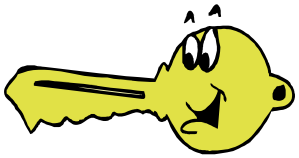
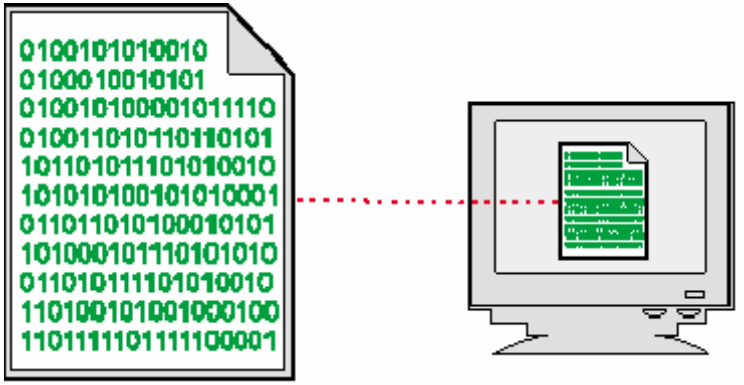
LA FIRMA DIGITALE

Materiale a cura di:

L.Chiumiento,F. Orfei, F. Baiani, S. Sallam

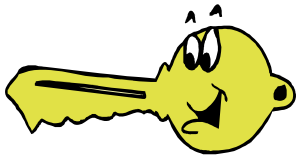


Il “documento informatico” (1)



La legge definisce “documento informatico la rappresentazione Informatica di atti, fatti o dati giuridicamente rilevanti” (DPR 513/97, art. 1/1).

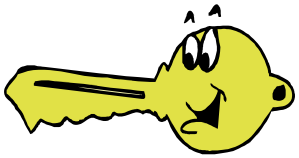
Un documento informatico è una sequenza di caratteri binari (in genere n bit) e può contenere testo, immagini, sequenze audio e video (Es. File word, excel, immagine jpg…).



Il documento informatico (2)

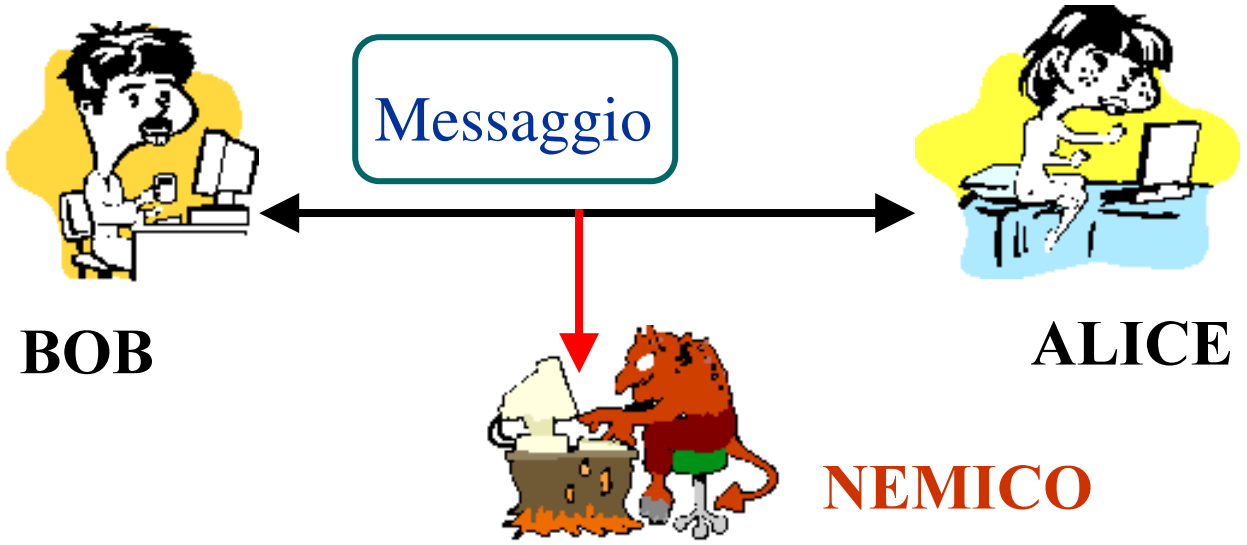
- Servizi offerti :

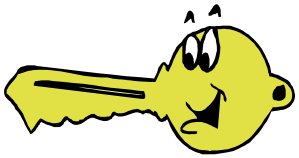
  - *Servizi su rete ;*
  - *Gestione della vita del documento ;*
  - *Trasmissioni più veloci.*



Il documento informatico (3)

- Problemi :
  - *Autenticazione ;*
  - *Integrità del documento ;*
  - *Facilmente riproducibile e modificabile ;*



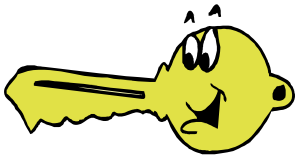


Algoritmi di crittografia

Algoritmi matematici in grado di trasformare (cifrare) reversibilmente un insieme di dati (nel nostro caso un documento) per renderlo non intelligibile.

Classificati in:

- SIMMETRICI — A chiave privata
- ASIMMETRICI. — A chiave pubblica



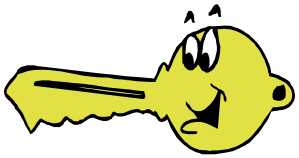
Algoritmi a chiave simmetrica

- Difetti:

➢ *Tenere segreta la chiave ;*

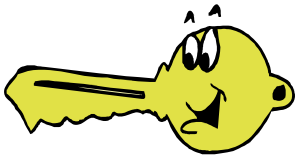
➢ *Impossibile decriptare senza avere la chiave ;*

➢ *Per N persone servono N(N-1)/2 chiavi segrete, una per ogni coppia di persone.*

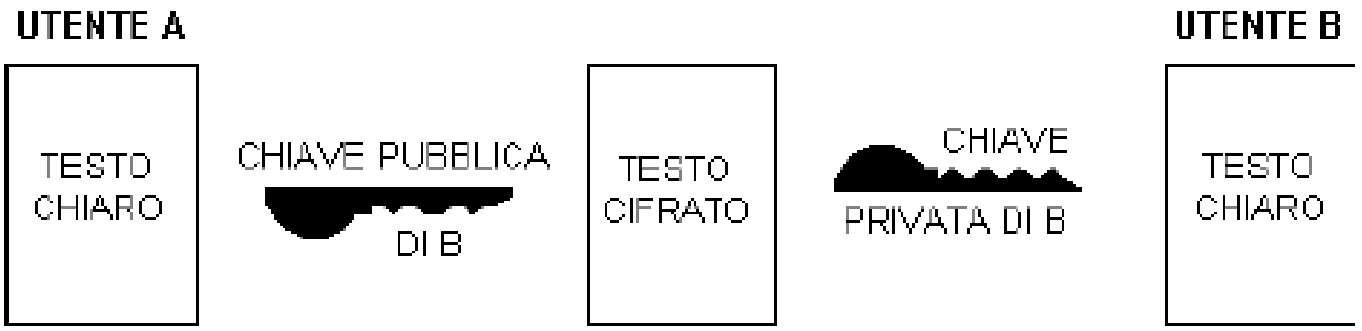


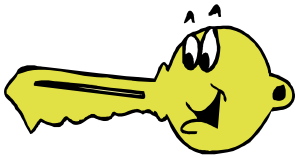
Algoritmi a chiave asimmetrica

- Non si può decifrare il testo con la stessa chiave usata per cifrarlo ;
- Le due chiavi sono generate con la stessa procedura e correlate univocamente ;
- Conoscendo una delle due chiavi non si può ricostruire l'altra.



Alcuni esempi



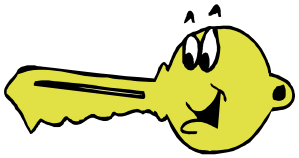


Cos'è la firma digitale

Tecnica che utilizza un algoritmo crittografico asimmetrico eseguibile da un computer e che offre integrità, autenticazione e non ripudio di un documento informatico.

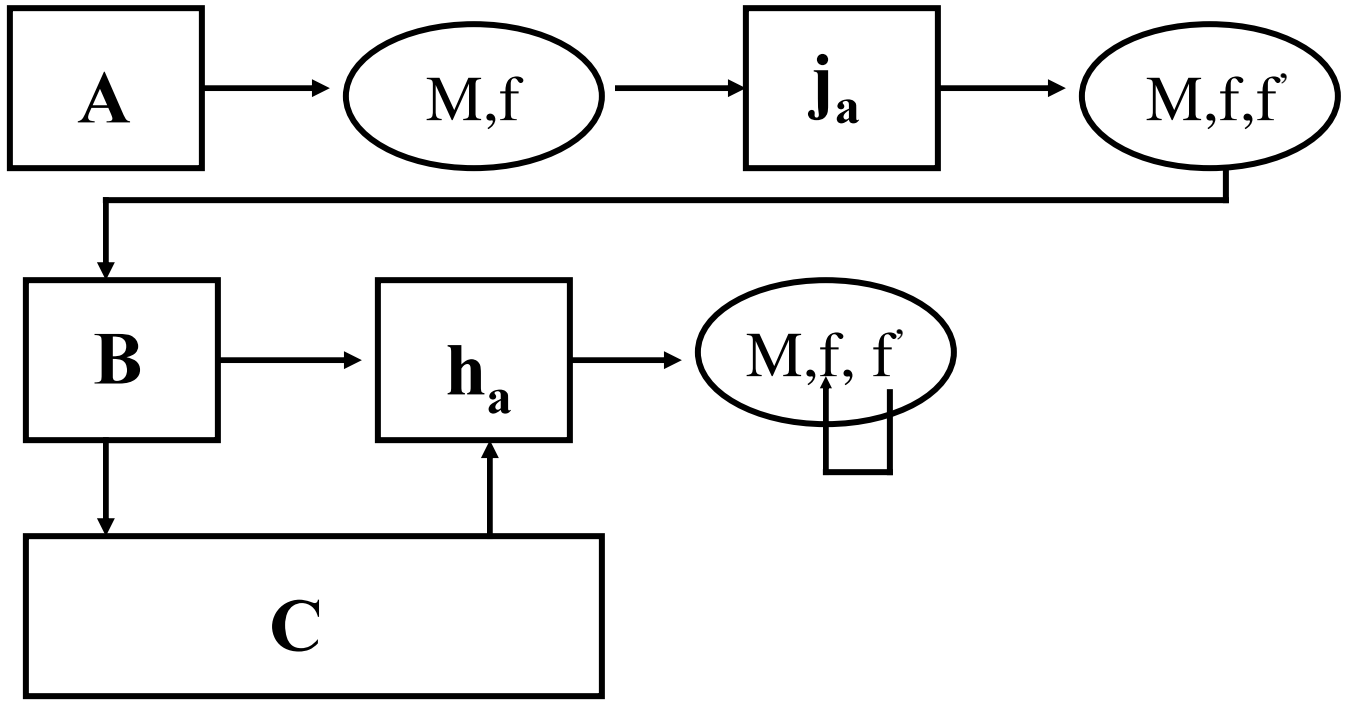
Il processo di firma digitale è composto da tre fasi :

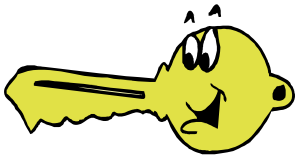
1. *Generazione dell'impronta digitale (digest message).*
2. *Generazione della firma.*
3. *Apposizione della firma.*



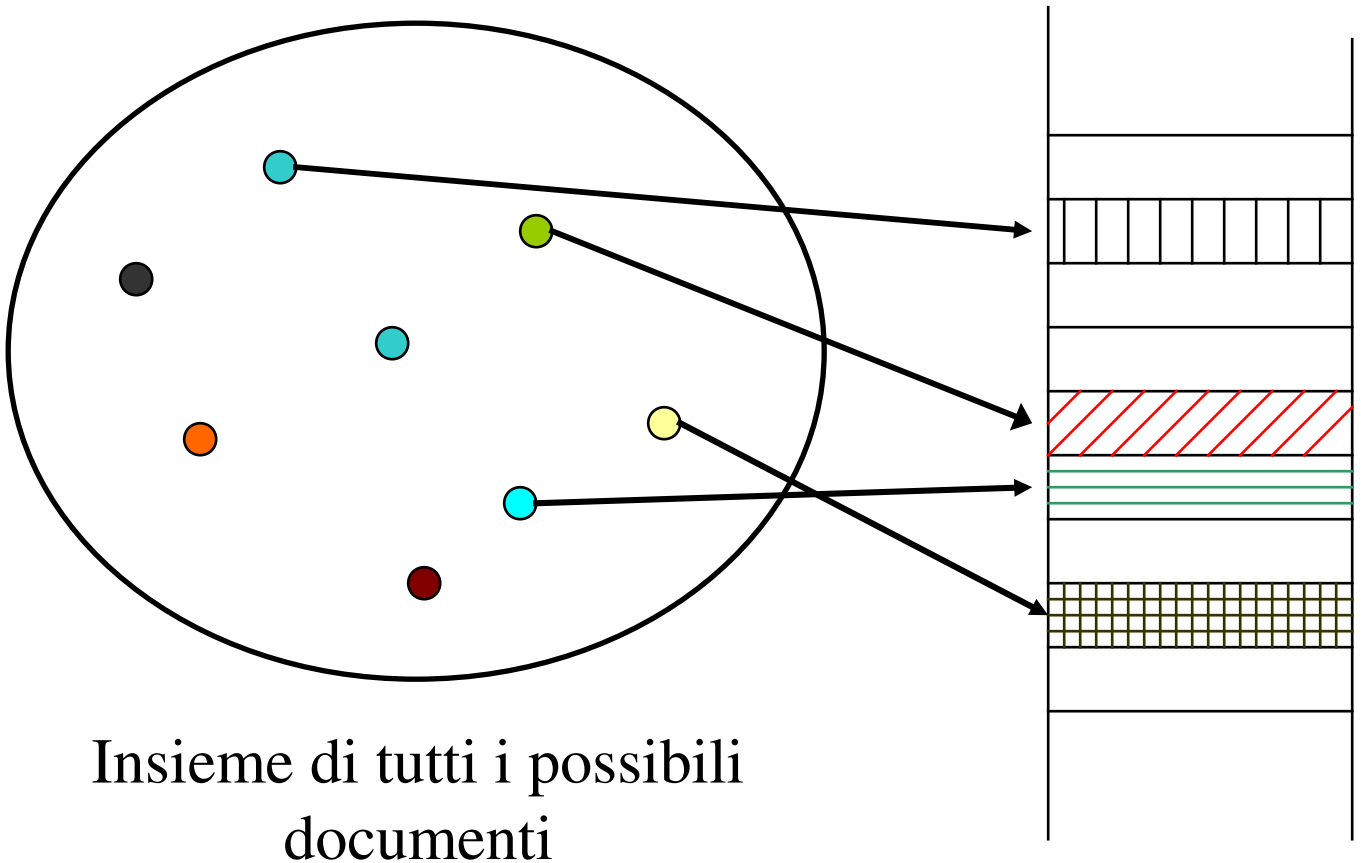
La firma digitale “debole”

Sistema di firma non contemplato nel nostro ordinamento giuridico, perché assicura solo la provenienza del documento ma non l’integrità del contenuto.



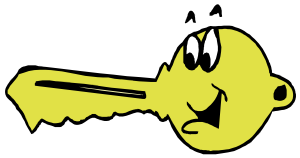


Generazione dell'impronta



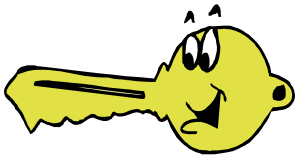
Insieme di tutti i possibili documenti

Insieme degli interi n bit

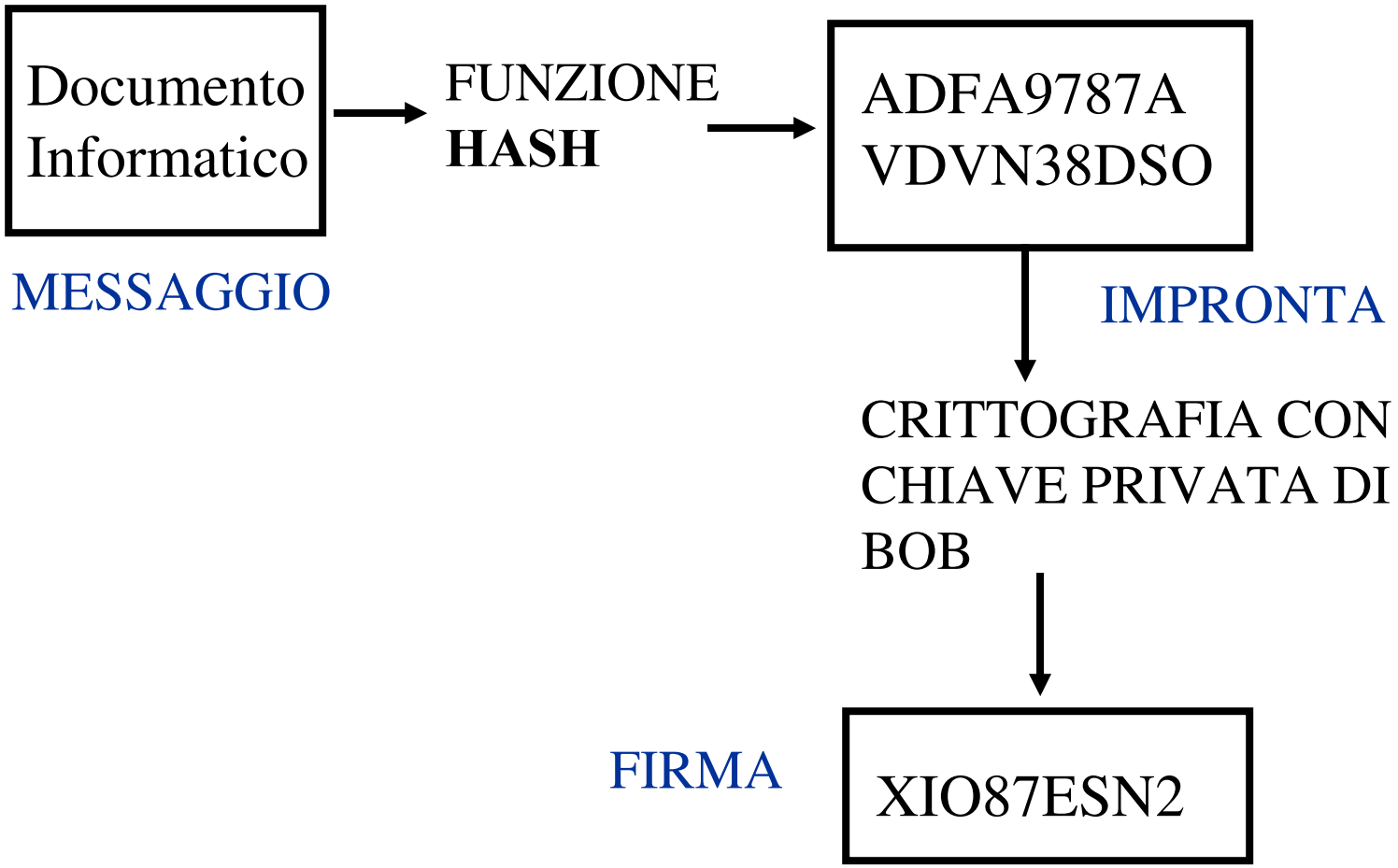


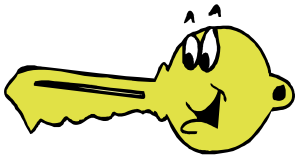
Considerazioni sull'impronta

- Si evita di applicare l'algoritmo di cifratura all'intero testo
- Una terza parte fidata riesce ad autenticare la sottoscrizione del documento.
- Basta modificare un solo bit del documento per generare un'impronta differente.
- Data un'impronta è quasi impossibile costruire un documento che la generi.

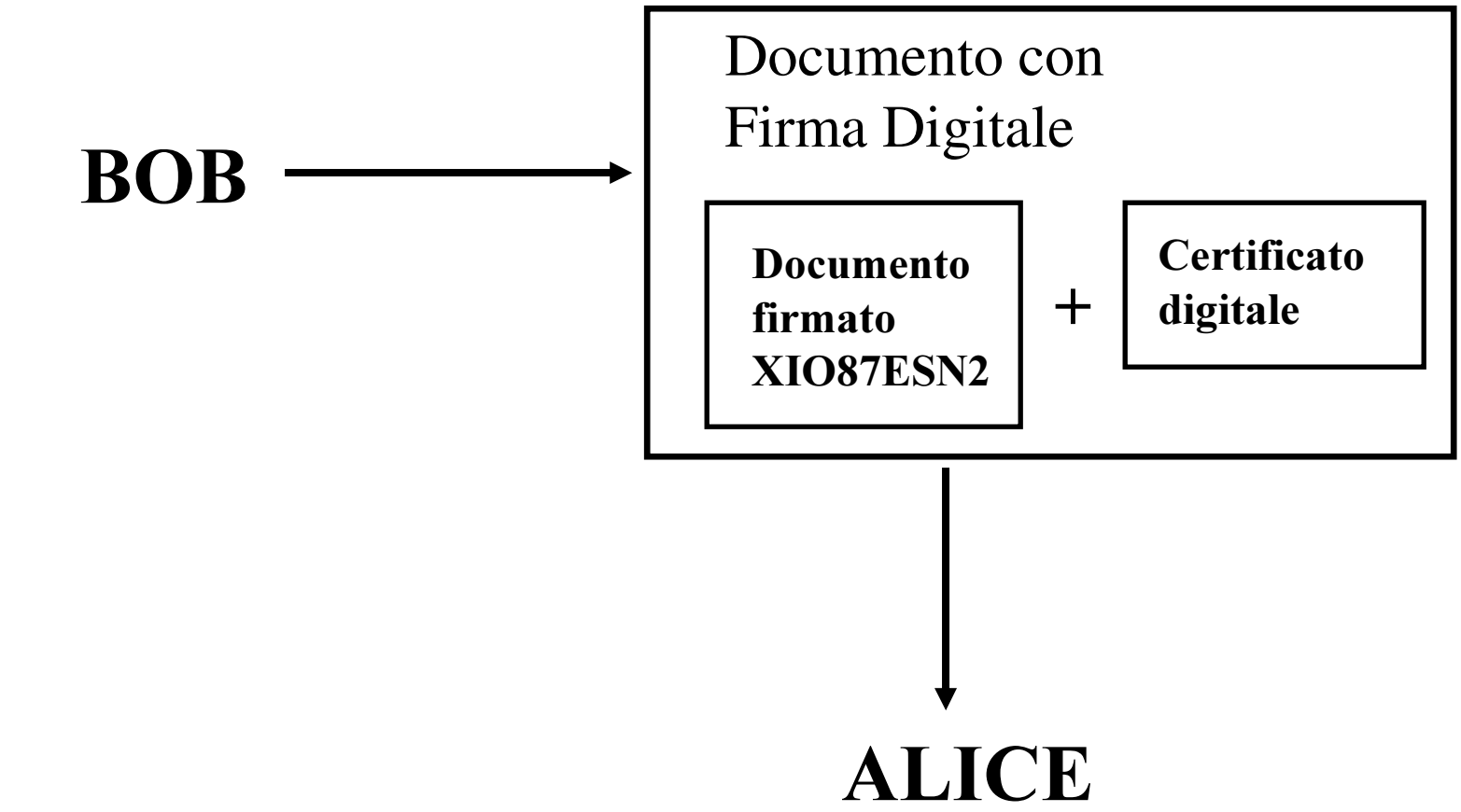


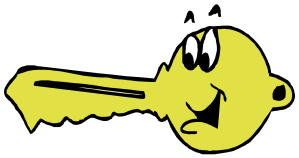
Generazione della firma



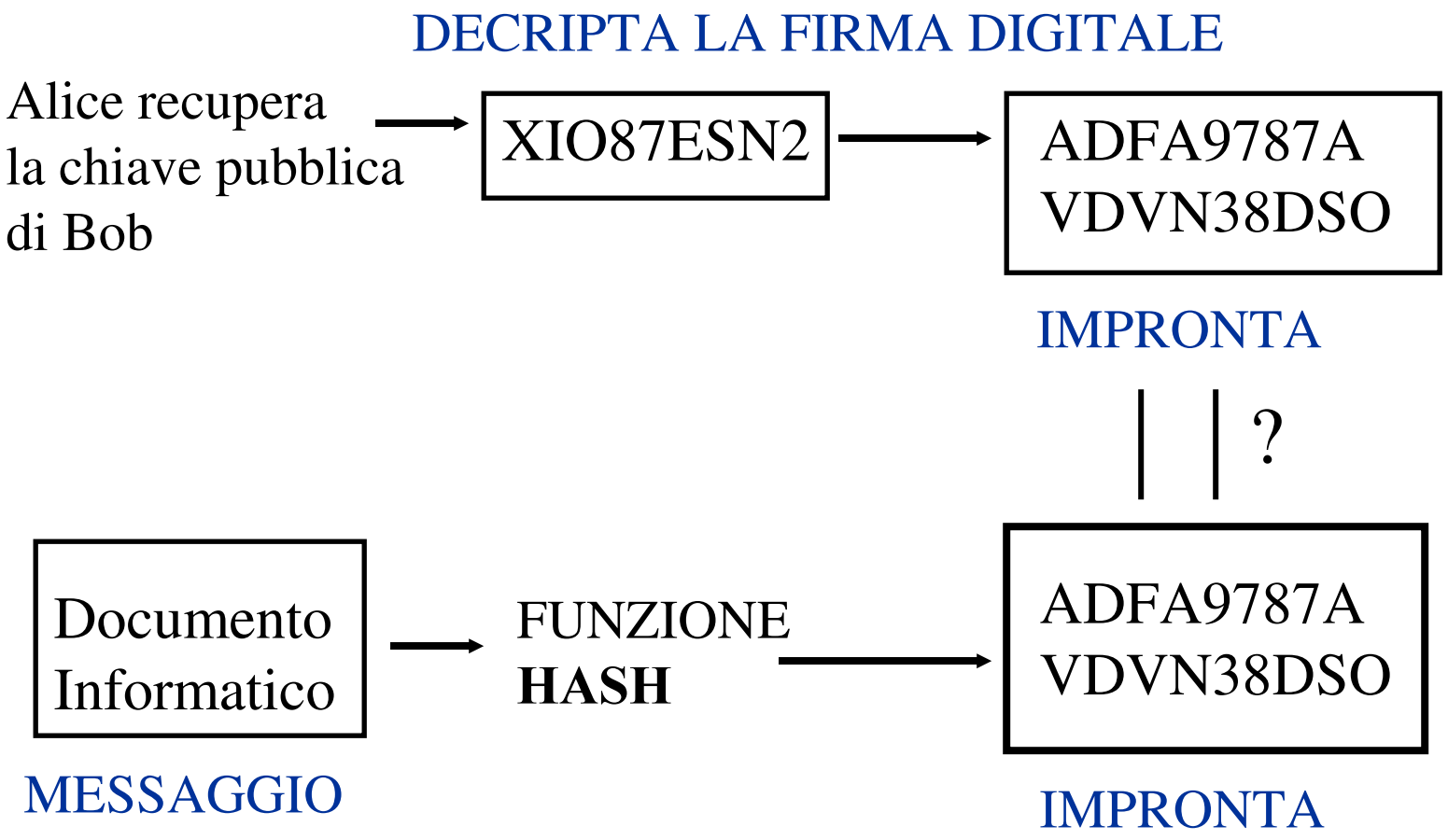


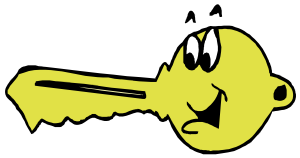
Invio del documento





Verifica dell'impronta



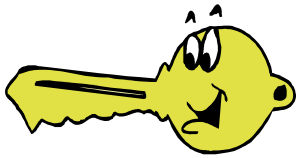


I certificati a chiave pubblica

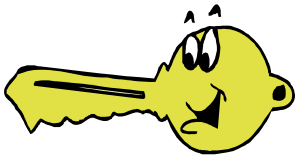
Strumenti affidabili (emessi da un **CA**) con i quali vengono distribuite le chiavi pubbliche e rese note agli utenti finali, garantendo autenticità ed integrità.
Composti almeno da :

- *Generalità del possessore della chiave pubblica.*
- *Valore della chiave pubblica.*
- *Validità temporale del certificato.*
- *La firma digitale del CA.*

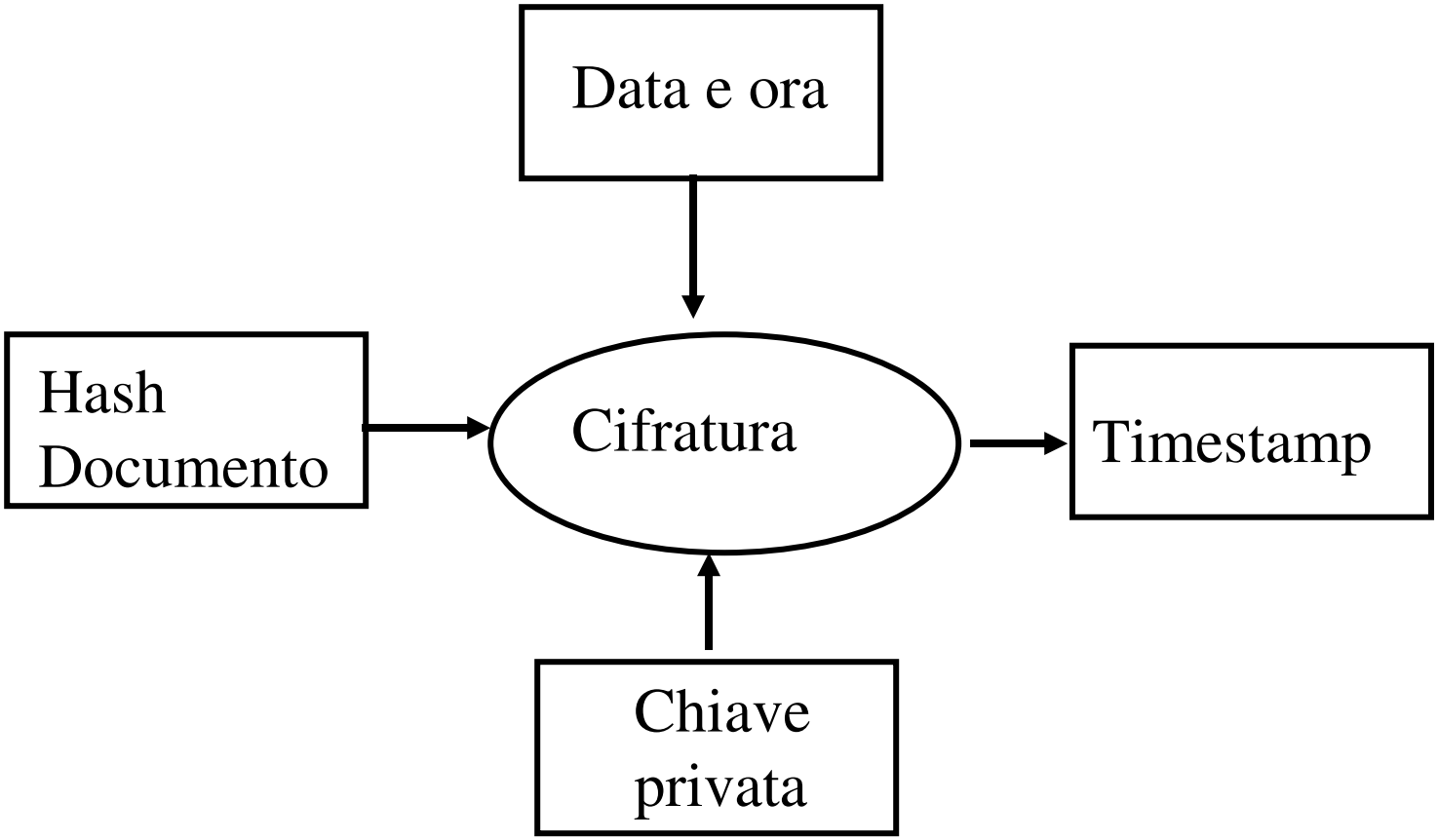
Marcatura temporale (TIMESTAMP)

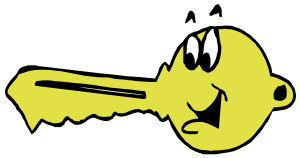


- Risponde all'esigenza di certificare data e ora di redazione/pubblicazione.
- Viene effettuata da un'autorità di certificazione.
- Data e ora più hash del documento cifrati con la chiave privata del CA.



Generazione della marca temporale



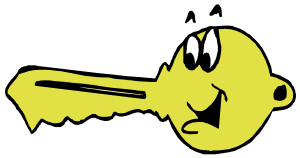


Algoritmi di cifratura e di hash

- Algoritmi di cifratura a chiave pubblica
  - Diffie-Hellman
  - RSA
  - ElGamal

- Algoritmi di hash (message digest)
  - MD5
  - SHA

Algoritmo di Diffie – Hellman

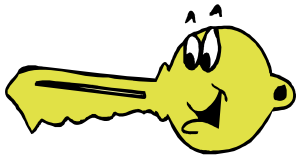


- Nato nel 1976 per risolvere il problema dello scambio di chiavi in un sistema simmetrico di cifratura (DES), pone le basi per la crittografia a chiave pubblica

- Si basa sul seguente protocollo:

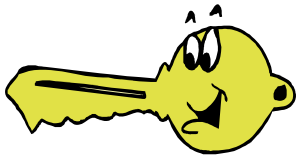
1. A e B scelgono pubblicamente un insieme di numeri **G=[0,N-1]** ed un suo elemento ***s***
2. A sceglie un numero ***a*** in G, calcola s^a e lo invia a B
3. B sceglie un numero ***b*** in G, calcola s^b e lo invia a A
4. A, ricevuto s^b , *calcola* $K = (s^b)^a$
5. B, ricevuto s^a , *calcola* $K = (s^a)^b$

- A e B possiedono la chiave segreta K, la forzatura della quale richiede la risoluzione del problema difficile del logaritmo discreto. Sul canale insicuro sono infatti transitati soltanto *N, s,* s^a, s^b



RSA (Rivest – Shamir - Adleman)

- Nato nel 1978 è il primo algoritmo a chiave pubblica
- La sua sicurezza si basa sulla difficoltà del problema della fattorizzazione di interi di grandi dimensioni
- Per utilizzarlo ciascun interlocutore deve compiere le seguenti azioni:
  1. Scegliere 2 interi ***p*** e ***q*** primi
  2. Calcolare ***N=p*q***
  3. Scegliere un intero ***e*** primo rispetto a ***Φ(N)=(p-1)(q-1)***
  4. Calcolare l'intero ***d*** per il quale risulta ***e*d mod Φ(N)=1, d=*** K_{priv}
  5. Rendere pubblica la coppia ***(e, N) =*** K_{pub}



RSA (Rivest – Shamir - Adleman)

Il messaggio cifrato x si ottiene dal testo in chiaro m attraverso la funzione di cifratura

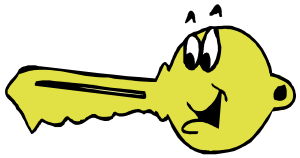
$$x = C(m, K_{pub}) = m^e \bmod N$$

La decifratura invece avviene secondo la relazione

$$m = D(x, K_{priv}) = x^d \bmod N$$

$$\downarrow$$

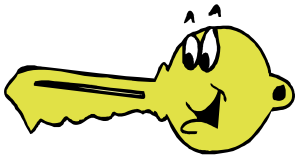
$$x^d \bmod N = (m^e \bmod N)^d = m^{ed} \bmod N = m$$



Curiosità e considerazioni

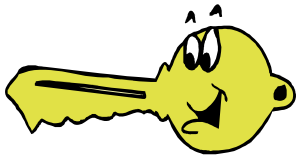
- Un docente dell'università dell'Indiana è riuscito a fattorizzare un numero con 167 cifre in un tempo pari a 100 mila ore di tempo computer. Questo fa riflettere sulla sicurezza dell' RSA che, con l'aumento della potenza di calcolo dei moderni calcolatori e la scoperta di nuovi algoritmi, va lentamente scemando

- Inoltre una chiave a 1024 bit di un sistema a chiave pubblica è paragonabile in termini di sicurezza ad una chiave a 64 bit di un sistema simmetrico



Algoritmo di ElGamal

- Proposto nel 1985 opera nel seguente modo:
  1. Si scelgono gli interi ***N*** ed ***s***, con ***1≤s≤N-1***, resi pubblici
  2. Si sceglie un intero ***a*** utilizzato come chiave privata
  3. Si calcola s^a e lo si utilizza come chiave pubblica
- La cifratura di un messaggio inviato da A a B avviene mediante le seguenti operazioni:
  1. A sceglie un intero ***h*** e calcola s^h
  2. Calcola $(K_{pub}^B)^h = s^{bh}$ (b= chiave privata di B)
  3. Calcola $x = m * s^{bh}$
  4. Invia come messaggio cifrato la coppia (s^h, x)
  5. B riceve la coppia (s^h, x)
  6. Calcola s^{bh} e recupera **m**
- La sicurezza di questo algoritmo si basa ancora una volta sulla complessità del calcolo del logaritmo discreto

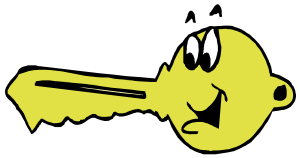


Funzioni Hash

Una **funzione di hash** (o message digest) è una funzione H che, dato un input *M* di dimensione qualsiasi, produce un output *h* (l'hash) di **dimensione fissa**:

$$h = H(M)$$

Lo scopo di una funzione di hash è di creare una **impronta** del messaggio. Dato che l'output ha lunghezza fissa (in genere dell'ordine di una o poche centinaia di bit), esisteranno necessariamente diversi messaggi che generano lo stesso hash.



Unidirezionalità

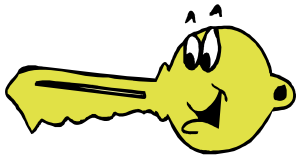
Una funzione di hash sicura deve essere **unidirezionale** (one-way), cioè:

M= messaggio H= funzione hash

1) M ⟹ H(M) facile
2) H(M) ⟹ M difficile

L'idea è che l'hash di un messaggio sia in sostanza una corrispondenza pseudocasuale, per cui non è possibile prevedere in alcun modo il risultato se non calcolando l'hash stesso.

Messaggi diversi, anche per un solo bit, devono generare hash **non correlati**



Assenza di collisioni (1)

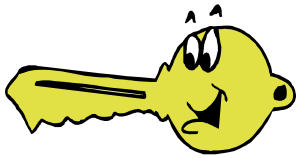
Una funzione di hash sicura deve essere **senza collisioni** (collision-resistant):

deve essere computazionalmente impraticabile trovare

Dato M, M′≠ M t.c. H(*M′*) = H(*M*)
(funzione **debolmente senza collisioni**);

(*M*, *M′*), con *M′≠ M*, t.c H(*M*) = H(*M′*)
(funzione **fortemente senza collisioni**).

L'idea è che, anche se necessariamente esistono messaggi diversi che producono lo stesso hash, non è praticabile trovarli.

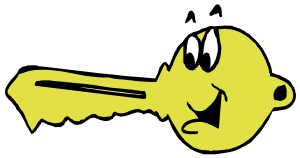


Assenza di collisioni (2)

Le due proprietà sull'assenza di collisioni (debole e forte), corrispondono a due diversi tipi di attacchi:

- **attacco a forza bruta "semplice"**:
  trovare un messaggio che produca un dato hash (cioè un hash uguale a quello di un messaggio dato);
- **attacco del compleanno** (birthday attack):
  trovare due messaggi che producano lo stesso hash, indipendentemente dal valore di questo hash.

La complessità dei due attacchi è molto diversa!
Se l'hash è lungo m bit, la complessità del primo è 2^m, quella del secondo è $2^{m/2}$.

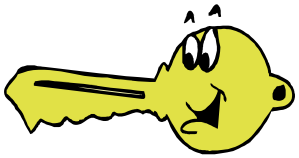


Che cosa NON è un hash

Un hash **non è né una firma né un MAC**, poiché di per sè non garantisce autenticazione e integrità.

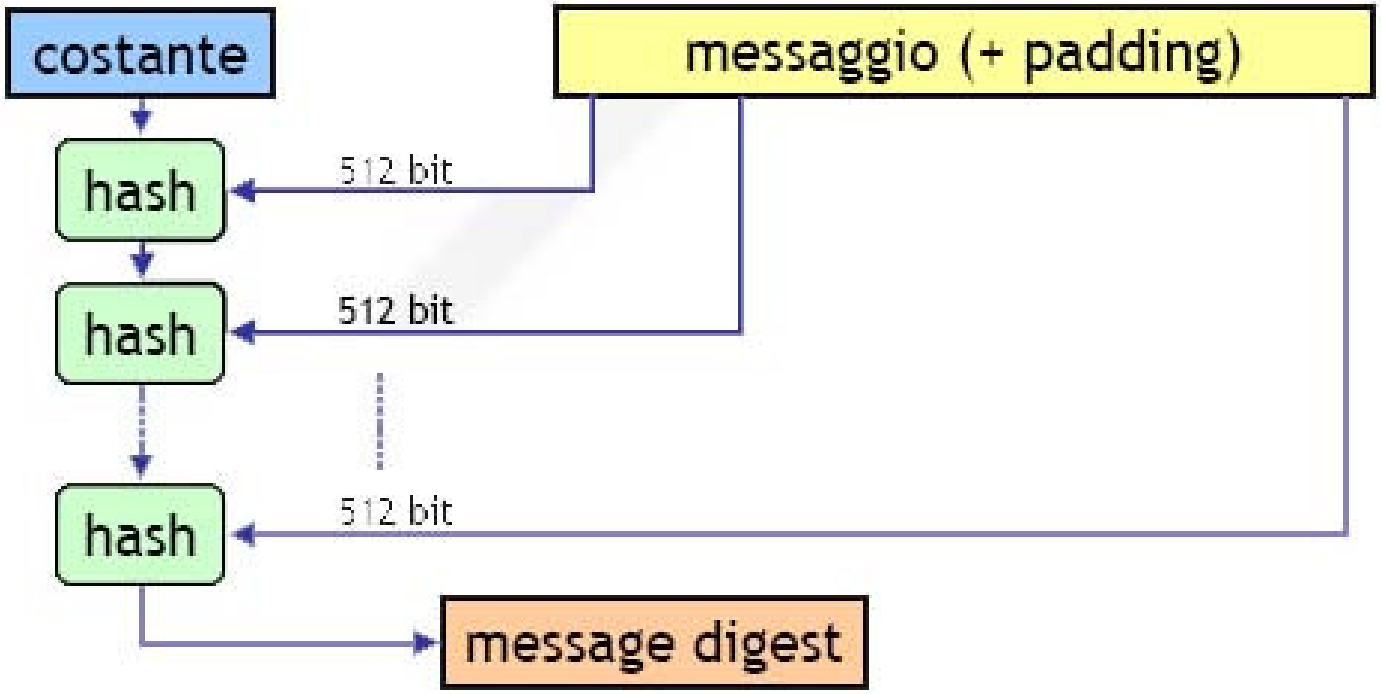
Nel calcolo di un hash non si inserisce nessuna informazione segreta, per cui chiunque può generare l'hash corretto di qualunque messaggio.

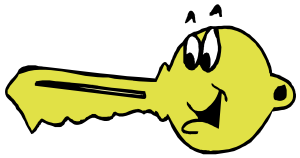
Una funzione di hash **non è un algoritmo di cifratura.** Calcolare l'hash di un messaggio non equivale a cifrarlo: il calcolo di un hash non include nessuna chiave, e soprattutto è un'operazione **non invertibile**.



MD5 e SHA-1

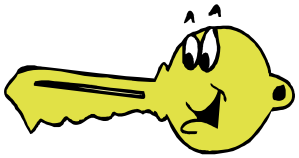
- Le due funzione di hash sicuro attualmente più in uso sono MD5 e SHA-1 ed hanno una struttura simile





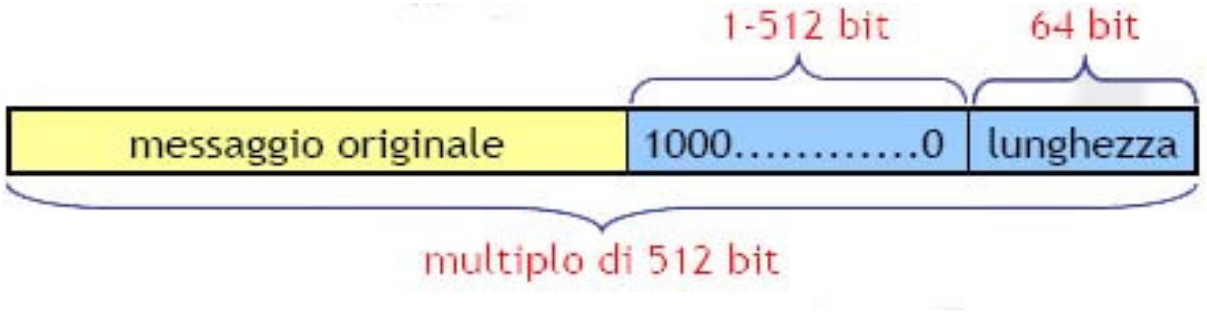
MD5

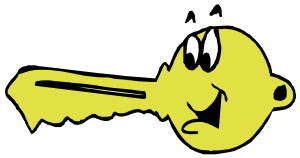
- **MD5** (Message Digest 5) è stato progettato da Ron Rivest, ed è definito nell' RFC 1321 (1992).
- Il messaggio viene elaborato a blocchi di 512 bit.
- L'hash è di **128 bit**.
- Ad ogni iterazione si calcola una funzione che prende in ingresso il blocco corrente del messaggio e il valore dell'hash all'iterazione precedente.
- L'hash finale è quello risultante dall'ultima iterazione.
- Non è più considerato molto sicuro (128 bit di hash non sono molti).



MD5: padding

- Prima di iniziare l'elaborazione, si aggiunge al messaggio un **padding** in modo che la lunghezza totale risulti un multiplo di 512 bit:
  - si aggiunge un bit a 1 e poi tanti bit a 0 quanto basta perché la lunghezza risulti di 64 bit minore rispetto a un multiplo di 512 bit (se la lunghezza originale è già corretta si aggiungono comunque 512 bit);
  - si aggiungono 64 bit contenenti la lunghezza originale del messaggio (modulo 2^{64}).





SHA-1 (Secure Hash Algorithm 1)

- **SHA-1** è stato progettato dal NIST (l'algoritmo SHA originale è stato sostituito con SHA-1 per via di una vulnerabilità non pubblicata).
- Il messaggio (che deve essere di lunghezza inferiore a 2^{64} bit) viene elaborato a blocchi di 512 bit.
- L'hash è di **160 bit**.
- Ad ogni iterazione si calcola una funzione che dipende dal blocco corrente del messaggio e dal valore dell'hash all'iterazione precedente; il risultato viene sommato al valore dell'iterazione precedente.

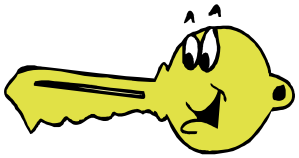
MD5 vs SHA-1

	MD5	SHA-1
lunghezza hash	128 bit	160 bit
lunghezza massima messaggio	illimitata	2^{64} -1 bit
dimensione blocco messsaggio	512 bit	512 bit
numero operazioni	64	80
numero costanti additive	64	4
numero funzioni primitive	4	4 (di cui 2 uguali)

Uso degli algoritmi cifrati asimmetrici per la firma digitale

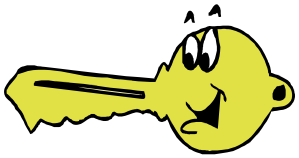
o Metodi Crittografici a chiavi pubbliche utilizzati per costruire strumenti per la firma digitale …. Vediamone alcune caratteristiche:

- Ruolo delle chiavi nella firma di un messaggio
- Necessità di un "autorità di certificazione" … garantisce certezza identità autore di un messaggio e non ripudiabilità
- Esistono algoritmi di cifratura e decifratura specificamente adatti per la generazione e la verifica della firma



Quali algoritmi vedremo….

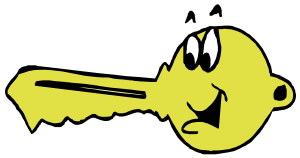
- L'algoritmo RSA (Rivest Shamir Adleman)
- L'algoritmo di ElGamal
- L'algoritmo di Schnorr
- L'algoritmo DSS (Digital Signature Standard)



Caratteristiche dell'algoritmo RSA …

- Si basa sull'inversione del ruolo delle chiavi rispetto a quello utilizzato per assicurare la riservatezza di un messaggio

- Intero testo non viene cifrato …. Viene "compresso" in una sorta di "riassunto" … l'impronta digitale!!!



… e il suo protocollo

A → (Messaggio = m) → B

$K_{a\ priv} = d_a$
$K_{a\ pub} = (n_a, e_a)$

$K_{b\ priv} = d_b$
$K_{b\ pub} = (n_b, e_b)$

- A vuole firmare il messaggio da mandare a B per far in modo che B sia sicuro che il messaggio che riceve è stato effettivamente mandato da A.

Per prima cosa A firma il messaggio utilizzando la funzione di decifratura

$$\mathbf{f = D(m,K_{a\ priv}) = m^{da} (mod\ n_a)}$$

e poi cifra il messsaggio utilizzando la funzione di cifratura

$$\mathbf{f' = C(f, K_{bpub}) = f^{eb} (mod\ n_b)}$$

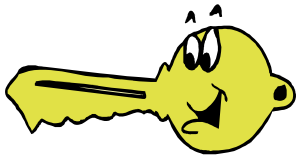
- A spedisce (A, f')

- B riceve il messaggio e per prima cosa decifra utilizzando la funzione di decifratura

$$\mathbf{D(f',K_{b\ priv}) = f'^{db} (mod\ n_b) = f}$$

e poi verifica la firma utilizzando la funzione di cifratura

$$\mathbf{C(f, K_{a\ pub}) = f'^{ea} (mod\ n_a) = m}$$



Il protocollo dell'algoritmo di ElGamal …

A —— Messaggio = M ——→ B

o Per firmare il messaggio A deve effettuare le seguenti operazioni:

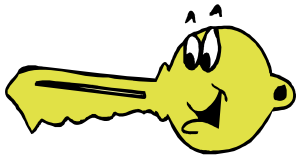
- Generare un intero h nell'intervallo [0, N-1] che sia primo rispetto N-1
- Calcolare $u = s^h \bmod N$
- Risolvere rispetto a v la relazione di congruenza

  $M = K_{apriv} u + h\, v \mod (N-1)$
- La firma del Messaggio M è la coppia f= (u,v)

o A invia (A,M,f)

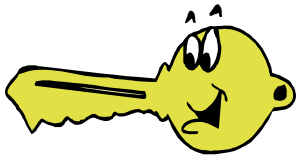
o B riceve il messaggio e per verificare la firma utilizza questa relazione:

$$s^M = K_{apub}{}^u \, u^v \bmod N$$



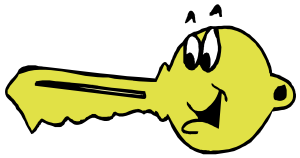
… e le sue caratteristiche

- E’ abbastanza diverso da quello già visto per la cifratura
- La sicurezza del sistema è basata sul fatto che la determinazione della coppia (u,v) richiede la soluzione del problema del logaritmo discreto nel caso in cui si fissi u e si cercasse di determinare v di conseguenza; nel caso opposto, ossia fissando v e cercando di calcolare u, si incappa in una congruenza esponenziale mista per la quale non si conoscono algoritmi efficienti di soluzione



Caratteristiche dell'algoritmo di Schnorr ...

- Metodo di generazione della firma simile a quello di ElGamal ... l'unica differenza è l'introduzione di una funzione di hash che associa ad ogni messaggio e ad ogni chiave un intero in un intervallo di ampiezza predefinita T
- Anche la sua sicurezza è legata alla complessità di soluzione del problema del calcolo del logaritmo discreto
- Rispetto all'algoritmo di ElGamal ha il vantaggio di poter scegliere la dimensione della firma selezionando opportunamente l'ampiezza T del codominio della funzione di hash

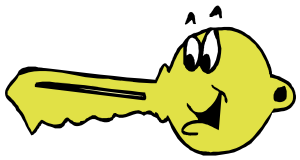


... e il suo protocollo

A —— Messaggio = M ——→ B

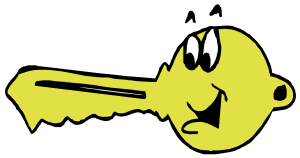
- Per generare la firma A deve effettuare le seguenti operazioni:
  - Generare casualmente un intero h nell'intervallo [0, N-1]
  - Calcolare $u = s^h \bmod N$
  - Calcolare il valore della funzione di Hash corrispondente ad M e u

    $e = H(M,u)$
  - Calcolare $v = K_{apriv} e + h \bmod N$
  - La firma di M e la coppia f=(v,e)
- B riceve il messaggio e per verificare la firma deve:
  - Calcolare s^v
  - Calcolare K_{apub}

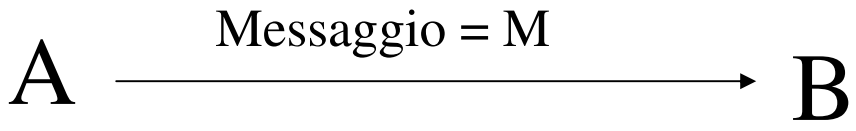


Caratteristiche dell'algoritmo DSS ...

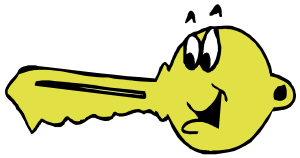
- Proposto da NIST (National Institute of Standard and Technology) nel 1991
- E' una variante del metodo di Schnorr in cui la funzione di Hash H ha come unico argomento M e quindi il suo valore non dipende dalla chiave di cifratura
- Il protocollo è più complesso rispetto ai precedenti in quanto prima di poter firmare un messaggio è necessario generarsi le chiavi



...e il suo protocollo



- Procedura di generazione delle chiavi:
  - Scegliere un intero P, da usare nelle operazioni di modulo, che deve essere un numero primo compreso tra 2^{511} e 2^{512}
  - Scegliere un intero Q, divisore primo di P-1, compreso tra 2^{159} e 2^{160}
  - Scegliere un intero G nell'intervallo [0, P-1]
  - Scegliere un intero x nell'intervallo (0,Q), che costituisce la chiave privata K_{apriv}
  - Calcolare l'intero $y = G^x$ utilizzato come chiave pubblica K_{apub}



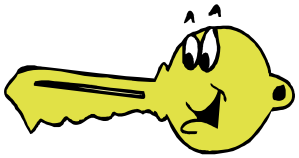
...ancora il protocollo del DSS

- Una volta ottenute le chiavi, per ottenere la firma del messaggio bisogna applicare le seguenti operazioni
  - Scegliere casulamente un intero h compreso nell'intervallo (0,Q)
  - Calcolare l'intero $u = (G^h \bmod P) \bmod Q$
  - Determinare il valore di v risolvendo la seguente relazione:
    $$H(M) = K_{apriv}\, u + hv \pmod Q$$
  - La firma di M è la coppia f=(u,v)
- Quando B riceve il messaggio verifica la firma applicando le seguenti operazioni:
  - Determinare il valore di w tale che wv=1
  - Calcolare i=H(M) w mod Q
  - Calcolare l=uw mod Q
  - Calcolare $t = ((G^i K_p) \bmod P) \bmod Q$
  - Verificare che t=u

Decreto del presidente del consiglio dei ministri 13 Gennaio 2004

Il presente decreto stabilisce:

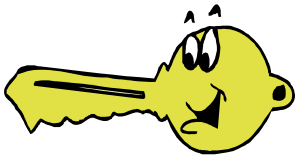
- Le regole tecniche per la generazione,apposizione e verifica delle firme digitali.
- Le disposizioni che si applicano ai certificatori che rilasciano al pubblico certificati qualificati.



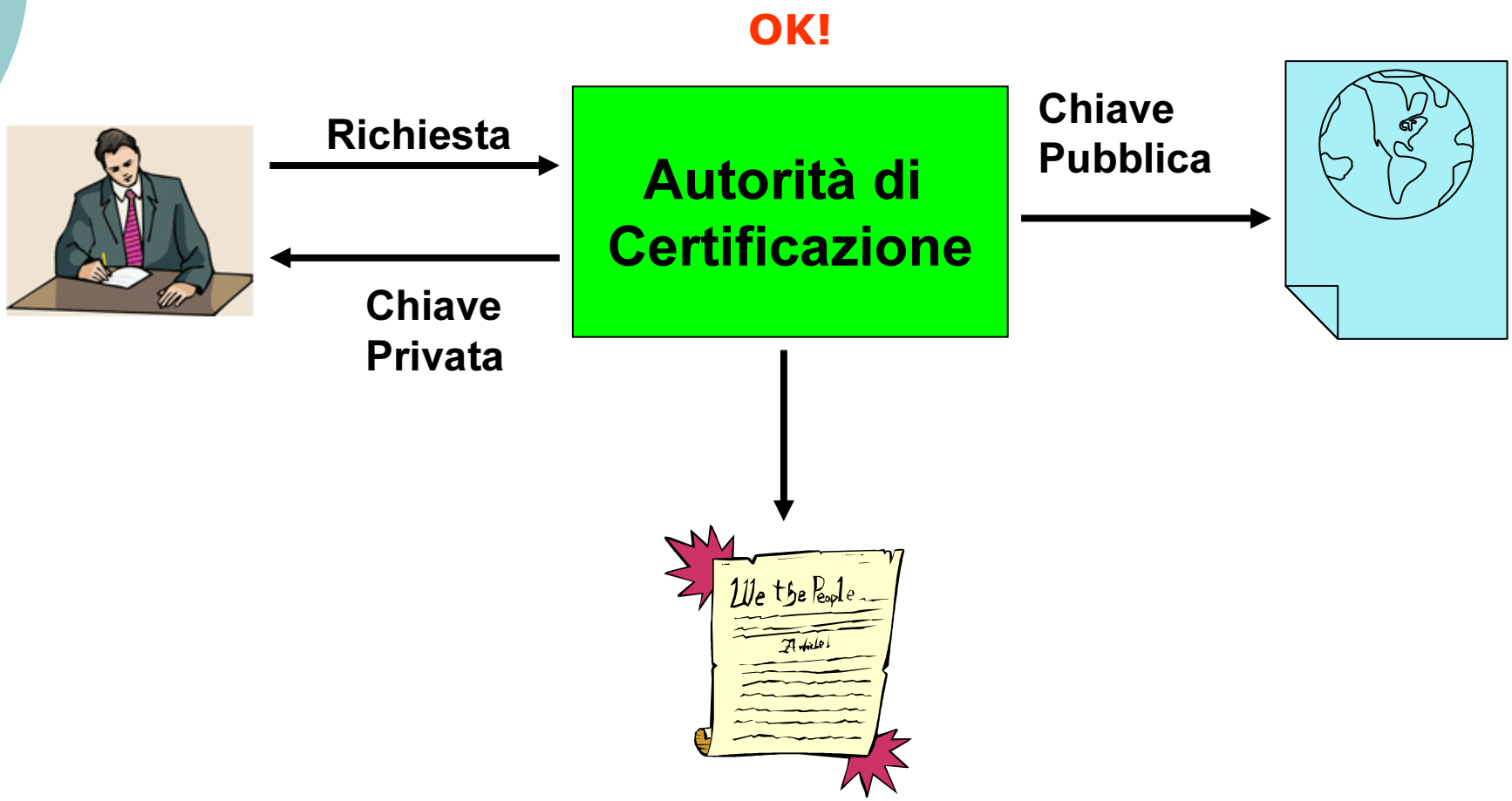
Chiavi

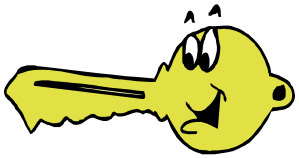
Le chiavi di creazione si distinguono in tre tipologie:

- Chiavi di sottoscrizione;
- Chiavi di certificazione;
- Chiavi di marcatura temporale.



Certificatori



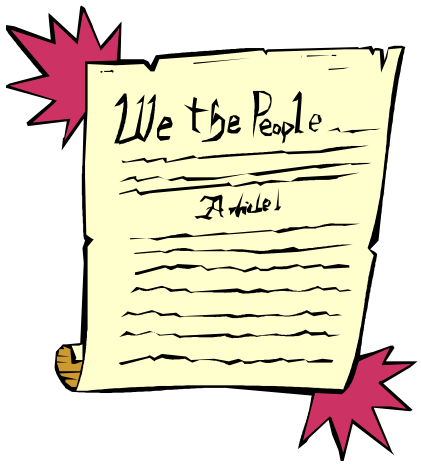


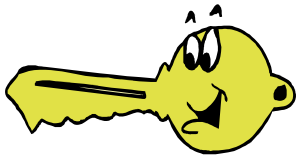
Certificato

Il certificato garantisce la corrispondenza biunivoca tra chiave pubblica,necessaria per la verifica della firma, chiave privata e soggetto titolare.

Il certificato contiene:

- generalità della persona;
- la chiave pubblica;
- termine di scadenza.



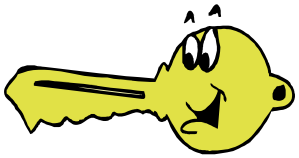


Firma digitale autenticata

L'autenticazione consiste nell'attestazione da parte di un pubblico ufficiale che la firma digitale è stata apposta in sua presenza dal titolare

L'autenticazione deve essere preceduta da :

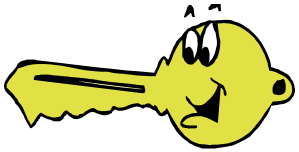
- Accertamento identità personale;
- Accertamento della validità della chiave utilizzata;
- Controllo che il documento corrisponda alla volontà della parte;
- Controllo che il documento non sia in contrasto con l'ordinamento giuridico.



Requisiti del Certificatore

I requisiti che un Certificatore deve soddisfare per essere inscritto nell'Albo :

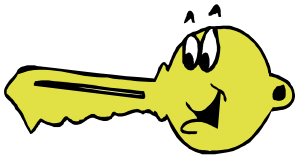
- soddisfare precise specifiche tecniche;
- curarsi della formazione di personale altamente qualificato;
- presentare strategie dettagliate per contrastare eventuali attacchi;
- attivare un sistema di elevata sicurezza e certificarlo.



Inizio attività

Per iniziare un'attività di certificatore bisogna:

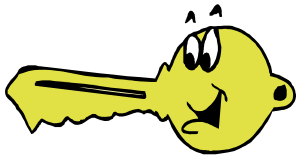
- Definire delle procedure operative;
- Realizzare delle specifiche per la realizzazione del servizio e delle procedure operative front-end e back-end.
- Attivare il sistema di qualità.



Sistema di qualità

Per sistema di qualità si intende avere:

- un manuale di qualità;
- procedure di gestione di qualità;
- integrazione all'interno del sistema di qualità di tutte le procedure operative;
- definire dei modelli per la registrazione di qualità;
- applicare il sistema di qualità e certificarlo (ISO 9002).



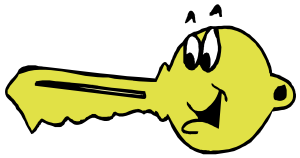
ITSEC

Il regolamento approvato dall'AIPA prescrive che la verifica dei requisiti di sicurezza delle componenti più critiche venga condotta in accordo ai criteri di valutazione di sicurezza informatica detti ITSEC.

ITSEC = Information Technology Security Evaluation Criteria.

Valutazione :

- Componenti dedicate alla generazione delle chiavi;
- Componenti dedicate alla generazione delle firme.



ITSEC

In ITSEC l'oggetto di valutazione si chiama Target Of Evaluation e viene indicato con TOE. L'insieme di specifiche rispetto al quale viene valutato il TOE si chiama Security Target.

I TOE vengono descritti a tre livelli:

- Obiettivi di sicurezza;
- Funzioni di sicurezza;
- Meccanismi di sicurezza.

Obiettivi della valutazione:

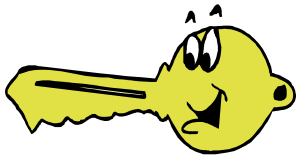
- valutare l'efficacia delle funzioni di sicurezza;
- valutare la correttezza della realizzazione delle funzioni di sicurezza.

Valutazione

Il livello di valutazione è da intendersi come misura, in senso probabilistico, delle garanzie offerte dalla valutazione circa la correttezza e l'efficacia del TOE.

Esistono 7 livelli di valutazione:

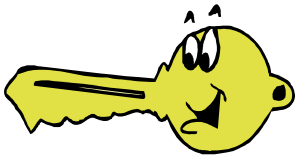
E0	**La valutazione non ha avuto successo**
E1	**Test funzionali per la verifica della rispondenza al Security Target**
E2	**Controllo del progetto e delle prove di correttezza fornite dallo sviluppatore**
E3	**Maggior rigore nel verificare i test eseguiti durante lo sviluppo e analisi del codice sorgente**
E4	**Metodi formali e semiformali da usarsi nello sviluppo del TOE**
E5	**Maggior rigore nella verifica di rispondenza al progetto e nell'analisi delle vulnerabilità**
E6	**Maggior uso di metodi formali nella progettazione**



Robustezza

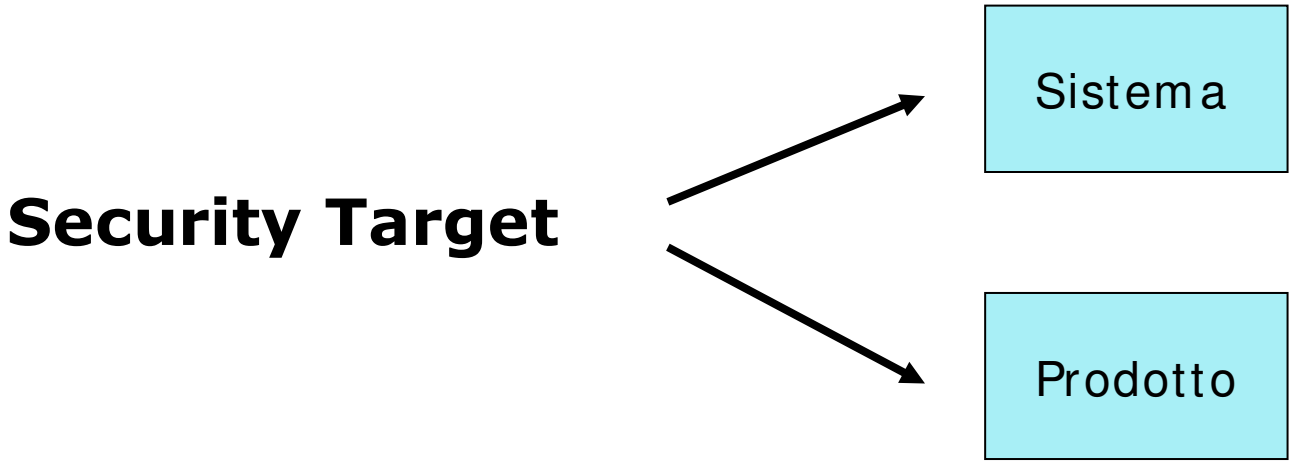
Secondo elemento di una valutazione ITSEC è il livello di robustezza dei meccanismi.

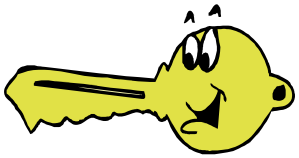
BASIC	**Il meccanismo fornisce protezione solo rispetto ad attacchi lanciati senza particolare determinazione e senza conoscenze specifiche.**
MEDIUM	**Il meccanismo è capace di resistere ad attacchi da parte di chi dispone di conoscenze e risorse limitate.**
HIGH	**Il meccanismo può essere aggirato solo da chi dispone di conoscenze e risorse al di sopra della norma.**



Security Target

Il livello di valutazione e il livello di robustezza dei meccanismi non sono da soli significativi se a questi non è affiancato il Security Target. L'analisi del Security Target può far capire all'utilizzatore del TOE se quest'ultimo è adatto o meno alle sue esigenze.

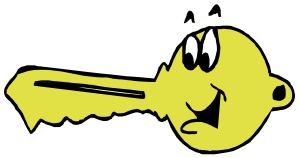




Security Target

Il Security Target di un SISTEMA è un documento contenente:

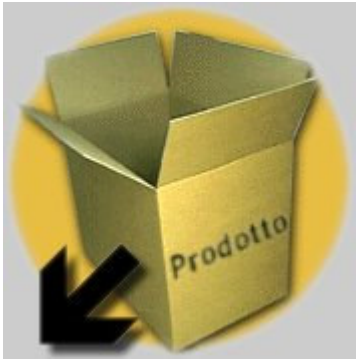
- La descrizione della politica di sicurezza;
- Obiettivi di sicurezza del TOE;
- Le minacce al quale il TOE è esposto;
- Le misure di sicurezza fisiche,procedurali e relative al personale.
- La descrizione delle funzioni di sicurezza;
- La dichiarazione del livello di robustezza dei meccanismi.



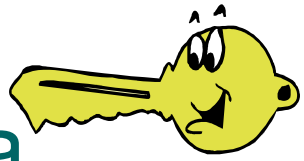
Security Target

Il Security Target di un PRODOTTO è un documento contenente:

- Giustificazione logica del prodotto (product rationale);
- Obiettivi di sicurezza del TOE;
- Le minacce al quale il TOE è esposto;
- Le misure di sicurezza fisiche,procedurali e relative al personale.
- La descrizione delle funzioni di sicurezza;
- La dichiarazione del livello di robustezza dei meccanismi.



Regole tecniche per la firma digitale

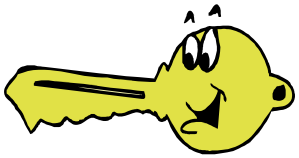


- Per componenti dedicate alla generazione delle chiavi e delle firme:

  Valutazione ITSEC a livello **E3** con robustezza dei meccanismi **High**.

- Per le componenti dedicate alla verifica delle firme e la validazione temporale:

  Valutazione ITSEC a livello **E2** con robustezza dei meccanismi **High**.



Firma digitale in Europa

- Due tipi di firma:
  - **Firma elettronica**: equivalente alla nostra firma digitale debole;
  - **Firma elettronica avanzata**: equivalente alla nostra firma digitale forte.
- L'attività di certificazione è libera.
- Entrambi le firme hanno valenza giuridica.