La Firma Digitale

Giorgio De Lucia

Carlo Saccone

Daria Mazza

Firma Digitale

- Introduzione
- Crittografia Asimmetrica
- Normative
- Protocollo Diffie-Hellman
- L'algoritmo RSA

Giorgio De Lucia

Introduzione

- Internet è sempre più insicuro
- Necessità di una tecnologia che possa garantire un certo livello di sicurezza per i dati che vengono trasmessi e archiviati. Al fine di proteggere i dati diffusi in rete dai pericoli derivanti da un uso illecito delle informazioni.

- E' quindi indispensabile fornire contromisure di sicurezza mirate a garantire
  - Integrità
  - Autenticità
  - Confidenzialità
  - Non ripudiabilità

Introduzione

INTEGRITA' DEI DATI: è possibile garantire che un documento non sia stato alterato dall'istante in cui è stato creato. Quanto ricevuto corrisponde a quanto inviato

AUTENTICAZIONE FORTE: il ricevete e/o il mittente possono essere identificati con certezza

CONFIDENZIALITA': garantire la privacy di informazioni e dati memorizzati mediante tecniche crittografiche

NON RIPUDIABILITA': fornire un meccanismo per garantire che un utente che abbia firmato un documento non possa, in seguito, negare di averlo fatto

La Firma Digitale

- E' il risultato di una procedura informatica che garantisce l'autenticità e l'integrità di messaggi e documenti scambiati e archiviati con mezzi informatici

- E' la tecnologia con cui possono essere effettivamente soddisfatti tutti i requisiti richiesti per dare validità legale ad un documento elettronico firmato digitalmente

- E' al pari di quanto svolto dalla firma autografa per i documenti tradizionali

La Firma Digitale

FIRMA AUTOGRAFA

- Riconducibile al soggetto direttamente
- Facilmente falsificabile
- Legata al documento attraverso il supporto fisico
- Deve essere autenticata per impedire il ripudio

FIRMA DIGITALE

- Riconducibile al soggetto solo attraverso il possesso di un segreto
- Non falsificabile senza conoscere il segreto
- Legata al contenuto del documento
- Ripudio impossibile

Crittografia Asimmetrica

Il problema della firma digitale è fondamentale su reti come Internet, dove non è poi così difficile inviare messaggi sotto falso nome.

es. email,bonifici o acquisti in linea.

Principio Base:
Chiave di codifica ≠ Chiave di decodifica

E' possibile quindi distribuire la propria chiave di cifratura (*chiave pubblica*) e mantenere segreta la chiave di decifratura (*chiave privata*).
La coppia di chiavi correlate tra loro può essere utilizzata nell'ambito dei sistemi di validazione o di cifratura di documenti informatici.

Crittografia Asimmetrica

- Ogni documento può essere cifrato con una qualunque delle due chiavi e decifrato con l'altra,a seconda del tipo di servizio richiesto.

- Con la chiave pubblica si può cifrare un messaggio ed essere sicuri che solo il possessore della relativa chiave privata possa decifrarlo e leggerlo (***Confidenzialità***).

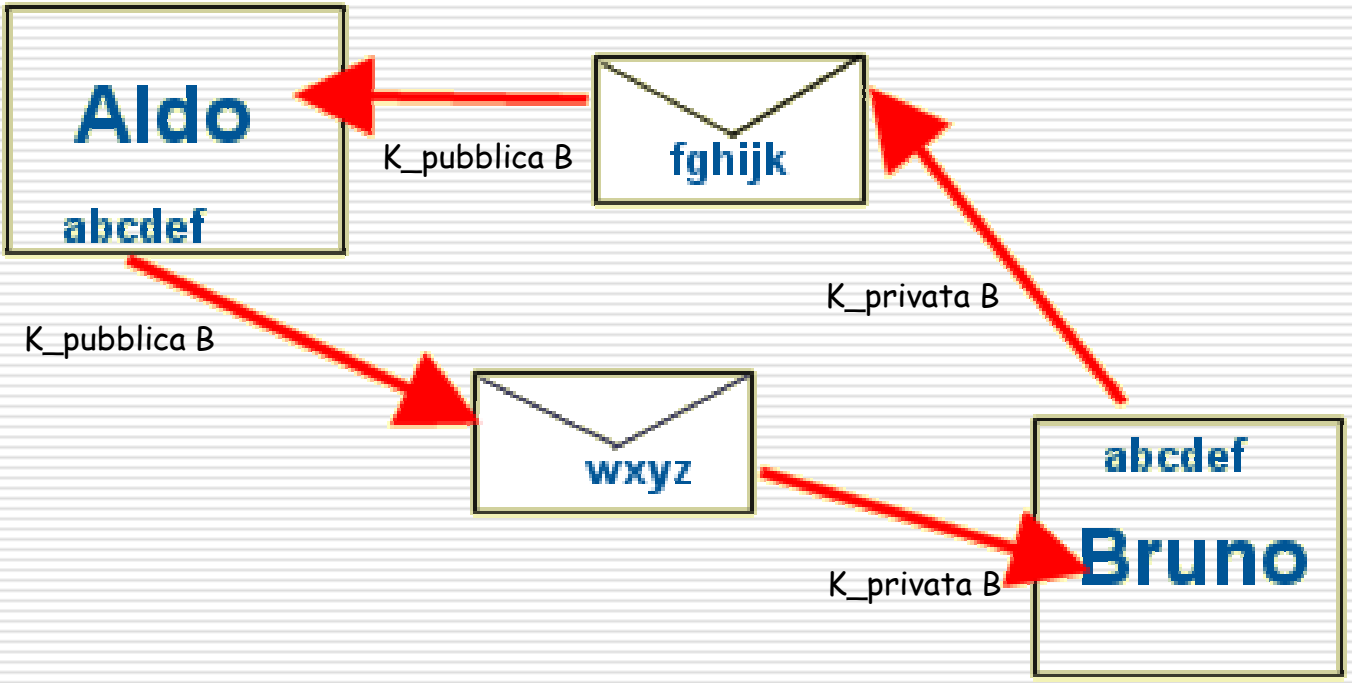
- Con la chiave privata si può cifrare un messaggio di modo che il ricevente possa sapere con certezza chi sia il mittente (***Autenticazione***)

Crittografia Asimmetrica

- Utilizzo della crittografia a chiave pubblica per realizzare un sistema di firma digitale sicuro.
  - Si potrebbe cifrare il messaggio da inviare con la propria chiave privata sfruttando il servizio di autenticazione offerto dalla crittografia asimmetrica.
    - *Sicuro ma molto lento.*

- Aldo invia una parola qualsiasi (**abcdef**) cifrata con la chiave pubblica di Bruno in **wxyz**, Bruno la decifra usando la propria chiave privata e riottiene **abcdef** e la rispedisce ad Aldo con il messaggio da inviare cifrandola con la sua chiave privata; Aldo la decifra usando la chiave pubblica di Bruno; Se la parola decifrata è identica a quella inviata Aldo è sicuro che il messaggio viene da Bruno.

Crittografia Asimmetrica



Crittografia Asimmetrica

Solo Bruno infatti poteva cifrare la parola inviata, in modo che Aldo potesse decifrarla.

Nemico:

- Non può decifrare la parola inviata da Aldo.
- Anche scoprendo la parola inviata da Aldo non può rinviarla spacciandosi per Bruno.
  - Non possiede la chiave privata di Bruno.

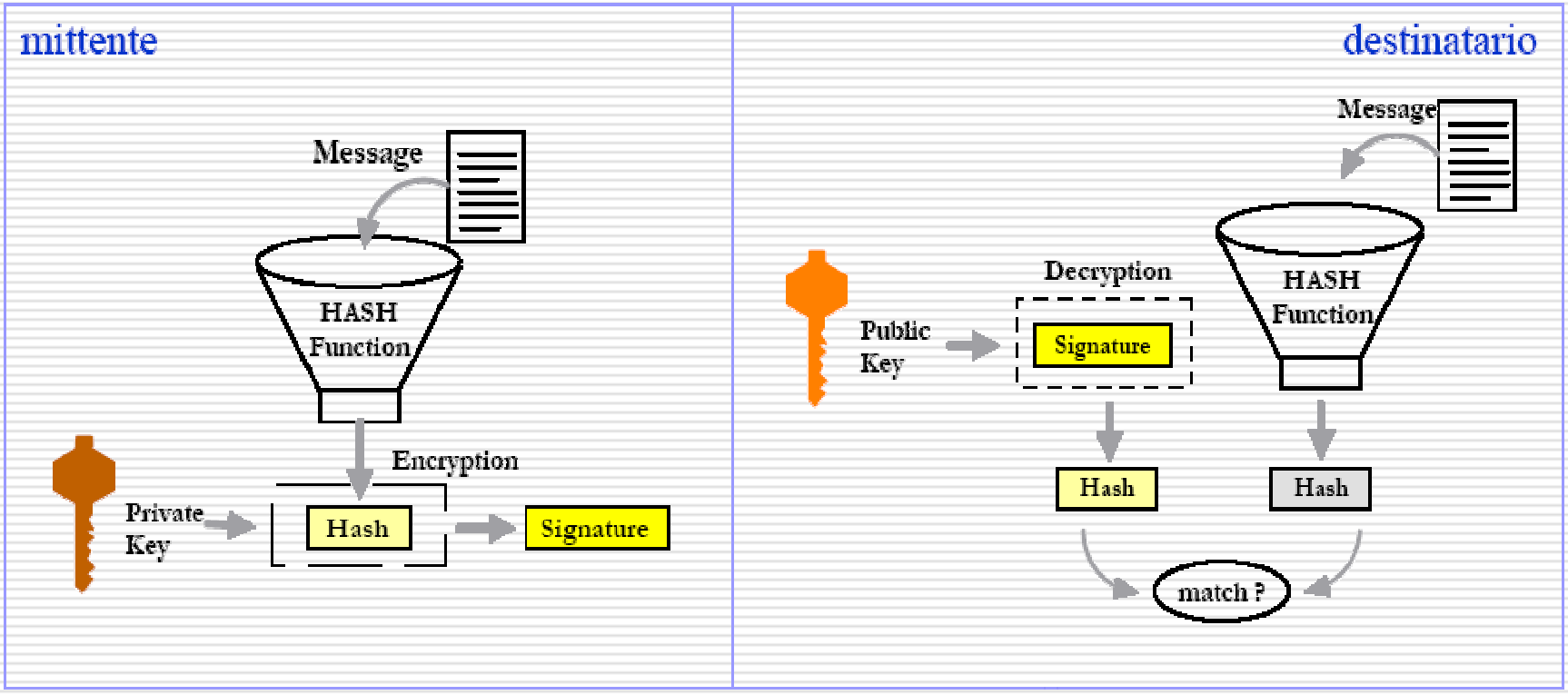
Problemi

- Questo ancora non va bene,ci serve qualcosa di più robusto ed efficiente
- Non ci garantisce che il messaggio arrivi integro a destinazione e che non venga sostituito

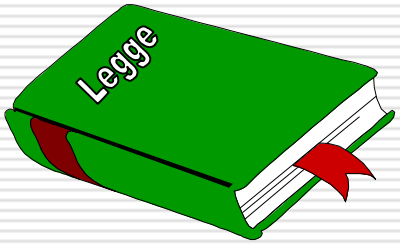
- Abbiamo bisogno di un qualcosa che sia legato da un lato al mittente e dall'altro all'informazione che questo vuole mandare

Firma Digitale

Visione d'Insieme



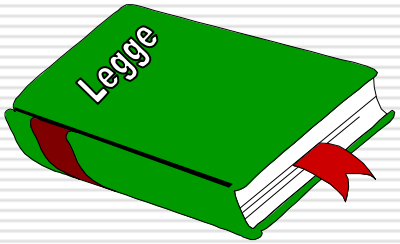
Quadro Normativo



Firma Digitale:

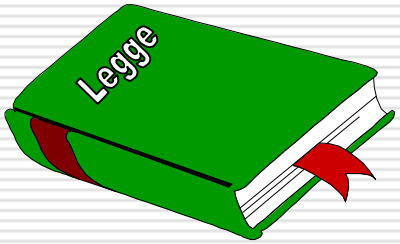
"... il risultato della procedura informatica (validazione) basata su un sistema di chiavi asimmetriche a coppia, una pubblica e una privata, che consente al sottoscrittore tramite la chiave privata e al destinatario tramite la chiave pubblica, rispettivamente, di rendere manifesta e di **verificare la provenienza e l'integrità di un documento informatico** o di un insieme di documenti informatici".

Quadro Normativo



- La firma digitale consente sempre di verificare anche la minima alterazione del documento
- Ha la stessa efficacia della firma autografa
- Rende il documento informatico "valido e rilevante a tutti gli effetti di legge"

Quadro Normativo



Algoritmi ammessi:

Art. 53. Norme transitorie

1. *In attesa della pubblicazione degli algoritmi per la generazione e verifica della firma digitale secondo quanto previsto dall'art. 3, i certificatori accreditati … , devono utilizzare l'algoritmo **RSA** (Rivest-Shamir-Adleman) con lunghezza delle chiavi non inferiore a 1024 bit.*
2. *In attesa della pubblicazione delle funzioni di hash secondo quanto previsto dall'art. 3, i certificatori accreditati … devono utilizzare uno dei seguenti algoritmi:*
   a) *dedicated hash-function 3 (**SHA-1**);*
   b) *dedicated hash-function 1 (**RIPEMD-160**).*

Definizioni Base

1. Una **firma digitale** è una stringa di dati che è associata ad un messaggio(in forma digitale).
2. Un **algoritmo per la generazione** della firma digitale è un metodo per produrre una firma digitale.
3. Un **algoritmo per la verifica** della firma digitale è un metodo per verificare che una firma digitale è autentica.
4. Uno **schema** di firma digitale consiste di un algoritmo per la generazione della firma e il relativo algoritmo di verifica.

Schema di Firma

DEFINIZIONE : Siano :

- D un insieme finito di possibili documenti
- F un insieme finito di possibili firme
- K un insieme finito di possibili chiavi

se $\forall$ k $\in$ K $\exists$: $\begin{cases} sig_k \rightarrow algoritmo\ di\ firma \\ ver_k \rightarrow algoritmo\ di\ verifica \end{cases}$

sig_k: D $\rightarrow$ F

ver_k: D×F $\rightarrow$ {vero, falso} tale che $\forall$ d $\in$ D, $\forall$ f $\in$ F :

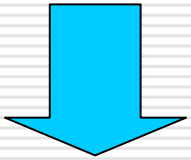
$ver_k(d,f)$ = { vero se f = $sig_k(d)$; falso se f ≠ $sig_k(d)$ }

allora

(D, F, K, sig_k, ver_k) costituisce uno **schema di firme**.

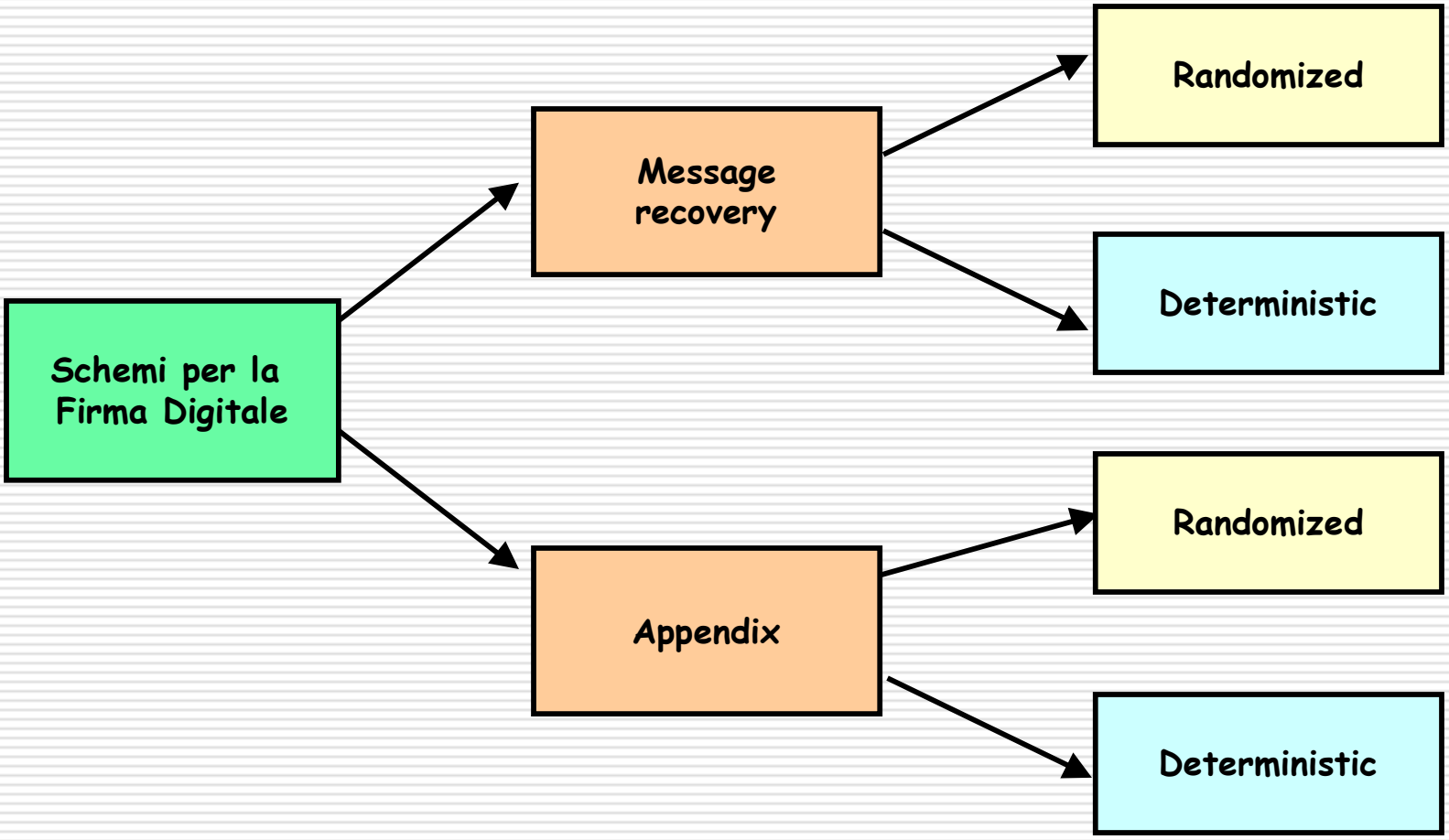
Schema di Firma

- DEFINIZIONE : Uno schema di firme è detto **incondizionatamente sicuro** se non esiste un modo per la falsificazione di una firma $f \in F$



- Non esistono schemi di firme incondizionatamente sicuri.
- Si può pensare di testare tutte le possibili firme $y \in F$ di un documento di un utente finché non si trova quella giusta
- Questo tipo di attacco alla sicurezza dello schema di firme risulta essere computazionalmente oneroso e praticamente irrealizzabile, poiché la cardinalità dell'insieme di possibili firme è molto elevata

Schema di Firma



Schema di Firma

- Due tipi di schemi per la firma digitale:
  - Message recovery
  - Appendix
- Altre due caratteristiche:
  - Deterministic
  - Randomized

Schema di Firma

- **Message Recovery**:
  - Le operazioni di verifica recuperano il messaggio dalla firma stessa.
  - Non richiesta la conoscenza a priori del messaggio dall'algoritmo di verifica.
  - Utilizzo di funzioni di hash.
  - Utilizzato soprattutto per messaggi brevi.
  - Esempio di algoritmi che realizzano firma digitale attraverso schemi con message recovery sono RSA, Rabin.

Schema di Firma

- **Appendix:**
  - E' lo schema di firma digitale più usato.
  - Schemi di firma digitale che richiedono il messaggio come input per l'algoritmo di verifica.
    - Devo trasmettere sia il messaggio originale che la firma
  - Si basa sull'uso di funzioni di hash
  - Maggiore resistenza agli attacchi.
  - Esempio di algoritmi che realizzano firma digitale attraverso schemi con appendix sono DSA, ElGamal e Schnorr

Protocollo Diffie-Hellman

Protocollo per lo scambio di una chiave segreta in maniera sicura sopra un canale insicuro.

Progettato per risolvere il problema dell'avvio di un normale sistema di cifratura a chiavi simmetriche,(es. il DES) in realtà ha posto le basi della crittografia a chiavi pubbliche.

La sua sicurezza si basa sulla complessità computazionale del calcolo del logaritmo discreto che è computazionalmente difficile.

Protocollo Diffie-Hellman

- In un'aritmetica finita di ordine **N** esistono in genere numeri **g** detti **generatori**: una base che elevata a potenza, genera tutti i numeri che sono primi con **N**.

Se **N** è primo esiste sicuramente un generatore **g** che in questo caso genera tutti i numeri compresi tra 1 e N-1.

DEF: La potenza di un numero in un'aritmetica finita è :

$$y = g^x \bmod n$$

DEF: Il logaritmo è *l'esponente che si deve dare alla base g per ottenere il valore y*:

$$x = \log_g y \bmod n$$

Tale logaritmo si dice *logaritmo discreto*.

Protocollo Diffie-Hellman

- Due interlocutori Aldo e Bruno vogliono scambiarsi una chiave segreta in modo sicuro su un canale insicuro.
- Scelgono e rendono pubblico un numero primo **N** molto elevato (es. con 1024 bit, circa 300 cifre decimali) e un *generatore* **g**.
- Aldo genera un numero casuale **a** < **N** e calcola $A = g^a \bmod N$. Il numero **A** viene comunicato pubblicamente a Bruno.
- Allo stesso modo Bruno genera il numero **b** e poi $B = g^b \bmod N$ e invia **B** ad Aldo.
- Aldo calcola il numero $k = B^a \bmod N$ mentre Bruno $k = A^b \bmod N$.

➔ Si noti che le due chiavi **k** sono uguali! Infatti entrambe valgono $g^{ab} \bmod N$.

- Entrambi possiedono la chiave K ma sopra il canale sono stati trasferiti solo **N**, **g**, **A** e **B** che non consentono di ricostruirla.

Protocollo Diffie-Hellman

ESEMPIO:

- Aldo e Bruno scelgono N = **17** e *g* = **3** (3 è generatore di 17)
- Aldo sceglie a = 6 e quindi A = 3^6 mod 17 = **15**.
- Bruno sceglie b = 11 e quindi B = 3^11 mod 17 = **7**.
- Aldo calcola k = **7**^6 mod 17 = **9**.
- Bruno calcola k = **15**^11 mod 17 = **9**.
- La chiave segreta è quindi **9**.

L'algoritmo RSA

- Il nome dell'algoritmo deriva dalla prima lettera dei cognomi di coloro che lo inventarono nell'Aprile del 1977: Ronald L. **R**ivest, Adi **S**hamir e Leonard M. **A**dleman.

- Fra le varie applicazioni che sfruttano l'algoritmo RSA possiamo ricordare:
  - Pretty Good Privacy (PGP)
  - Secure Socket Layer (SSL)
  - Secure Electronic Transactions (SET)

L'algoritmo RSA

L' RSA si basa su un procedimento che utilizza i numeri primi e funzioni matematiche che è quasi impossibile invertire:

- Il principio di Eulero ($m^{(p-1)(q-1)} \bmod n = 1$)
- Il principio di Fermat ($m^{(p-1)} \bmod p = 1$)

Dati due numeri primi,è facile calcolarne il prodotto,mentre è difficile determinare, a partire da un determinato numero,quali numeri primi hanno prodotto quel risultato.

Difficile derivare la chiave segreta da quella pubblica

L'algoritmo RSA

- **Generazione della coppia di chiavi:**
  - Scelta di due numeri primi **p**, **q** molto grandi
  - Calcolo di **n=p*q**
  - Calcolo della funzione di Eulero **Φ(n) = (p - 1)(q - 1)**
  - Scelta di un intero **E** minore di **Φ(n)** e primo con esso;
  - Calcolo,utilizzando la versione estesa dell'algoritmo di Euclide, l'intero **D** così da avere **E*D = 1 mod Φ(n)**;
  - Pubblicazione dei valori **E,n** che costituiscono la chiave pubblica.
    - Viene mantenuto segreto **D** che utilizzato con **n** rappresenta la chiave privata

L'algoritmo RSA...in pratica

Calcoliamo il valore di n, prodotto di p e q, due numeri primi molto elevati (sono consigliati valori maggiori di 10^{100}).

Esempio: p=7 q=5

n = p*q = 7*5 = 35

Φ(n) =(p-1)(q-1) = 24

Si sceglie un intero D che sia primo rispetto a **Φ(n)**

D = 7

Trovare un numero E tale che E*D mod **Φ(n)** = 1

E = 7 --> 49 mod 24 = 1

Come si vede per la cifratura si devono conoscere **E** ed **n** (chiave pubblica), mentre per decifrare è necessario conoscere **D** ed **n** (chiave privata).

L'algoritmo RSA...uso delle chiavi

Dopo aver calcolato questi parametri inizia la cifratura.

Il testo in chiaro viene visto come una stringa di bit e viene diviso in blocchi costituiti da k-bit, dove k è il più grande intero che soddisfa la disequazione $2^k < n$.

Per ogni blocco M

Cifratura: $\mathbf{M_c = M^E \bmod n}$.

Decifratura: $\mathbf{M = M_c^D \bmod n}$.

Funzioni Hash e firma digitale

Carlo Saccone

Funzioni e algoritmi Hash

- Utilizzati per produrre un'impronta del messaggio, il Digest
- Un'impronta ha le seguenti caratteristiche:
  - Ha lunghezza prestabilita
  - Non è invertibile

Caratteristiche dell'impronta

- Lunghezza prestabilita, breve rispetto al messaggio
  - E' meno costosa da trasmettere
  - Facilita le manipolazioni (es. cifratura)
- Non invertibilità
  - non esiste un algoritmo noto che, dato un digest, sia in grado di generare un messaggio che gli corrisponde
  - è estremamente difficile produrre un messaggio che abbia un digest predeterminato

Funzioni hash crittografiche

Il requisito di sicurezza è stringente

- Devono essere Strongly Collision Resistant
  - A questo scopo si deve creare un meccanismo che crei una corrispondenza one-way tra messaggio e valore hash relativo. Però:
    - Si deve mappare un numero infinito di messaggi su un insieme finito di possibili impronte (2^n dove n è il numero di bit del digest)
    - è certo che vi saranno delle collisioni
  - Fondamentale è ridurre ad un valore irrisorio la probabilità che queste si verifichino

Algoritmi utilizzati

- Tra gli algoritmi digest più noti ci sono: MD4, MD5, SHA-1, RIPEMD-160
- Solo SHA-1 e RIPEMD-160 sono validi per la legge italiana.
- E' interessante analizzare anche MD5 per il suo elevato valore storico e per fare un parallelo con SHA-1

Algoritmo MD5

- Ideato da uno degli autori di RSA (Rivest) è uno dei più noti algoritmi per la produzione di digest
- Genera impronte di 128 bit
- Recentemente sono stati sollevati dubbi sulla sua robustezza. Hans Dobbertin ha mostrato due versioni dello stesso messaggio che differiscono per un solo carattere ma che producono lo stesso digest MD5

Algoritmo MD5 – qualche dettaglio

- Cinque fasi per generare l'impronta
  - 1 - Aggiunta bits di riempimento
    - Il messaggio viene portato tramite padding ad una lunghezza congrua a 448 mod 512
  - 2 - Aggiunta della lunghezza
    - Viene aggiunta al messaggio una rappresentazione a 64 bits della lunghezza iniziale, prima del padding. Il risultato è un messaggio perfettamente divisibile in blocchi da 448+64=512 bits

Algoritmo MD5 – qualche dettaglio

- 3 - Inizializzazione del buffer MD (initial variable/chaining variable)
  - Si tratta di un buffer di 4 words (A,B,C,D) da 32 bits, collegate in una catena, ciascuna con valori predefiniti
- 4 - Elaborazione del messaggio (compression function)
  - Opera delle trasformazioni sulle 4 variabili di chaining in base a 4 funzioni che vengono applicate alternativamente secondo i valori dei singoli bit dei blocchi in ingresso (quindi non secondo una logica predeterminata)
- 5 - Output
  - Ricavato leggendo a partire dal bit meno significativo le 4 parole precedentemente trasformate

Lo Standard SHS

- Secure Hash Signature Standard
- Pubblicato dalla **Federal Information Processing Standards** nel 2002
- Definisce quattro algoritmi sicuri SHA-1, SHA-256, SHA-384 e SHA-512 per la rappresentazione condensata di messaggi
- Differiscono per dimensioni dei blocchi processati (da 512 a 1024 bits), lunghezza dell'impronta di output (da 160 a 512 bits), e funzioni applicate.

SHA-1

- Analizziamo in dettaglio il primo e maggiormente diffuso degli algoritmi dello standard SHS
- Processamento del messaggio articolato secondo due fasi:
  1. Preprocessamento
  2. Scheduling e calcolo del valore hash

Preprocessamento - Padding

- Padding:
  - Aggiunta di bit in coda ai dati per portarli a una lunghezza prefissata
    - Un multiplo di 512 bit in questo caso

Preprocessamento - Parsing

- Parsing:
  - Il messaggio viene scomposto in blocchi da 512 bits, ciascuno rappresentato come 16 parole da 32 bits

Preprocessamento - Variabili

- Inizializzazione delle variabili da usare nel calcolo dell'hash
  - Vengono inizializzate con valori predefiniti 5 parole da 32 bits:

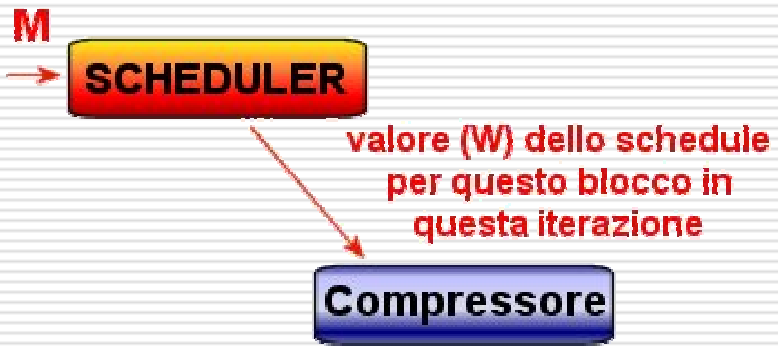
$$H_0^{(0)} = 67452301$$
$$H_1^{(0)} = efcdab89$$
$$H_2^{(0)} = 98badcfe$$
$$H_3^{(0)} = 10325476$$
$$H_4^{(0)} = c3d2e1f0$$

SHA-1 Hash Computation

- Si calcola un "message schedule" da applicare per il calcolo del valore hash
  - 80 "variazioni" per ciascun blocco
  - Si può pensare a questi valori come operatori che modificano il piano delle operazioni da effettuare
- Le trasformazioni subite dalle variabili di supporto H non sono fisse, ma vengono influenzate dal messaggio:
  - Aumento della sicurezza!

Calcolo del Message Schedule

- Siano M_i i blocchi del messaggio dopo il preprocessamento
- Quando un blocco attraversa lo scheduler alla t-esima iterazione viene calcolato il valore (W_t) secondo questa procedura:

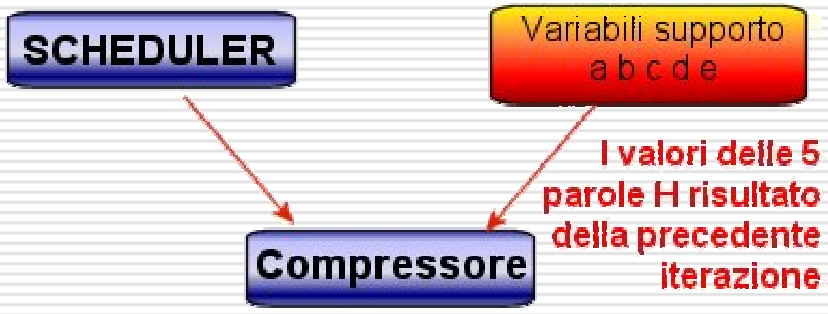


$$W_t = \begin{cases} M_t^{(i)} & 0 \le t \le 15 \\ ROTL^1(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) & 16 \le t \le 79 \end{cases}$$

ROTL^n indica uno shift a sinistra di n posizioni

Variabili di Supporto

- Aggiorna le 5 variabili di supporto
  - Con i valori (i-1)-esimi delle 5 variabili H, dove i indica il numero del blocco che si sta analizzando
  - In pratica memorizza (per riusarli) i valori intermedi delle 5 parole di hash restituiti dall'esecuzione dell'algoritmo sul blocco precedente



$$a = H_0^{(i-1)} \qquad b = H_1^{(i-1)}$$

$$c = H_2^{(i-1)} \qquad d = H_3^{(i-1)}$$

$$e = H_4^{(i-1)}$$

Fase di compressione

- Per ogni blocco itera da 0 a 79 eseguendo:

$$T = ROTL^5(a) + f_t(b,c,d) + e + K_t + W_t$$

$$e = d$$

$$d = c$$

$$c = ROTL^{30}(b)$$

$$b = a$$

$$a = T$$

Utilizza i valori passati dagli strati superiori, con l'aggiunta di costanti di iterazione, per calcolare la funzione f e modificare la sequenza degli H

- Kt è la costante per il calcolo dell'hash da utilizzare all'iterazione t

Funzione di compressione

- La funzione f che compare nel terzo passo è composta dai seguenti predicati booleani:

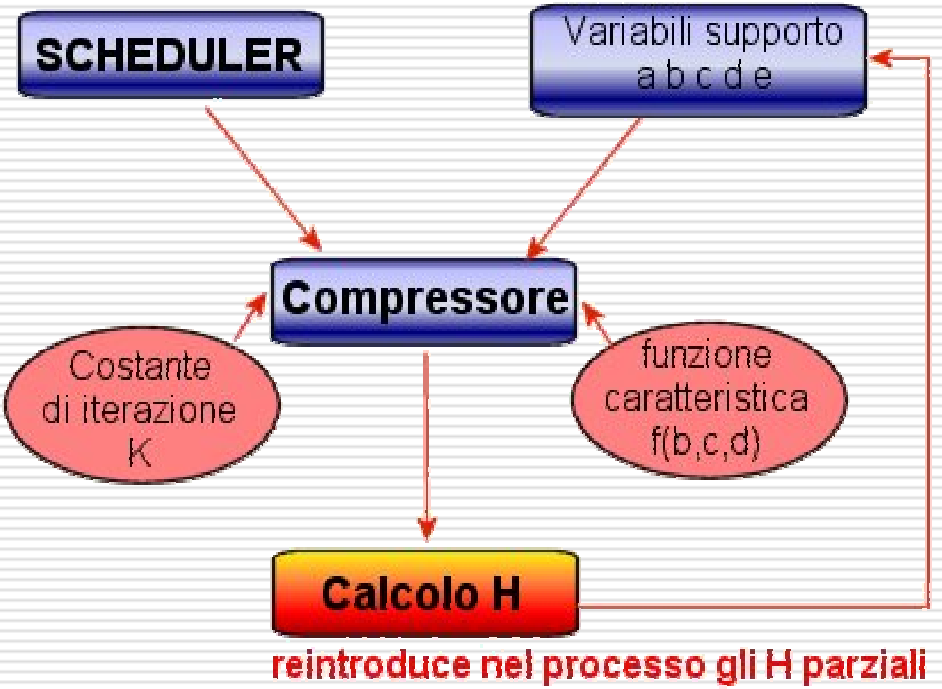
$$f_t(x,y,z) = \begin{cases} Ch(x,y,z) = (x \wedge y) \oplus (\neg x \wedge z) & 0 \le t \le 19 \\ Parity(x,y,z) = (x \oplus y \oplus z) & 20 \le t \le 39 \\ Maj(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) & 40 \le t \le 59 \\ Parity(x,y,z) = (x \oplus y \oplus z) & 60 \le t \le 79 \end{cases}$$

- Prende in input 3 parole da 32 bit e restituisce una sola parola da 32 bit utilizzando uno XOR bit a bit (e varianti)

Calcolo dei valori H

- Calcola i cinque valori hash ottenuti in seguito all'elaborazione del blocco corrente
  - Dati dalla somma delle variabili di supporto e dei rispettivi valori hash del ciclo precedente

$$H_0^{(i)} = a + H_0^{(i-1)}$$
$$H_1^{(i)} = b + H_1^{(i-1)}$$
$$H_2^{(i)} = c + H_2^{(i-1)}$$
$$H_3^{(i)} = d + H_3^{(i-1)}$$
$$H_4^{(i)} = e + H_4^{(i-1)}$$



Estrazione dell'Output

- Una volta ripetuti i passi da 1 a 4 per un numero di volte N pari al numero di blocchi del messaggio, si ottiene l'impronta dalla concatenazione dei valori delle 5 variabili hash restituiti dall'ultima iterazione

$$H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)}$$

SHA-1 vs MD5

- MD5:
  - Output di 128 bits
  - 5 round di elaborazione
  - Le operazioni sulle variabili di supporto vengono modificate tramite predicati sui bit del blocco
- SHA-1
  - Output di 160 bits
  - 80 cicli di elaborazione per ciascun blocco
  - Le operazioni sulle variabili di supporto vengono determinate con l'ausilio di uno schedule legato al contenuto del blocco mediante una funzione complessa

RIPEMD-160

- Nasce da una delle linee di sviluppo originate a seguito degli studi relativi a MD4 e MD5 e ai relativi punti deboli
- La dimensione del digest viene portata a 160 bit
- Vengono aggiunti due round e modificate le operazioni booleane bit a bit
- La velocità di esecuzione è molto minore rispetto agli antenati MD e persino rispetto a SHA-1

RIPEMD-160

- E' organizzato in 5 round che operano su 5 parole da 32 bits
- In ciascun round vengono applicate le seguenti operazioni:
  - Shift binario
  - Operazioni booleane bit a bit
  - Ordinamento delle parole
  - Addizione di costanti

Sicurezza degli algoritmi hash

- La letteratura è ricca di documenti che descrivono e testimoniano attacchi agli algoritmi menzionati
  - In particolare sulla possibilità di trovare due messaggi con lo stesso digest
- MD5 è stato messo in disuso dalla legislazione italiana esattamente per questo motivo

Sicurezza degli algoritmi hash

- La resistenza di una particolare funzione hash ad una serie di strategie di attacco note fornisce una misura del suo livello di sicurezza
- Diverse tipologie di attacchi
  - Indipendenti dall'algoritmo
    - Es. birthday attack
  - Legati alla natura degli algoritmi
    - Es. chaining attack

Yuval's Birthday attack

- Si basa sul paradosso del compleanno
  - Scegliendo casualmente in un insieme di n elementi, con buona probabilità un elemento ripetuto verrà trovato dopo O(sqrt(n)) estrazioni
- Significativo nel caso delle funzioni hash one way perché è più semplice trovare due collisioni piuttosto che cercare una controimmagine dato il valore hash
- Si può applicare a tutte le funzioni hash senza chiave in un tempo $O(2^{m/2})$ dove m è la lunghezza dell'impronta

Yuval's Birthday attack

- Idea: posso ottenere la firma su un certo documento x1 originale, e voglio applicarla in seguito al documento x2, contraffatto, in modo che risulti firmato in modo autentico
- Creo un insieme di messaggi ottenuti da piccole variazioni di x1, ne calcolo l'hash e lo memorizzo per una ricerca successiva
- Allo stesso modo creo un insieme di messaggi varianti di x2.
- Cerco nel secondo insieme un messaggio che produca una delle impronte del primo insieme
- Se il tentativo riesce posso richiedere una firma autentica per il documento x1 modificato ed applicarla al documento x2 modificato

Yuval's Birthday attack

- Ad ogni modo per un algoritmo che produce un digest di 128 bit, considerando una serie di 64 piccole modifiche, ad esempio sostituzioni di tabulazioni con spazi o simili, l'attacco necessita di un tempo $O(2^{64})$ e spazio in memoria per $O(2^{64})$ messaggi, il che è pensabile solo avendo a disposizione numerosi calcolatori che lavorano in parallelo

Chaining attack

- Sono gli attacchi che sfruttano l'iteratività degli algoritmi di hash, ed in particolare l'uso delle "chaining variables"
- Si concentrano sulla funzione di compressione piuttosto che su tutto il processo di hash

Chaining attack in teoria

- Supponiamo di avere un algoritmo hash h con blocchi x_i di 512 bits e variabili di chaining H_i di 128 bit e una funzione di compressione f che produce un output $H_{i+1} = f(H_i, x_i)$ che dipende dal ciclo precedente
- Dato un messaggio di 10 blocchi si può supporre di voler sostituire uno di questi blocchi senza modificare il valore di output
- Se f si comporta come una distribuzione casuale il numero di blocchi in cui è possibile trovarne uno che abbia questa proprietà è $2^{512} / 2^{128} = 2^{384}$
- Un qualsiasi metodo che consenta di trovare un blocco simile in modo efficiente rappresenta un potenziale attacco all'algoritmo
- Si può pensare di applicare il metodo per sostituire ciascun blocco, e modificare l'intero messaggio

Sicurezza degli algoritmi hash

- Ulteriori attacchi efficaci sono quelli di analisi statistica delle impronte ottenute da messaggi simili
  - In pratica varianti dell'attacco del compleanno, con vincoli meno restrittivi sulle variazioni applicate al messaggio
- Nel febbraio 2005 alcuni ricercatori cinesi hanno individuato, pare utilizzando questa tecnica, una falla in SHA-1 che permetterebbe di ridurre di circa 2000 volte il tempo necessario per la ricerca di una collisione

La vostra firma è in pericolo?

- Assolutamente no!!!
  - Le collisioni trovate presentano caratteri del tutto privi di senso, che non sono "piccole variazioni" sul messaggio
    - È alquanto improbabile che una banca accetti il seguente messaggio: "msg. Da Alice #àà@@]++[ì?^ a banca: ç°è*?$)&&£ versare !"!"!??$)%(^§]#[500 dollari sul conto di Bob"
  - Il calcolo necessario per la ricerca di una collisione richiederebbe comunque circa 1000 anni con un PC di ultima generazione

La vostra firma è in pericolo? (2)

- I messaggi importanti solitamente hanno un formato prestabilito, che riduce al minimo le variazioni accettate.
  - Per cui il messaggio stesso è la sua prima arma difesa da un attacco
- Ad ogni modo si potrebbe decidere di rimpiazzare SHA-1 per mantenere comunque un livello di sicurezza del tutto adeguato alle garanzie che la firma digitale deve offrire

Il processo di Firma Digitale

Mazza Daria

Azioni Preliminari

- Registrazione dell'utente presso una Certification Authority
- Generazione di una coppia di chiavi Ks e Kp
  - Da utilizzare una per la generazione della firma, che verrà mantenuta segreta: Ks, e l'altra, destinata alla verifica e che verrà resa pubblica: Kp
- Certificazione della chiave pubblica Kp
- Registrazione della chiave pubblica Kp
  - Il certificato è reso disponibile in uno o più cataloghi ai quali può accedere chiunque abbia bisogno di accertare la validità di una sottoscrizione digitale

Registrazione utente

La registrazione presso la CA ha due scopi:

- rendere questa certa della sua identità
- instaurare con esso un canale di comunicazione sicuro

Procedura di registrazione:

1. L'utente richiede alla CA la registrazione fornendo la documentazione richiesta da questa per accertare l'identità del richiedente
2. La CA attribuisce all'utente un identificatore di cui essa garantisce l'univocità

Registrazione utente

3. La CA inserisce l'utente nei cataloghi di utenti registrati
4. La CA fornisce attraverso un canale sicuro la chiave crittografica che l'utente dovrà utilizzare per le richieste di certificazione delle chiavi
   - La necessità di un canale sicuro nasce dal fatto che, sebbene le richieste di certificazione contengono chiavi pubbliche per le quali non è richiesta una protezione ai fini della riservatezza, la CA deve essere certa che ciascuna richiesta provenga effettivamente dell'utente in essa indicato e non da un altro soggetto che lo sta impersonando

Azioni Preliminari

- Registrazione dell'utente presso una Certification Authority
- **Generazione di una coppia di chiavi Ks e Kp**
  - Da utilizzare una per la generazione della firma, che verrà mantenuta segreta: Ks, e l'altra, destinata alla verifica e che verrà resa pubblica: Kp
- Certificazione della chiave pubblica Kp
- Registrazione della chiave pubblica Kp
  - Il certificato è reso disponibile in uno o più cataloghi ai quali può accedere chiunque abbia bisogno di accertare la validità di una sottoscrizione digitale

Certificazione chiave pubblica

Ha lo scopo di rassicurare chiunque riceva un documento correttamente firmato circa l'identità del soggetto che ha apposto la firma

L'operazione avviene attraverso tre passi:

1. L'utente invia alla CA la richiesta di certificazione per la chiave Kp generata nella fase precedente, autenticandola mediante la chiave ricevuta dalla CA durante il processo di registrazione
2. La CA genera il certificato
3. Il certificato viene inviato al richiedente

Azioni Preliminari

- Registrazione dell'utente presso una Certification Authority
- Generazione di una coppia di chiavi Ks e Kp
  - Da utilizzare una per la generazione della firma, che verrà mantenuta segreta: Ks, e l'altra, destinata alla verifica e che verrà resa pubblica: Kp
- Certificazione della chiave pubblica Kp
- **Registrazione della chiave pubblica Kp**
  - Il certificato è reso disponibile in uno o più cataloghi ai quali può accedere chiunque abbia bisogno di accertare la validità di una sottoscrizione digitale

Firma Digitale

Viene realizzata tramite :

- utilizzo di particolari funzioni matematiche, chiamate **funzioni hash unidirezionali**
- **tecniche crittografiche a chiave pubblica** dove il mittente utilizza la funzione di cifratura e la sua chiave privata per generare un'informazione da associare al messaggio e dove chiunque può accertare la provenienza del messaggio utilizzando la chiave pubblica del mittente

Processo di firma

Il processo di firma digitale passa attraverso quattro fasi :

- Generazione dell'impronta digitale
- Generazione della firma
- Apposizione della firma
- Verifica della firma

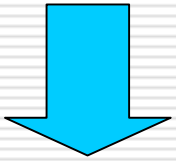
Generazione dell'impronta

- Viene applicata al documento in chiaro una funzione di hash che produce una stringa binaria di lunghezza costante e piccola (normalmente 128 o 160 bit) chiamata *digest message*; l'impronta del messaggio.

- Queste funzioni devono avere due proprietà:
  - **unidirezionalità,** ossia dato x è facile calcolare f(x), ma data f(x) è computazionalmente difficile risalire a x
  - **prive di collisioni** (collision-free), ossia a due testi diversi deve essere computazionalmente impossibile che corrisponda la medesima impronta

Generazione della firma

Consiste semplicemente nella cifratura con la propria chiave privata dell'impronta generata in precedenza. (Non si cifra l'intero documento!)



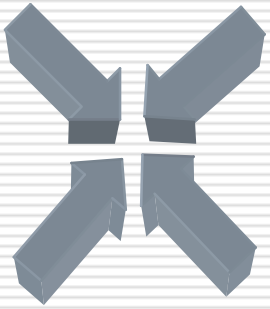
La firma risulta legata, da un lato (attraverso la chiave privata usata per la generazione) al soggetto sottoscrittore, e dall'altro (tramite l'impronta) al testo sottoscritto

Apposizione della firma

- La firma digitale generata precedentemente viene aggiunta in una posizione predefinita, normalmente alla fine del testo del documento e corredata con il suo certificato, più eventualmente quello della CA

- Al destinatario viene spedita la busta digitale composta da:
  - il documento in chiaro
  - la "firma digitale" in calce al documento
  - il certificato rilasciato dalla competente autorità di certificazione a garanzia della titolarità della chiave pubblica necessaria per decriptare la firma digitale

Punto della situazione - Mittente

DOCUMENTO +  FUNZIONE HASH =  IMPRONTA +  CODIFICA CON CHIAVE PRIVATA = 

FIRMA

Punto della situazione - Mittente

26GS5TY

Verifica della Firma

Il processo di verifica consiste nelle seguenti fasi:

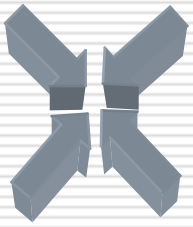
- chi riceve il messaggio firmato si procura il certificato del mittente (spesso è in allegato al messaggio stesso) e, dopo averne controllato la validità, ne estrae la chiave pubblica che è contenuta
- con questa chiave pubblica il ricevente può decriptare la firma digitale ed estrarre il digest che il mittente aveva calcolato per il messaggio
  - se l'operazione riesce significa che il messaggio era stato criptato, al momento della spedizione, con l'unica chiave privata corrispondente a quella pubblica contenuta nel certificato. Questo garantisce l'identità del mittente

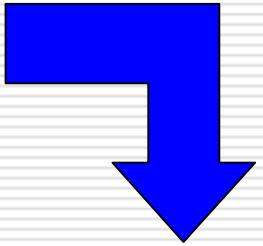
Verifica della Firma

Il ricevente:

- Calcola un suo digest per lo stesso messaggio, avendo cura di usare lo stesso sistema del mittente (SHA-1 o RIPEMD-160)
- Confronta i due digest, quello che ha appena calcolato e quello estratto dalla firma digitale: se sono uguali significa che il messaggio non è stato in alcun modo alterato durante la spedizione

Punto della situazione - Destinatario

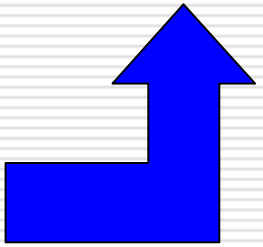
DOCUMENTO ORIGINARIO +  FUNZIONE HASH =  IMPRONTA



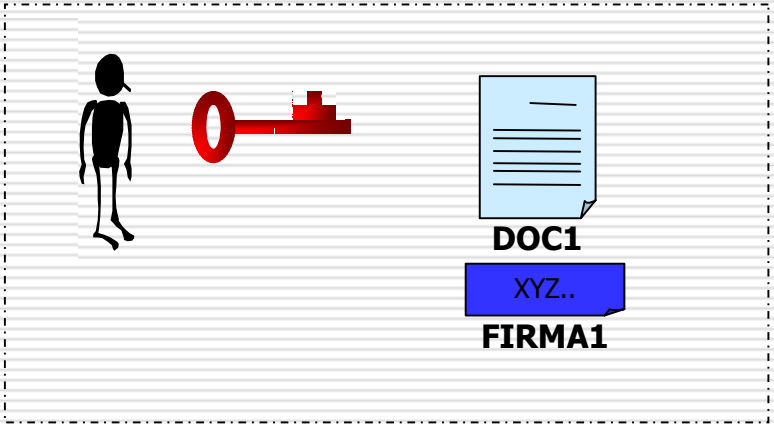
= Ok!

26GS5TY

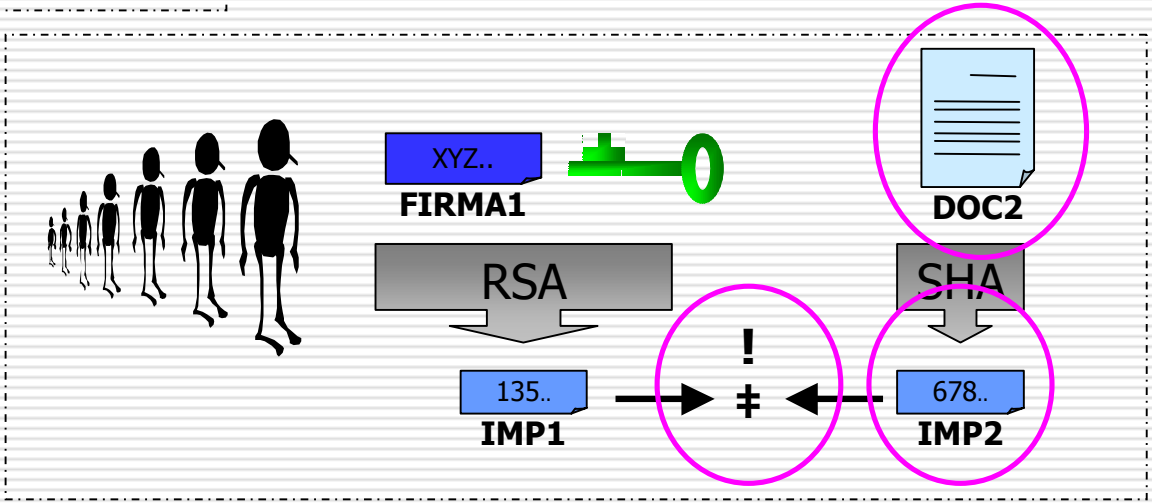
FIRMA +  DECODIFICA CON CHIAVE PUBBLICA =  IMPRONTA



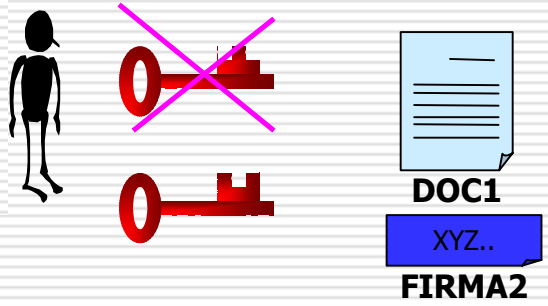
Mancata Verifica – caso 1



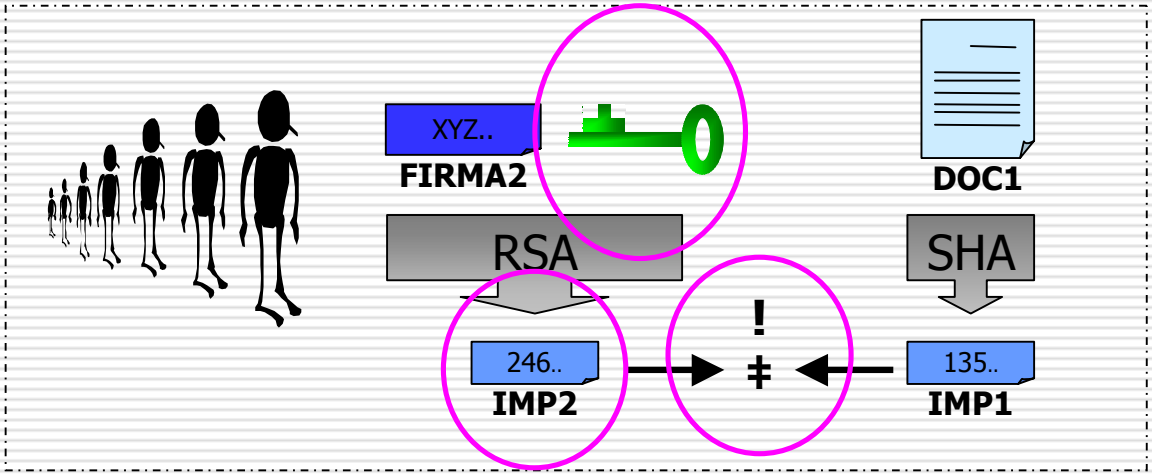
- La verifica non va a buon fine, non si ottiene l'uguaglianza !
- Modifica del documento da parte di terzi



Mancata Verifica – caso 2



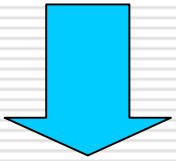
- La verifica non va a buon fine, non si ottiene l'uguaglianza !
- Invio da parte di terzi del documento firmato !



Certificati

Certification Authority

La verifica di una firma digitale con la chiave pubblica del «sottoscrittore», pur fornendo la certezza che il documento non ha subito alterazioni e che può provenire solo da una persona che conosca la chiave privata corrispondente, non da ancora alcuna indicazione certa circa la reale identità di quest'ultima persona



- **Certification Authority** e il sistema dei **certificati**
  - Documenti elettronici che associano una chiave pubblica ad una carta d'identità personale
  - Lo standard più utilizzato è X.509.

Certification Authority

Le CA svolgono i seguenti compiti fondamentali:

- verificano ed attestano, emettendo un apposito certificato digitale, l'identità del titolare di una determinata chiave pubblica
- stabiliscono il termine di scadenza dei certificati
- pubblicano il certificato e la chiave pubblica
- ricevono la segnalazione di eventuali smarrimenti, furti, cancellazioni, divulgazioni improprie, ecc. di chiavi private e pubblicano quindi la lista dei certificati revocati o sospesi in conseguenza di tali fatti

Revoca e sospensione di un certificato

Revoca del certificato :

- il certificatore annulla la validità del certificato da un dato momento in poi. Il provvedimento non è retroattivo.

Sospensione del certificato elettronico:

- il certificatore sospende la validità del certificato per un determinato periodo di tempo.

I certificati revocati e sospesi sono inseriti in due liste:

CRL: Certificate Revocation List

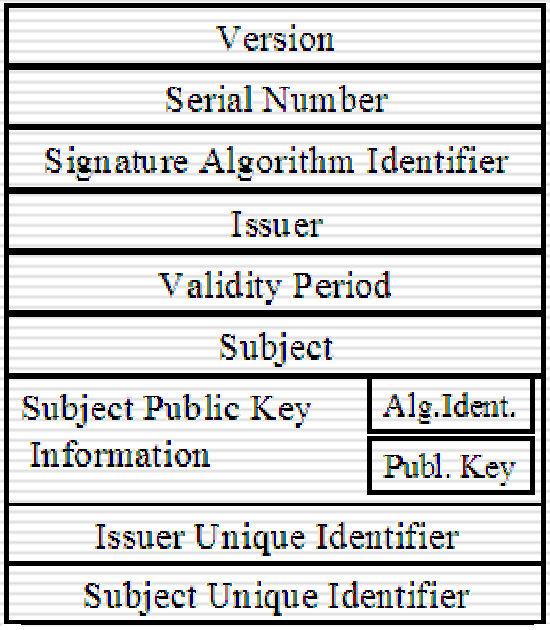
CSL: Certificate Suspension List

Standard X.509

- Appartiene alla serie di documenti X.500 dell'ITU-T (International Telecommunications Union)
- X.509 specifica come fornire un servizio di autenticazione agli utenti

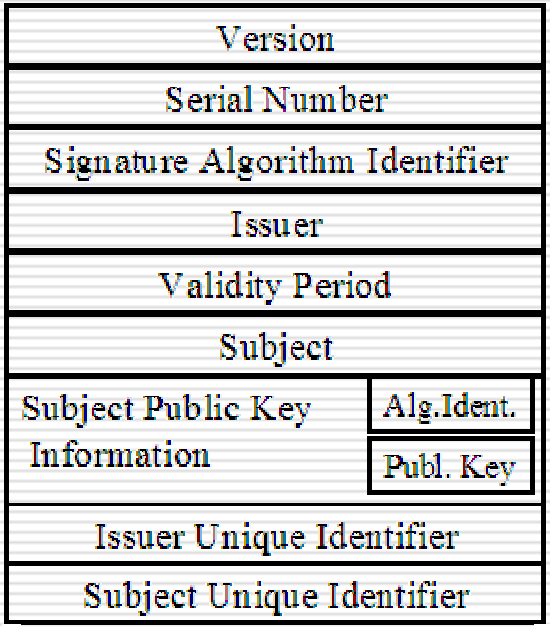
- E' stato emesso nel 1988
  - X.509v2 nel 1993
  - X.509v3 nel 1995

Standard X.509



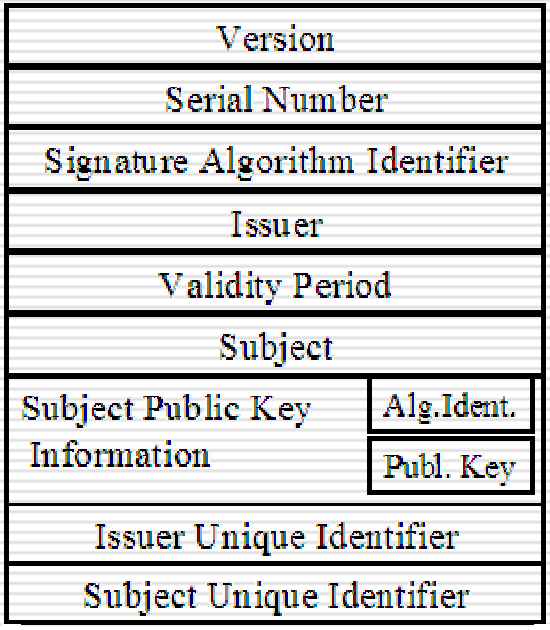
- ***Version:*** Indica la Versione dello standard applicata al certificato
- ***Serial Number:*** Numero univoco assegnato dall'autorità di certificazione che lo ha emesso. Tale attributo, unito al nome dell'autorità che lo ha emesso, identifica univocamente un certificato
- ***Signature Algorithm Identifier:*** Identifica l'algoritmo di firma digitale utilizzato dalla CA per firmare i certificati pubblici

Standard X.509



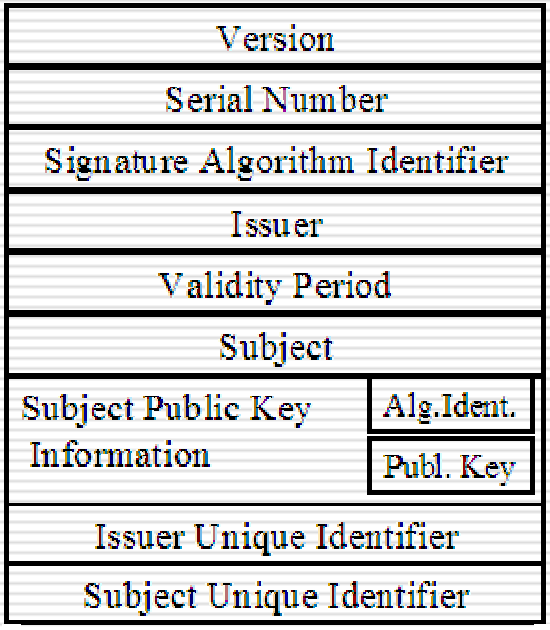
- ***Issuer:*** Nome della CA che emette i certificati
- ***Validity Period:*** Data e ora d'inizio e fine validità del certificato. La data di scadenza si riferisce all'utilizzo della chiave privata e non di quella pubblica. Infatti questa può essere utilizzata anche successivamente alla scadenza per verificare la veridicità di quei file immagazzinati prima della scadenza

Standard X.509



- ***Subject:*** Nome del soggetto a cui è rilasciato il certificato
- ***Subject public key information:*** identifica sia la chiave pubblica dell'intestatario del certificato che l'identificativo dell'algoritmo di hashing e di crittografia pubblica con cui la chiave può essere utilizzata

Standard X.509



- ***Issuer unique identifier:*** Una stringa di bit opzionale per identificare univocamente una CA nel caso che lo stesso nome sia stato assegnato ad un'altra entità
- ***Subject unique identifier:*** Una stringa di bit opzionale per identificare univocamente il soggetto, Subject, nel caso che lo stesso nome sia stato assegnato ad un'altra entità

Marcatura Temporale

Marcatura Temporale

- La marcatura temporale o **Time Stamping** serve per attribuire certezza ad un documento circa il momento in cui questo è stato redatto ed è divenuto valido
- E' essenziale per evitare che documenti redatti utilizzando certificati revocati o scaduti vengano utilizzati a scopi fraudolenti

- Nella marca sono contenute le seguente informazioni:
  - data e ora della creazione della marca
  - nome dell'emittente della marca
  - impronta del documento cui la marca fa riferimento

Marcatura Temporale

L'operazione avviene secondo la seguente procedura:

1. L'impronta del documento viene inviata al servizio di marcatura temporale. Tale impronta costituisce il riferimento certo al testo originale ma non ne consente la ricostruzione pertanto la marcatura può essere effettuata senza compromettere la confidenzialità del testo

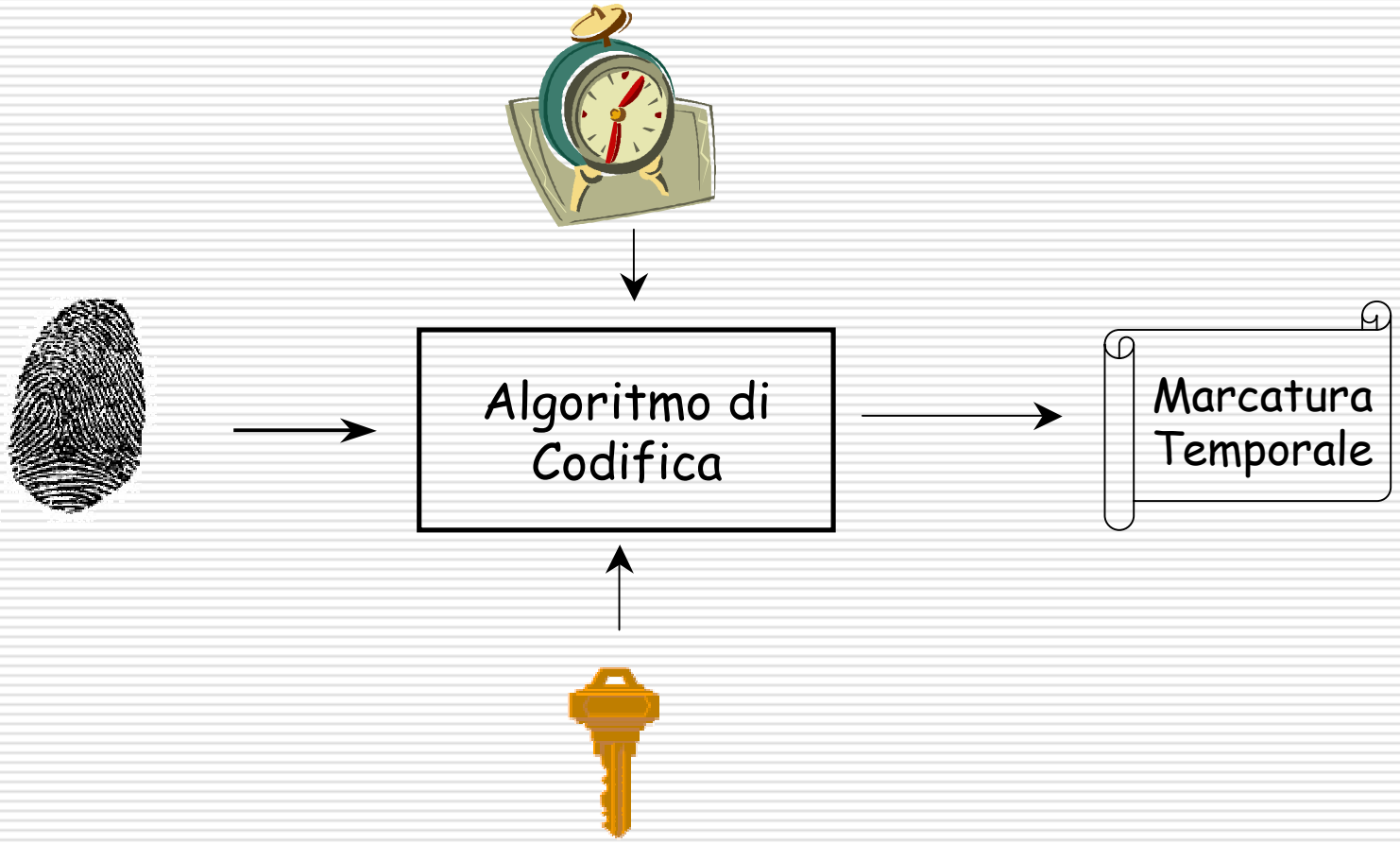
2. Il servizio di marcatura aggiunge all'impronta ricevuta la data e l'ora, ottenendo una impronta marcata

Marcatura Temporale

3. Tale impronta marcata viene cifrata con la chiave privata del servizio, ottenendo la marca temporale da cui è possibile recuperare, mediante la relativa chiave pubblica, l'impronta del documento e la data e l'ora della sua generazione

4. La marca temporale viene inviata al richiedente il quale la allega al documento

Marcatura Temporale



Dimostrazione pratica

Dispositivi di firma

Per la normativa italiana con dispositivo di firma si intende:

- *"un apparato elettronico programmabile solo all'origine, facente parte del sistema di validazione, in grado almeno di conservare in modo protetto la chiave privata e generare al suo interno le firme digitali"*

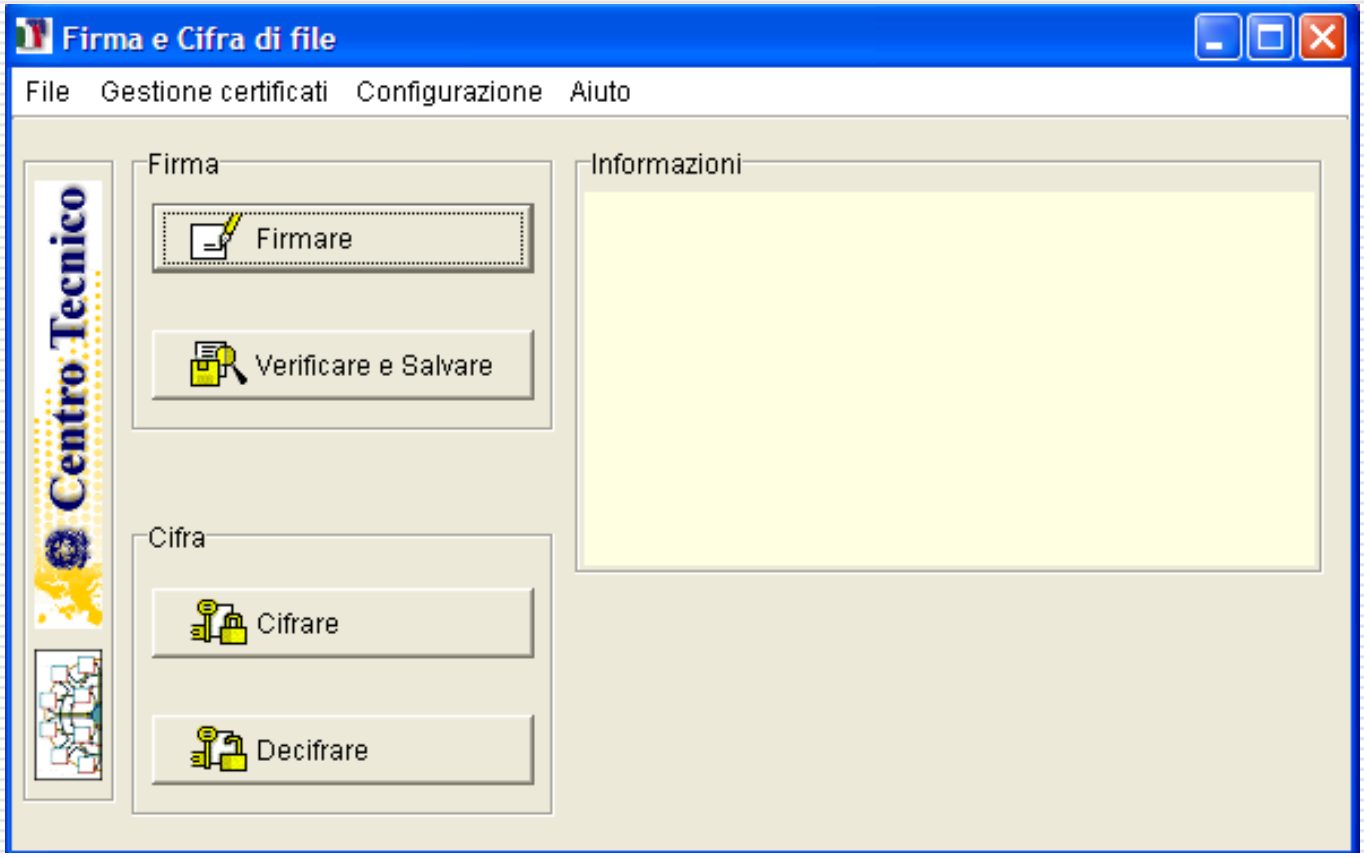
Uno strumento che è possibile utilizzare come dispositivo di firma è la **smart card crittografica**

Smart Card

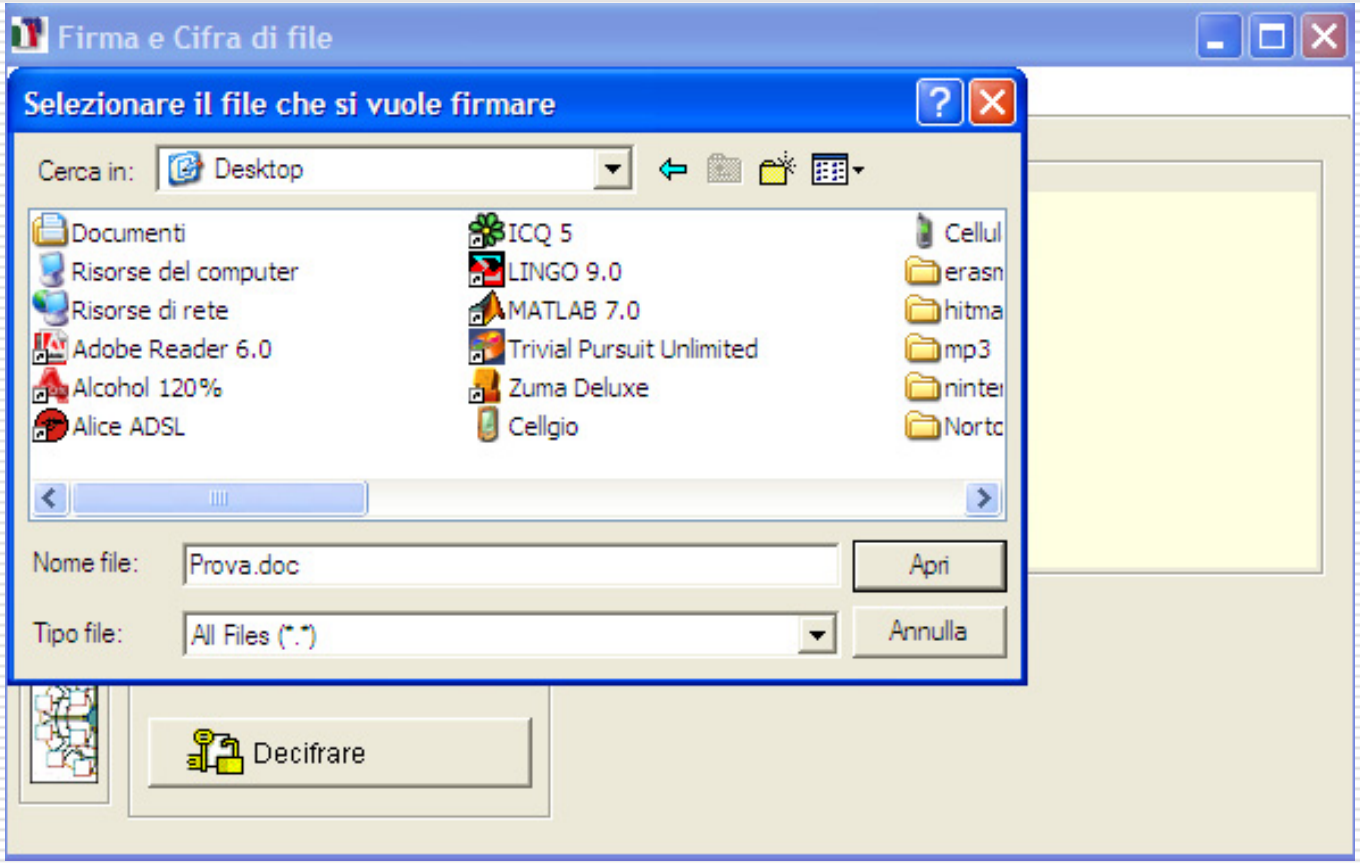
- E' simile ad una tradizionale carta di credito ma incorpora un processore in grado di memorizzare dati ed informazioni a cui è possibile accedere tramite un codice di sicurezza (PIN)
- E' in grado di eseguire:
  - un algoritmo di inizializzazione in grado di generare e memorizzare stabilmente una coppia di chiavi pubblica/privata
  - un algoritmo di cifratura asimmetrica in grado di cifrare i dati in ingresso con la chiave privata memorizzata internamente
- Si collega con il computer tramite un apposito lettore ed il relativo software di interfaccia

Dispositivi di firma

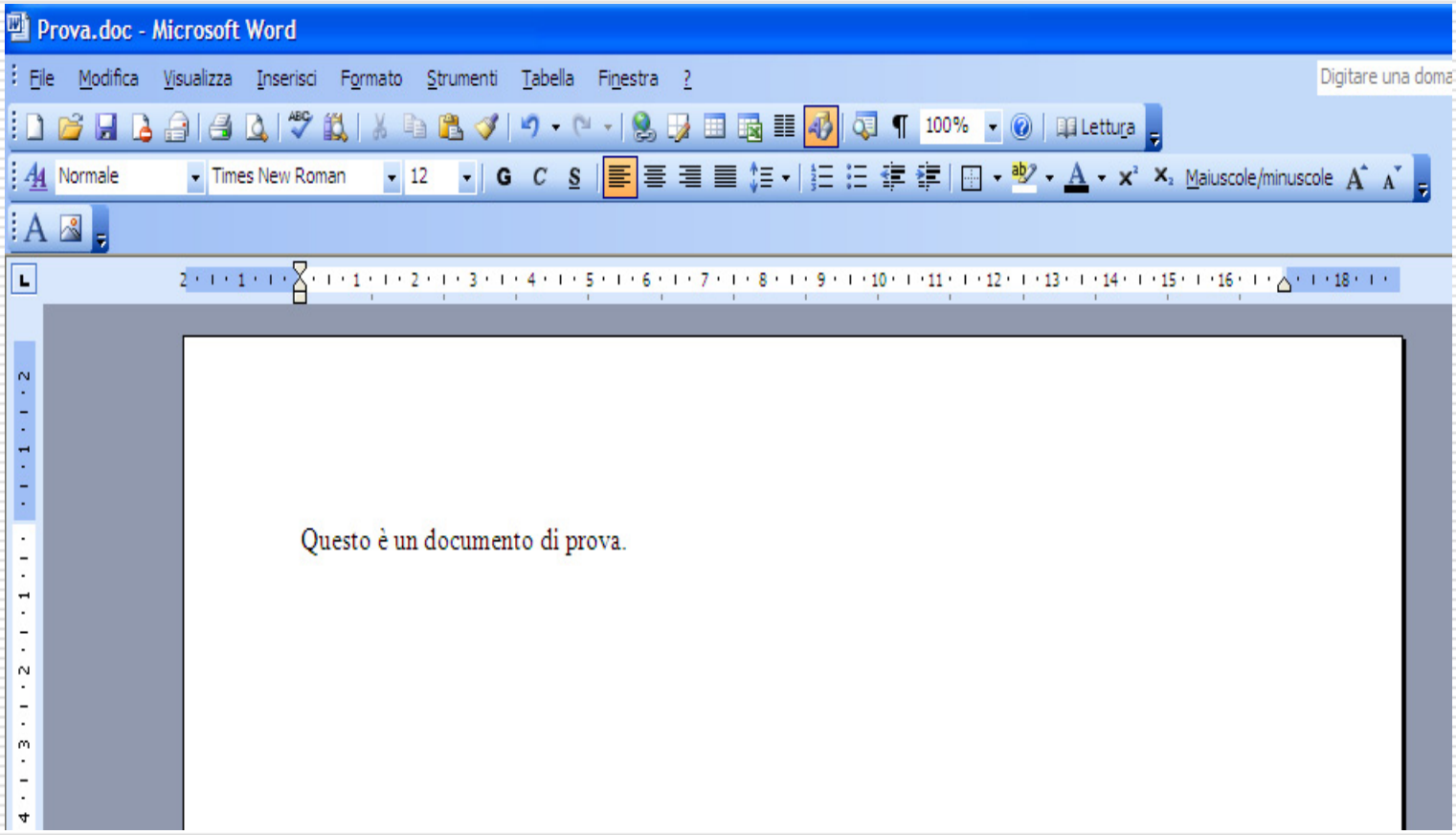
Dimostrazione pratica



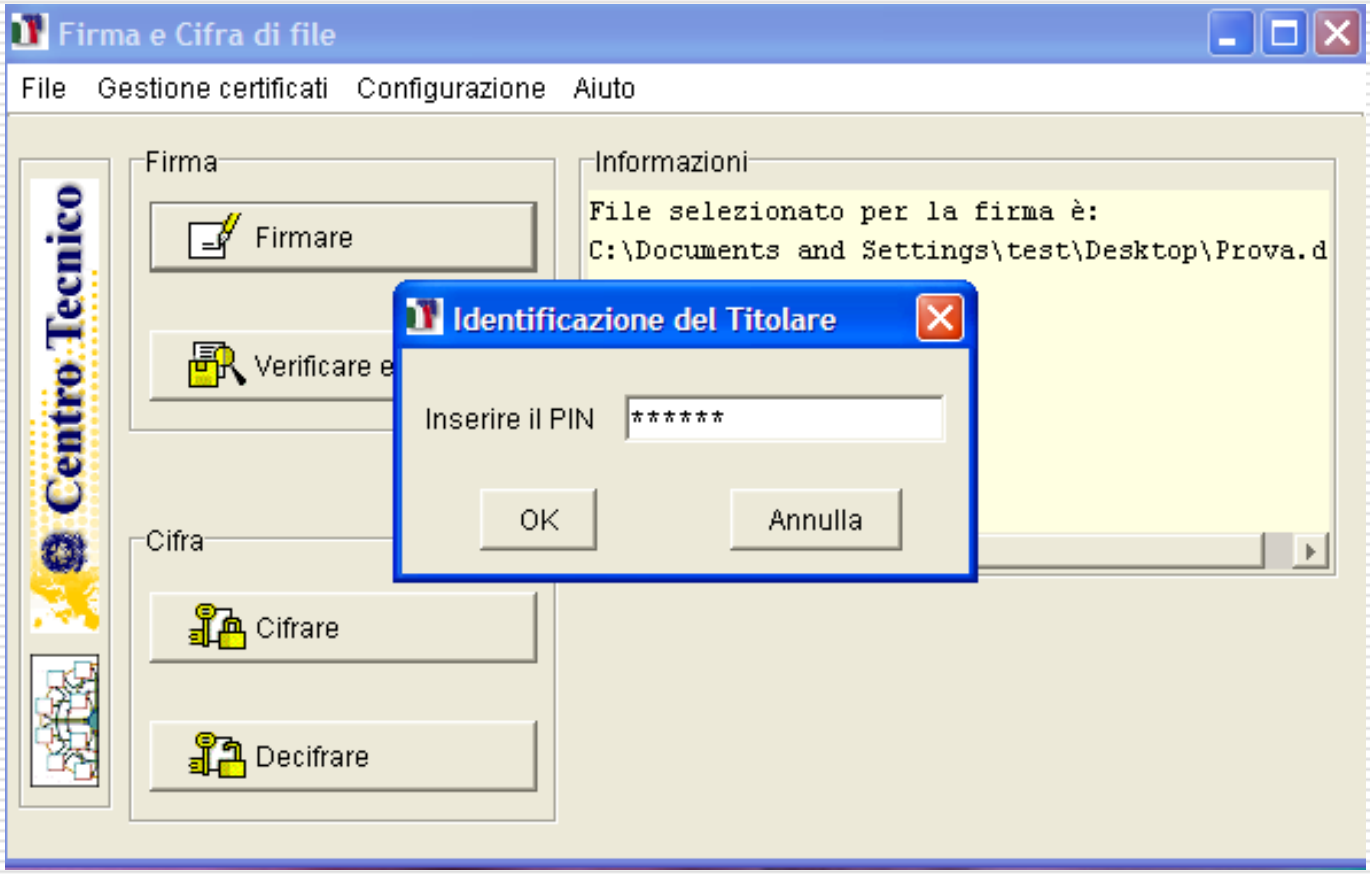
Dimostrazione pratica



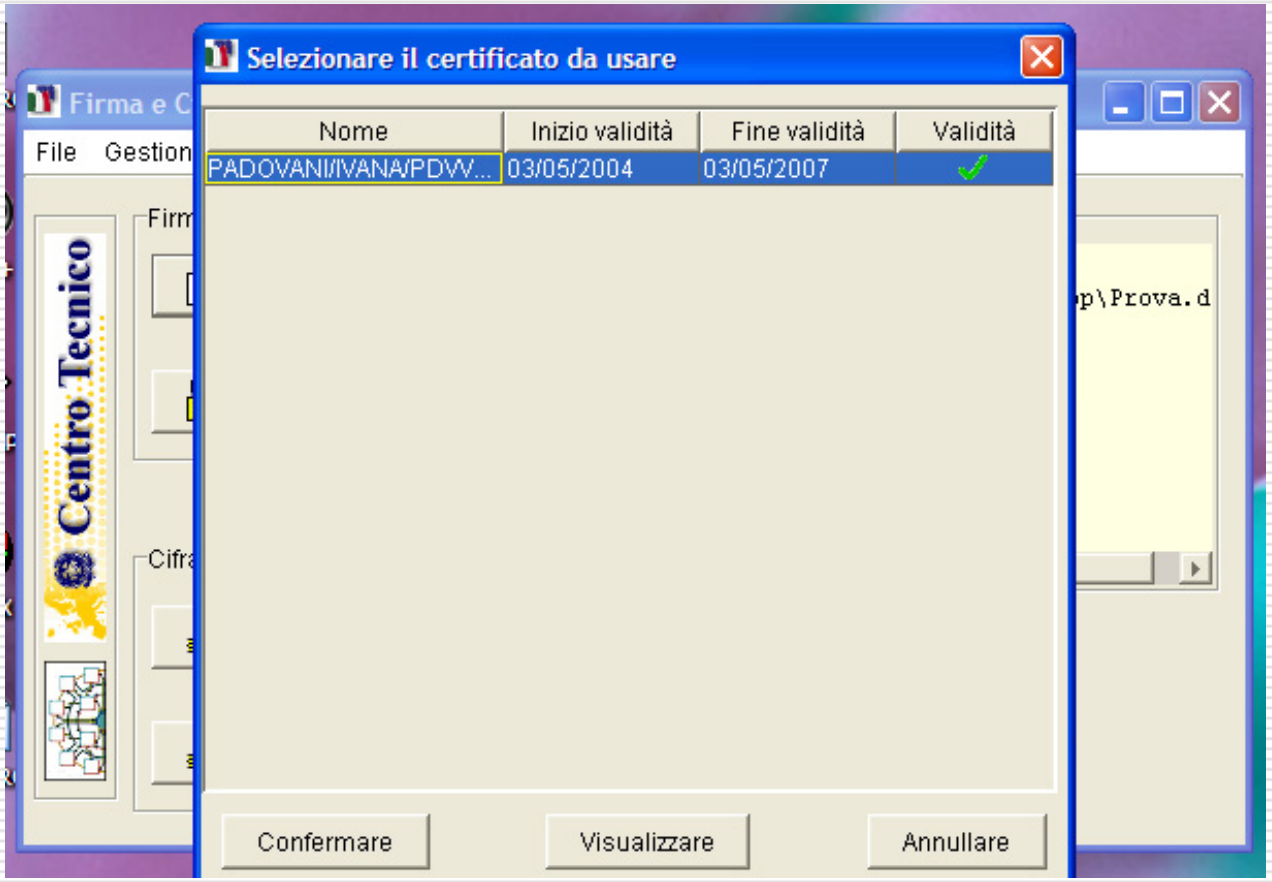
Dimostrazione pratica



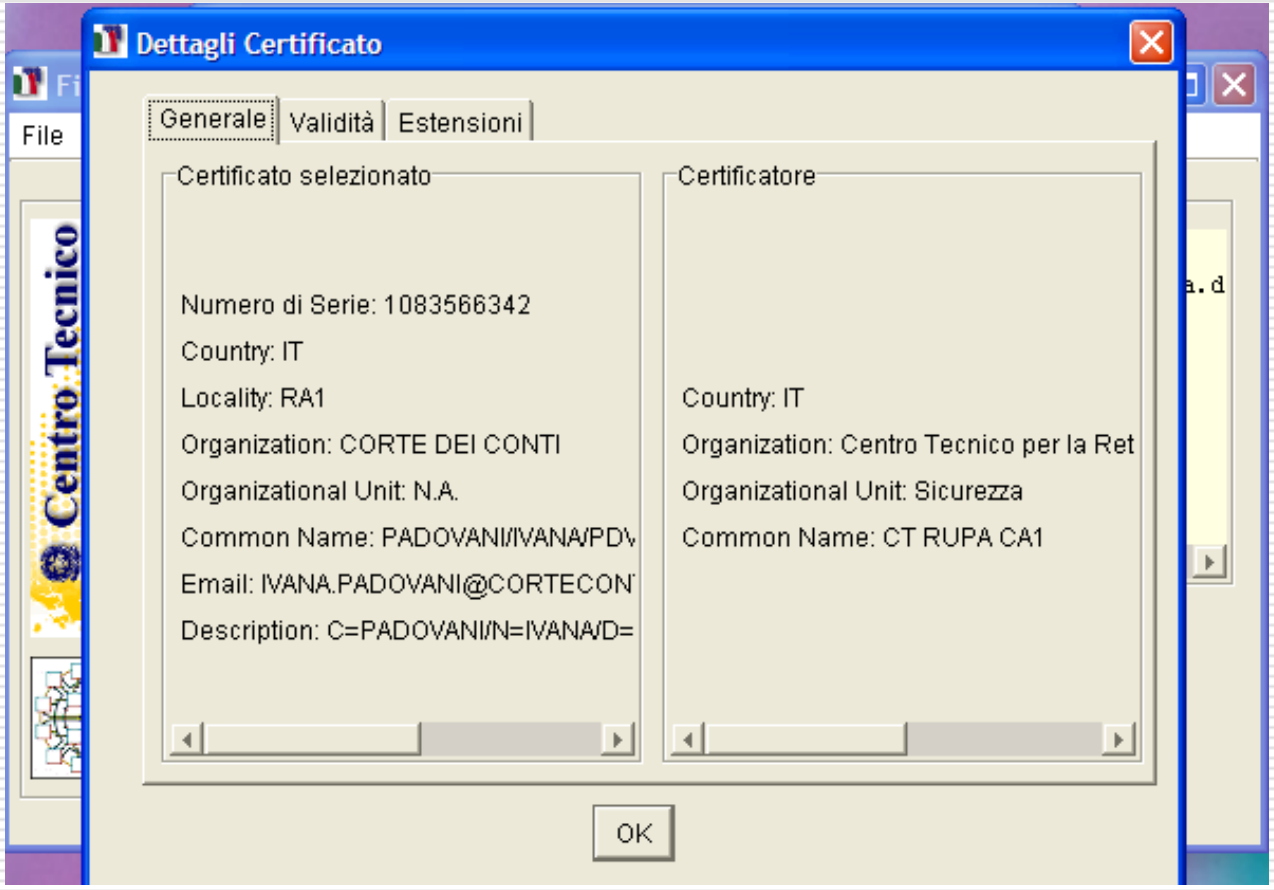
Dimostrazione pratica



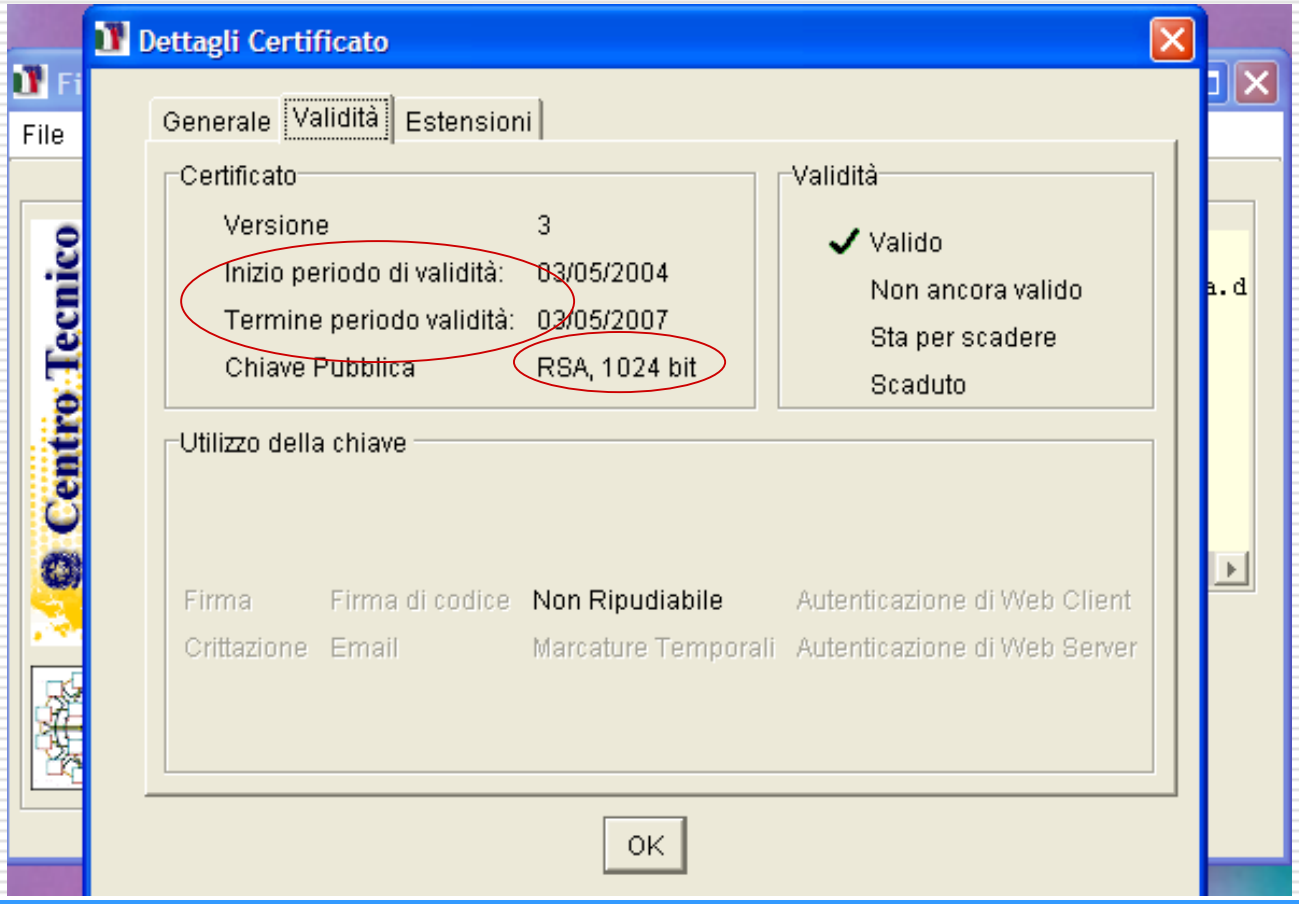
Dimostrazione pratica



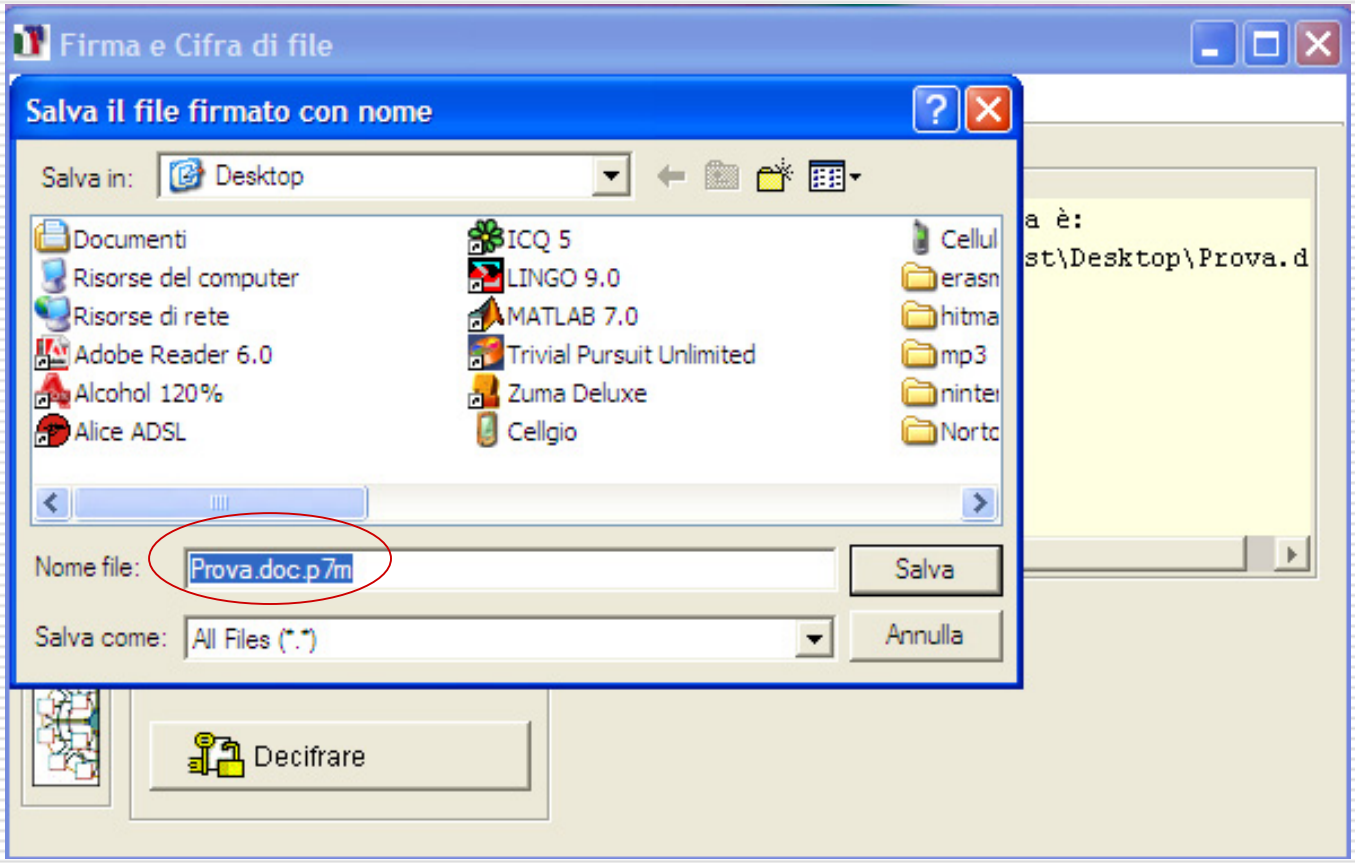
Dimostrazione pratica



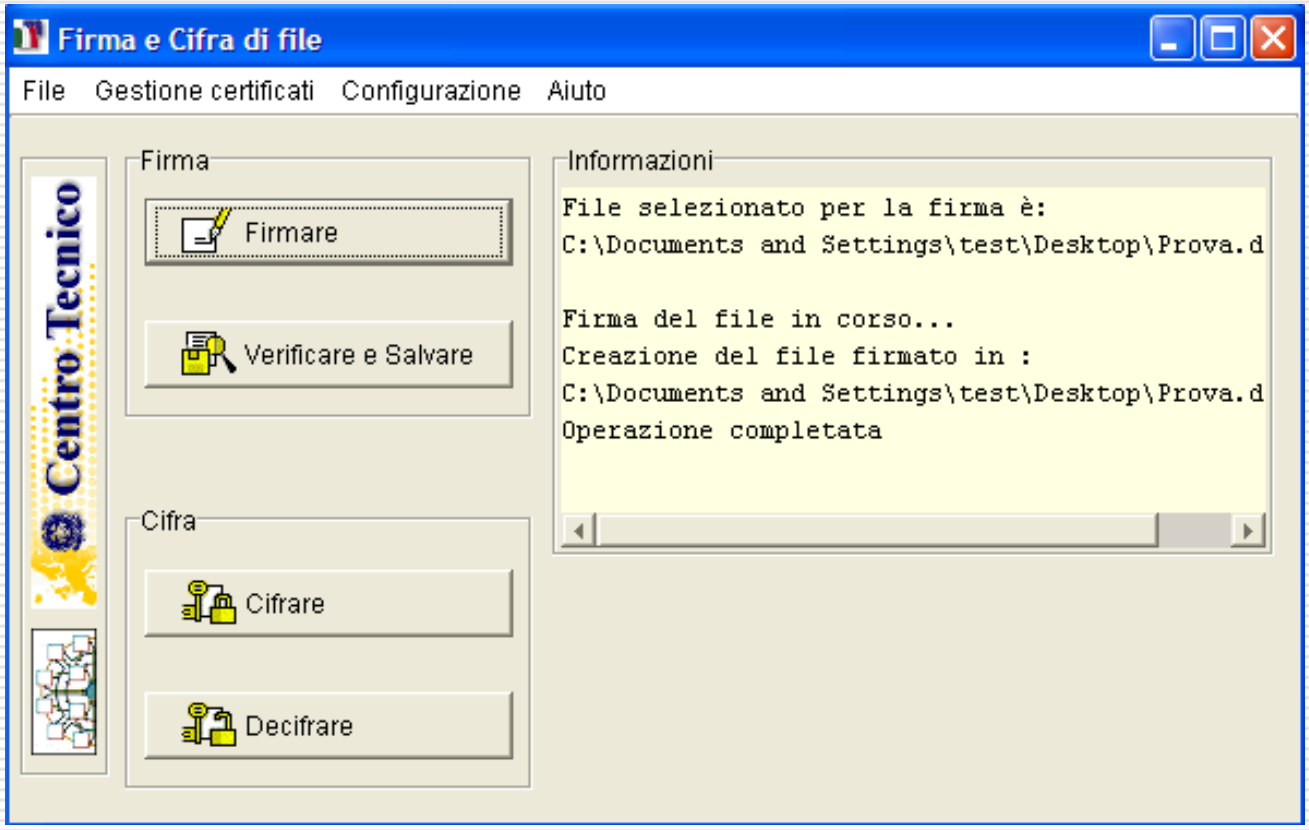
Dimostrazione pratica



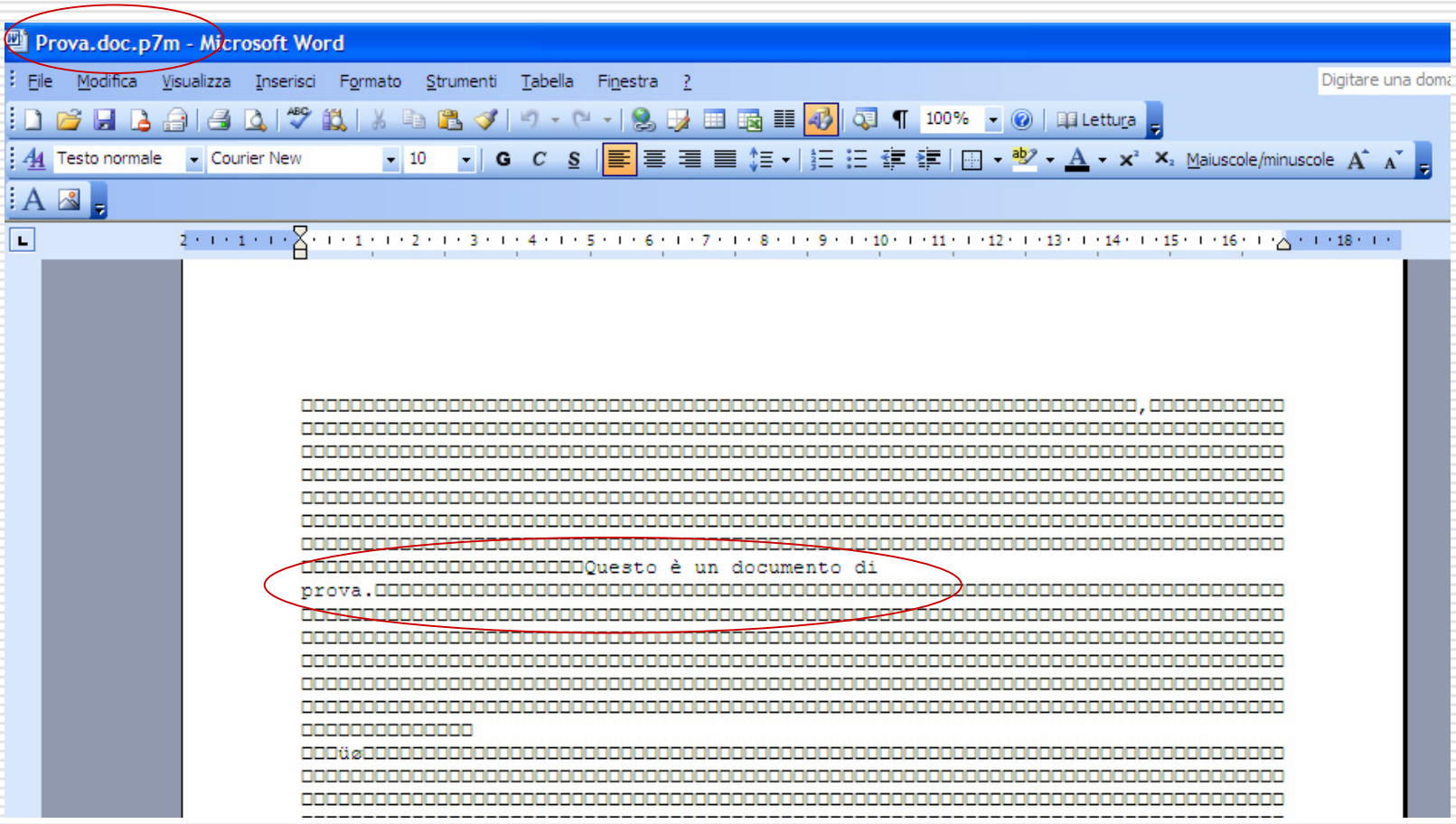
Dimostrazione pratica



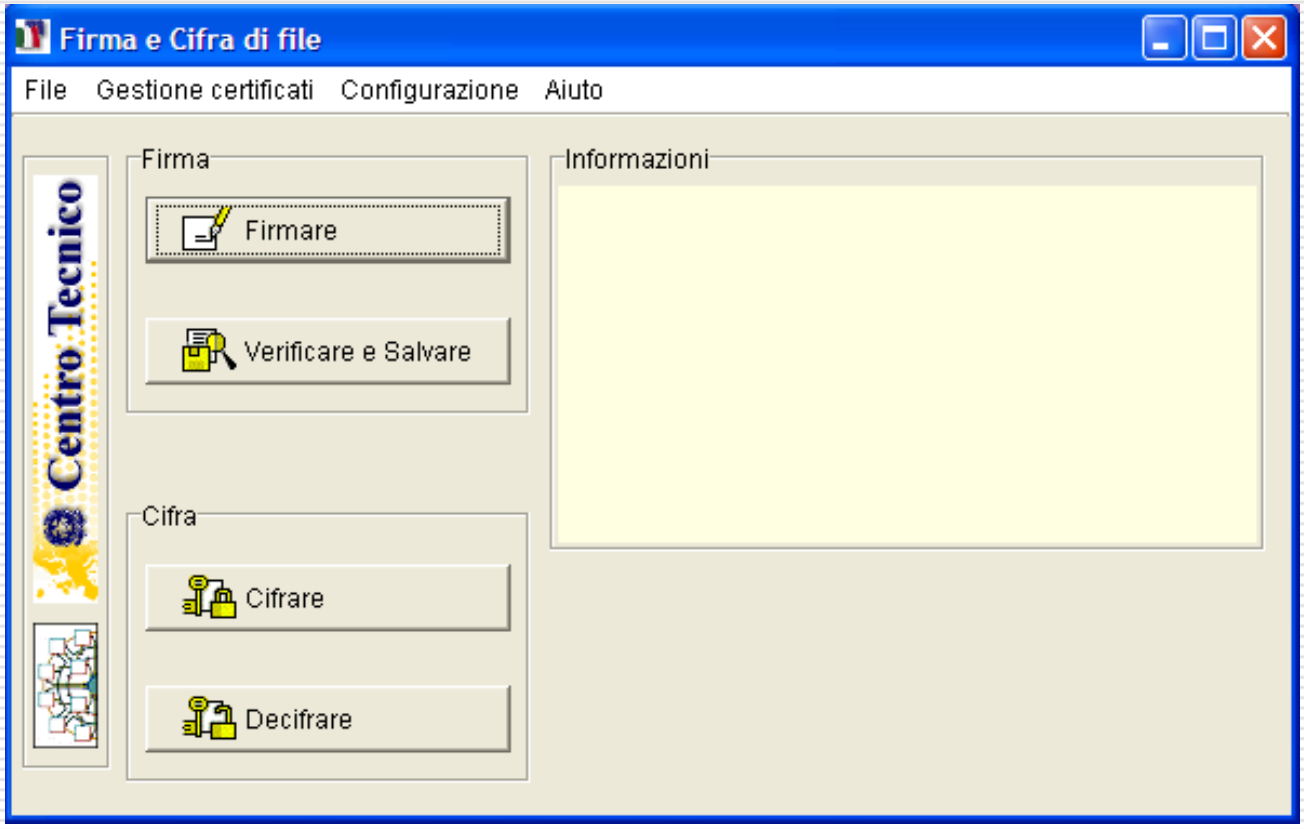
Dimostrazione pratica



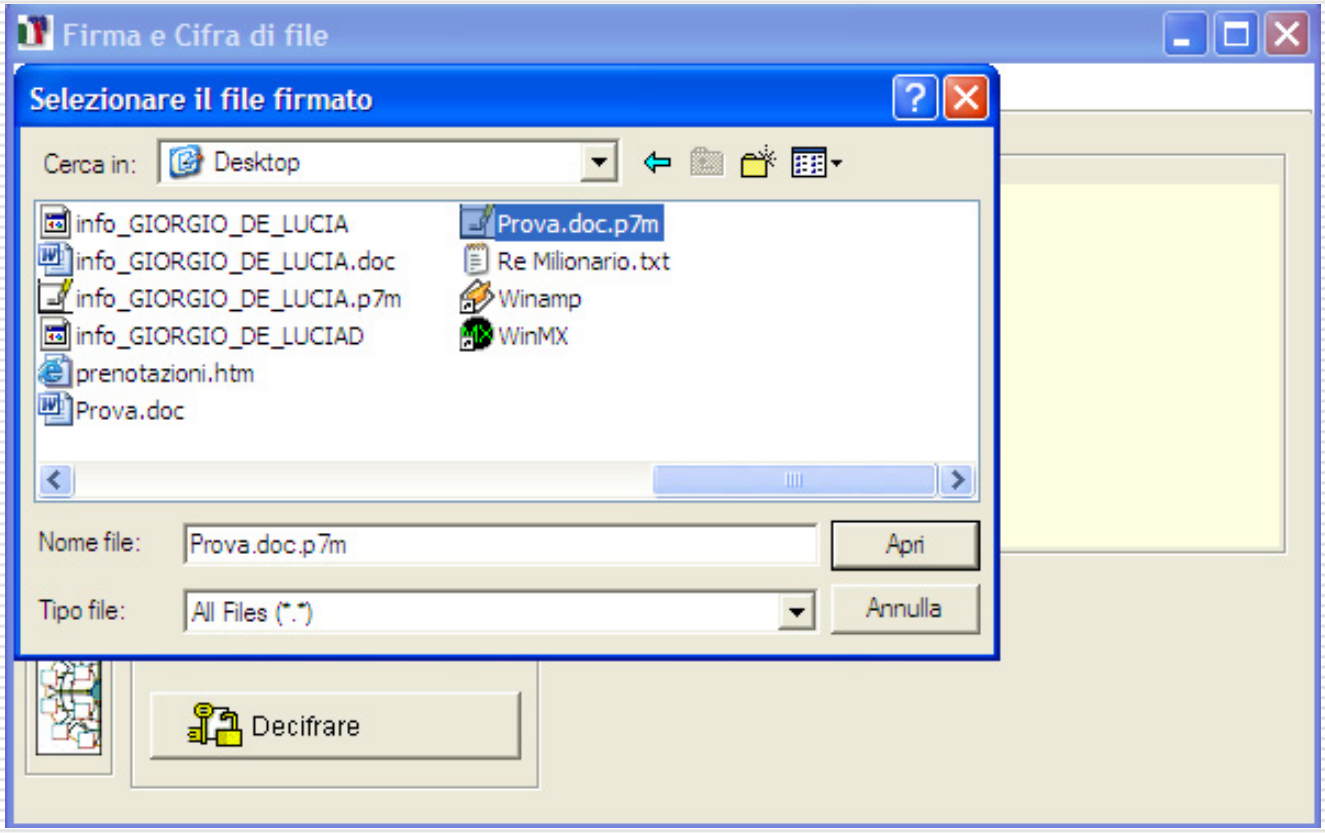
Dimostrazione pratica



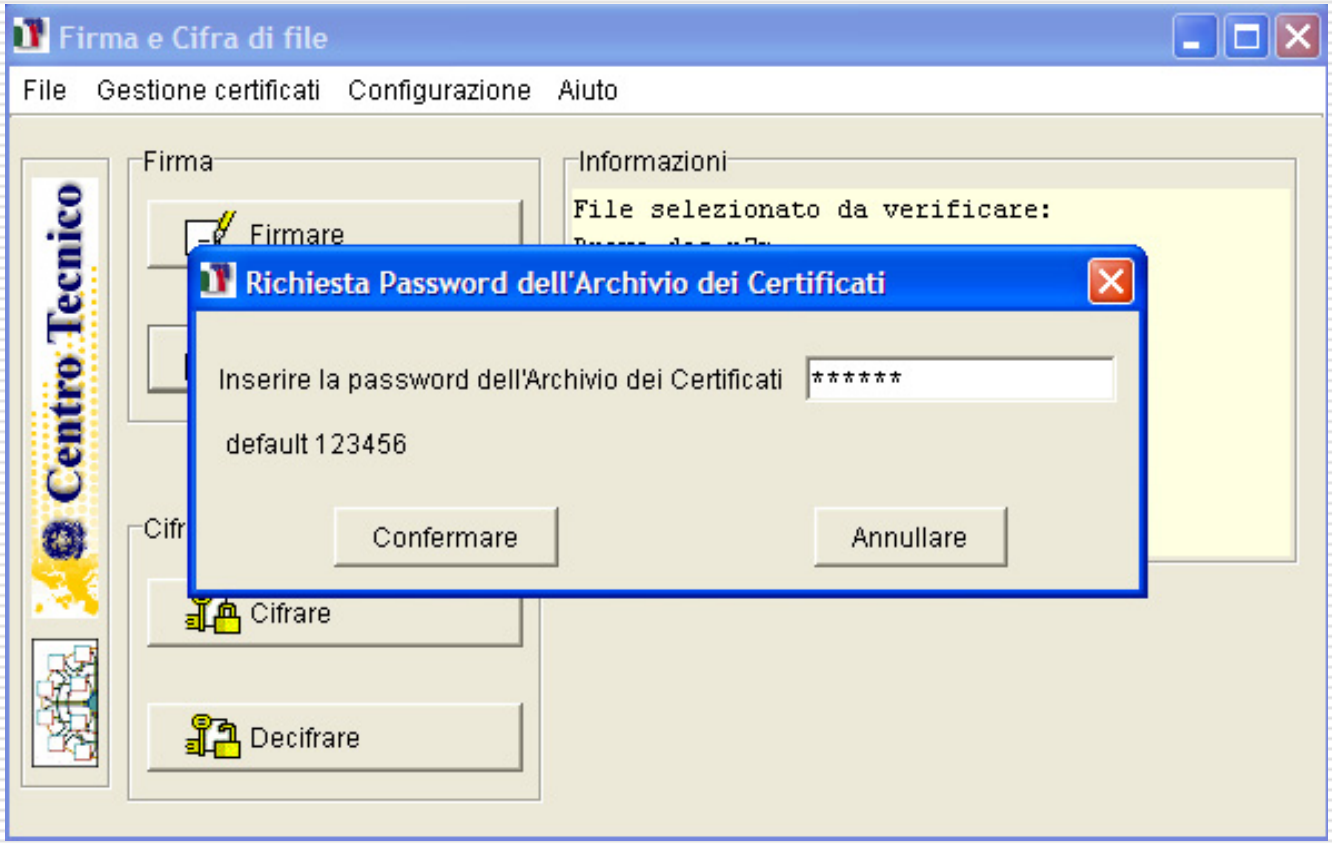
Dimostrazione pratica - Verifica



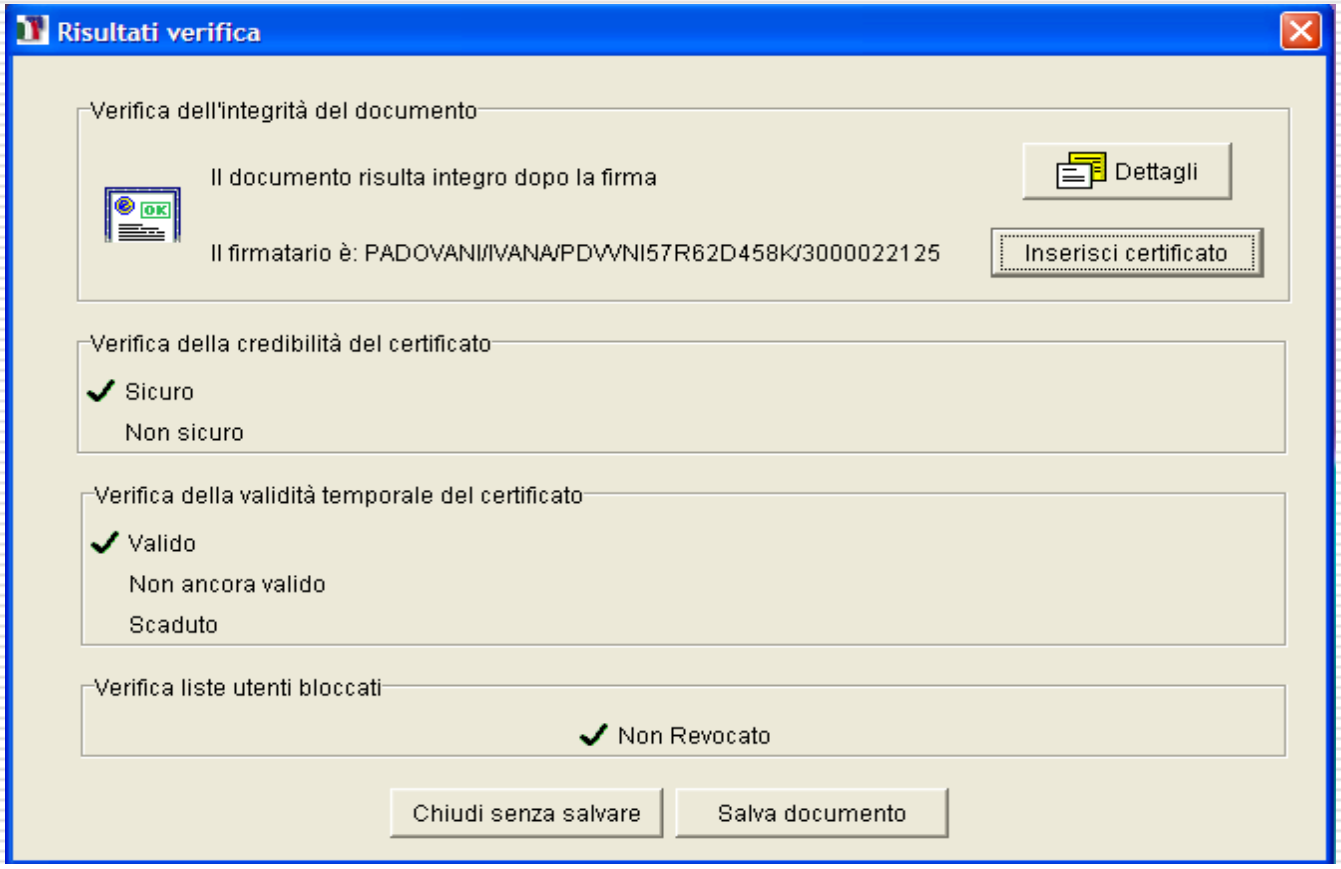
Dimostrazione pratica - Verifica



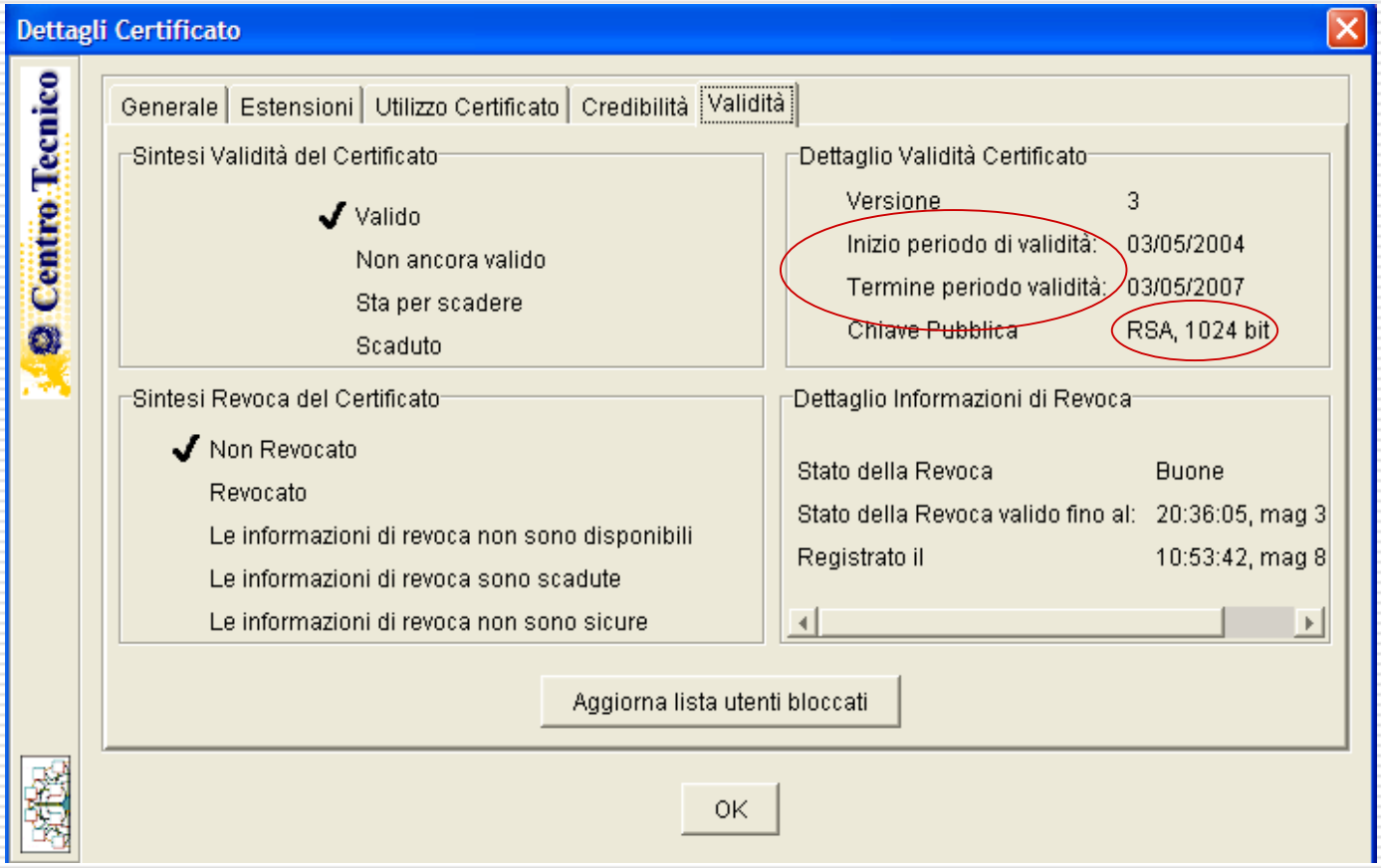
Dimostrazione pratica - Verifica



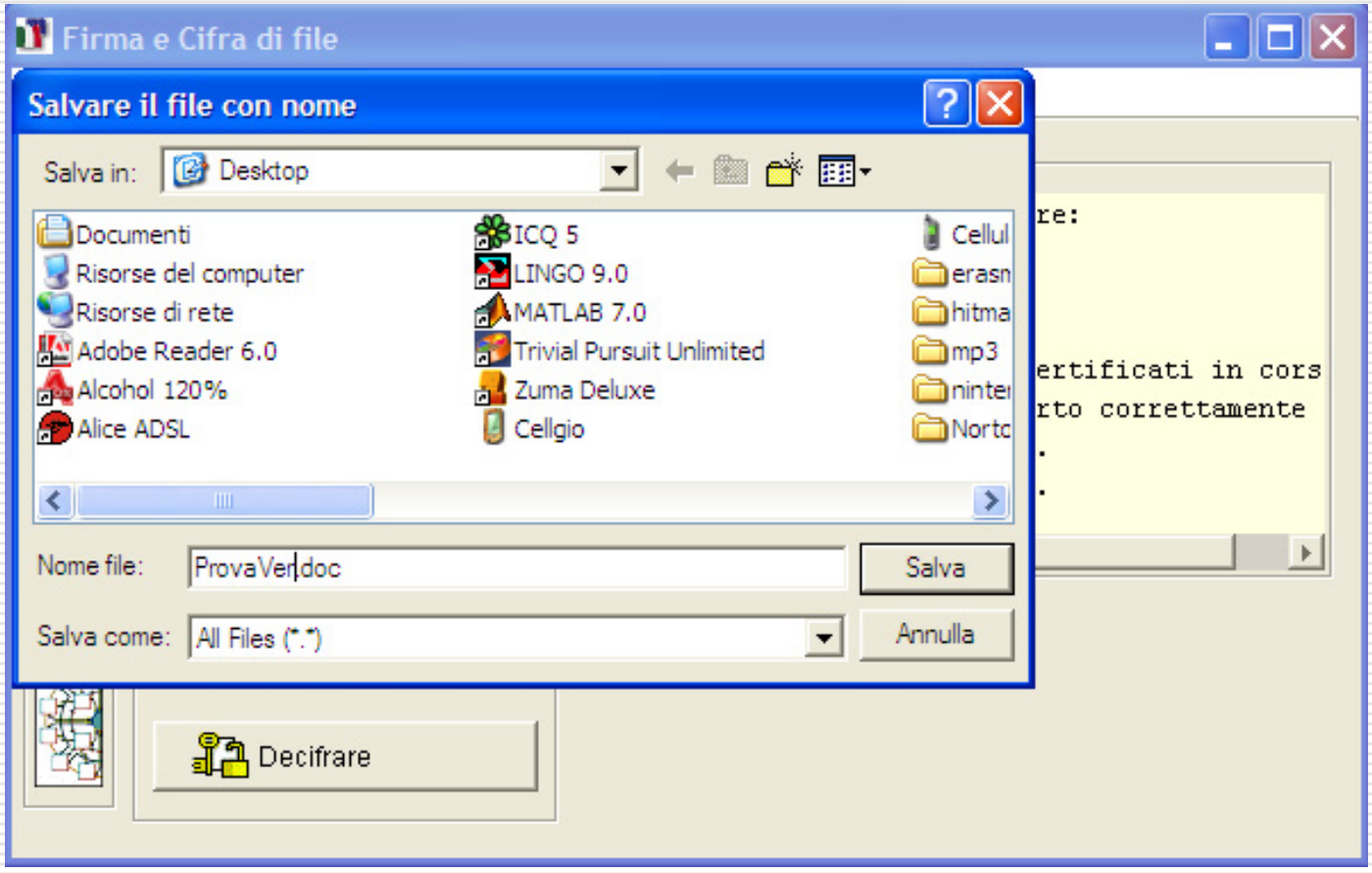
Dimostrazione pratica - Verifica



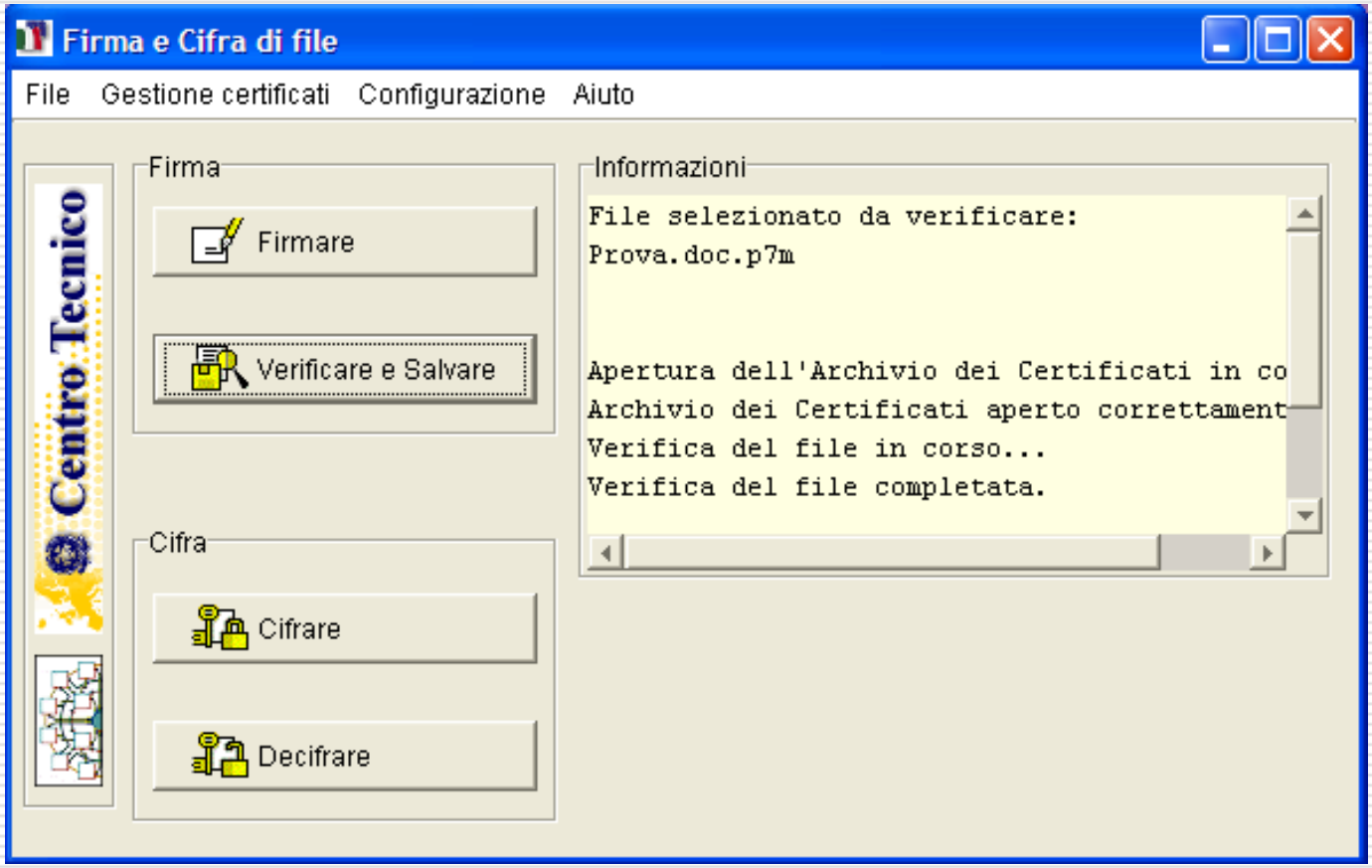
Dimostrazione pratica - Verifica



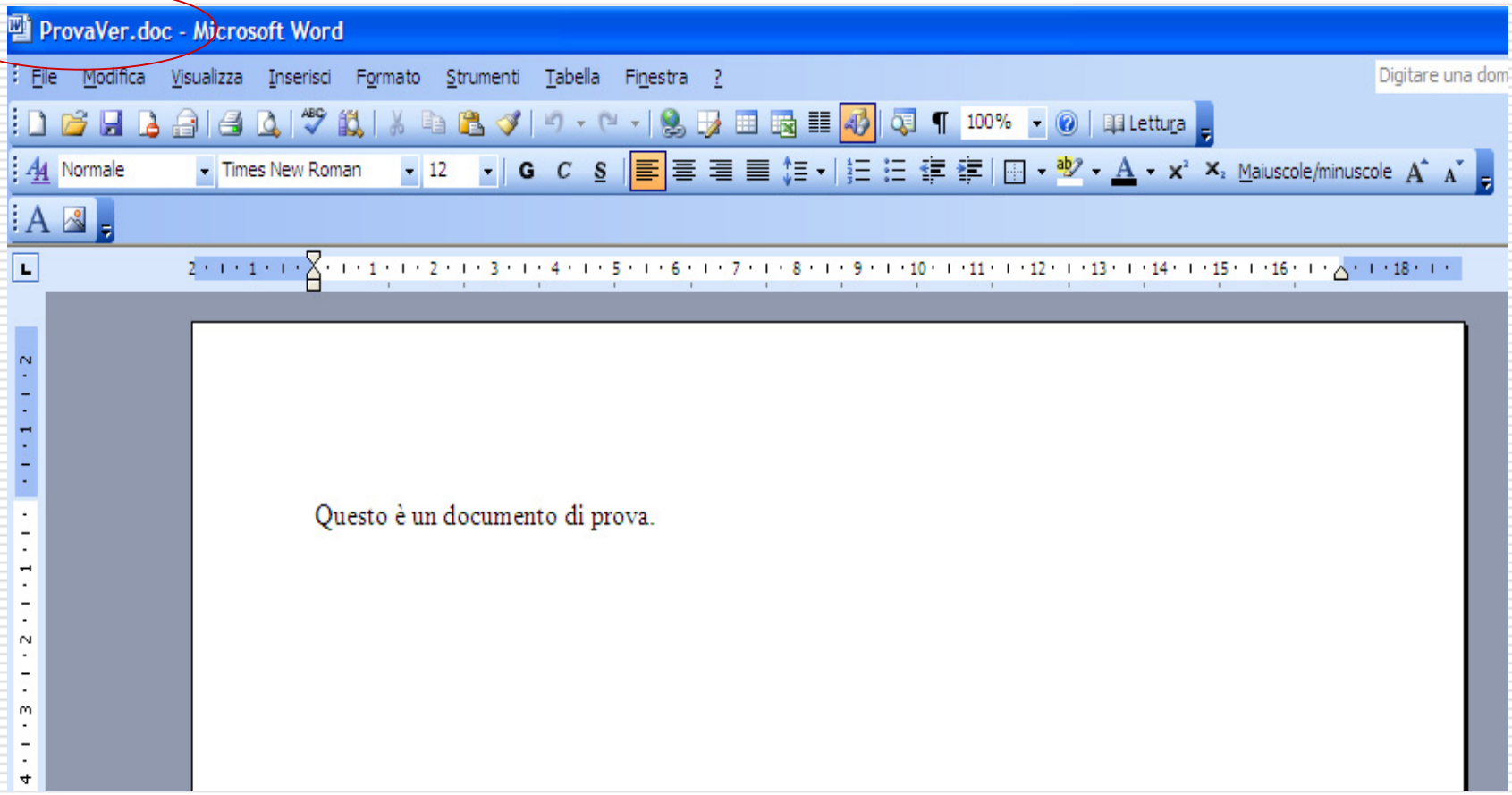
Dimostrazione pratica - Verifica



Dimostrazione pratica - Verifica



Dimostrazione pratica - Verifica



GRAZIE PER L'ATTENZIONE