



CryptoAPI & CAPICOM

di Arcieri Raffaele, Carbone Dora, Gatto Angelica, Parrella Silvana, Vigorito Vincenzo

13/06/2002



Architettura del sistema di crittografia di Windows

CryptoAPI

- Chiavi crittografiche
- Codifica e decodifica dei dati
- Cifratura e decifratura dei dati
- Hash e firme digitali
- Certificati digitali
- Esempi
- Microsoft CAPICOM
- Operazioni di firma e verifica
- Cifratura
- I certificati
- Esempio di cifratura

13/06/2002

CryptoAPI e CAPICOM

2



Architettura del sistema di crittografia di Windows

L'architettura del sistema di crittografia di Windows comprende tre elementi fondamentali:

- Il client (applicazione).
- Il sistema operativo.
- I Cryptographic Service Provider (CSP).



13/06/2002

CryptoAPI e CAPICOM

3



Cryptographic Service Provider

I CSP sono dei moduli indipendenti che eseguono tutte le operazioni di crittografia, come

- la generazione di chiavi di sessione (chiavi simmetriche),
- la generazione di chiavi asimmetriche (pubblica/privata) e
- le operazioni di firma.

Fisicamente i CSP sono composti almeno:

- da un file con estensione .DLL che consente le operazioni di crittografia, e
- da una firma elettronica che Windows utilizza per verificare l'integrità del sistema.

13/06/2002

CryptoAPI e CAPICOM

4



Cryptographic Service Provider

Ogni CSP ha un database delle chiavi chiamato key database.



I client possono connettersi ai contenitori tramite **CryptAcquireContext**.

13/06/2002

CryptoAPI e CAPICOM

5



CSP forniti da Microsoft

- Microsoft Base Cryptographic Provider.
- Microsoft Enhanced Cryptographic Provider.
- Microsoft DSS Cryptographic Provider.
- Microsoft Base DSS and Diffie-Hellman Cryptographic Provider.
- Schannel Cryptographic Provider.

Cifrario	Base CSP	Enhanced CSP	
RSA public key signature algorithm	Key length: 512 bit	Key length: 1024 bit	
RSA public key exchange algorithm	Key length: 512 bit	Key length: 1024 bit	
RC2 block encryption algorithm	Key length: 40 bit	Key length: 128 bit	Salt length: settable
RC4 stream encryption algorithm	Key length: 40 bit	Key length: 128 bit	Salt length: settable
DES	Not supported	Key length: 56 bit	
DES triple (due chiavi)	Not supported	Key length: 112 bit	
DES triple (tre chiavi)	Not supported	Key length: 168 bit	

13/06/2002

CryptoAPI e CAPICOM

6



CSPI

Ogni funzione nella tabella corrisponde in maniera diretta ad una funzione della CryptoAPI col prefisso CP invece di Crypt.

Funzione	Significato
CPAcquireContext	Acquisisce un handle ad un particolare contenitore di chiavi all'interno del CSP.
CPCreateHash	Crea un oggetto hash e restituisce un handle ad esso.
CPDevert	Decifra una sezione di testo cifrato usando la chiave di cifratura specificata.
CPDeriveKey	Crea una chiave da una password.
CPDestroyHash	Distrugge un oggetto hash.
CPDestroyKey	Distrugge una chiave.
CPDuplicateHash	Crea una copia esatta di un oggetto hash con all'interno lo stato dell'hash.
CPDuplicateKey	Crea una copia esatta di una chiave, incluso lo stato della chiave.
CPEnvert	Cifra una sezione di testo in chiaro usando la chiave di cifratura specificata.
CPExportKey	Trasferisce una chiave dal CSP ad un BLOB della chiave nella memoria dell'applicazione.
CPGenKey	Crea una chiave casuale.

13/06/2002

CryptoAPI e CAPICOM

7



CSPI

Funzione	Significato
CPGenRandom	Genera dati casuali.
CPGetHashParam	Restituisce i parametri di un oggetto hash.
CPGetKeyParam	Restituisce i parametri di una chiave.
CPGetProvParam	Restituisce gli attributi del CSP.
CPGetUserKey	Ottiene un handle alla chiave di scambio od alla chiave firma.
CPHashData	Crea un hash di un blocco di dati e gli aggiunge l'oggetto hash specificato.
CPHashSessionKey	Crea un hash di una chiave di sessione e le aggiunge l'oggetto hash specificato.
CPImportKey	Trasferisce una chiave da un BLOB della chiave ad un CSP.
CPReleaseContext	Rilascia l'handle acquisito con CPAcquireContext.
CPSetHashParam	Stipa i parametri di un oggetto hash.
CPSetKeyParam	Specifica i parametri di una chiave.
CPSetProvParam	Stipa gli attributi specifici di un CSP.
CPSignHash	Firma l'oggetto hash specificato.
CPVerifySignature	Verifica una firma digitale.

13/06/2002

CryptoAPI e CAPICOM

8



Architettura del sistema di crittografia di Windows

CryptoAPI

- Chiavi crittografiche
 - Codifica e decodifica dei dati
 - Cifratura e decifratura dei dati
 - Hash e firme digitali
 - Certificati digitali
 - Esempi
- Microsoft CAPICOM
- Operazioni di firma e verifica
 - Cifratura
 - I certificati
 - Esempio di cifratura

13/06/2002

CryptoAPI e CAPICOM

9



CryptoAPI

La **CryptoAPI** è un'interfaccia di programmazione che astrae le applicazioni crittografiche dalla comunicazione diretta con il sistema operativo.

Inoltre

- Fornisce servizi che consentono agli sviluppatori di applicazioni di aggiungere sicurezza basata sulla crittografia alle applicazioni.
- Include funzionalità
 - per codificare e decodificare da ASN.1,
 - per l'hashing,
 - per la cifratura e la decifratura dei dati,
 - per l'autenticazione usando i certificati digitali e
 - per la gestione dei certificati nei propri registri.

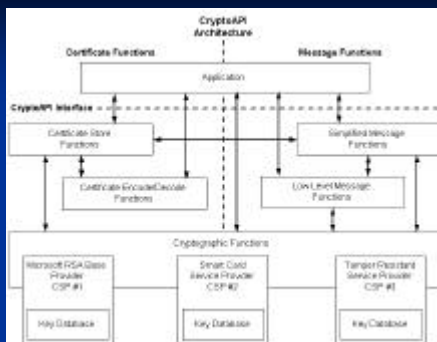
13/06/2002

CryptoAPI e CAPICOM

10



Architettura della CryptoAPI



13/06/2002

CryptoAPI e CAPICOM

11



Architettura del sistema di crittografia di Windows

CryptoAPI

- Chiavi crittografiche
 - Codifica e decodifica dei dati
 - Cifratura e decifratura dei dati
 - Hash e firme digitali
 - Certificati digitali
 - Esempi
- Microsoft CAPICOM
- Operazioni di firma e verifica
 - Cifratura
 - I certificati
 - Esempio di cifratura

13/06/2002

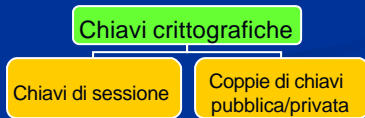
CryptoAPI e CAPICOM

12



Chiavi crittografiche

Le **chiavi crittografiche** sono importanti per le operazioni di crittografia. Esse devono essere mantenute segrete, perché chiunque possieda una data chiave ha accesso a qualunque dato a cui essa è associata.



13/06/2002

CryptoAPI e CAPICOM

13



Chiavi di sessione

Le **chiavi di sessione**

- vengono usate con gli **algoritmi di cifratura simmetrica**;
- usualmente, vengono cambiate per ogni messaggio cifrato;
- vengono create usando o la funzione **CryptGenKey** o **CryptDeriveKey**;
- sono mantenute interne al CSP.

Gli algoritmi simmetrici sono più veloci degli algoritmi a chiave pubblica.

A differenza delle coppie di chiavi pubblica/privata, le chiavi di sessione sono volatili.

13/06/2002

CryptoAPI e CAPICOM

14



Coppie di chiavi pubblica/privata

Le **coppie di chiavi pubblica/privata** vengono usate in un procedimento di cifratura più sicuro chiamato **cifratura asimmetrica**.

La cifratura asimmetrica

- viene usata principalmente per cifrare e decifrare le chiavi di sessione e le firme digitali;
- usa gli **algoritmi di cifratura chiave pubblica**.

Ogni utente, generalmente, ha due coppie di chiavi pubblica/privata:

- **exchange key pair**. Viene usata per cifrare le chiavi di sessione.
- **signature key pair**. Viene usata per creare la firma digitale.

13/06/2002

CryptoAPI e CAPICOM

15



Memorizzazione e scambio di chiavi crittografiche

Quando le chiavi devono essere esportate dal CSP per essere inserite nello spazio dati di un'applicazione, vengono memorizzate in strutture BLOB cifrate.

Ci sono due situazioni specifiche in cui è necessario esportare le chiavi:

- Per salvare una chiave di sessione per un utilizzo successivo da parte di un'applicazione.
- Per mandare una chiave a qualcuno.

In entrambi i casi, un'applicazione deve memorizzare una chiave di sessione al di fuori del CSP per un certo periodo di tempo.



13/06/2002

CryptoAPI e CAPICOM

16



Key database

Ogni CSP ha un **key database** nel quale memorizza le chiavi crittografiche.



Generalmente, viene creato un contenitore di default per ogni utente.

È anche possibile per un'applicazione creare un suo contenitore delle chiavi e le sue coppie di chiavi.

13/06/2002

CryptoAPI e CAPICOM

17



BLOB della chiave

Le chiavi sono memorizzate all'interno del CSP e le applicazioni hanno permesso di accesso alla chiave solo tramite un **handle**.

BLOB della chiave

- sono l'unica eccezione a questa regola;
- consistono di un'intestazione standard, seguita dalla chiave cifrata;
- sono cifrati con la exchange public key del destinatario designato.
- vengono a volte firmati con la exchange private key dell'utente originario, per renderli a prova di intrusione.

Se il BLOB della chiave contiene una chiave di sessione, questi dati sono sempre tenuti cifrati.

13/06/2002

CryptoAPI e CAPICOM

18



Alternative alla memorizzazione delle chiavi di sessione

Invece di memorizzare una chiave di sessione casuale, può essere usata una **chiave derivata**, creata da una password usando la funzione **CryptDeriveKey**.

Invece di memorizzare una particolare chiave derivata, un'applicazione può creare una chiave derivata chiedendo la password all'utente.



Alternative alla memorizzazione delle chiavi di sessione

La stabilità dei BLOB della chiave dipende da quella delle coppie e di chiavi pubblica/privata.

Se queste coppie di chiavi vengono, per qualche motivo, perse, i BLOB della chiave non possono essere decifrati.

Per prevenire questo tipo di perdite, si deve usare una backup authority.



Alternative alla memorizzazione delle chiavi di sessione

Una **backup authority** è un'applicazione fidata in esecuzione su un computer sicuro che fornisce memoria per le chiavi di sessione dei suoi client.

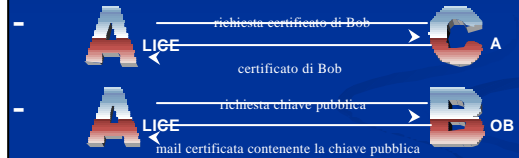
Tutte le chiavi di sessione memorizzate vengono cifrate sotto forma di BLOB della chiave usando la chiave pubblica della backup authority.



Scambio di coppie di chiavi pubbliche

Due utenti effettuanti una comunicazione cifrata hanno bisogno di scambiarsi le chiavi pubbliche.

Ci sono due modi principali per ottenere la chiave pubblica dell'altro:



Questo metodo può anche essere usato per validare i valori della chiave pubblica che è stata scambiata in altri modi.



Certificati e certification authority

Un **certificato** è un pacchetto di dati che contiene la chiave pubblica di un utente ed i dati che lo identificano nel mondo reale, come il nome dell'utente.

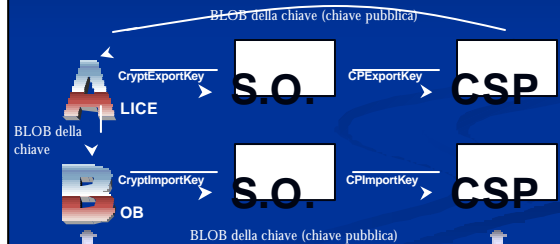
Ogni certificato viene creato e firmato da un'entità fidata conosciuta come **certification authority (CA)**.

Visto che i certificati vengono firmati da una CA, per maggiore sicurezza, si deve verificare che la firma di ogni certificato utilizzi la chiave pubblica della CA.



Scambio manuale delle coppie di chiavi pubbliche

Se non è disponibile una CA, oppure se gli utenti non hanno registrato le loro chiavi pubbliche in una CA, questi devono scambiarsi le chiavi pubbliche in altro modo (**scambio manuale**).





Scambio di chiavi di sessione

Per inviare ad un altro utente un messaggio cifrato, la chiave di sessione usata per eseguire la cifratura deve essere mandata all'utente che dovrà decifrare il messaggio.

Ci sono due modi d'applicarlo:

- L'utente trasmittente può creare una chiave di sessione casuale, cifrata, e la manda, sotto forma di BLOB della chiave, al ricevitore.
- Gli utenti trasmittente e ricevente possono reciprocamente accordarsi su una chiave di sessione scambiandosi una serie di messaggi avanti e indietro.

13/06/2002

CryptoAPI e CAPICOM

25



Architettura del sistema di crittografia di Windows

CryptoAPI

Chiavi crittografiche

Codifica e decodifica dei dati

Cifratura e decifratura dei dati

Hash e firme digitali

Certificati digitali

Esempi

Microsoft CAPICOM

Operazioni di firma e verifica

Cifratura

I certificati

Esempio di cifratura

13/06/2002

CryptoAPI e CAPICOM

26

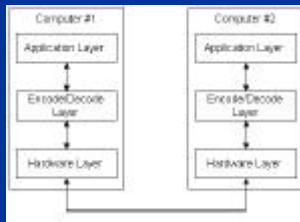


Codifica e decodifica dei dati

Per inviare dati su un mezzo di comunicazione, questi devono essere convertiti in una stringa di 0 e 1 che possa essere trasmessa serialmente sulla linea.

La serializzazione deve essere fatta in un modo che il computer che riceve i dati possa riconvertirli nel loro formato originale.

Il modo in cui viene effettuata la serializzazione è chiamato protocollo di comunicazione.



13/06/2002

27



ASN.1

Un principio di progettazione software accettato è l'uso di **astrazione**

Usando l'astrazione, un progettista può specificare un oggetto software che ha qualità specifiche senza entrare nei dettagli dell'implementazione.

La maggior parte dei protocolli di comunicazione contiene astrazione (gli oggetti agli strati più alti sono implementati usando quelli di strati più bassi).

13/06/2002

CryptoAPI e CAPICOM

28



ASN.1

Abstract Syntax Notation One (ASN.1): un metodo comune (definito in **CCITT X.208**) di specificare oggetti astratti.

Distinguished Encoding Rules (DER): un insieme di regole (definito in **CCITT X.509**) di ASN.1 per rappresentare tali oggetti come stringhe binarie.

La CryptoAPI richiede che le strutture C usate per codificare e decodificare i dati rispecchino i tipi di dati ASN.1.

13/06/2002

CryptoAPI e CAPICOM

29



ASN.1

Un valore di un tipo di dato ASN.1 è un elemento dell'insieme del tipo definito.

ASN.1 ha quattro specie di tipi:

- **Tipi semplici.**
- **Tipi strutturati.**
- **Tipi aggiuntivi (derivati).**
- **Altri tipi.**

Tipi e valori possono essere assegnati con l'operatore d'assegnamento ($::=$), e quei nomi possono essere usati nella definizione di altri tipi e valori.

13/06/2002

CryptoAPI e CAPICOM

30

Codifica e decodifica con la CryptoAPI

General Decode Model

General Encode Model

Fra gli item che le funzioni **CryptEncodeObject** e **CryptDecodeObject** possono codificare e decodificare ci sono i certificati, le estensioni dei certificati, le richieste di certificato.

13/06/2002 CryptoAPI e CAPICOM 31

Architettura del sistema di crittografia di Windows CryptoAPI

- Chiavi crittografiche
- Codifica e decodifica dei dati
- Cifratura e decifratura dei dati
- Hash e firme digitali
- Certificati digitali
- Esempi
- Microsoft CAPICOM
- Operazioni di firma e verifica
- Cifratura
- I certificati
- Esempio di cifratura

13/06/2002 CryptoAPI e CAPICOM 32

Cifratura e decifratura dei dati

Gli attuali algoritmi di cifratura usati nel processo di cifratura/decifratura dipendono dal CSP utilizzato.

Per tutte le cifrature effettuate usando le funzioni della CryptoAPI, viene usato un algoritmo simmetrico, indifferentemente da quale CSP è installato.

Le applicazioni che invocano le funzioni della CryptoAPI non hanno bisogno di conoscere i dettagli dell'attuale algoritmo usato.

Cifrario	Tipo di cifrario	Key setup time (µs)	Velocità di cifratura (byte/s)
DES	cifrario a blocchi da 64 bit	460	1138519
RC2	cifrario a blocchi da 64 bit	40	286888
RC4	stream cipher	151	2511725

Questi dati sono generati da un'applicazione che utilizza la CryptoAPI su un Pentium a 120 MHz.

13/06/2002 CryptoAPI e CAPICOM 33

Cifratura di file e messaggi usando la CryptoAPI

Per cifrare un file in modo che solo l'utente corrente possa accedere ai dati, viene utilizzato un cifrario simmetrico.

L'applicazione dovrà memorizzare i seguenti dati:

- I dati cifrati ("imbottiti" se si usa un cifrario a blocchi).
- Uno o più BLOB della chiave, ognuno contenente la chiave di sessione utilizzata per cifrare il messaggio.
- Tutti i valori aggiuntivi che sono stati specificati nel momento in cui i dati sono stati cifrati.
- Tutti i vettori di inizializzazione che sono stati specificati nel momento in cui i dati sono stati cifrati.

13/06/2002 CryptoAPI e CAPICOM 34

Decifratura di file e messaggi usando la CryptoAPI

CryptGenKey crea una chiave di sessione casuale nella fase di cifratura.

Prima di decifrare il messaggio, il BLOB contenente la chiave di sessione deve essere inserito nel CSP tramite la funzione **CryptImportKey**.

Se il messaggio è stato cifrato in modo che ogni possessore di password possa accedere ai dati:

- la funzione **CryptImportKey** non viene usata;
- si crea (dalla password) una chiave di sessione di decifratura tramite la funzione **CryptDeriveKey**.

13/06/2002 CryptoAPI e CAPICOM 35

Decifratura di file e messaggi usando la CryptoAPI

CryptSetKeyParam configura tutti i parametri della chiave di sessione come sono stati configurati durante la cifratura.

CryptDecrypt decifra il messaggio (se il messaggio è ampio più chiamate a funzione).

CryptDestroyKey distrugge la chiave di sessione e libera le risorse CSP.

13/06/2002 CryptoAPI e CAPICOM 36

Cifratura e decifratura simultanee usando la CryptoAPI

Posso cifrare e decifrare due flussi di dati simultaneamente con la stessa chiave crittografica?

La stessa chiave di sessione non deve essere usata per entrambe le operazioni visto che la chiave di sessione contiene informazioni sullo stato interno, che verrebbero mescolate.

Posso creare una copia della chiave di sessione!!!

13/06/2002 CryptoAPI e CAPICOM 37

Cifratura e decifratura simultanee usando la CryptoAPI

```

HCRYPTPROV hProv;
HCRYPTKEY hKey;
HCRYPTKEY hCopyKey = 0;
HCRYPTKEY hPubKey = 0;
BYTE pbBlob[256];
DWORD dwBlobLen;

CryptGetUserKey(hProv, AT_KEYEXCHANGE, &hPubKey);

dwBlobLen = 256;
CryptExportKey(hKey, hPubKey, SIMPLEBLOB, 0, pbBlob, &dwBlobLen);

CryptImportKey(hProv, pbBlob, dwBlobLen, 0, 0, &hCopyKey);
...

```

Obtiene un handle ad un CSP

Obtiene un handle ad una chiave di sessione

Ottiene un handle alla exchange public key dell'utente corrente

Esporta la chiave di sessione in un BLOB della chiave

Importa la chiave di sessione nel CSP.

13/06/2002 CryptoAPI e CAPICOM 38

Architettura del sistema di crittografia di Windows

CryptoAPI

- Chiavi crittografiche
- Codifica e decodifica dei dati
- Cifratura e decifratura dei dati
- Hash e firme digitali
- Certificati digitali
- Esempi

Microsoft CAPICOM

- Operazioni di firma e verifica
- Cifratura
- I certificati
- Esempio di cifratura

13/06/2002 CryptoAPI e CAPICOM 39

Hash e firme digitali

Con le funzioni di hashing e di firma digitale, un utente può firmare i dati digitalmente in modo che ogni altro utente possa verificare che i dati non sono stati modificati una volta firmati.

Una **firma digitale** consiste di un piccolo quantitativo di dati binari, tipicamente meno di 256 byte.

Questa firma può essere impacchettata con il messaggio firmato o memorizzata separatamente.

Il Microsoft Base Cryptographic Provider crea firme digitali conformi al **RSA Public-Key Cryptography Standard (PKCS)**.

13/06/2002 CryptoAPI e CAPICOM 40

Hash di dati

Le funzioni della CryptoAPI creano un hash di un testo, e lo utilizzano come identificatore unico dei dati.

Per ottenere un valore hash, si deve:

1. Creare un oggetto hash usando **CryptCreateHash**.
2. Aggiungere i dati all'oggetto hash tramite la funzione **CryptHashData**.
3. Ottenere il valore hash dei dati, dopo che l'ultimo blocco di dati è stato aggiunto all'hash, tramite la funzione **CryptGetHashParam**.
4. Distruggere l'oggetto hash con **CryptDestroyHash** non appena il valore hash è stato ottenuto, per una sicurezza migliore.

13/06/2002 CryptoAPI e CAPICOM 41

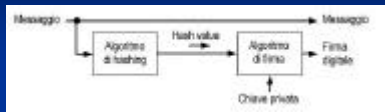
Firme digitali

13/06/2002 CryptoAPI e CAPICOM 42

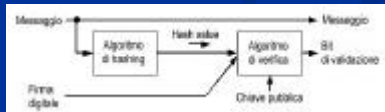


Firme digitali

Ci sono due passi coinvolti nella creazione di una firma digitale e di un messaggio.



Per verificare una firma, sono richiesti sia il messaggio che la firma.



13/06/2002

CryptoAPI e CAPICOM

43



Firme di messaggi e verifica della firma

Per applicare una firma digitale a dei dati:



La CryptoAPI astrae il metodo di firma, in modo che gli sviluppatori di applicazioni non hanno bisogno di sapere i dettagli del meccanismo di firma.

13/06/2002

CryptoAPI e CAPICOM

44



Messaggi firmati

Per firmare un messaggio od ogni altro dato, si deve:

- creare un oggetto hash usando **CryptCreateHash** ;
- usare la funzione **CryptSignHash** per firmare il hash;
- distruggere l'oggetto hash con **CryptDestroyHash** , dopo aver ottenuto la firma digitale

Gli hash possono essere firmati o con la **signature private key** o con la **exchange private key**.

Un singolo messaggio può essere firmato da più di un firmatario.

Uno o più firmatari possono fornire un hash al messaggio originale e cifrarlo.

13/06/2002

CryptoAPI e CAPICOM

45



Verifica di una firma

Per verificare una firma, si deve:

- creare un oggetto hash usando **CryptCreateHash** ;
- usare **CryptVerifySignature** per verificare la firma.
- distruggere l'oggetto hash tramite la funzione **CryptDestroyHash** .

13/06/2002

CryptoAPI e CAPICOM

46



Architettura del sistema di crittografia di Windows CryptoAPI

Chiavi crittografiche
Codifica e decodifica dei dati
Cifratura e decifratura dei dati
Hash e firme digitali

Certificati digitali

Esempi

Microsoft CAPICOM

Operazioni di firma e verifica

Cifratura

I certificati

Esempio di cifratura

13/06/2002

CryptoAPI e CAPICOM

47



Certificati digitali

Gli utenti devono essere in grado di provare la loro identità a coloro con i quali essi comunicano e devono essere in grado di verificare l'identità degli altri.

Il Certificato digitale

- è un mezzo di verifica dell'identità.
- è formato da:
 - un insieme di dati che identifica un'identità (soggetto del certificato).
 - almeno una chiave pubblica del soggetto.
 - firma di un'autorità che lega chiave e identità.

I certificati digitali includono anche estensioni e proprietà estese che forniscono informazioni aggiuntive sul soggetto del certificato.

13/06/2002

CryptoAPI e CAPICOM

48



Certificati digitali

Certification authority (CA): organizzazione fiduciaria che

- rilascia il certificato.
- firma il certificato garantendo il legame tra chiave e soggetto del certificato.
- rilascia un certificato solo dopo aver verificato l'identità di un soggetto del certificato.

Su una rete, c'è spesso un'applicazione fidata conosciuta come **certificate server**, gestita da una CA.

Questa applicazione ha accesso alla chiave pubblica di tutti i suoi clienti e quindi è in grado di distribuire certificati.

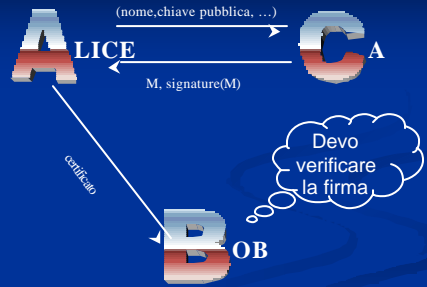
13/06/2002

CryptoAPI e CAPICOM

49



Certificazione digitale X.509



13/06/2002

CryptoAPI e CAPICOM

50



Certificazione digitale X.509

I certificati digitali **X.509** includono le seguenti informazioni:

Campo	Descrizione
Version	Numero di versione del certificato
Serial number	Numero seriale del certificato
Algorithm identifier	Algoritmo di firma usato dal firmatario del certificato
Issuer name	Nome dell'emittente del certificato
Not before	Data prima della quale il certificato non è valido
Not after	Data dopo la quale il certificato non è valido
Subject name	Nome della persona o dell'entità alla quale il certificato è stato emesso
Algorithm	Algoritmo usato per la chiave pubblica
Subject public key	Chiave pubblica attuale
Issuer unique ID	Campo opzionale. Se presente, la versione è la seconda
Subject unique ID	Campo opzionale. Se presente, la versione è la seconda
Extensions	Campo opzionale. Contiene dati aggiuntivi che un emittente può voler aggiungere al certificato, come l'indirizzo e-mail o l'autorizzazione ad emettere certificati. Se presente, la versione è la terza.

13/06/2002

CryptoAPI e CAPICOM

51



Certificati e CryptoAPI

Un certificato può scadere e non sarà più valido.

Una CA può revocare un certificato per molte ragioni.

Per manipolare le revoche, una CA mantiene e distribuisce una lista di certificati revocati, chiamata **certificate revocation list (CRL)**.

La CryptoAPI ha implementato una metodologia per permettere di verificare automaticamente i certificati tramite una lista predefinita di certificati fidati (CTL).

La CryptoAPI:

- supporta la codifica e la decodifica dei certificati,
- supporta la struttura dei certificati secondo lo standard X.509.

13/06/2002

CryptoAPI e CAPICOM

52



Contesti dei certificati

Il **contesto di un certificato**, **CERT_CONTEXT**, è una struttura C che contiene

- un membro codificato,
- un handle ad un **archivio dei certificati**,
- un puntatore al BLOB contenente la chiave pubblica,
- un puntatore ad una struttura C **CERT_INFO**.

CERT_INFO	C Structure
DWORD dwVersion	
CRYPT_INTEGER_BLOB SerialNumber	
CRYPT_ALGORITHM_IDENTIFIER SignatureAlgorithm	
CERT_NAME_BLOB Issuer	
FILETIME NotBefore	
FILETIME NotAfter	
CERT_NAME_BLOB Subject	
CERT_PUBLIC_KEY_INFO SubjectPublicKeyInfo	
CRYPT_BIT_BLOB IssuerUniqueId	
CRYPT_BIT_BLOB SubjectUniqueId	
DWORD cExtensions	
PCERT_EXTENSION* rgExtensions	

13/06/2002

CryptoAPI e CAPICOM

53



Contesti dei certificati



13/06/2002

CryptoAPI e CAPICOM

54



Operazioni con i certificati

La CryptoAPI fornisce funzioni per lavorare con:

- i certificati.
- le **certificate revocation list** (CRL)
- le **certificate trust list** (CTL).

Queste includono funzioni per:

- convertire i tipi codificati in tipi contesto;
- duplicare gli oggetti;
- rilasciare questi oggetti.

13/06/2002

CryptoAPI e CAPICOM

55



Operazioni con i certificati

In particolare per convertire i tipi codificati in tipo contesto abbiamo:

- **CertCreateCertificateContext**
- **CertCreateCRLContext**
- **CertCreateCTLContext**

Per duplicare i certificati, le CRL ed i CTL sono:

- **CertDuplicateCertificateContext.**
- **CertDuplicateCRLContext.**
- **CertDuplicateCTLContext .**

Le funzioni della CryptoAPI che rilasciano i contesti sono:

- **CertFreeCertificateContext.**
- **CertFreeCRLContext.**
- **CertFreeCTLContext.**

13/06/2002

CryptoAPI e CAPICOM

56



Operazioni con i certificati

Le funzioni duplicanti:

- non allocano spazio aggiuntivo
- non copiano i dati da un contesto ad una nuova locazione di memoria.

Le funzioni della CryptoAPI per duplicare i certificati, le CRL ed i CTL incrementano il **reference counter** del contesto e restituiscono un puntatore al contesto.

Le funzioni della CryptoAPI che rilasciano i certificati, le CRL e le CTL decrementano il reference counter di un contesto.

Se il reference count raggiunge zero, la memoria allocata per il contesto viene rilasciata.

13/06/2002

CryptoAPI e CAPICOM

57



Proprietà estese dei certificati

Le proprietà predefinite, identificate dai **propertyID**, includono:

- **CERT_KEY_PROV_HANDLE_PROP_ID.**
- **CERT_KEY_PROV_INFO_PROP_ID.**
- **CERT_KEY_CONTEXT_PROP_ID.**
- **CERT_SHA1_HASH_PROP_ID.**
- **CERT_MD5_HASH_PROP_ID.**

I dati contenuti nel contesto sono a sola lettura.

Ai certificati della CryptoAPI (sulle piattaforme Microsoft) sono associate anche proprietà dinamiche estese che possono essere aggiunte e cambiate.

13/06/2002

CryptoAPI e CAPICOM

58



Certificati e messaggi

Le funzioni fondamentali di crittografia usano

- I certificati.
- Le funzioni a basso livello per i messaggi.
- Le funzioni semplificate per i messaggi.

Per mandare o ricevere messaggi che incorporano certificati, si devono usare le funzioni a basso livello per i messaggi.

13/06/2002

CryptoAPI e CAPICOM

59



Funzioni a basso livello

Le **funzioni a basso livello per i messaggi**

- codificano dati per la trasmissione.
- decodificano i dati che sono stati ricevuti.
- verificano le firme dei messaggi ricevuti.

L'utilizzo delle funzioni a basso livello per i messaggi

- richiede molte chiamate a funzione;
- implica lavoro aggiuntivo per effettuare delle chiamate ad altre funzioni.



13/06/2002

CryptoAPI e CAPICOM

60



Funzioni semplificate

È stato fornito un gruppo di funzioni ad alto livello per semplificare la manipolazione dei messaggi.

Queste funzioni sono chiamate **funzioni semplificate per i messaggi**.

I nomi di tutte le funzioni semplificate per i messaggi contengono la parola "Message".

Le funzioni semplificate per i messaggi sono ad 'alto livello'.

Esse contengono in una singola funzione molte funzioni crittografiche di base, funzioni a basso livello per i messaggi e funzioni per i certificati.



13/06/2002

CryptoAPI e CAPICOM

61



Gestione dei certificati con gli archivi dei certificati

In un certo periodo di tempo, i certificati si accumulano sul computer di un utente.

La CryptoAPI fornisce dei tool per gestire questi certificati:

- Archivio dei certificati: sul quale possiamo effettuare operazioni di store, cancellazione, elencazione e verifica dei certificati.

Oltre ai certificati, anche le CRL e le CTL possono essere mantenuti negli **archivi dei certificati** da dove possono essere recuperate, per essere usate nei processi di autenticazione.

13/06/2002

CryptoAPI e CAPICOM

62



Gestione dei certificati con gli archivi dei certificati



13/06/2002

63



Gestione dei certificati con gli archivi dei certificati

In un sistema solitamente abbiamo due store per i certificati:

- **Il MY store:** contenente i certificati dell'utente, usati per firmare e decifrare i messaggi dell'utente.
 - può stare in qualunque delle varie locazioni fisiche.
- **Il ROOT store:** contenente i certificati delle emittenti fidate.
 - persiste in una sottochiave del registro.

Nel contesto della CryptoAPI, il ROOT store viene protetto e dei dialog speciali dell'interfaccia utente ricordano a questi di porre, in quell'archivio, solo certificati fidati.

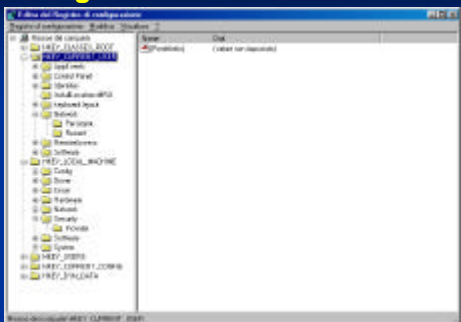
13/06/2002

CryptoAPI e CAPICOM

64



Gestione dei certificati con gli archivi dei certificati



13/06/2002

CryptoAPI e CAPICOM

65



Architettura del sistema di crittografia di Windows CryptoAPI

Chiavi crittografiche
Codifica e decodifica dei dati
Cifratura e decifratura dei dati
Hash e firme digitali
Certificati digitali

Esempi
Microsoft CAPICOM
Operazioni di firma e verifica
Cifratura
I certificati
Esempio di cifratura

13/06/2002

CryptoAPI e CAPICOM

66

Cifratura di un file (linguaggio C)

```

hSource = fopen(szSource, "rb");
hDestination = fopen(szDestination, "wb");

CryptAcquireContext(&hCryptProv, NULL, MS_ENHANCED_PROV, PROV_RSA_FULL, 0);

if (!szPassword)
{
    CryptGenKey(hCryptProv, ENCRYPT_ALGORITHM, KEYLENGTH | CRYPT_EXPORTABLE, &hKey);
    CryptGetUserKey(hCryptProv, AT_KEYEXCHANGE, &hXchgKey);
    CryptExportKey(hKey, hXchgKey, SIMPLEBLOB, 0, NULL, &dwKeyBlobLen);
    pbKeyBlob = (BYTE *)malloc(dwKeyBlobLen);
    CryptExportKey(hKey, hXchgKey, SIMPLEBLOB, 0, pbKeyBlob, &dwKeyBlobLen);
    CryptDestroyKey(hXchgKey);
    hXchgKey = 0;
    fwrite(&dwKeyBlobLen, sizeof(DWORD), 1, hDestination);
    fwrite(pbKeyBlob, 1, dwKeyBlobLen, hDestination);
}

```

Ottiene un handle al CSP

Crea una chiave di sessione casuale

Ottiene un handle alla exchange public key del decifratore

Determina la dimensione del BLOB della chiave ed alloca la memoria

Cifra ed esporta la chiave di sessione in un BLOB della chiave semplice

Rilascia l'handle alla exchange public key

13/06/2002 CryptoAPI e CAPICOM 67

Cifratura di un file

```

else
{
    CryptCreateHash(hCryptProv, CALG_MD5, 0, 0, &hHash);
    CryptHashData(hHash, (BYTE *)szPassword, strlen(szPassword), 0);
    CryptDeriveKey(hCryptProv, ENCRYPT_ALGORITHM, hHash, KEYLENGTH, &hKey);
    CryptDestroyHash(hHash);
    hHash = 0;
}

dwBlockLen = 1000 - 1000 % ENCRYPT_BLOCK_SIZE;
dwBufferLen = dwBlockLen + ENCRYPT_BLOCK_SIZE;
else
dwBufferLen = dwBlockLen;

pbBuffer = (BYTE *)malloc(dwBufferLen);

```

crea un oggetto hash

hashing della password

Deriva una chiave di sessione dall'oggetto hash

distruge l'oggetto hash

determina la dimensione del blocco

determina la dimensione del buffer

Alloca del memoria

13/06/2002 CryptoAPI e CAPICOM 68

Cifratura di un file

```

do
{
    dwCount = fread(pbBuffer, 1, dwBlockLen, hSource);
    CryptEncrypt(hKey, 0, feof(hSource), 0, pbBuffer, &dwCount, dwBufferLen);
    fwrite(pbBuffer, 1, dwCount, hDestination);
} while (!feof(hSource));
fclose(hSource);
fclose(hDestination);
free(pbBuffer);
CryptDestroyKey(hKey);
CryptReleaseContext(hCryptProv, 0);

```

Cifra i dati

Legge la lunghezza del BLOB della chiave dal file sorgente, e alloca memoria

Legge il BLOB dal file sorgente

Importazione del BLOB nel CSP

crea un oggetto hash

hashing della password

Deriva una chiave di sessione dall'oggetto hash

13/06/2002 CryptoAPI e CAPICOM 69

Decifratura di un file (linguaggio C)

```

hSource = fopen(szSource, "rb");
hDestination = fopen(szDestination, "wb");

CryptAcquireContext(&hCryptProv, NULL, MS_ENHANCED_PROV, PROV_RSA_FULL, 0);

if (!szPassword)
{
    fread(&dwKeyBlobLen, sizeof(DWORD), 1, hSource);
    pbKeyBlob = (BYTE *)malloc(dwKeyBlobLen);
    fread(pbKeyBlob, 1, dwKeyBlobLen, hSource);
    CryptImportKey(hCryptProv, pbKeyBlob, dwKeyBlobLen, 0, 0, &hKey);
}
else
{
    CryptCreateHash(hCryptProv, CALG_MD5, 0, 0, &hHash);
    CryptHashData(hHash, (BYTE *)szPassword, strlen(szPassword), 0);
    CryptDeriveKey(hCryptProv, ENCRYPT_ALGORITHM, hHash, KEYLENGTH, &hKey);
}

```

Ottiene un handle al CSP

Legge la lunghezza del BLOB della chiave dal file sorgente, e alloca memoria

Legge il BLOB dal file sorgente

Importazione del BLOB nel CSP

crea un oggetto hash

hashing della password

Deriva una chiave di sessione dall'oggetto hash

13/06/2002 CryptoAPI e CAPICOM 70

Decifratura di un file

```

CryptDestroyHash(hHash);
hHash = 0;

dwBlockLen = 1000 - 1000 % ENCRYPT_BLOCK_SIZE;
dwBufferLen = dwBlockLen;

pbBuffer = (BYTE *)malloc(dwBufferLen);

do
{
    dwCount = fread(pbBuffer, 1, dwBlockLen, hSource);
    CryptDecrypt(hKey, 0, feof(hSource), 0, pbBuffer, &dwCount);
    fwrite(pbBuffer, 1, dwCount, hDestination);
} while (!feof(hSource));

```

Distruge l'oggetto hash

Determina la dimensione del blocco

Determina la dimensione del buffer

Alloca del memoria

Legge dwBlockLen byte dal file sorgente

Decifra i dati

Chiude il file sorgente, chiude il file di destinazione, rilascia la memoria, distruge la chiave di sessione, rilascia l'handle al CSP

13/06/2002 CryptoAPI e CAPICOM 71

Architettura del sistema di crittografia di Windows CryptoAPI

- Chiavi crittografiche
- Codifica e decodifica dei dati
- Cifratura e decifratura dei dati
- Hash e firme digitali
- Certificati digitali
- Esempi
- Microsoft CAPICOM
- Operazioni di firma e verifica
- Cifratura
- I certificati
- Esempi di cifratura

13/06/2002 CryptoAPI e CAPICOM 72



Il metodo Sign

Il metodo *Sign* richiede tre parametri:

1. *Signer as Signer*

rappresenta il firmatario che deve avere accesso alla chiave privata relativa al certificato utilizzato;

2. *bDetached as Boolean*

- TRUE

- FALSE



3. *EncodingType*

rappresenta il tipo di codifica utilizzata per il buffer contenente la firma.

13/06/2002

CryptoAPI e CAPICOM

79



Operazione di verifica

La verifica avviene tramite un metodo esposto da **SignedData**

```
SignedData.Verify(SignedMessage as String,
                 bDetached as boolean,
                 VerifyFlag as CAPICOM_SIGNED_DATA_VERIFY_FLAG)
```

I primi due parametri sono equivalenti a quelli presenti in **SignedData.Sign** mentre l'ultimo indica la politica di controllo.

Il tipo di dato **CAPICOM_SIGNED_DATA_VERIFY_FLAG** può assumere due valori:

- **CAPICOM_VERIFY_SIGNATURE_ONLY**
- **CAPICOM_VERIFY_SIGNATURE_AND_CERTIFICATE**

13/06/2002

CryptoAPI e CAPICOM

80



Architettura del sistema di crittografia di Windows CryptoAPI

- Chiavi crittografiche
- Codifica e decodifica dei dati
- Cifratura e decifratura dei dati
- Hash e firme digitali
- Certificati digitali
- Esempi
- Microsoft CAPICOM
- Operazioni di firma e verifica
- Cifratura
- I certificati
- Esempio di cifratura

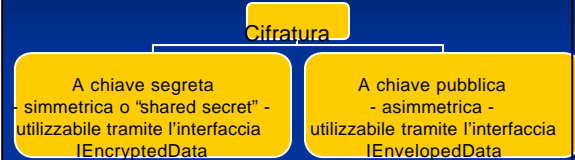
13/06/2002

CryptoAPI e CAPICOM

81



Cifratura



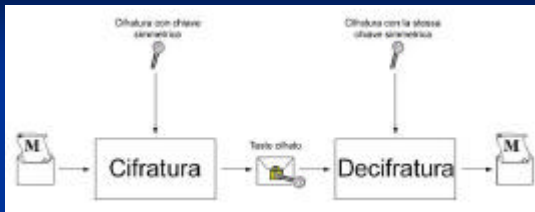
13/06/2002

CryptoAPI e CAPICOM

82



Cifratura a chiave simmetrica



Queste chiavi sono un numero casuale con dimensioni che possono variare da 40 a 2000 bit, spesso generate partendo da un hash.

13/06/2002

CryptoAPI e CAPICOM

83



CAPICOM e le chiavi di sessione

CAPICOM quando genera le chiavi di sessione tramite la cifratura a blocchi, le calcola in CBC mode (cipher block chaining) con un vettore di inizializzazione uguale a zero.

Questa è la modalità di default della funzione **CryptDeriveKey** utilizzata appunto per la generazione delle chiavi di sessione non casuali.

```
Dim oCifra As New EncryptedData
...
oCifra.SetSecret "password segreta"
oCifra.Content = "Questo testo viene cifrato"
oCifra.Algorithm.Name = CAPICOM_ENCRYPTION_ALGORITHM_3DES
Msgbox oCifra.Encrypt
```

13/06/2002

CryptoAPI e CAPICOM

84



Cifratura a chiave asimmetrica

Innanzitutto CAPICOM internamente genera una chiave di sessione con la quale cifra il documento.

La chiave di sessione viene cifrata con la chiave pubblica presente nel certificato del destinatario.

E IL DESTINATARIO???????????

Decifra la chiave di sessione con la propria chiave privata e successivamente può riottenere il documento in chiaro.

13/06/2002

CryptoAPI e CAPICOM

85



Interfaccia IEnvelopedData

Gli utenti CAPICOM hanno a disposizione l'interfaccia **IEnvelopedData**, la quale espone tre proprietà:

- **Algorithm**. Indica l'algoritmo di cifratura.
- **Content**. Indica il testo in chiaro.
- **Recipients**. Una collezione di destinatari identificati tramite il loro certificato.

La modalità di verifica dell'interfaccia IEnvelopedData avviene tramite il metodo **Decrypt** il quale verifica che nel MY storesia presente il certificato contenuto nel messaggio.



13/06/2002

CryptoAPI e CAPICOM

86



Architettura del sistema di crittografia di Windows CryptoAPI

- Chiavi crittografiche
- Codifica e decodifica dei dati
- Cifratura e decifratura dei dati
- Hash e firme digitali
- Certificati digitali
- Esempi
- Microsoft CAPICOM
- Operazioni di firma e verifica
- Cifratura
- I certificati
- Esempio di cifratura

13/06/2002

CryptoAPI e CAPICOM

87



I certificati

Sono documenti il cui scopo è quello di "certificare" l'identità di una persona, di un servizio o di un server.

Vengono rilasciati da una certification authority (CA).

L'identità viene preventivamente verificata tramite procedure predefinite da un altro servizio: la **registration authority**(RA).

IL CERTIFICATO E' PUBBLICO E DEVE ESSERE CONSULTATO DURANTE LE OPERAZIONI DI VERIFICA!!!!!!!!!!!!!!

13/06/2002

CryptoAPI e CAPICOM

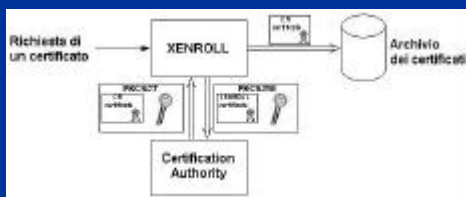
88



XENROLL

CAPICOM è in grado di gestire lo store dei certificati ed i certificati stessi, però non ha le funzionalità di registrazione di certificati negli store software e sulle smart card.

Se il programmatore ha necessità di coprire anche la parte di richiesta e generazione dei certificati deve utilizzare il componente COM **Certificate Enrollment Control XENROLL**.



13/06/2002

CryptoAPI e CAPICOM

89



Architettura del sistema di crittografia di Windows CryptoAPI

- Chiavi crittografiche
- Codifica e decodifica dei dati
- Cifratura e decifratura dei dati
- Hash e firme digitali
- Certificati digitali
- Esempi
- Microsoft CAPICOM
- Operazioni di firma e verifica
- Cifratura
- I certificati
- Esempio di cifratura

13/06/2002

CryptoAPI e CAPICOM

90

Cifrare un messaggio (Visual Basic)

```
Sub EncryptMessage(ByVal ToBeEncrypted As String,  
                  ByVal Hidden As String,  
                  ByVal FileName As String)  
  
    Dim Message As New EncryptedData  
    Message.Content = ToBeEncrypted  
    Message.SetSecret Hidden  
  
    Message.AlgorithmName = CAPICOM_ENCRYPTION_ALGORITHM_DES  
  
    Dim EncryptedMessage As String  
  
    EncryptedMessage = Message.Encrypt  
  
    If Len(EncryptedMessage) > 1 then  
        Open FileName for Output as #1  
        Write #1, EncryptedMessage  
        Close #1  
        MsgBox "Messaggio cifrato scritto nel file."  
    End If  
  
    Set Message = Nothing  
  
Exit Sub  
End Sub
```

- Stringa da cifrare
- Password da usare per la chiave

- Nome del file di output

Dichiarazione ed inizializzazione dell'oggetto EncryptedData

Cifatura del messaggio

Apertura del file di output e scrittura del messaggio cifrato nel file

Rilascia l'oggetto EncryptedData