Il processo di autenticazione

Roma, 2 febbraio 2000

Indice

1. CHANGE LOG

Data	*Versione*	*Descrizione*	*Note*
29 settembre 1999	00	Prima emissione	--
22 ottobre 1999	01		--
15 novembre 1999	02		
2 febbraio 2000	03		--

2. INTRODUZIONE

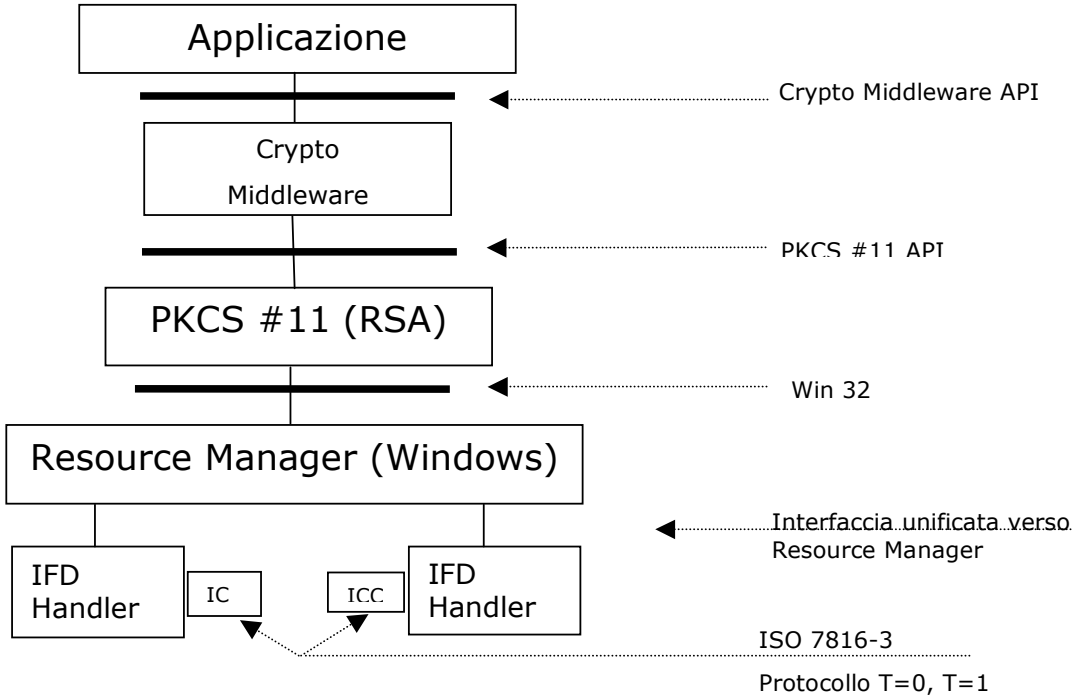
2.1 Il riconoscimento in rete per la fruizione dei servizi

In considerazione dell'architettura definita per la Carta d' Identità Elettronica e dell'utilizzo della componente chip per il riconoscimento in rete della carta nei confronti di un server applicativo che eroga dei servizi, la soluzione che si è scelta è quella della Strong Authentication.

La Strong Authentication richiede l'utilizzo di funzioni tipiche di una Public Key Infrastructure per cui, oltre alle piattaforme standard per l'utilizzo delle Smart Card, è necessario considerare cosa è normalmente disponibile sul mercato per interagire con una PKI utilizzando le Smart Card come dispositivi di cifratura.

La Figura [1.1] schematizza un esempio tipico di applicazione utente che interagisce con una Infrastruttura a Chiave Pubblica.

Figura 1.1



3 CRYPTO MIDDLEWARE ED API PKCS#11

Le Infrastrutture a Chiave Pubblica mettono a disposizione dei *Client,* che operano su reti aperte, apposite applicazioni (piattaforme) per gestire i servizi di cifratura/decifratura.

Tali piattaforme, chiamate generalmente Crypto Middleware, svolgono le seguenti funzioni:

- Accesso LDAP ai servizi di Directory;
- Gestione in *Cache* della *Cerificate Revocation List*;
- *Parsing* dei Certificati Digitali;
- Costruzione di strutture PKCS#7;
- Richiesta di certificazione di chiavi pubbliche;
- Richiesta di revoca di certificati
- Interfaccia ad alto livello verso le funzioni di cifratura.

Queste piattaforme, a loro volta, poggiano su strati software, o API, che le isolano dai dispositivi di cifratura, nel nostro caso le Smart Card. Le API più diffuse sono le PKCS#11. Le caratteristiche salienti di queste API sono:

- Consentire ai Crypto Middleware di prescindere dai dispositivi che memorizzano chiavi e sviluppano crittografia;
- Fornire ai Crypto Middleware una interfaccia standard;
- Rendere portabili le applicazioni negli ambienti in cui la crittografia è trattata con queste API.

La Figura [1.1] suggerisce le seguenti considerazioni:

- la funzionalità di cifratura principale richiesta alla Carta di Identità Elettronica è la Autenticazione Forte in rete;
- la applicazione di autenticazione risiede in parte sulla Workstation che interfaccia la Smart Card ed in parte sulla macchina che eroga il servizio per cui la autenticazione è richiesta.

Le considerazioni precedenti inducono ad analizzare nel dettaglio il processo di Autenticazione Forte per stabilire cosa è realmente necessario per realizzare in modo interoperabile questo processo.

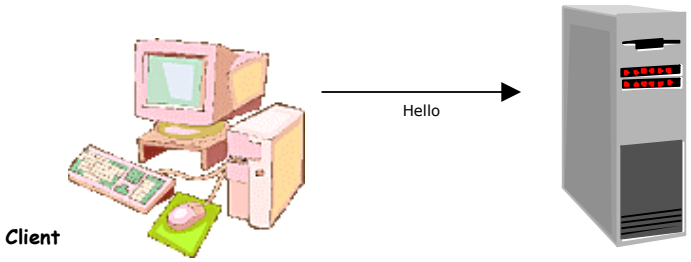
Nei paragrafi successivi saranno esaminati gli step procedurali del processo di *Strong Authentication* ed i comandi ISO di gestione della Smart Card al fine di determinare una interfaccia semplice, efficiente e che consenta sia di interoperare con le applicazioni che di rivolgersi ad un mercato aperto di fornitori di chip

Particolare attenzione è data alla interoperabilità con le PKI di mercato al fine di non vincolare alle scelte tecnologiche del Ministero dell'Interno il processo di autenticazione

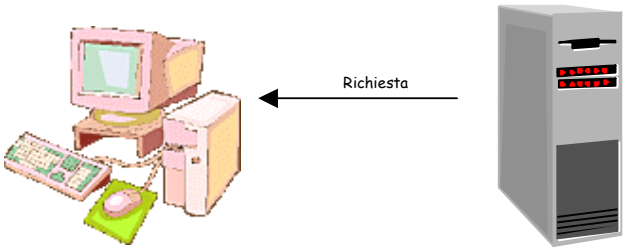
4 PROCESSO DI STRONG AUTHENTICATION

Questo processo consente la identificazione da remoto della carta per la fruizione dei servizi erogati da una applicazione residente presso una Pubblica Amministrazione Centrale. Riferendoci alla struttura dei dati in essa contenuti (vedere documento …) e tenendo presente il vincolo di definire un processo che utilizzi librerie standard ed aperte, i passi previsti dalla procedura sono:

L'applicazione client stabilisce la comunicazione con l'applicazione server.



L'applicazione server richiede all'applicazione client il file "C_Carta" contenente il

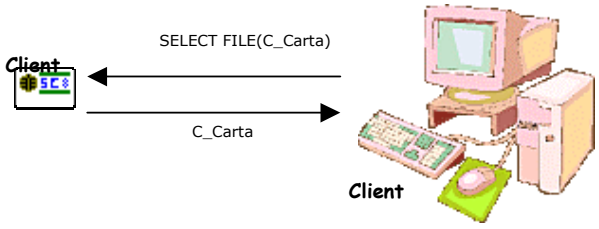


certificato (ID_Carta più la chiave pubblica $K_{pub\ della\ carta}$).

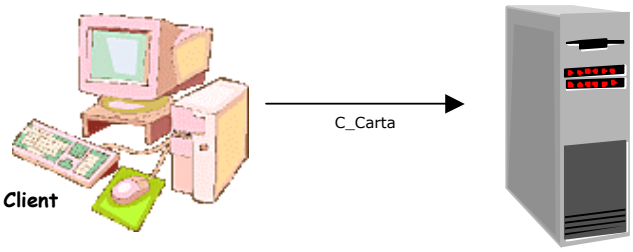
L'applicazione client interroga la carta e legge tale file mediante i comandi APDU:

SELECT FILE (C_Carta)

READ BINARY

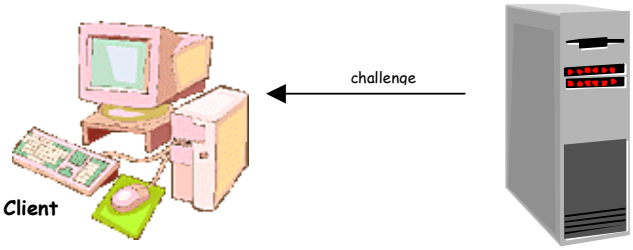


L'applicazione client invia il file "C_Carta" al server



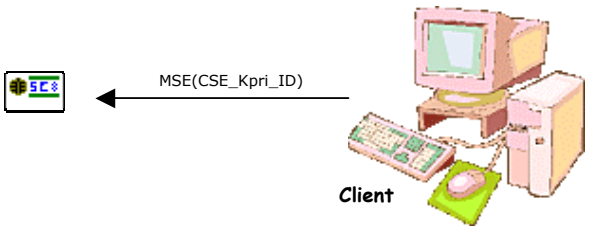
L'applicazione server verifica la validita' del certificato mediante MI_{pub} ed estrae da esso ID_Carta e K_{pub}.

L'applicazione server genera una stringa di challenge e la invia al client rimanendo in attesa della risposta.

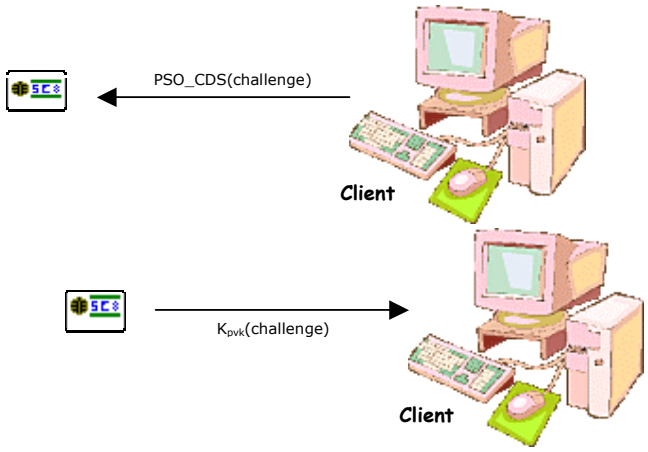


L'applicazione client seleziona K_{pri} mediante il comando *MSE(Manage Security Environment)*

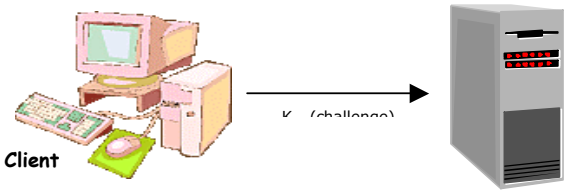
In tal modo K_{pri} e' attivata e verra' usata in tutte le successive operazioni di cifratura effettuate dalla carta.



Mediante il comado PSO (*Perform Security Operation*)la carta esegue la cifratura del *challenge* usando K_{priv} precedentemente attivata, e restituisce all'applicazione *client* la stringa ottenuta. La chiave privata che e' stata generata dalla carta in fase di inizializzazione, risulta invisibile dall'esterno e comunque impossibile da estrarre dalla carta.



Il client invia al server in attesa il challenge firmato ricevuto dalla carta.



L'applicazione server verifica la stringa ricevuta e la confronta con il challenge precedentemente generato.

Se tale confronto ha esito positivo la carta e' autenticata. A questo proposito è necessario che l'algoritmo di verifica, residente sul server, sia compatibile con quello usato dalla carta per cifrare il challenge.

5 COMANDI DI GESTIONE

La norma ISO 7816 parte 4 e parte 8 definisce, oltre alla struttura del File System , anche i comandi per interagire a livello applicativo. Tali comandi sono chiamati APDU (Application Protocol Data Unit).

La struttura degli APDU è relativamente semplice ed è costituita da:

- un *header* che comprende classe, tipo di istruzione e parametri di controllo;
- un *body* che comprende il campo lunghezza dei dati applicativi, i dati applicativi e la lunghezza massima dei dati inerenti la risposta al comando.

Ad ogni comando corrisponde un APDU *Response* strutturato in un campo dati ed in un campo "stato" della lunghezza di 2 Byte in cui sono contenute tutte le informazioni necessarie a gestire la risposta.

In funzione dei passi procedurali del processo di Autenticazione sono individuati i seguenti comandi APDU:

SELECT FILE, per selezionare l'Elementary File che contiene il certificato della Carta di Identità Elettronica (C_Carta);

READ BINARY, per leggere il certificato;

MSE (Manage Security Environment), per attivare la chiave privata di autenticazione;

PSO (Perform Security Operation), per cifrare il *challenge* da inviare alla applicazione server.

Dalle considerazioni precedenti si evince che è necessaria solo una parte molto modesta delle funzionalità della architettura di Figura [1.1], almeno per quanto concerne l'accoppiamento della componente applicativa che gestisce la Carta di Identità Elettronica.

Un approccio efficiente, che consente alle applicazioni di gestire in modo interoperabile la componente Chip della Carta di Identità Elettronica è quello di realizzare una libreria di interfaccia che implementi i comandi descritti precedentemente.

Tale libreria è realizzabile tanto nella architettura PC/SC quanto in altri environment operativi quali UNIX , MAC OS ecc.

In ultima analisi le funzioni della libreria sono:

- servizi di amministrazione
- funzioni di interfaccia verso la CIE
- identificazione Utente
- selezione File
- Read File
- selezione chiave
- autenticazione Interna
- gestione errori ed anomalie

5.1 Considerazioni sulla interoperabilità

La interoperabilità si ottiene definendo l'algoritmo crittografico di autenticazione ed il formato del messaggio autenticato. Una volta scelti algoritmo e formato non esiste ambiguità sui messaggi di autenticazione.

Di seguito sono esposti i razionali per la scelta di questi elementi.

5.1.1 Gli algoritmi

Gi algoritmi asimmetrici comunemente impiegati dalle Smart Card ed idonei per realizzare la autenticazione sono: l'algoritmo RSA e l'algoritmo DSA

Questi algoritmi sono onerosi dal punto di vista computazionale e quindi sono realizzati utilizzando un coprocessore aritmetico. La lunghezza della chiave dipende dalla capacita del coprocessore di effettuare moltiplicazioni in modulo. Questo comporta una lunghezza massima di chiave pari al massimo modulo supportato dal coprocessore per l'algoritmo DSA ed una lunghezza massima pari al doppio del modulo per l'algoritmo RSA grazie alla possibilità di utilizzare il Chinese Remaider Theorem.

In virtù delle considerazioni precedenti si ritiene opportuno optare per l'algoritmo RSA in quanto consente di:

- poter scegliere tra una vasta gamma di fornitori ;
- estendere in futuro la lunghezza della chiave.

5.1.2 I Formati

I formati utilizzati dalla crittografia asimmetrica sono:

- il formato ISO 9796 parte 2;
- il formato PKCS#1.

Il formato ISO 9796-2 è adottato dallo standard EMV per la autenticazione statica e dinamica.

In applicazioni non EMV questo formato è consigliabile quando l'intero processo di autenticazione comporta l'utilizzo di due Smart Card (Mutua autenticazione interna ed esterna).

Il formato PKCS#1 può essere considerato "standard de facto" ed i messaggi di autenticazione (Response) costruiti secondo questo formato possono essere verificati dalle applicazioni che utilizzano gli strumenti tipici delle Public Key Infrastructure.

Da quanto esposto si evince che è opportuno utilizzare RSA come algoritmo di autenticazione e PKCS#1 come formato.

5.2 Considerazioni sulle Smart Card Crittografiche

In questo paragrafo sono riesaminati i comandi ISO 7816-4/8 adatti ad effettuare la autenticazione forte e vengono fatte alcune considerazioni inerenti le modalità con cui le Smart Card implementano questi comandi .

Lo scopo di questa analisi è quello di poter emettere in modo preciso le specifiche della libreria e di non limitare lo spettro delle Smart Card utilizzabili.

Internal Authentication **.**

E' un processo di Signature con elaborazione del formato all'interno della Smart Card (Hashing, Padding e Signature). In generale le Smart Card implementano uno solo dei due formati visti precedentemente e non sempre con gli stessi parametri APDU.

Perform Security Operation**.**

E' un processo di *Signature* trasparente ovvero con formato calcolato all'esterno della carta.

La *Signature* trasparente è implementata da tutte le Smart Card con coprocessore aritmetico, ma non in generale con gli stessi parametri APDU.

APDU Response

Non tutte le Smart Card restituiscono direttamente nella risposta ai comandi (APDU response) i dati processati ma richiedono un ulteriore comando di READ_RESULT o GET_DATA.

Il seguente approccio consente di superare facilmente le difformità nei comandi APDU:

- Nella CIE è presente un Elementary File (Parametri_APDU) di tipo Read Only che contiene i parametri APDU della carta;
- La libreria compila in modo dinamico gli APDU con i parametri contenuti nell'Elementary File.

Queste operazioni sono trasparenti alla applicazione e non appesantiscono la libreria di interfaccia.

6 STRONG AUTHENTICATION LATO SERVER

Quanto affermato nei precedenti paragrafi è un solido punto di partenza per risolvere il problema della autenticazione forte in rete per quanto concerne il *Client* e la Carta di Identità Elettronica. E' ora necessario definire la componente server del processo di Autenticazione.

Per quanto concerne il *Server* le funzionalità sono più complesse ed è ragionevole pensare a strumenti software di interfaccia più sofisticati che, oltre a garantire la interoperabilità con le informazioni prodotte dai Client non vincolino la scelta degli strumenti di produzione dei Certificati Digitali.

La figura [5.1] illustra i componenti che intervengono nel processo di autenticazione**.**

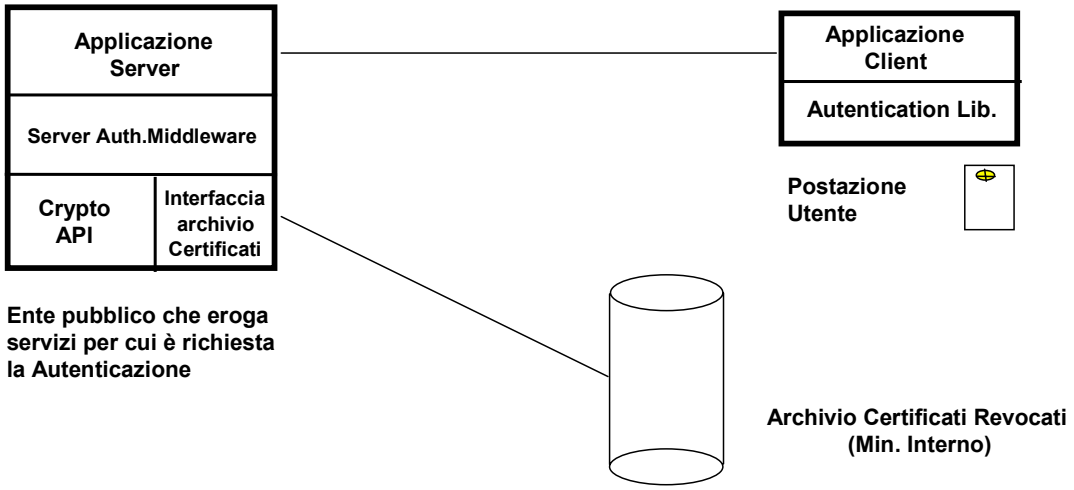


Fig. 5.1

6.1.Server Authentication Middleware

Il Server Authentication Middleware è lo strato software che fornisce i servizi crittografici alla Applicazione.

Le funzioni che questo strato deve rendere disponibili sono:

- Generazione di quantità random;
- Funzioni di Hash;
- Gestione di Certificati digitali in formato X509v3 ;
- Verify Certificate (per validare il certificato della CIE)
- Verify Signature (per validare il messaggio di Autenticazione)

- Caricamento della Certficate Revocation List
- Gestione della Revocation List.

Quelle descritte sono solamente un subset delle funzionalità Client di una Infrastruttura a Chiave Pubblica ma sono comunque sufficienti per considerare la possibilità di utilizzo di software di mercato.

I requisiti di questa componente software sono:

1. Servizi crittografici come descritto nei punti precedenti;
2. Interoperabilità con i Client;
3. Indipendenza degli strumenti di produzione dei certificati.

Il primo requisito è una funzionalità tipica dei *Middleware* crittografici, il secondo requisito è soddisfatto dalla scelta fatta per Algoritmo e Formato che rende univoca la struttura del messaggio di Autenticazione mentre il terzo requisito è garantito dal circuito di emissione della Carta di Identità Elettronica.

Il circuito di emissione della Carta di Identità Elettronica prevede che la attività di registrazione e la richiesta di certificazione siano centralizzate e quindi non esiste nessuna interazione diretta tra servizio di certificazione e Server Autentication Middleware.

In altri termini l'Autentication Middleware è indipendente dal Servizio di Certificazione una volta definita la Policy di generazione dei Certificati.

6.2 Modalità di Envelope

Nei paragrafi precedenti è stato definito il formato dei dati di autenticazione per consentire un livello minimo di interoperabilità infatti i dati di autenticazione in formato PKCS#1 sono trattabili dai "Crypto Middleware" di mercato.

E' importante osservare che le informazioni di autenticazione trattate dai "Crypto Middleware" sono normalmente incapsulate in un "Envelope" secondo il formato PKCS#7 e quindi ,al fine di estendere il campo di interoperabilità ,è opportuno adottare questa modalità di incapsulamento dei dati di autenticazione.

Un Envelope PKCS#7 contiene in modo strutturato tutte le informazioni necessarie al processo di verifica della autenticazione ovvero: il formato di encryption (PKCS#1), l'algoritmo di hash utilizzato,idati di autenticazione ed i dati autenticati ,il certificato della CA (Certificato del Ministero dell'Interno) il certificato della CIE.

Alla luce delle precedenti considerazioni si propone l'adozione dello "Envelope" PKCS#7 per strutturare i dati di autenticazione e quindi la libreria di interfaccia della CIE dovrà contenere un modulo SW per il PKCS#7 encoding.

7. IL RICONOSCIMENTO IN RETE PER L'AGGIORNAMENTO DEI DATI RELATIVI ALLA FRUIZIONE DEI SERVIZI

Nei paragrafi precedenti è stato approfondito il tema della autenticazione della CIE verso un Ente in grado di erogare servizi , in questo paragrafo viene completato il processo di Autenticazione specificando le procedure che permettono alla CIE di verificare l'autenticità del servizio remoto con cui sta interagendo. Questo processo verrà chiamato :Autenticazione Esterna

Un altro tema trattato in questo paragrafo è il caricamento remoto sicuro di dati nella CIE da parte dell'Ente che eroga il servizio , questo processo verrà chiamato Secure Messaging.

I processi di Autenticazione e di Secure Massaging garantiscono la interazione diretta tra Ente e Carta di Identità Elettronica e prevengono dai tentativi di intrusione che possono essere condotti sulla rete.

Questi processi sono sviluppati con l'utilizzo di crittografia simmetrica e sono proposti i seguenti due metodi per la gestione delle chiavi :

- Un metodo basato sullo utilizzo di "*diversified key*" derivate da una o più *"master key"* e caricate nella CIE durante la fase di emissione;

- Un metodo basato sullo scambio di una chiave di sessione generata in modo casuale dalla CIE e crittografata ed autenticata dalla CIE stessa.

Il primo metodo è consolidato e comunemente impiegato nelle applicazioni *"Smart Card Based"* ma richiede particolare attenzione nella custodia e distribuzione delle chiavi.

Il secondo metodo ,in fase di valutazione , richiede la scrittura di un comando ad hoc che consente la generazione, la crittografia e la autenticazione della chiave di sessione all'interno della CIE al fine di garantire alla Applicazione Server che quella chiave può essere decrittografata solo da lei e generata solamente dalla CIE.

7.1 Autenticazione Esterna

Il processo di Autenticazione Esterna è attivato dall'Applicazione remota che deve poter accedere ai file della Carta di Identità Elettronica per aggiornarne i dati.
Questo processo utilizza la chiave simmetrica di sessione $\mathbf{K_S}$ che , in funzione delle modalità descritte precedentemente, sarà derivata dalla Applicazione

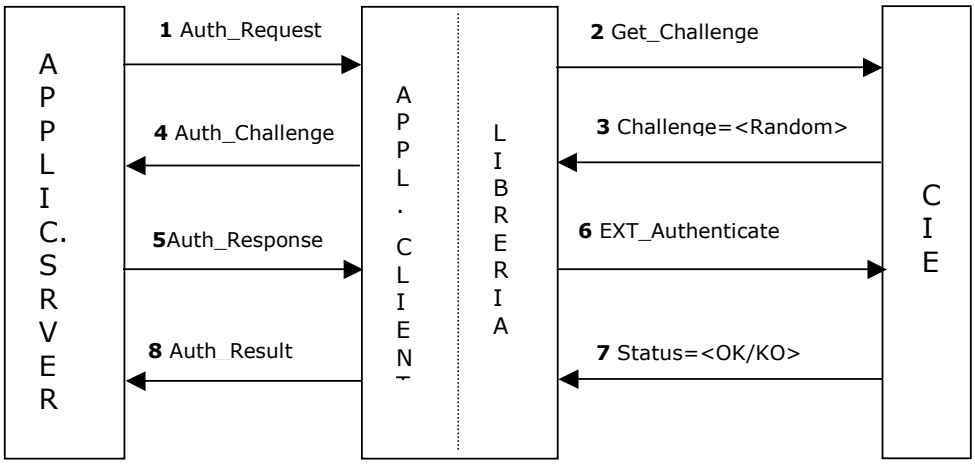
Server a partire dalla "*master key*" $\mathbf{K_M}$ per mezzo dello identificativo della CIE oppure scambiata con modalità crittografiche asimmetriche.

La Autenticazione Esterna coinvolge I seguenti moduli:

- Applicazione Server
- Applicazione Client
- Libreria di interfaccia e CIE.

La Figura [7.1] schematizza il flusso di informazioni scambiate tra i vari moduli che concorrono al processo di autenticazione esterna.

Il processo di Autenticazione Esterna è attivato dalla Applicazione Server dopo che è stata riconosciuta (autenticata) la Carta ed il titolare.



Il processo di Autenticazione si svolge secondo i seguenti passi procedurali:

1. L'Applicazione Server richiede alla Applicazione Client di essere autenticata dalla CIE tramite il messaggio "**Aut_Request**" ;
2. L'applicazione Client ,servendosi della LIBRERIA , invia alla CIE il comando "**Get_Challenge**" ;
3. La risposta della CIE è un numero random (il Challenge)
4. L'Applicazione Client invia alla Applicazione Server il messaggio "**Aut_Challenge**" che contiene il numero random generato dalla CIE;

5. L'Applicazione Server crittografa il Challenge con la chiave di sessione $\mathbf{K_S}$ e lo invia alla Applicazione Client con il messaggio "**Auth_Response**"

6. La applicazione Client , servendosi della libreria , invia alla CIE il comando "**EXT_Authentication**"

7. La CIE utilizza la chiave di sessione $\mathbf{K_S}$, relativa alla directory a cui si vuole accedere, per verificare la autenticità del Response;se la verifica è positiva viene inviato un messaggio di consenso alla Applicazione Client tramite la libreria e viene reso disponibile l'accesso ai file appartenenti a quella directory;

8. L'Applicazione Client invia alla Applicazione Server il messaggio "Auth_Result" per comunicare l'esito del processo.

Nella descrizione del processo di Autenticazione Esterna si sono trascurati dettagli procedurali quali la gestione delle eventuali anomalie e le "Retry" tipiche di questi processi in quanto non incidono sulle funzionalità della CIE.

7.2 Secure Messaging

Il processo di Secure Messaging è attivato dopo i processi di Autenticazione e consente lo scambio dati crittografato tra CIE ed Applicazione Server .

Esso utilizza una chiave di sessione diversificata $\mathbf{K_D}$ che è:

- derivata da $\mathbf{K_S}$ attraverso la generazione di una quantità Random ,qualora venga scelto di distribuire le chiavi secondo metodi convenzionali durante la fase di emissione;

- coincidente con $\mathbf{K_S}$. qualora venga scelta la distribuzione delle chiavi di sessione dalla CIE alle Applicazioni Server con un apposito comando basato sull'utilizzo di crittografia asimmetrica.

Il comando di Secure Messaging sarà implementato secondo la norma ISO 7816-4 nella modalità "Secure Messaging for Confidentiality".

8 SINTESI DELLE FUNZIONALITÀ DELLA LIBRERIA

In questo capitolo sono riassunte le funzionalità della libreria di interfaccia verso la CIE per la realizzazione della procedura di autenticazione lato Client in modalità Challenge Response.

Funzionalità a basso livello

La libreria dovrà implementare verso la CIE i seguenti comandi a basso livello:

- Select_File
- Read_Binary
- Get_Challenge
- Write Binary
- Read Record
- Write Record

Questi comandi , realizzati secondo la norma ISO7816-4 , consentiranno l'accesso in lettura/scrittura ai Files della CIE e saranno lo strumento base per lo APDU Constructor.

APDU Constructor

La libreria costruirà i comandi del protocollo applicativo APDU utilizzando i dati contenuti nel Public File di configurazione. I comandi APDU costruiti dovranno consentire di:

- Selezionare la chiave privata di autenticazione interna;
- Attivare le procedure di crittografia asimmetrica (RSA) per il calcolo dell'Authentication Response ;
- Selezionare la chiave di autenticazione esterna $\mathbf{K_S}$ secondo la modalità che verrà adottata;
- Attivare le procedure Autenticazione Esterna;
- Selezionare/derivare la chiave di sessione esterna $\mathbf{K_D}$ secondo la modalità che verrà adottata;

- Attivare le procedure di Secure Messaging in modalità Confidentiality.

PKCS#1 Formatter

Questo modulo della libreria calcolerà la funzione di hash (SHA1) sui dati di autenticazione (Challenge) e preparerà i dati da inviare alla CIE secondo il formato PKCS#1.I dati strutturati secondo questa modalità saranno inviati alla CIE con il comando APDU realizzato tramite lo APDU Constructor.

PKCS#7 Encoder

Questo modulo della libreria costruirà il messaggio di Response secondo le modalità di Envelope PKCS#7.ed un security profile definito.

9 STANDARD DI RIFERIMENTO

1. CCITT X 208 per le Abstract Sintax Notation One(ASN.1)

2. CCITT X 209 per le Basic Encoding Rules (BER) della sintassi ASN.1

3. RSA Laboratories Technical Notes : A Layman's Guide to a Subset of ASN.1 , BER and DER (Distinguished Encoding Rules)

4. CCITT X509 versione 3 per il formato dei Certificati Digitali, le estensioni e le policy;

5. 2 FIPS 180-1 per la funzione di Hash SHA-1;

6. FIPS 46 per il Data Encryption Standard;

7. RSA 78 Rivest,Shamir,Aldeman. A method for obtaining digital signatures and public key cryptosystems.

8. PKCS#1 per il formato dei dati da sottoporre ad autenticazione;

9. PKCS#7 per la sintassi dei dati da sottoporre ad autenticazione;

10. PKCS#9 per i "selected attribute type " da utilizzare nella sintassi PKCS#7 e PKCS#10;

11. PKCS#10 per la sintassi delle richieste di certificazione di chiavi pubbliche