chiamante.

FUN-VERIEST:
richiama il servizio VERIRID.
se esito negativo restituisci il codice di errore della VERIRID.

Poni la label
"**extensions.cRLDistributionPoints.DistributionPoint.1.distributionPoint.fullName.uniformResourceIdentifier.1**" in MBM-LABEL-VAL.
Poni la lunghezza della label (**107**) in MBM-LABEL-LEN.
Chiama **F4: estrazione proprietà da certificato.**
Se ERRORE in chiamata MBM :
assegna codice di ritorno a SXPK071-RET-COD

ritorna al chiamante.
Poni MBM-FIELD-VAL in MBM-FIELD-ASCII
chiama **F6 – conversione da ASCII ad EBCDIC**
se ERRORE in chiamata MBM :
assegna codice errore MBM a SXPK071-RET-COD
ritorna al chiamante.
Ricerca in SXPK071-URL il valore di MBM_FIELD_VAL.
Se non trovato
assegna codice di ritorno della routine a RET-COD
ritorna al chiamante

Scandisci la tabella delle CRL:
Per ogni CRL la cui **AuthorityKeyIdentifier,** presente in SXPK071-CRL-AKI, coincide con la AuthorityKeyIdentifier del certificato:
Poni l'handle relativo della CRL in MBM-HANDLE
Chiama **F9 - verifica presenza estremi certificato in CRL**.
Se errore MBM
Assegna codice di ritorno della routine a RET-COD
Ritorna al chiamante.
Altrimenti se certificato revocato
Assegna codice di ritorno della routine a RET-COD
Ritorna al chiamante.
Poni la label "**subject.commonName**" in MBM-LABEL-VAL.
Poni la lunghezza della label (**18**) in MBM-LABEL-LEN.
Chiama **F4: estrazione proprietà da certificato.**
Se ERRORE in chiamata MBM :
assegna codice di ritorno a SXPK071-RET-COD
ritorna al chiamante.
Poni MBM-FIELD-VAL in MBM-FIELD-ASCII
chiama **F6 – conversione da ASCII ad EBCDIC**
se ERRORE in chiamata MBM :
assegna codice errore MBM a SXPK071-RET-COD
ritorna al chiamante.
Estrai da MBM-FIELD-VAL, contenente il commonName del soggetto, il codice fiscale, che si trova fra il "secondo" ed il "terzo" "/" e ponilo in SXPK071-EST-COD-FISCALE.
il commonName ha la struttura "Cognome/Nome/codiceFiscale/descrizione".
assegna SXPK071-RET-SUCCESS a SXPK071-RET-COD.
ritorna al chiamante.

FUN-VERICRL:

Per ogni certificato di CA presente nell'archivio MBMCA esegui le stesse operazioni descritte nella FUN-INIZIA (caricamento, verifica, prelievo info...).
Per ogni DDname relativo alle CRL presente nella configurazione della routine:

leggi la CRL
se ERRORE in lettura
assegna codice errore routine a SXPK071-RET-COD
ritorna al chiamante.
poni la CRL letta in MBM-BUFIN-VAL
poni la lunghezza della CRL in MBM-BUFIN-LEN
chiama **F7- caricamento CRL in ambiente MBM**
se ERRORE in chiamata MBM :
assegna codice errore MBM a SXPK071-RET-COD
ritorna al chiamante
poni la label "crlExtensions.authorityKeyIdentifier.keyIdentifier" in MBM-LABEL-VAL.
poni la lunghezza della label (50) in MBM-LABEL-LEN.
chiama **F8 - estrazione proprietà da CRL.**
se ERRORE in chiamata MBM :
assegna codice errore MBM a SXPK071-RET-COD
ritorna al chiamante.
poni MBM-FIELD-VAL in MBM-FIELD-ASCII
chiama **F6 – conversione da ASCII ad EBCDIC**
se ERRORE in chiamata MBM :
assegna codice errore MBM a SXPK071-RET-COD
ritorna al chiamante.
Individua, dal confronto fra la **AuthorityKeyIdentifier** reperita, presente in MBM-FIELD, e le **SubjectKeyIdentifier** trovate in precedenza dall'esame dei certificati di CA e presenti in SXPK071-CA-SKI, la PublicKey della CA da usare per la verifica della CRL e ponila in MBM-KEY.
Se chiave CA non trovata in tabella
assegna codice di ritorno della routine a SXPK071-RET-COD
ritorna al chiamante.
chiama **FB -verifica firma CRL**
se ERRORE in chiamata MBM :
assegna codice di ritorno a SXPK071-RET-COD
ritorna al chiamante.
poni la label "nextUpdate" in MBM-LABEL-VAL.
poni la lunghezza della label (10) in MBM-LABEL-LEN.
chiama **F8 - estrazione proprietà da CRL.**
se ERRORE in chiamata MBM :

assegna codice errore MBM a SXPK071-RET-COD
ritorna al chiamante.
poni MBM-FIELD-VAL in MBM-FIELD-ASCII
chiama **F6 – conversione da ASCII ad EBCDIC**
se ERRORE in chiamata MBM :
assegna codice errore MBM a SXPK071-RET-COD
ritorna al chiamante.
controlla che la data reperita (in MBM-FIELD) non sia anteriore più di 3 ore alla data ricevuta in SXPK071-VER-DATAORA..
se la data è più vecchia:
assegna codice errore della routine a SXPK071-RET-COD
ritorna al chiamante.
chiama **FA liberazione CRL da ambiente MBM**
se ERRORE in chiamata MBM :
assegna codice di ritorno a SXPK071-RET-COD
ritorna al chiamante.

5. MODALITA' DI RILASCIO DELLA ROUTINE

La routine SXPK07 è stata implementata utilizzando il set di librerie C0T2.MI20.* della partizione di PLT1.
In particolare:

- Il sorgente è contenuto in C0T2.MI20.SOURCE;
- Il load corrispondente in C0T2.MI20.LOAD;
- Il JCL di compilazione ($CMPCOBS) in C0T2.MI20.CNTL.

Nella libreria C0T2.MI20.CNF è presente un esempio di tabella di conversione ASCII-EBCDIC (membro AE3270).
Tale tabella di conversione biunivoca è stata ottenuta eseguendo i seguenti passi:

- creando un file su PC contenente tutti i caratteri ASCII (ordinati da X'00' a X'FF'),
- trasferendo su Host tale file con la funzione di File Transfer dell'emulatore Personal Communication, con modalità ASCII e senza CRLF[7].

5.1. DISPONIBILITÀ DELLA ROUTINE IN AMBIENTE CCC/LCM

Per permettere la richiamabilità dei programmi dell'area Spese sviluppati in ambiente CCC/LCM:

- è stato copiato il load su H0T5.RGSLIB,
- è stata aggiornata l'Exclusion List del prodotto CCC/LCM (inserendo un'entrata relativa a SXPK07),
- la COPY FSXPK071 è stata inserita in ambiente CCC/LCM[8] (System=SPE, Configuration=SVIL),
- per evitare problemi di omonimie è stata inserita un'entrata corrispondente a SXPK07 nella base dati di CCC/LCM (D01.TDLCOGG).

[7] Tale tabella è strettamente dipendente dai code-page dei sistemi coinvolti nel file trasfer.
[8] Questo comporta che il trasferimento della COPY sia a carico applicativo.

6. MODALITA' DI RICHIAMO DELLA ROUTINE DA PROGRAMMA COBOL

6.1. INVOCAZIONE

La routine COBOL SXPK07 è richiamabile da programma COBOL in ambiente MVS/BATCH.
La chiamata da programma COBOL viene effettuata nel modo descritto nei paragrafi seguenti dettagliati in base alla tipologia del chiamante.

6.1.1. Ordini di Pagare – Verifica della firma

Per effettuare lo sbustamento e la verifica della firma dei PKCS#7 da trattare il programma Cobol applicativo chiamante operante nel batch dell'applicazione 'Ordini di Pagare' dovrà seguire il seguente schema di chiamate della routine SXPK07.

```
WORKING-STORAGE SECTION.
.....
.....
 W-MBMPKCS7          PIC X(8).
 W-MBMDATO           PIC X(8).
 COPY 'FSXPK071'.
.....
.....
PROCEDURE DIVISION.
.....
INIZIA.
......
   MOVE 'INIZIA  ' TO SXPK071-COD-FUNZ
   CALL 'SXPK07'  USING SXPK071
.....
VERIRID.
.....
.....
    MOVE 'VERIRID ' TO SXPK071-COD-FUNZ
    MOVE W-MBMPKCS7 TO SXPK071-RID-DD-PKCS7
    MOVE W-MBMDATO TO SXPK071-RID-DD-DATO.
    CALL 'SXPK07'  USING SXPK071.
```

```
TERMINA.
......
   MOVE 'TERMINA ' TO SXPK071-COD-FUNZ
   CALL 'SXPK07'  USING SXPK071
.....
```

Il servizio INIZIA deve essere richiamato, prima di ogni altro servizio, per permettere l'esecuzione delle operazioni di inizializzazione.

Il servizio VERIRID può essere richiamato più volte, dopo aver richiamato INIZIA: tale servizio produce, in output, il dato firmato sbustato e convertito in EBCDIC. Nell'esempio precedente *W-MBMPKCS7* e *W-MBMDATO* devono contenere i DDname degli archivi allocati allo step nel JOB di esecuzione. Il primo deve contenere il PKCS#7 da trattare, nel secondo la routine restituisce il dato sbustato. Entrambi gli archivi devono essere chiusi all'atto della chiamata di SXPK07.

Il servizio TERMINA deve essere richiamato come ultimo servizio. Per riavviare il processo di verifica occorre rieseguire INIZIA.

6.1.2. Mandato Informatico – Verifica della firma

Per effettuare lo sbustamento e la verifica della firma dei PKCS#7 da trattare il programma Cobol applicativo chiamante operante nel batch dell'applicazione 'Mandato Informatico' dovrà seguire il seguente schema di chiamate della routine SXPK07.

```
WORKING-STORAGE SECTION.
.....
.....
 W-MBMPKCS7          PIC X(8).
 W-MBMDATO           PIC X(8).
 COPY 'FSXPK071'.
.....
.....
PROCEDURE DIVISION.
.....
INIZIA.
......
   MOVE 'INIZIA  ' TO SXPK071-COD-FUNZ
   CALL 'SXPK07'  USING SXPK071
.....
```

```
VERIEST.
.....
.....
    MOVE 'VERIEST ' TO SXPK071-COD-FUNZ
    MOVE W-MBMPKCS7 TO SXPK071-RID-DD-PKCS7
    MOVE W-MBMDATO TO SXPK071-RID-DD-DATO.
    CALL 'SXPK07'  USING SXPK071.

TERMINA.
......
   MOVE 'TERMINA ' TO SXPK071-COD-FUNZ
   CALL 'SXPK07'  USING SXPK071
.....
```

Il servizio INIZIA deve essere richiamato, prima di ogni altro servizio, per permettere l'esecuzione delle operazioni di inizializzazione.

Il servizio VERIEST può essere richiamato più volte, dopo aver richiamato INIZIA: tale servizio produce, in output, il dato firmato sbustato e convertito in EBCDIC. Nell'esempio precedente *W-MBMPKCS7* e *W-MBMDATO* devono contenere i DDname degli archivi allocati allo step nel JOB di esecuzione. Il primo deve contenere il PKCS#7 da trattare, nel secondo la routine restituisce il dato sbustato. Entrambi gli archivi devono essere chiusi all'atto della chiamata di SXPK07.

Il servizio TERMINA deve essere richiamato come ultimo servizio. Per riavviare il processo di verifica occorre rieseguire INIZIA.

6.1.2. Mandato Informatico – Verifica della validità delle CRL

Per effettuare, invece, il controllo di validità delle CRL nella fase iniziale del batch, il programma Cobol chiamante relativo dovrà richiamare la routine SXPK07 solo una volta, nel modo seguente.

```
WORKING-STORAGE SECTION.
.....
.....
 W-DATA-ORA          PIC 9(12).
 COPY 'FSXPK071'.
.....
.....
PROCEDURE DIVISION.
```

```
.....
VERICRL.
.....
.....
    MOVE 'VERICRL ' TO SXPK071-COD-FUNZ
    MOVE W-DATA-ORA TO SXPK071-VCRL-DATAORA
    CALL 'SXPK07'  USING SXPK071.

.....
```

6.2. JCL DI ESECUZIONE

Lo step di esecuzione del programma richiamante la routine SXPK07 deve contenere il parametro REGION impostato a 4M e l'allocazione dei data set descritti nei punti seguenti.

Ordini di Pagare – Verifica della firma

- MBMCA: DDname del data set contenente i certificati di CA;
- MBMCRLnn, con nn=01,…,10: DD DUMMY;
- MBMTCONV: DDname del dataset contenente la tabella di conversione ASCII-EBCDIC;
- MBMCERT: DDname del data set di appoggio del certificato utente 'sbustato' dalla routine. Tale data set deve essere un membro di un partitioned[9] (allocato con DISP=SHR o OLD)[10] Deve essere allocato in uno step precedente con DSORG=PO, LRECL=30004, RECFM=VB ed un'allocazione di spazio con una dimensione primaria di 150 CYLS;
- *Mbmdato*: DDname del data set di appoggio del dato. Deve avere LRECL=30004, RECFM=VB. Deve essere allocato in uno step precedente[10];
- *Mbmpkcs7*: DDname del data set di appoggio del file PKCS#7. Deve avere LRECL=30004. L'impostazione di RECFM può essere VB o FB;
- MBMURL: DD DUMMY.

Inoltre, in tale step non devono essere presenti schede DD associate a DDname del tipo: MBMTEMPn.

9 E' fondamentale che MBMCERT sia definito come membro di partitioned, altrimenti il prodotto MBM ne forza il valore di LRECL a 132.

10 E' fondamentale che tale data set sia allocato allo step con DISP ≠ NEW altrimenti il software MBM segnala errore.

E' consigliabile, per ottimizzare i tempi di esecuzione, che i data set *mbmdato*, MBMCERT e *mbmpkcs7* siano allocati temporanei.

Mandato Informatico – Verifica della firma

- MBMCA: DDname del data set contenente i certificati di CA;
- MBMCRLnn, con nn=01,…,10: DDname dei data set contenenti le CRL;
- MBMTCONV: DDname del dataset contenente la tabella di conversione ASCII-EBCDIC;
- MBMCERT: DDname del data set di appoggio del certificato utente 'sbustato' dalla routine. Tale data set deve essere un membro di un partitioned[11] (allocato con DISP=SHR o OLD)[10] Deve essere allocato in uno step precedente con DSORG=PO, LRECL=30004, RECFM=VB ed un'allocazione di spazio con una dimensione primaria di 150 CYLS;
- *Mbmdato*: DDname del data set di appoggio del dato. Deve avere LRECL=30004, RECFM=VB. Deve essere allocato in uno step precedente[12];
- *Mbmpkcs7*: DDname del data set di appoggio del file PKCS#7. Deve avere LRECL=30004. L'impostazione di RECFM può essere VB o FB;
- MBMURL: DDname dell'archivio (LRECL=80, RECFM= FB) contenente la lista degli URL da cui vengono scaricate le CRL.

Inoltre, in tale step non devono essere presenti schede DD associate a DDname del tipo: MBMTEMPn.

E' consigliabile, per ottimizzare i tempi di esecuzione, che i data set *mbmdato*, MBMCERT e *mbmpkcs7* siano allocati temporanei.

Mandato Informatico – Verifica della validità delle CRL

- MBMCA: DDname del data set contenente i certificati di CA;
- MBMCRLnn, con nn=01,…,10: DDname dei data set contenenti le CRL;
- MBMTCONV: DDname del dataset contenente la tabella di conversione ASCII-EBCDIC.

Inoltre, in tale step non devono essere presenti schede DD associate a DDname del tipo: MBMTEMPn.

11 E' fondamentale che MBMCERT sia definito come membro di partitioned, altrimenti il prodotto MBM ne forza il valore di LRECL a 132.

12 E' fondamentale che tale data set sia allocato allo step con DISP ≠ NEW altrimenti il software MBM segnala errore.

6.3. CODICI DI ERRORE

Il codice di errore è restituito nella variabile SXPK071-RET-COD.

La tabella seguente mostra un prospetto dei valori che possono essere assunti da tale campo.

Valore	Descrizione
0	Elaborazione corretta
8	Errore valori di input
10	Una delle CRL è scaduta da più di 3 ore
12	Errore interno (es. numero certificati CA maggiore di 30)
16	Certificato non valido (es. scaduto)
1nn	Errore di I/O (nn=file-status)
1nnn	Errore MBM (nnn= return code MBM)

7. PROGETTAZIONE DEI CASI DI TEST

Prog./ Tipol. di Test(*)	Funzione	Valori di input	Risultato atteso	Esito
1/M	INIZIA	SXPK071-COD-FUNZ = 'INIZIA' e archivio MBMCA contenente un numero di certificati maggiore di 30	Elaborazione termina con ret-cod=12	OK
2/M	INIZIA	SXPK071-COD-FUNZ = 'INIZIA' e archivio MBMCA omesso nel jcl di lancio	Elaborazione termina con ret-cod=135	OK
3/M	INIZIA	SXPK071-COD-FUNZ = 'INIZIA' e archivio MBMCA contiene un certificato non valido (scaduto)	Elaborazione termina con ret-cod=16	OK
4/M	INIZIA	SXPK071-COD-FUNZ = 'INIZIA' MBMCRLi vuote	RC = 0	OK
5/M	INIZIA	SXPK071-COD-FUNZ = 'INIZIA' MBMCRLi in alternativa vuote e piene	RC = 0 SXPK071-CRL riempita in modo alterno	OK
6/M	INIZIA	SXPK071-COD-FUNZ = 'INIZIA' MBMCRLi più di 10	RC = 12 Messaggio:' 'SFONDAMENTO TABELLA IN UNA DELLE DUE DIMENSIONI'	OK
7/M	INIZIA	SXPK071-COD-FUNZ = 'INIZIA' MBMCRLi contenente un CRL firmato da una CA non conosciuta	RC = 12 Messaggio:' 'NON TROVATO CERT. CA CON SKI CORR. AD AKI DI CRL'	OK
8/M	INIZIA	SXPK071-COD-FUNZ = 'INIZIA' MBMCRLi contenente una CRL senza AKI	RC = 12 Messaggio:' 'ESTENSIONE AKI ASSENTE IN FILE DI CRL'	OK

9/M	INIZIA	SXPK071-COD-FUNZ = 'INIZIA' MBMURL con più di dieci entrate	RC =12 Messaggio:' 'SFONDAMENTO TABELLA URL NUM. ENT. MAGG. 10'	OK
10/M	INIZIA	SXPK071-COD-FUNZ = 'INIZIA' MBMURL vuoto	RC =8 Messaggio: 'MBMURL VUOTO'	OK
11/M	INIZIA	SXPK071-COD-FUNZ = 'INIZIA' e archivio MBMCA vuoto	Elaborazione termina con ret-cod=8	OK
12/M	INIZIA	SXPK071-COD-FUNZ = 'INIZIA' e archivio MBMCA contenente certificati validi	Elaborazione termina con ret-cod = 0 e la tabella SXPK071-CA contiene i dati dei certificati di CA (SKI e chiave pubblica) correttamente caricati	OK
13/M	VERIRID	SXPK071-COD-FUNZ = 'VERIRID', SXPK071-RID-DD-DATO = 'MBMDATO' e SXPK071-CA contiene i dati dei certificati di CA correttamente caricati. Tutti gli archivi della funzione contengono dati consistenti.	Elaborazione termina con ret-cod = 0 e il dato 'sbustato' è restituito nel dataset MBMDATO	OK
14/M	VERIRID	SXPK071-COD-FUNZ = 'VERIRID', SXPK071-RID-DD-DATO = 'MBMDATO' e SXPK071-CA contiene i dati dei certificati di CA correttamente caricati. L'archivio SXPK071-RID-DD-PKCS#7 contiene un certificato scaduto.	Elaborazione termina con ret-cod=16	OK
15/M	VERIRID	SXPK071-COD-FUNZ = 'VERIRID', SXPK071-RID-DD-DATO = 'MBMDATO' e SXPK071-CA contiene i dati dei certificati di CA correttamente caricati. L'archivio SXPK071-RID-DD-PKCS#7 contiene un	Elaborazione termina con ret-cod=12	OK

		certificato avente un AKI che non coincide con nessun SKI della tabella SXPK071-CA		
16/M	INIZIA	SXPK071-COD-FUNZ = 'INIZIA' e archivio MBMCA contenente un certificato inconsistente (formato asn.1 non valido)	Elaborazione termina con ret-cod=1029	OK
17/M	VERIRID	SXPK071-COD-FUNZ = 'VERIRID', SXPK071-RID-DD-DATO = 'MBMDATO' e SXPK071-CA contiene i dati dei certificati di CA correttamente caricati. L'archivio SXPK071-RID-DD-PKCS#7 contiene un file dal formato asn.1 non valido	Elaborazione termina con ret-cod=1198	OK
18/M	VERIRID	SXPK071-COD-FUNZ = 'VERIRID', SXPK071-RID-DD-DATO = 'MBMDATO' e SXPK071-CA contiene i dati dei certificati di CA correttamente caricati. L'archivio SXPK071-RID-DD-PKCS#7 non è allocato allo step di esecuzione	Elaborazione termina con ret-cod=1200	OK
19/M	VERIRID	SXPK071-COD-FUNZ = 'VERIRID', SXPK071-RID-DD-DATO = 'MBMDATO' e SXPK071-CA contiene i dati dei certificati di CA correttamente caricati. MBMTCONV non è allocato allo step di esecuzione.	Elaborazione termina con ret-cod=1207	OK
20/M	VERIRID	SXPK071-COD-FUNZ = 'VERIRID', SXPK071-RID-DD-DATO = 'MBMDATO' e SXPK071-CA contiene i dati dei certificati di CA correttamente caricati. L'archivio SXPK071-RID-DD-DATO non è allocato allo step di esecuzione	Elaborazione termina con ret-cod=1203	OK
21/M	VERIRID	SXPK071-COD-FUNZ = 'VERIRID', SXPK071-RID-DD-DATO = 'MBMDATO' e SXPK071-CA contiene i dati dei	Elaborazione termina con ret-cod=139	OK

		certificati di CA correttamente caricati. L'archivio MBMCERT è allocato con DSORG=PS		
22/M	VERIRID	SXPK071-COD-FUNZ = 'VERIRID', SXPK071-RID-DD-DATO = 'MBMDATO' e SXPK071-CA contiene i dati dei certificati di CA correttamente caricati. L'archivio SXPK071-RID-DD-PKCS#7 non contiene il certificato utente	Elaborazione termina con ret-cod=1196	OK
23/M	VERIEST	SXPK071-COD-FUNZ = 'VERIEST' MBMCERT con distr point non tra quelli presenti in MBMURL	RC = 12 Messaggio: 'NON TROVATA IN TABELLA URL CORRISPONDENTE AL CRL. DISTR. NAME'	
24/M	VERIEST	SXPK071-COD-FUNZ = 'VERIEST' MBMCER contenente certificato revocato come risulta da MBMCRLi	RC = 16 Messaggio: CERTIFICATO NON VALIDO - REVOCATO	
25/M	VERIEST	SXPK071-COD-FUNZ = 'VERIEST' MBMCER contenente certificato il cui AKI non corrisponde a nessuna CRL.	RC = 12 Messaggio: NON TROV. NESSUNA CRL CORR.AD AKI DI CERTIF. UT'	
26/M	VERIEST	SXPK071-COD-FUNZ = 'VERIEST' MBMCER contenente certificato con distr point corretto e non revocato.	RC = 0	
27/M	VERICRL	SXPK071-COD-FUNZ = 'VERICRL' data nextupd 28/02/2001/230101 data nextupd 29/02/2004/230101 data nextupd 30/06/2001/230101 data nextupd 31/08/2001/230101	Data con aggiunta elapsed di 3 ore corretta.	

		data nextupd 31/12/2001/230101		
28/M	VERICRL	SXPK071-COD-FUNZ= 'VERICRL' MBMCRLi data nextupd + 3 < data input	RC = 10 Messaggio 'CRLi NON VALIDA – PROVVEDERE AL SUO SCARICO'	OK
29/M	VERICRL	SXPK071-COD-FUNZ= 'VERICRL' MBMCRLi campo 'NextUpdate' non presente in CRL.	RC = 12 Messaggio 'CAMPO NEXTUPDATE NON PRESENTE IN CRLi '	OK
30/M	VERICRL	SXPK071-COD-FUNZ= 'VERICRL' archivi MBMCRLi tutti vuoti.	RC = 10 Messaggio 'TUTTE LE CRL SONO VUOTE'	OK

(*) Tipologia di test: 'M'= modulo elementare; 'I'= integrazione; 'S' sistema

ALLEGATO

COPY COBOL FSXPK071

```
*
*   COPY COBOL FSXPK071
*   AREE DI COMUNICAZIONE FRA PGM. COBOL E
*   SUBROUTINE SXPK07
*
  01 SXPK071.
     05 SXPK071-COD-FUNZ             PIC X(7).
        88  SXPK071-COD-VERICRL  VALUE 'VERICRL'.
        88  SXPK071-COD-INIZIA   VALUE 'INIZIA '.
        88  SXPK071-COD-TERMINA  VALUE 'TERMINA'.
        88  SXPK071-COD-VERIRID  VALUE 'VERIRID'.
        88  SXPK071-COD-VERIEST  VALUE 'VERIEST'.
*
     05 SXPK071-RET-COD              PIC S9(4) COMP.
        88  SXPK071-RET-SUCCESS  VALUE +0.
*
     05 SXPK071-CA.
        08 SXPK071-CA-DATI OCCURS 30 INDEXED CA-PUNT.
           10 SXPK071-CA-SKI.
              15 SXPK071-CA-SKI-VAL         PIC X(256).
              15 SXPK071-CA-SKI-LEN         PIC S9(9) COMP.
           10 SXPK071-CA-KEY.
              15 SXPK071-CA-KEY-N           PIC S9(4) COMP.
              15 SXPK071-CA-KEY-E.
                 20 SXPK071-CA-KEY-E-EL OCCURS 3 PIC S9(4) COMP.
              15 SXPK071-CA-KEY-M.
                20 SXPK071-CA-KEY-M-EL OCCURS 1025 PIC S9(4) COMP.
*
     05 SXPK071-CRL.
        08 SXPK071-CRL-DATI OCCURS 2 INDEXED CRL-PUNT.
           10 SXPK071-CRL-SKI.
              15 SXPK071-CRL-SKI-VAL        PIC X(256).
              15 SXPK071-CRL-SKI-LEN        PIC S9(9) COMP.
           10 SXPK071-CRL-HANDLE OCCURS 5   PIC S9(9) COMP.
*
     05 SXPK071-URL.
        08 SXPK071-URL-VAL OCCURS 10 INDEXED URL-PUNT PIC X(80).
*
     05 SXPK071-VERICRL.
        10 SXPK071-VCRL-DATAORA.
           15 SXPK071-VCRL-DATAORA-AA  PIC 9(2).
           15 SXPK071-VCRL-DATAORA-MM  PIC 9(2).
           15 SXPK071-VCRL-DATAORA-GG  PIC 9(2).
           15 SXPK071-VCRL-DATAORA-HH  PIC 9(2).
           15 SXPK071-VCRL-DATAORA-MI  PIC 9(2).
           15 SXPK071-VCRL-DATAORA-SS  PIC 9(2).
*
     05 SXPK071-VERIRID.
```

```
           10 SXPK071-RID-DD-PKCS7       PIC X(8).
           10 SXPK071-RID-DD-DATO        PIC X(8).
*
        05 SXPK071-VERIEST.
           10 SXPK071-EST-COD-FISC       PIC X(16).
```