

# Sistemi di Crittografia

## Classificazione dei sistemi crittografici

- Tipo di operazioni usate per trasformare il testo in chiaro in testo cifrato
  - Sostituzione:
    - Ogni elemento del testo in chiaro è trasformato in un altro elemento
  - Trasposizione:
    - Gli elementi del testo in chiaro sono riorganizzati
- Numero di chiavi (distinte) utilizzate
  - Chiave singola: crittografia a chiave simmetrica (o a chiave segreta)
    - Le chiavi del mittente e del destinatario sono identiche
  - Due chiavi: crittografia a chiave asimmetrica (o a chiave pubblica)
    - La chiave di cifratura è pubblica; la chiave di decifratura è privata
- Il modo in cui il testo in chiaro è elaborato
  - Cifrario a blocchi:
    - Elabora in blocchi di dimensione fissa
  - Cifrario a flusso:
    - Elabora senza una lunghezza predefinita

# Cifrario di Cesare

## Esempio: cifrario di Cesare

- Tipo di operazioni:
  - sostituzione
- Numero di chiavi utilizzate:
  - Chiave singola ( $1 \leq k \leq 26$ )
- Modo in cui il testo in chiaro è elaborato:
  - cifrario a flusso

Plaintext letter:	a b c d e f g h i j k l m n o p q r s t u v w x y z
$C_1(k = 5)$ :	f g h i j k l m n o p q r s t u v w x y z a b c d e
$C_2(k = 19)$ :	t u v w x y z a b c d e f g h i j k l m n o p q r s

# Crittoanalisi

## Crittoanalisi

- Processo con cui si tenta di risalire al testo in chiaro o alla chiave usata
  - Diversi attacchi in base alle informazioni a disposizione dell'intruso
- Un algoritmo di cifratura è progettato per resistere a un attacco basato su testo in chiaro conosciuto
  - Per scoprire la chiave occorre provare tutte le chiavi possibili (**attacco a forza bruta**)

Un sistema di cifratura è **computazionalmente sicuro** se il testo cifrato soddisfa uno dei seguenti criteri:

- Il costo per rendere inefficace il cifrario supera il valore dell'informazione cifrata
- Il tempo richiesto per rendere inefficace il cifrario supera l'arco temporale in cui l'informazione è utile

Key Size (bits)	Number of Alternative Keys	Time required at $10^6$ Decryption/ $\mu$ s
32	$2^{32} = 4.3 \times 10^9$	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	10 hours
128	$2^{128} = 3.4 \times 10^{38}$	$5.4 \times 10^{18}$ years
168	$2^{168} = 3.7 \times 10^{50}$	$5.9 \times 10^{30}$ years

# Chiave simmetrica

## Crittografia a chiave simmetrica

- Mittente e destinatario condividono una chiave segreta
  - Come si concorda la chiave?  
(problema della distribuzione delle chiavi)

# Chiave simmetrica

## La crittografia simmetrica

- Introduciamo un parametro chiamato **k** (key= chiave) all'interno delle funzioni di cifratura **C(m,k)** e decifrazione **D(c,k)**.
- Si parla di crittografia simmetrica perchè si utilizza la stessa chiave **k** per le operazioni di cifratura e decifrazione.
- La robustezza del cifrario dipende, a differenza di prima, solo dalla segretezza della chiave **k**.



# Chiave simmetrica

## Il problema della trasmissione della chiave

- Volendo utilizzare un cifrario simmetrico per proteggere le informazioni tra due interlocutori come posso scambiare la chiave segreta? Devo utilizzare una "canale sicuro" di comunicazione.



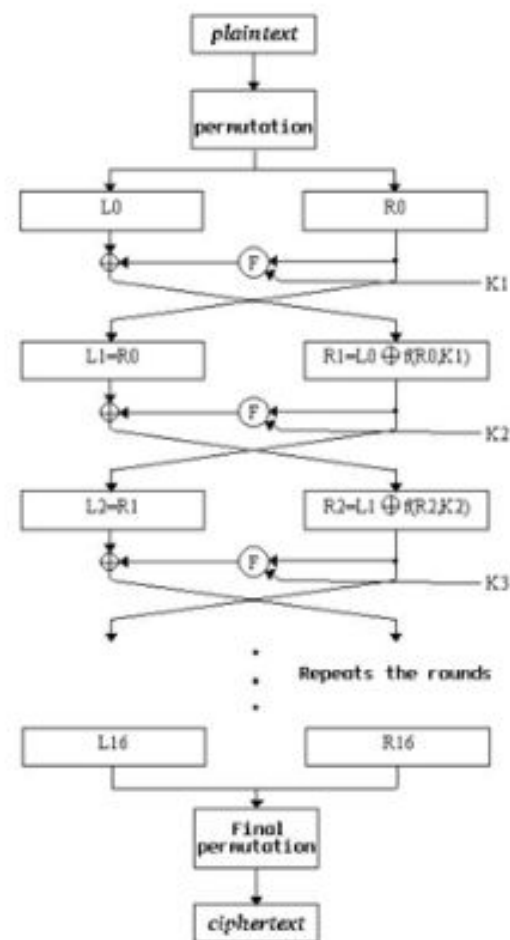
- Ma tale "canale sicuro" esiste nella realtà?
- Per una comunicazione sicura tra  $n$  utenti si dovranno scambiare in tutto  $(n-1)*n/2$  chiavi, ad esempio con 100 utenti occorreranno 4950 chiavi, il tutto per ogni comunicazione!



# Algoritmo simmetrico

## DES (Data Encryption Standard)

- Sviluppato dall'IBM nel 1970 diventato standard nel 1976.
- Utilizza chiavi di 56 bit, divide il testo in chiaro in blocchi di 64 bit, effettua delle permutazioni iniziali e finali ed un ciclo di 16 iterazioni di permutazioni e xor (Feistel network, tecniche di confusione e diffusione).
- Il 17 Luglio 1998, l'EFF (Electronic Frontier Foundation) costruisce un sistema dedicato in grado di violare il DES in meno di 3 giorni, tramite un attacco di tipo "brute-force".
- Morale della favola: non utilizzate sistemi di cifratura basati sul DES!



Shannon

Feistel

# Algoritmo simmetrico

## Crittografia a chiave simmetrica: 3DES

Triple DES (o TDEA)

- Usa tre chiavi e tre esecuzioni di DES (cifra-decifra-cifra)
- Lunghezza effettiva della chiave: 168 (=3x56) bit
- Modalità *Cipher block chaining* (cifatura a blocchi concatenati)
  - Operazione di XOR sull'iesimo blocco in ingresso con il precedente blocco di testo cifrato

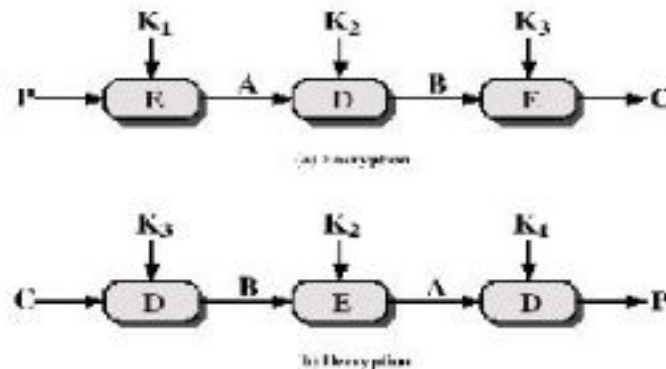


Figure 2.6 Triple DEA

$$C = E_{K3}[D_{K2}[E_{K1}[P]]]$$

C = ciphertext

P = Plaintext

EK[X] = encryption of X using key K

DK[Y] = decryption of Y using key K



# Algoritmo simmetrico

- **IDEA**
- **AES** (si usa nei dispositivi con basse risorse come le smart card)

Trovare i dettagli sul libro

# Limiti di un algoritmo simmetrico

Se si vuole effettuare un acquisto on-line  
bisogna recarsi dal commerciante per avere la  
chiave?

Il limite fondamentale è determinato dalla  
condivisione della chiave

# Cifratura asimmetrica

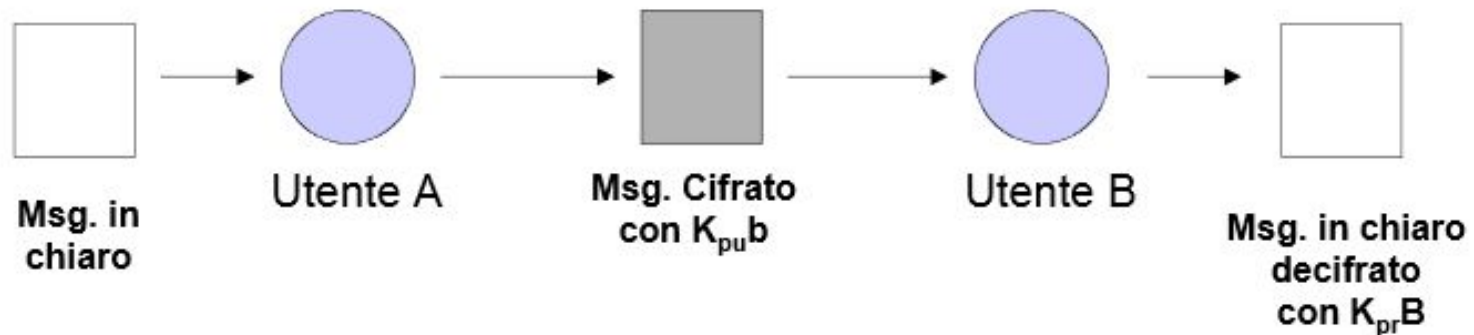
## La crittografia a chiave pubblica

- Utilizza una coppia di chiavi per le operazioni di cifratura (*encryption*) e decifrazione (*decryption*).
- Una chiave detta pubblica (**public key**) viene utilizzata per le operazioni di encryption.
- L'altra chiave, detta privata (**private key**), viene utilizzata per le operazioni di decryption.
- A differenza dei cifrari simmetrici non è più presente il problema della trasmissione delle chiavi.
- Sono intrinsecamente sicuri poiché utilizzano tecniche di tipo matematico basate sulla teoria dei numeri, sulla teoria delle curve ellittiche, etc.

# Cifratura asimmetrica

## La crittografia a chiave pubblica

- Esempio di encryption (trasmissione sicura):



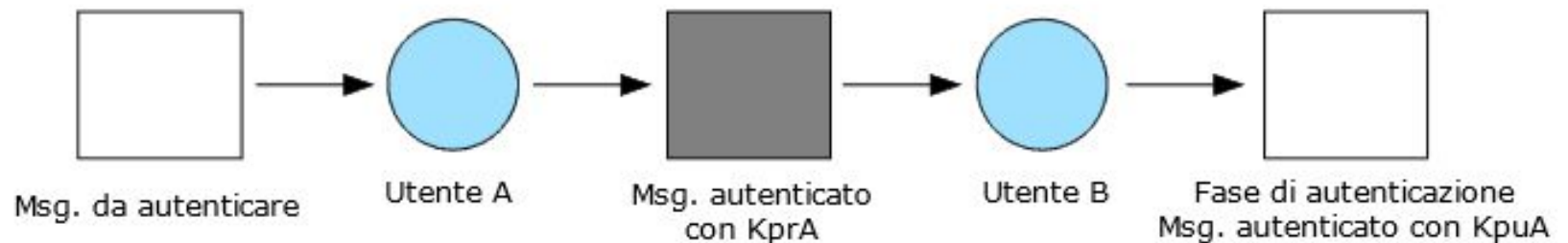
$K_{pu}B$  = chiave pubblica dell'utente B

$K_{pr}B$  = chiave privata dell'utente B

# Cifratura asimmetrica

## La crittografia a chiave pubblica

- Esempio di autenticazione:

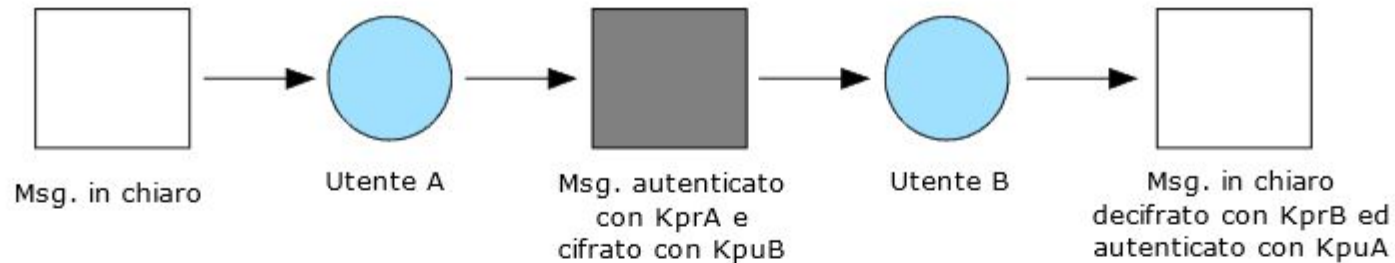


KprA = chiave privata dell'utente A  
KpuA = chiave pubblica dell'utente A

# Cifratura asimmetrica

## La crittografia a chiave pubblica

- Esempio di encryption ed autenticazione:



KprA = chiave privata dell'utente A  
KpuA = chiave pubblica dell'utente A  
KprB = chiave privata dell'utente B  
KpuB = chiave pubblica dell'utente B



# Algoritmo RSA

inventato nel 1977 da [Ronald Rivest](#), [Adi Shamir](#) e [Leonard Adleman](#)

La questione fondamentale è che nonostante le due chiavi siano fra loro dipendenti, non sia possibile risalire dall'una all'altra, in modo che se anche si è a conoscenza di una delle due chiavi, non si possa risalire all'altra, garantendo in questo modo l'integrità della crittografia.

# Algoritmo RSA

Per ottenere una discreta sicurezza è necessario utilizzare chiavi binarie di almeno 2048 bit. Quelle a 512 bit sono ricavabili in poche ore. Le chiavi a 1024 bit, ancora oggi largamente utilizzate, non sono più consigliabili. La [fattorizzazione](#) di interi grandi, infatti, è progredita rapidamente mediante l'utilizzo di hardware dedicati, al punto che potrebbe essere possibile fattorizzare un intero di 1024 bit in un solo anno di tempo, al costo di un milione di dollari (un costo sostenibile per qualunque grande organizzazione, agenzia o intelligence).

# Cosa significa fattorizzazione

Il problema della fattorizzazione di un numero nei suoi fattori primi è noto fin dai tempi di Euclide.

In particolare è noto che dato un qualsiasi numero intero positivo esiste un solo prodotto di numeri primi uguale al numero dato. L'individuazione dei fattori di questo prodotto prende appunto il nome di fattorizzazione.

- 2300 anni dopo Euclide il problema della fattorizzazione resta un problema arduo, nel senso che non si conosce un algoritmo che sia in grado di fattorizzare un numero intero in tempi *ragionevoli* (per ragionevoli qui intendiamo tempi polinomiali, o meglio ancora lineari, o meglio ancora logaritmici).

# Cosa significa fattorizzazione

Il metodo più generale per trovare i fattori primi di un intero resta quello delle divisioni successive, ben noto nelle scuole medie:

Iniziando da  $t = 2$  si tenta di dividere  $N$  per  $t$ ; se la divisione è esatta si è trovato un fattore; il quoziente prende allora il posto di  $N$  e si continua per tentativi, prima con  $t = 2$ , poi con  $t = 3$ , con  $t = 5$  ...

# Cosa significa fattorizzazione

Fattorizziamo 2142

2142   2	for i = 2, 3, 5, 7, ..., p
1071   3	for i = 2, 3, 5, 7, ..., p
357   3	for i = 2, 3, 5, 7, ..., p
119   7	for i = 2, 3, 5, 7, ..., p
17   17	for i = 2, 3, 5, 7, ..., 17, ...p
1	

$$2142 = 2 * 3 * 3 * 7 * 17$$

# Come si impostano le chiavi in RSA

## Calcolo Key e Cifratura

Le chiavi Pubbliche e Private sono definite:

$$K_{pub}=(e,n) \text{ e } K_{prv}=(d,n)$$

$$c = m^e \bmod n \quad \text{e} \quad m = c^d \bmod n$$

Affinché ciò accada devono essere verificate le seguenti condizioni (T.Euclide):

dati **a** e **b** numeri primi tali che  $n = ab$ ,  $z = (a-1)(b-1)$

- **e** non deve avere **fattori** comuni (*coprimo*) con **z**
- $(e*d) \bmod z = 1$

es:  $a=17$ ,  $b=5$  da cui  $n=85$  e  $z=64$

**e=13** non ha fattori comuni con 64, **d=5** ( $13*5 \bmod 64 = 1$ )

$K_{pub}=(13,85)$   $K_{prv}=(5,85)$  Rapp.Alfabeto={A=1,...Z=21}



# Come si utilizzano le chiavi in RSA

MITT (13,85): "EUROPA" 5,19,16,13,14,1

"E" = 5  $\rightarrow c = 5^{13} \bmod 85 = 20 \dots\dots\dots$

RIC (5,85): 20,49,16,13,39,1

$m = 20^5 \bmod 85 = 5 \rightarrow$  "E"  $\dots\dots\dots$

# Come si calcolano le chiavi in RSA

Svolgere il seguente esercizio: Individuare chiave pubblica e privata partendo dai numeri  $a=3$  e  $b=11$

Lavoro di gruppo:

Dividere la classe in gruppi formati da due studenti:

Ogni gruppo trova la coppia di chiavi e cifra un messaggio a piacimento

Una volta rese pubbliche tutte le chiavi pubbliche, ogni gruppo deve decifrare il messaggio di un altro gruppo

# Modo per calcolare la chiave privata

$$n=a*b \quad a,b \text{ PRIMI}$$

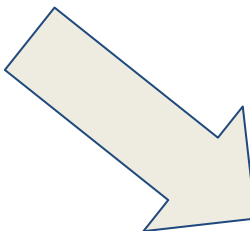
(**d**,n) chiave privata, (**e**,n) chiave pubblica

$$d*e \bmod (a-1)(b-1) = 1$$

quindi vuol dire che:

$$d*e = k(a-1)(b-1) + 1$$

da cui:


$$d = (k(a-1)(b-1) + 1) / e$$

Provo  $k=1, 2, 3 \dots$

Finché non trovo un valore INTERO per d

# Definizione di coprimo

Cosa significa 'coprimo'?

- $a$  è coprimo di  $b$

se il massimo comune divisore tra  $a$  e  $b$  è 1

- Ad es. 7 e 15 sono coprimi, mentre 8 e 10 no (hanno in comune il divisore 2)

Nota: se  $a$  è primo, allora è coprimo di qualsiasi numero che non sia diviso da  $a$

- Ad es. 7 è coprimo di tutti i numeri che non sono multipli di 7

# Forza dell'algoritmo RSA

Non è possibile risalire alla chiave privata dalla pubblica, in quanto servirebbe conoscere

**$(p-1)(q-1)$**  e questo implica fattorizzare  **$n$** .

Questo è un problema computazionalmente difficile.

dati due numeri primi  $p$  e  $q$   
è facile calcolare  $n = p * q$ ,

ma dato  $n$  è difficile (tempo di computazione esponenziale)  
risalire ai suoi fattori  $p$  e  $q$

# Chi gestisce le chiavi pubbliche?

RSA ha risolto il problema delle chiavi private.

Ma chi gestisce le chiavi pubbliche?

**public key infrastructure (PKI)**, è un insieme di processi e mezzi tecnologici che consentono a terze parti fidate di verificare e/o farsi garanti dell'identità di un utente, oltre che di associare una chiave pubblica a un utente, normalmente per mezzo di software distribuito in modo coordinato su diversi sistemi. Le chiavi pubbliche tipicamente assumono la forma di certificati digitali.

(li analizzeremo in seguito)



# Vantaggi e svantaggi

Crittografia simmetrica	Crittografia asimmetrica
Pro: molto veloce	Pro: non serve un canale sicuro per lo scambio delle chiavi
Contro: problema scambio di chiavi	Contro: molto lenta a causa dei calcoli complessi da effettuare

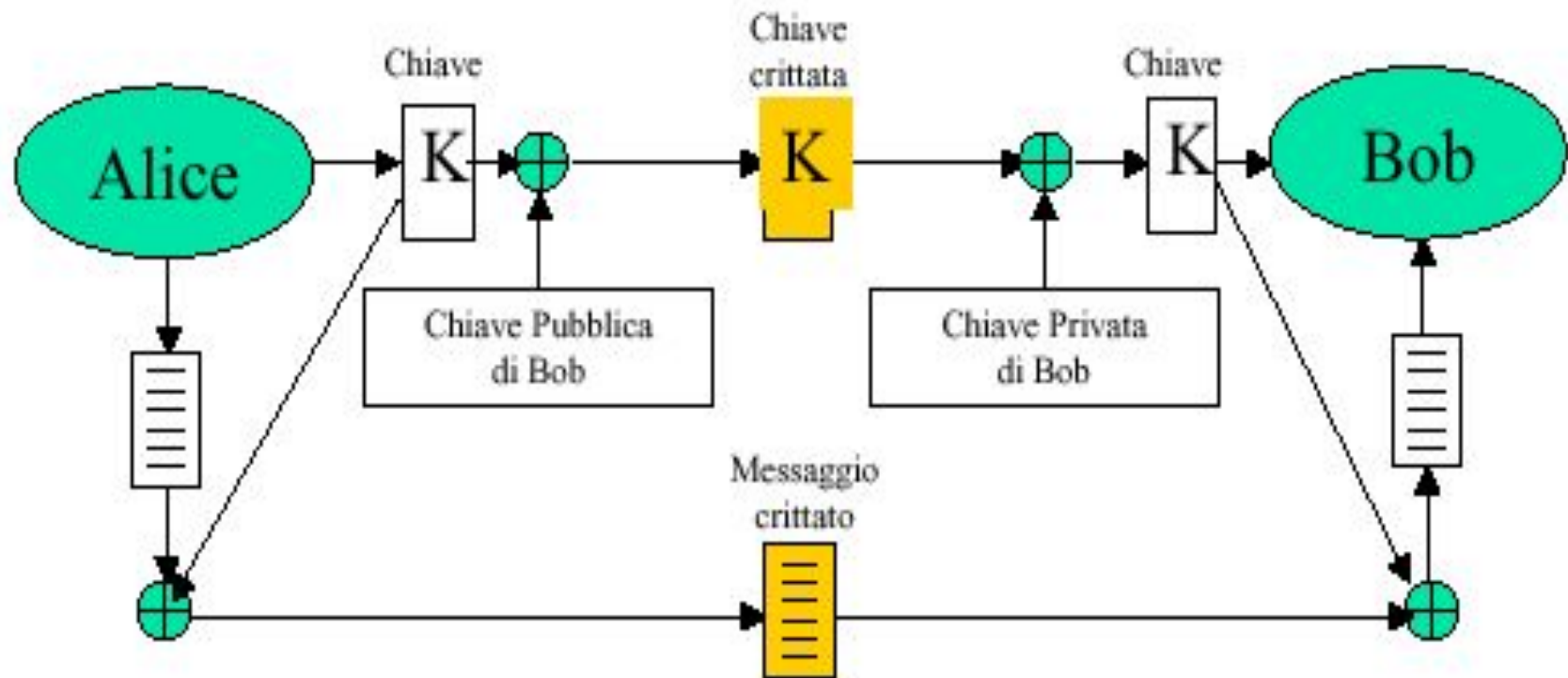
# Sistema ibrido (asimmetrico e simmetrico)

Un algoritmo ibrido utilizza sia un sistema simmetrico che uno a chiave pubblica. In particolare esso funziona utilizzando un algoritmo a chiave pubblica per condividere una chiave per il sistema simmetrico. Il messaggio effettivo è quindi criptato usando tale chiave e successivamente spedito al destinatario.

Si utilizza la chiave pubblica solo per inviare la chiave privata da utilizzare successivamente per la cifratura delle informazioni.

La chiave privata (chiamata chiave di sessione) si scambia solo una volta

# Sistema ibrido (asimmetrico e simmetrico)



# Firma digitale

La Firma Digitale è l'equivalente informatico di una tradizionale firma autografa apposta su carta e possiede le seguenti caratteristiche:

- ☐ autenticità: la firma digitale garantisce l'identità del sottoscrittore
- ☐ integrità: la firma digitale assicura che il documento non sia stato modificato dopo la sottoscrizione
- ☐ non ripudio: la firma digitale attribuisce piena validità legale al documento, pertanto il documento non può essere ripudiato dal sottoscrittore

# Firma digitale

Per generare una firma digitale è necessario utilizzare una coppia di chiavi digitali asimmetriche attribuite in maniera univoca ad un soggetto, detto titolare.

La chiave privata è conosciuta solo dal titolare ed è usata per generare la firma digitale da apporre al documento.

Viceversa, la chiave da rendere pubblica è usata per verificare l'autenticità della firma.

L'impiego della Firma Digitale pertanto, permette di snellire significativamente i rapporti tra Pubbliche Amministrazioni, i cittadini o le imprese, riducendo drasticamente la gestione in forma cartacea dei documenti, proprio come indicato nelle Linee Guida per l'utilizzo della Firma Digitale, emanate da AGID (Agenzia per l'Italia Digitale, ex DigitPA)

# Firma digitale

I metodi crittografici a chiave pubblica possono essere utilizzati per la costruzione di strumenti per la firma digitale, variamente concepiti. Mentre nella crittografia la chiave pubblica viene usata per la cifratura, ed il destinatario usa quella privata per leggere in chiaro il messaggio, nel sistema della firma digitale il mittente utilizza la funzione di cifratura e la sua chiave privata per generare un'informazione che (associata al messaggio) ne verifica la provenienza, grazie alla segretezza della chiave privata. Chiunque può accertare la provenienza del messaggio utilizzando la chiave pubblica.

La firma digitale viene realizzata tramite tecniche crittografiche a chiave pubblica insieme all'utilizzo di particolari funzioni matematiche, chiamate funzioni **hash unidirezionali**.



# Firma digitale

Il processo di firma digitale passa attraverso tre fasi:

1. Generazione dell'impronta digitale.
2. Generazione della firma.
3. Apposizione della firma.

Nella prima fase viene applicata al documento in chiaro una funzione di hash appositamente studiata che produce una stringa binaria di lunghezza costante e piccola, normalmente 128 o 160 bit, chiamata digest message, ossia impronta digitale. Queste funzioni devono avere due proprietà:

# Firma digitale

- unidirezionalità (one-way), 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.  
( se  $x \neq y$  allora  $f(x) \neq f(y)$  )

# Firma digitale

Poiché la dimensione del digest message è fissa, e molto più piccola di quella del messaggio originale; la generazione della firma risulta estremamente rapida.

# Firma digitale

Quando A vuole mandare a B un messaggio autenticato ed integro, calcola fingerprint, la cifra con la sua chiave privata e la invia insieme al messaggio in chiaro. Riassumendo A:

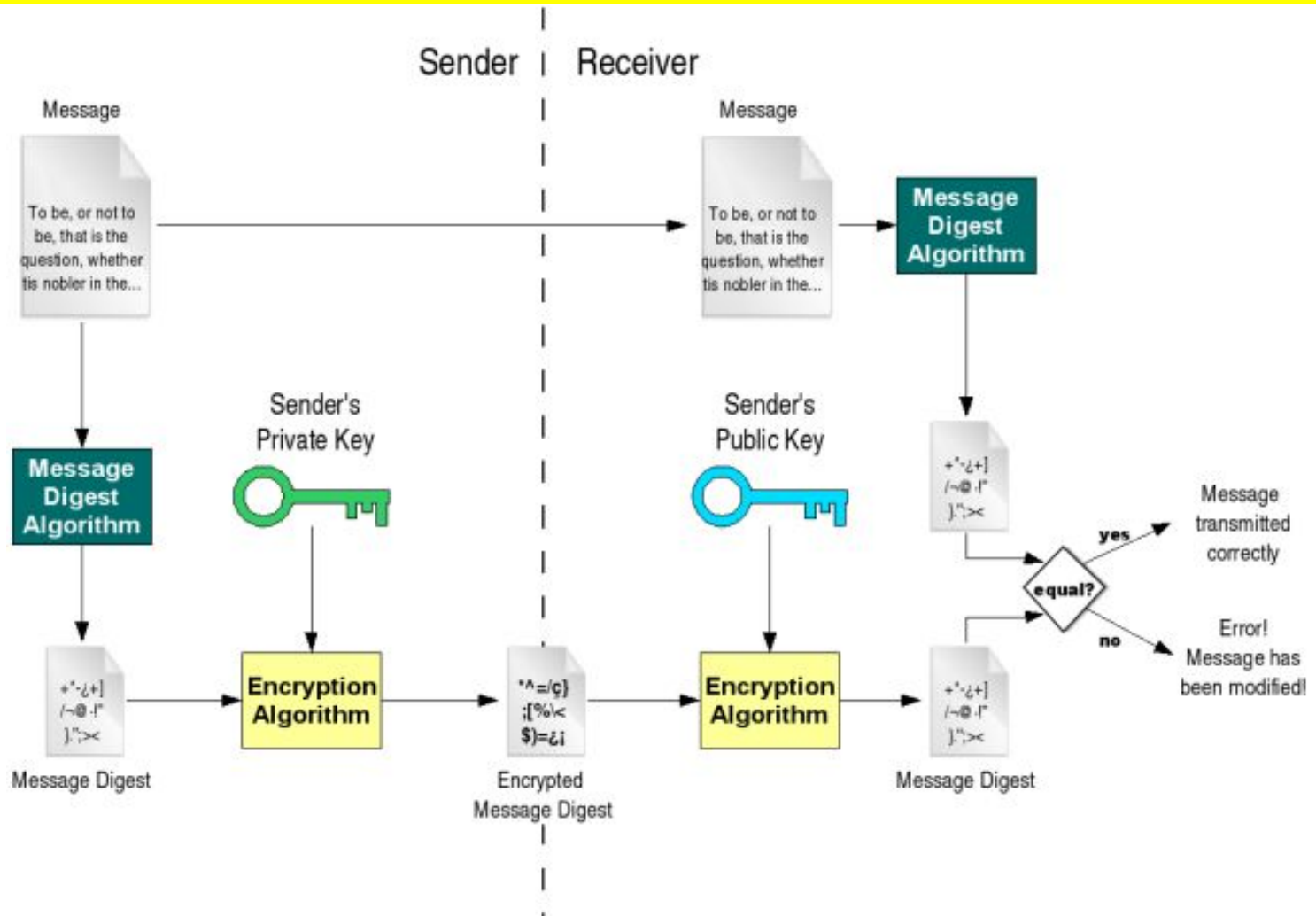
- Dal documento estrae l'impronta in chiaro (fingerprint)
- L'impronta viene cifrata con la chiave privata
- L'impronta crittografata viene accodata al messaggio in chiaro e tutto inviato a B

# Firma digitale

Il destinatario B:

- Decifra la firma digitale utilizzando la chiave pubblica del mittente (quest'ultimo viene identificato come certo)
- Ricalcola la fingerprint partendo dal documento originario. Se è uguale a quella decifrata, il documento è integro

# Rappresentazione grafica



# Funzioni Hash

**Hash** è un termine della lingua inglese (*to hash* sminuzzare, pasticciare) che designa originariamente una polpettina fatta di avanzi di carne e verdure; per estensione indica un composto eterogeneo cui viene data una forma incerta: "*To make a hash of something*" vuol dire infatti creare confusione, o fare una cosa piuttosto male.

Le funzioni Hash più utilizzate sono:

- **MD5** (standard per internet, veloce ma non molto sicuro)
- **SHA** (standard americano, non molto veloce ma sicuro)

# Funzioni Hash

Le funzioni Hash hanno tanti campi di applicazione:

- Crittografia
- Controllo degli errori nel trasferimento dei dati
- Creazione di hash table nella gestione degli archivi e data base
- Offuscamento della password o passphrase

L'algoritmo di hash elabora qualunque mole di bit (in informatica si dice che elabora dati "grezzi").



# Funzioni Hash

Dato che i testi possibili, con dimensione finita maggiore dell'hash, sono più degli hash possibili, ad almeno un hash corrisponderanno più testi possibili. Quando due testi producono lo stesso hash, si parla di *collisione*, e la qualità di una funzione di hash è misurata direttamente in base alla difficoltà nell'individuare due testi che generino una collisione.

Cosa è una **collisione**?

Essendo l'alfabeto di destinazione più piccolo del sorgente, nasce il fenomeno delle collisioni. La collisione è quando la stessa funzione di hash, applicata a due input diversi, restituisce lo stesso valore in output.

# Funzioni Hash

Le funzioni hash ritenute più resistenti richiedono attualmente un tempo di calcolo per la ricerca di una collisione molto elevato.

Un hash crittograficamente sicuro non dovrebbe permettere di risalire, in un tempo confrontabile con l'utilizzo dell'hash stesso, ad un testo che possa generarlo.

# Funzioni Hash

## *Crittografia, Controllo Errori*

- La lunghezza dei valori di hash varia a seconda degli algoritmi utilizzati. Il valore più comunemente adottato è di 128 bit, che offre una buona affidabilità in uno spazio relativamente ridotto. Tuttavia va registrata la possibilità d'uso di hash di dimensione maggiore (SHA, ad esempio, può anche fornire stringhe di 224, 256, 384 e 512 bit).
- Le funzioni hash svolgono un ruolo essenziale nella crittografia: sono utili per verificare l'integrità di un messaggio, poiché l'esecuzione dell'algoritmo su un testo anche minimamente modificato fornisce un ***message digest*** completamente differente rispetto a quello calcolato sul testo originale, rivelando la tentata modifica.

# Funzioni Hash

## *Hash Table*

L' unidirezionalità non è indispensabile se si usano gli hash per controllare gli errori nei trasferimenti dei dati.

Nelle applicazioni di **basi di dati** la funzione hash è usata per realizzare una particolare struttura dati chiamata hash table. In questa applicazione non occorrono proprietà crittografiche e generalmente l'unica proprietà richiesta è che non ci siano hash più probabili di altri.

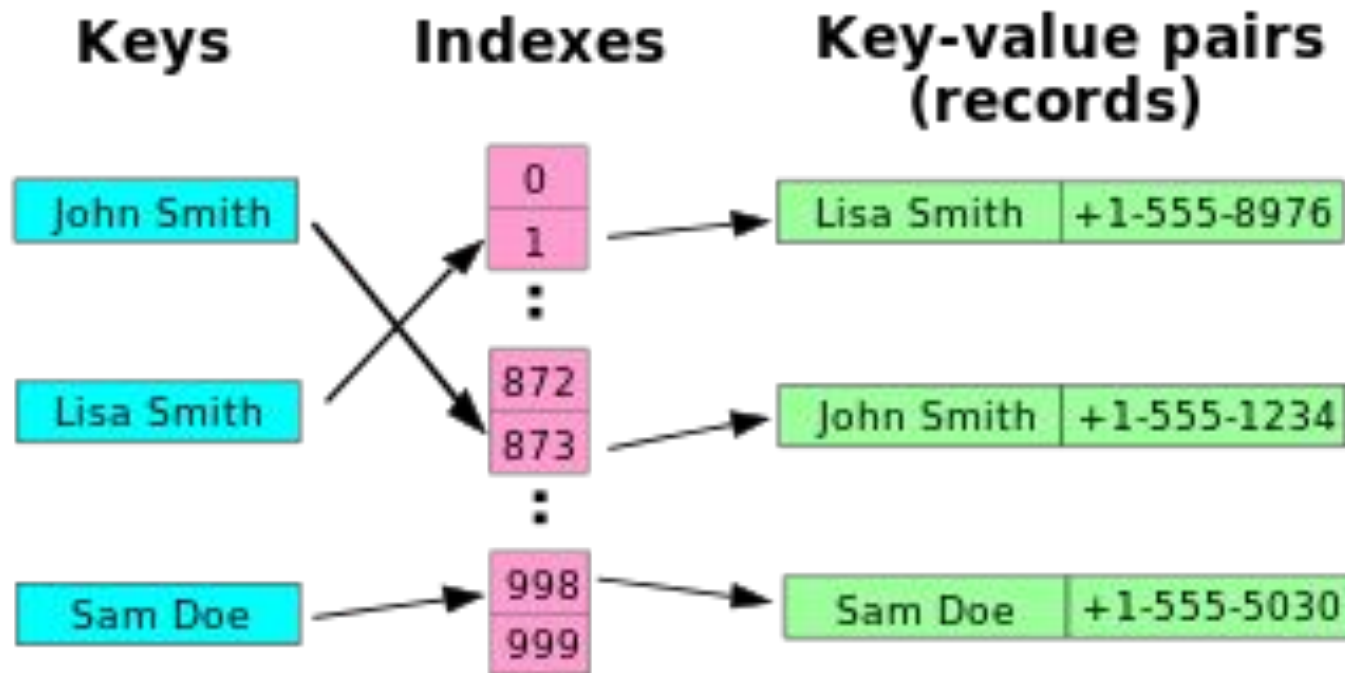
# Funzioni Hash

## *Hash Table*

La **Hash table** è una struttura dati molto efficiente per le operazioni di ricerca. La hash table contiene dati associati ad una chiave di ricerca e viene spesso utilizzato nei database per indicizzare gli elementi che saranno oggetto di ricerca. Questa tecnica (detta di hashing) permette di realizzare funzioni di ricerca che riescono ad individuare l'elemento desiderato in un tempo costante, indipendente (almeno in teoria) dal numero di elementi presenti nell'indice.

# Funzioni Hash

## *Hash Table*



# Funzioni Hash

## *Secure Hashing*

Le funzioni hash possono essere utilizzate per offuscare le **password** o le **passphrase** (in questo caso si chiamano funzioni di secure hashing)

Es. la password specificata dall'utente viene offuscata con una funzione hash e il risultato viene memorizzato nel database delle password. Quando l'utente effettua un nuovo accesso e bisogna controllare la correttezza della password inserita, si applica la funzione hash alla password e si controlla che il valore ottenuto sia uguale a quello precedentemente memorizzato nel database

# IMPORTANTE

Le funzioni hash non sono funzioni crittografiche... perché lo scopo della crittografia è avere un sistema sicuro per lo scambio di informazioni che permetta ai legittimi interlocutori di mettere in chiaro il testo. Le funzioni di *secure hashing* non hanno questo obiettivo. Una volta offuscato, il dato resta offuscato e l'unico modo di metterlo in chiaro è un attacco di forza bruta o con dizionario.



# Come violare le password?

L'attaccante ha una serie di hash. Nel suo arsenale d'attacco vi saranno un bel po' di dizionari contenenti password di uso comune alle quali verrà applicato l'algoritmo di hash per vedere se il valore offuscato è nel database appena trafugato.

Il particolare algoritmo di hash viene desunto dalla lunghezza dell'hash stesso. Quindi è un'informazione che dobbiamo dare per scontata sia in mano a chi ci attacca.

# Come violare le password?

Terminati i ***dizionari***, si parte enumerando tutte le possibili combinazioni di lettere e numeri. E qui è solo una questione di tempo .

Si può rendere più complesso la violazione delle password introducendo il **salt** nella fase iniziale di calcolo della funzione hash applicata alla password. Il salt è una sequenza casuale di bit utilizzata assieme ad una password come input a una funzione hash, il cui output è conservato al posto della sola password, e può essere usato per autenticare gli utenti.

Usando dati salt si complicano gli attacchi a dizionario, quella classe di attacchi che sfruttano una precedente cifratura delle voci di un elenco di probabili parole chiave per confrontarle con l'originale: ogni bit di salt utilizzato raddoppia infatti la quantità di memorizzazione e di calcolo necessari all'attacco.

# Come violare le password?

per aumentare la sicurezza è abitudine tenere segreto il valore salt e conservarlo separatamente dal database delle password. Ciò fornisce un vantaggio quando viene rubato il database con le password, ma non il salt. Per scoprire una password da un hash rubato, infatti, un utente malintenzionato non può limitarsi a provare password comuni (ad esempio le parole o i nomi della lingua italiana), ma è costretto a calcolare gli hash di caratteri casuali (almeno per la parte dell'input che si pensa essere il salt), il che è molto più lento.

# MD5

**MD5** è l'acronimo di ***Message Digest 5***, un algoritmo di sintesi a senso unico (cioè che non prevede di essere decrittato), creato nel 1991 da **Ronald Rivest** a causa dell'inefficienza del suo predecessore: MD4.

Tecnicamente, MD5 è una ***funzione di compressione***, che dato un input composto da una stringa di lunghezza arbitraria, restituisce una fingerprint che consiste in una stringa da **128** bit. Si presuppone che l'hash ( ovvero l'output ) restituito dalla funzione sia univoco o, più precisamente, che sia molto improbabile ottenere due hash identici da due input diversi.

# MD5

Le funzioni di hashing, hanno anche altre funzioni oltre alla sicurezza, infatti permettono di verificare l'***integrità dei downloads***, tramite un checksum da confrontare: Assieme al file del download, viene fornita la stringa che corrisponde all'hash dei file immessi come input. Se l'hash ottenuto non corrisponde a quello fornito dal download, potrebbe essersi verificata una perdita di dati.

# MD5

L'elaborazione prevede 4 fasi:

1. Padding, cioè aggiunta di bit 0 preceduti da un 1 fino a raggiungere **448** mod 512
2. Si aggiungono **64** bit che contengono la lunghezza del messaggio originale  $|messaggio|1|00000 \dots 0|lunghezza - del - messaggio|$
3. Inizializzazione del buffer **MD**: questo buffer è composto da **4** word a **32** bit, inizializzate come segue:

A: 01 23 45 67

B: 89 ab cd ef

C: fe dc ba 98

D: 76 65 32 10

**MD=ABCD**

# MD5

4) Elaborazione del messaggio: vengono definite quattro funzioni che ricevono in ingresso tre word a 32 bit e ne restituiscono una:

$$F(x,y,z) = (x \text{ AND } y) \text{ OR } (\text{NOT } x \text{ OR } z)$$

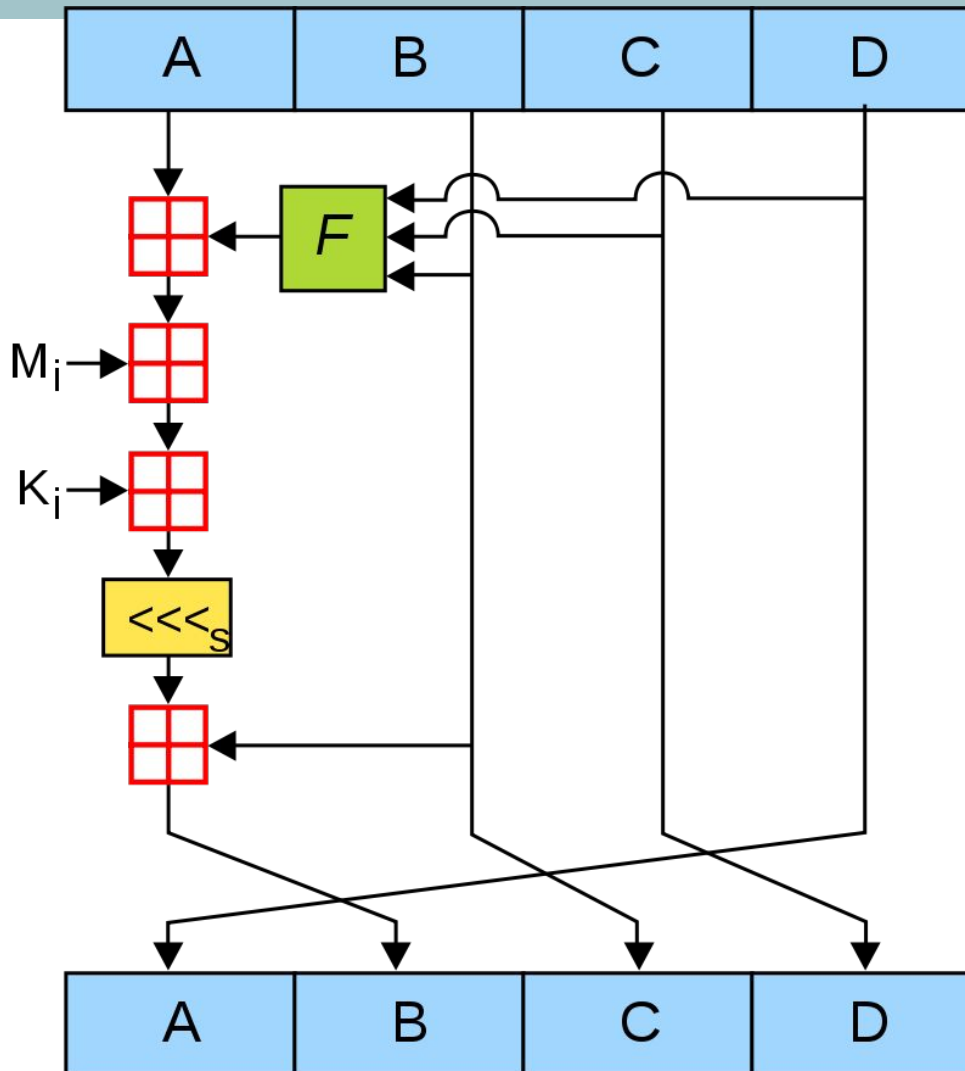
$$G(x,y,z) = (x \text{ AND } z) \text{ OR } (y \text{ OR } \text{NOT } z)$$

$$H(x,y,z) = (x \text{ XOR } y \text{ XOR } z)$$

$$I(x,y,z) = y \text{ XOR } (x \text{ OR } \text{NOT } z)$$

Vengono poi applicati degli algoritmi per inserire tutte le word ottenute nei **128** bit del digest finale

# Come funziona MD5



L'algoritmo consta di 64 di queste operazioni, raggruppate in gruppi di 16. Nello schema, "F" è una funzione non lineare (ne viene usata una per ogni passaggio F,G,H,I), " $M_i$ " indica un blocco a 32 bit del messaggio, " $K_i$ " indica una costante a 32 bit, differente per ogni operazione.

[Algoritmo](#)

$$b = b + ((a + F(b,c,d) + M[i] + K[i]) \lll s).$$



# SHA

Secure Hash Algorithm

SHA-0 obsoleto, violato

SHA-1 160 bit, violato nel 2017

SHA-2 in uso (non è stato ancora violato)

SHA-3 introdotto nel 2012

SHA-2 si utilizza nel **Pretty Good Privacy** (algoritmo di crittografia a chiave pubblica per crittografare le **e-mail**) e **Secure Sockets Layers**

Algoritmo e variante		Dimensione dell'output (bit)	Dimensione dello stato interno (bit)	Dimensione del blocco (bit)	Max. dimensione del messaggio (bit)	Dimensione della word (bit)	Passaggi	Operazioni	Collisioni trovate
SHA-0		160	160	512	$2^{64} - 1$	32	80	+,and,or,xor, rotl	Sì
SHA-1		160	160	512	$2^{64} - 1$	32	80	+,and,or,xor, rotl	Attacco $2^{53}$ [11]
SHA-2	SHA-256/224	256/224	256	512	$2^{64} - 1$	32	64	+,and,or,xor,shr, rotr	Nessuna
	SHA-512/384	512/384	512	1024	$2^{128} - 1$	64	80	+,and,or,xor,shr, rotr	Nessuna

# SHA

Secure Hash Algorithm

L'algoritmo SHA funziona come MD5 con **4 fasi** delle quali le prime 2 identiche mentre il passo 3 viene effettuato con uno schema a 8 registri e nel passo 4 la sequenza viene divisa in blocchi di 512 o 1024 bit. Su ogni blocco vengono effettuati 80 cicli di operazioni.

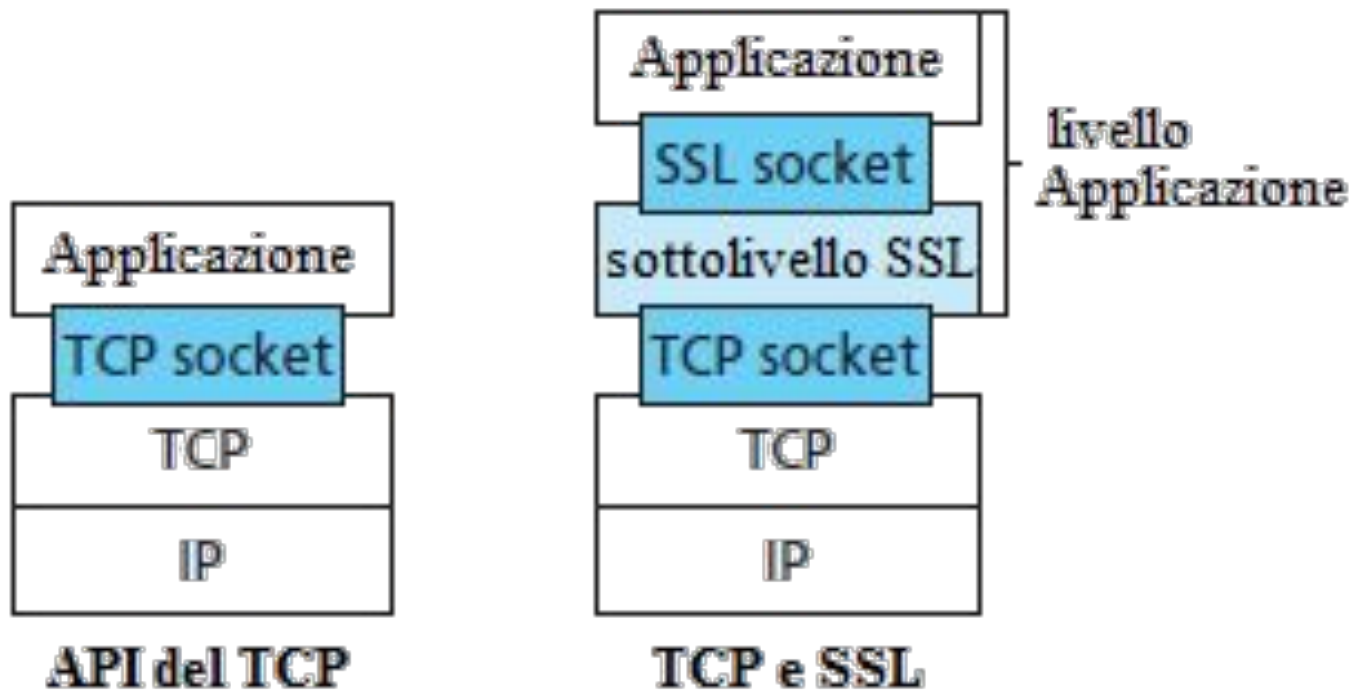
E' alla base dei protocolli **SSL**.

# PGP e SSL

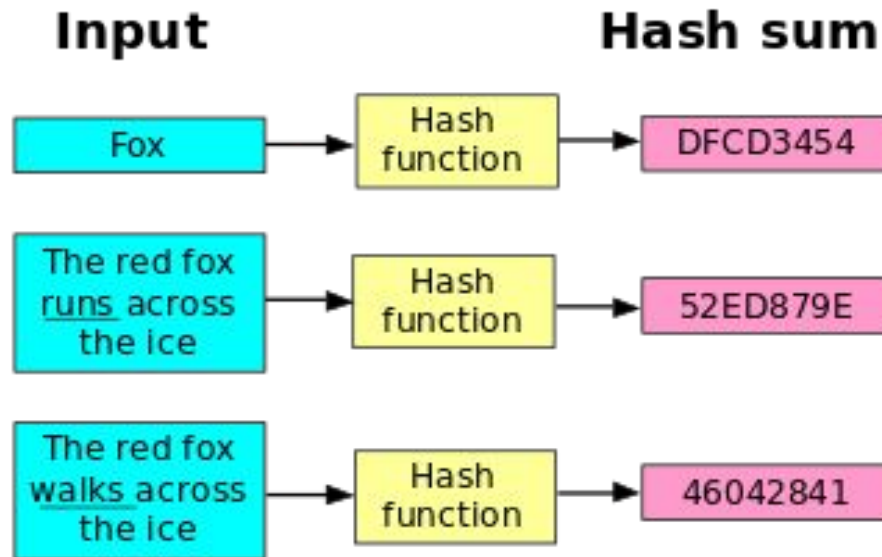
**PGP** applicazione usa la crittografia a **chiave asimmetrica**, nella quale il destinatario del messaggio ha generato precedentemente una coppia di chiavi collegate fra loro; una chiave pubblica ed una privata. La chiave pubblica del destinatario serve al mittente per cifrare una chiave comune (detta anche chiave segreta o convenzionale) per un algoritmo di crittografia **simmetrica**; questa chiave viene quindi usata per cifrare il testo in chiaro del messaggio che abbiamo già visto e classificato come **Sistema IBRIDO**.

**SSL** sono dei protocolli crittografici di **presentazione** usati nel campo delle telecomunicazioni e dell'informatica che permettono una comunicazione sicura dalla sorgente al destinatario (*end-to-end*) su reti TCP/IP (come ad esempio Internet) fornendo autenticazione, integrità dei dati e cifratura operando al di sopra del livello di trasporto; utilizzato da siti di e-commerce e finanziari per proteggere le transazioni online;

# PGP e SSL



# Esempio di applicazione di SHA



Risultato dei primi quattro [byte](#) della funzione hash [SHA-1](#).

[Hash di un File](#)

# Cosa serve per fare la firma digitale?

- Dispositivo di firma sicuro (smart card, token USB) rilasciato da appositi enti certificatori, i quali accertano l'identità del mittente prima di rilasciare la carta e fornire il PIN.
- Vedere cos'è la carta nazionale dei servizi e cosa bisogna fare per richiederla (cercare su internet)

# Cosa serve per fare la firma digitale?

Durante l'apposizione della firma il file viene incapsulato in una busta “crittografica” e il nuovo file ha una estensione **.p7m**

Ci sono dei programmi che permettono di verificare l'identità del firmatario e verificare l'integrità del file di partenza.

[Differenti formati firme digitali.](#) ([dike6](#))

# Dove troviamo la chiave pubblica?

Nei sistemi di cifratura abbiamo trascurato ***come facciamo a conoscere la chiave pubblica e ad essere sicuri che sia quella corretta.***

Ci serve un documento che garantisca l'identità del mittente e che mi permetta di ottenere tutte le informazioni che mi servono sul mittente: **il certificato digitale**



# Cos'è un certificato digitale?

I certificati digitali sono dei file, con una validità temporale limitata, usati per garantire l'identità di un soggetto, sia esso un server o una persona.

## **A cosa servono**

I certificati digitali, rappresentano quello che i documenti d'identità costituiscono nella vita reale; servono per stabilire con esattezza, in una comunicazione, l'identità delle parti.

# Chi crea i certificati digitali?

I certificati digitali vengono rilasciati dalle cosiddette autorità di certificazione.

Un' ***Autorità di Certificazione*** (Certification Authority, solitamente abbreviato con C.A.) rilascia i certificati a chi ne fa richiesta dopo averne attestato l'identità.

Svolge il ruolo di garante dell'identità di chi usa il certificato da lei rilasciato, così come le autorità di pubblica sicurezza (prefettura, comune, etc...) che emettono documenti di identificazione quali il passaporto o la carta d'identità.

# Chi crea i certificati digitali?

Chiunque può verificare la validità di un certificato, in quanto le C.A. devono mantenere un pubblico registro dei Certificati Emessi e una Lista dei Certificati Revocati (Certification Revocation List) disponibile per la verifica per via telematica da parte di tutti gli utenti.

L'elenco delle più diffuse autorità di certificazione è solitamente precaricato nei browser più diffusi al momento dell'installazione.

# Chi crea i certificati digitali?

Esistono due tipi di CA :

***Pubbliche (Trusted):*** sono enti pubblicamente riconosciuti su Internet, abilitati all'emissione univoca dei certificati che a loro volta sono pubblicamente riconosciuti ed utilizzabili per servizi diversi (certificati acquistabili per certificare siti, firmare posta e documenti ecc.).

# Chi crea i certificati digitali?

***Private (Untrusted):*** sono enti privati che emettono certificati per un uso non pubblico e dedicato a servizi circoscritti in un dominio specifico (singola applicazione, all'interno di una intranet ecc..).

# Come funziona il CD

Tutto il sistema si fonda sull'affidabilità della **Certificati on Authority**.

E' a questa che ci si rivolge per acquistare un certificato digitale ed anch'essa ha una propria *coppia di chiavi*.

Quando la **Certification Authority** ti fornisce i dati del tuo certificato. Quindi sarà possibile decifrarle con la corrispondente chiave pubblica della Certification Authority.

Questo garantisce che, una volta consegnate al legittimo proprietario, non sarà possibile alterare le chiavi.

# Come funziona il C.D.

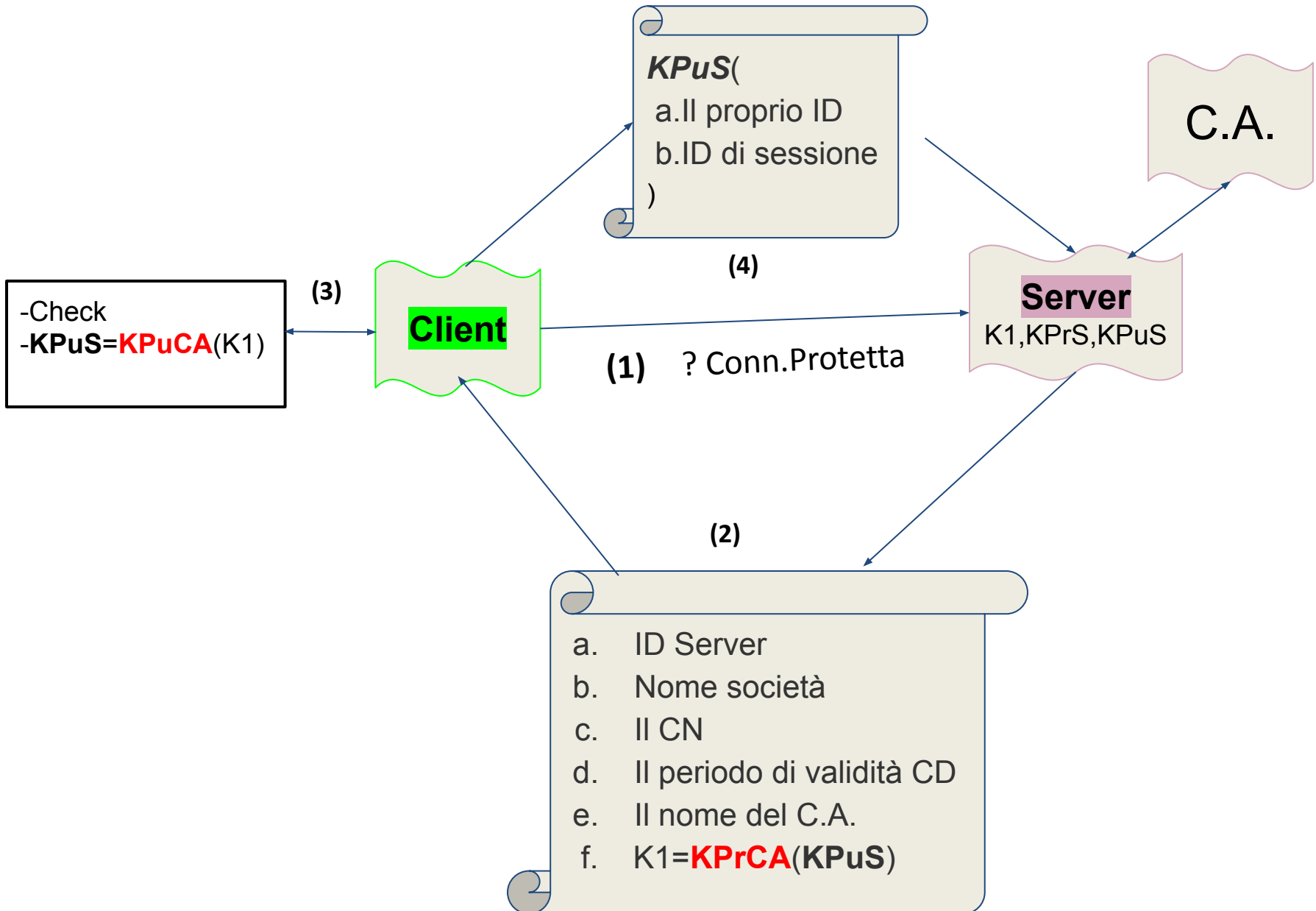
In concreto succede questo:

1. Il **Client** richiede l'apertura di una connessione protetta al **Server**;
2. Il **Server** risponde al **Client** inviandogli:
  - a. Il proprio ID
  - b. Il nome della società per la quale è stato emesso il certificato
  - c. Il proprio common name, che contiene il nome di dominio per il quale il certificato è valido.
  - d. Il periodo di validità del certificato
  - e. Il nome della Certification Authority che ha rilasciato il certificato
  - f. La propria chiave pubblica cifrata con la chiave privata della Certification Authority
3. Il **Client** verifica la validità dei dati che gli sono stati inviati dal **Server** e ne decifra la chiave pubblica utilizzando la chiave pubblica della Certification Authority.
4. Il **Client** usa la chiave pubblica del **Server** appena ottenuta per cifrare ed inviargli:
  - a. Il proprio ID
  - b. Un ID di sessione (che permette al **Server** di distinguere un Client dagli altri)
5. Le presentazioni tra Client e Server sono finite, ora i due si conoscono e sono in grado di trasmettere e ricevere dati cifrati perchè si sono scambiati prima le chiavi per codificare la comunicazione.

Da questo momento in poi tutti i dati saranno assolutamente protetti.

Quella esposta sopra è una generica procedura di comunicazione cifrata.

# Come funziona il C.D.





# Standard X.509

*Lo standard x.509 è stato definito nel 1988 da **ITU-T** (International Telecommunication Union – Telecommunication Standardization Sector, l'ente regolatore per gli standard nelle telecomunicazioni) ed ha subito una revisione nel 1993 (versione 2) e nel 1995 (versione 3).*

*Un certificato x.509 serve ad abbinare un nome distintivo (conosciuto come **Distinguished name** , ovvero l'acronimo **DN**) a una **chiave pubblica**; questo nome distintivo è in pratica una raccolta di informazioni su una certa persona in un certo contesto.*

# Standard X.509

I campi tipici di un nome distintivo nei certificati x.509 sono:

## **CAMPO**

CN

O

OU

C

ST

L

## **DESCRIZIONE**

Nome comune, o *Common name*

Organizzazione

Dipartimento

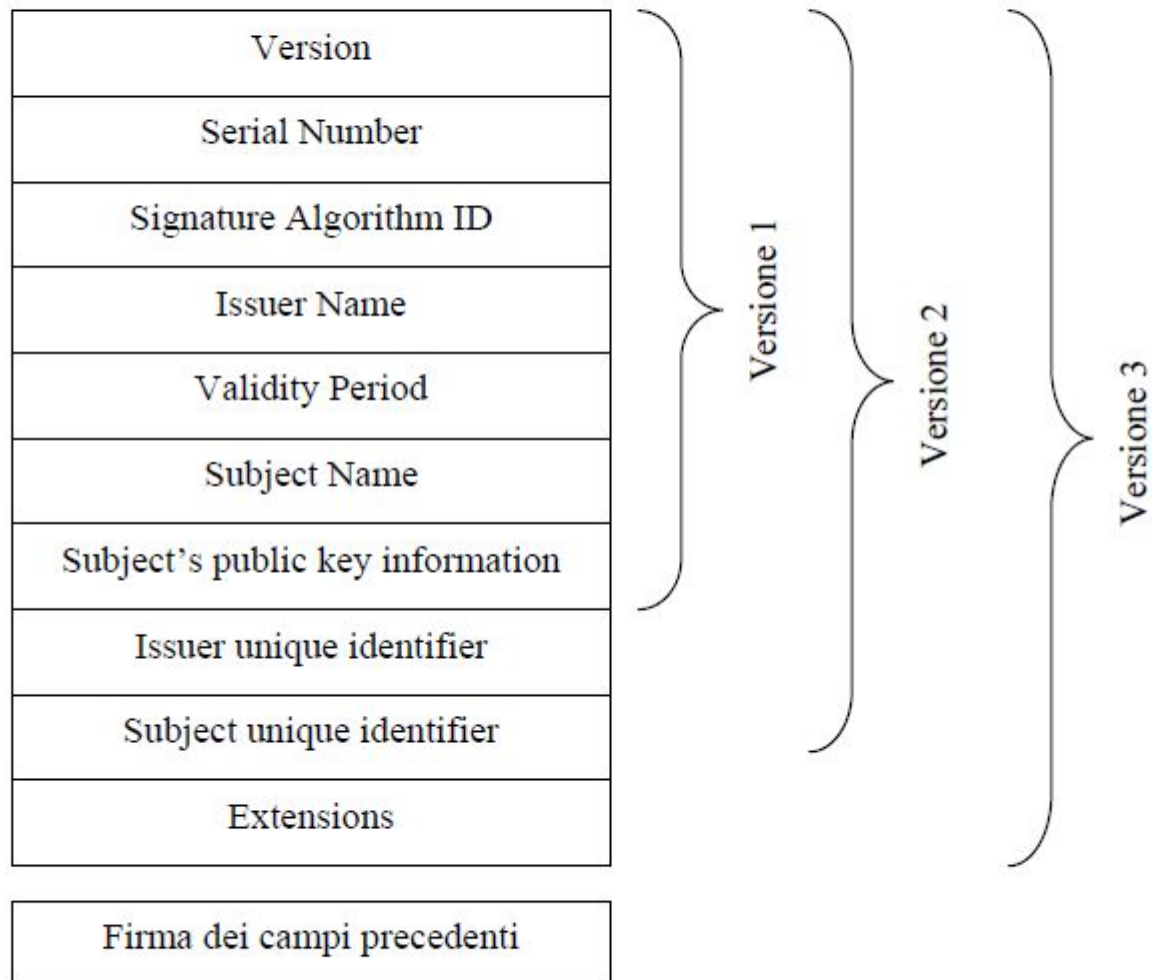
Sigla del paese (nazione)

Regione o provincia

Località

# Standard X.509

Un certificato x.509 è strutturato come segue:



# Standard X.509

**Version** è un valore intero; le possibili alternative sono:

0. default (v1)

1. se presente Issuer unique identifier oppure Subject unique identifier (v2)

2. se ci sono estensioni (v3)

**Serial number** è un valore intero, unico per ogni CA; identifica senza ambiguità il certificato

**Signature algorithm ID** rappresenta l'algoritmo e la funzione hash usati dalla CA per firmare il certificato (es. md5WithRSAEncryption, sha-1WithRSAEncryption)

**Issuer name** DN della CA che ha creato e firmato il certificato

**Validity period** contiene 2 date: la data di inizio di validità del certificato e la data di scadenza del certificato

# Standard X.509

**Subject name** DN dell'utente (proprietario) del certificato (cioè chi conosce la chiave privata corrispondente)

**Subject's public key information** chiave pubblica del proprietario del certificato e relativo algoritmo (es. rsaEncryption)

**Issuer unique identifier** è usato per distinguere univocamente la CA nel caso che il DN (della CA) sia stato riutilizzato

**Subject unique identifier** è usato per distinguere univocamente il proprietario del certificato nel caso che il DN (dell'utente) sia stato riutilizzato

**Extensions** diversi campi di estensione: ogni estensione comprende un identificatore, un indicatore di criticità, un valore; le estensioni si dividono nelle tre categorie key and policy information (ad es. key usage: scopi per cui un certificato può essere usato, private-key usage period: periodo d'uso della corrispondente chiave privata) subject and issuer attributes (ad es. subject/issuer alternative name: nomi alternativi del soggetto/emettitore in varie forme) e certification path constraints (ad es. se il soggetto può agire da CA).

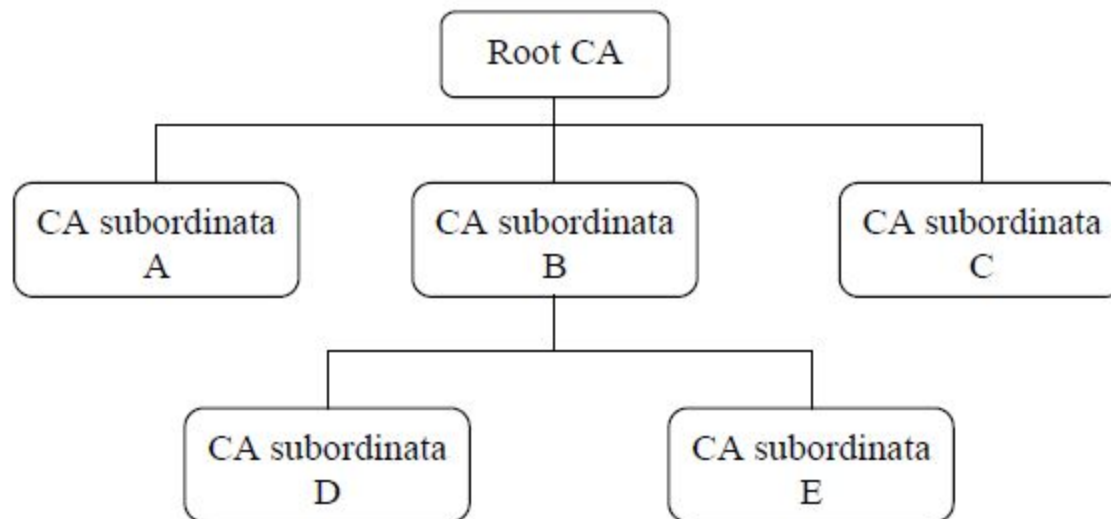
**L'hash dei suddetti campi è firmato tramite la chiave privata della CA.**

# Struttura della CA

I certificati vengono rilasciati da una Autorità di Certificazione (CA) la cui firma apposta sul certificato garantisce il legame tra chiave ed entità; ogni autorità di certificazione stabilisce e impone la propria procedura per ottenere la propria certificazione: questo significa che ogni autorità definisce il proprio ambito di competenza, quali tipi di certificazione elettronica è in grado di fornire (si fa riferimento al formato del certificato elettronico) e quali siano le informazioni che devono essere fornite in modo preciso: sarà poi compito dell'autorità la verifica della veridicità di tali informazioni.

# Struttura della CA

La struttura delle CA è di tipo gerarchico, ad albero:



L'autorità principale (Root CA), che si autocertifica, demanda e organizza il compito di certificazione a strutture inferiori, firmando il loro certificato (con la propria chiave privata): queste autorità inferiori possono avere a loro volta la responsabilità sulla certificazione di altre autorità di livello inferiore e così via.

# Richiesta di un certificato

Per ottenere un certificato da un'autorità, utilizzando lo standard X.509, si parte dalla creazione di una richiesta di certificato, che in pratica è un certificato avente già tutte le informazioni, tranne la firma del garante; le operazioni da compiere sono:

- l'utente prova la sua identità alla CA cui si è rivolto per ottenere il rilascio di un certificato
- viene generata una coppia di chiavi (questa operazione può essere effettuata dall'utente, dalla CA stessa, o da una terza parte)
- l'utente sottopone una richiesta di certificato alla CA
- la CA crea un certificato digitale che contiene la chiave pubblica dell'utente ed i dati identificativi
- il certificato digitale è firmato elettronicamente con la chiave privata della CA



# Revoca di un certificato

Un certificato non può essere valido per sempre, così come accade con un documento di riconoscimento, una carta di identità o un passaporto: un'informazione fondamentale che deve avere un certificato elettronico è la scadenza; questa è sempre l'informazione che viene controllata per prima, chiunque sia il titolare del certificato.

Tuttavia, anche nel periodo di validità di un certificato possono cambiare tante cose, per cui deve essere previsto un meccanismo di revoca, sia su richiesta del titolare, sia a seguito di una decisione dell'autorità di certificazione che lo ha firmato.

Infatti, il titolare del certificato potrebbe trovarsi in una condizione diversa rispetto a quella in cui si trovava nel momento del rilascio del certificato, per cui i dati in esso contenuti potrebbero non corrispondere più; dall'altra parte, l'autorità di certificazione potrebbe avere verificato un utilizzo irregolare del certificato e di conseguenza potrebbe decidere il suo ritiro.

Le ragioni che possono portare alla revoca di un certificato sono:

- Compromissione chiave privata, ovvero sia presente uno dei seguenti casi:
  - sia stato smarrito o rubato il dispositivo di firma che contiene la chiave;
  - sia venuta meno la segretezza della chiave o del suo codice di attivazione (PIN);
  - si sia verificato un qualunque evento che abbia compromesso il livello di affidabilità della chiave;
- Informazioni non più valide (es. cambiamento dei dati del titolare presenti nel certificato)

# Certificate Revocation List

La CA aggiorna e archivia l'elenco dei certificati revocati in una ***Certificate Revocation List*** (CRL) che è una lista, firmata dalla CA, contenente i numeri seriali dei certificati emessi revocati (ma non ancora scaduti), la data di quando è avvenuta la revoca ed, eventualmente, i motivi della revoca: alla CRL è associata la data di ultimo aggiornamento della lista.

# Chi controlla se un certificato è in CRL?

Nella configurazione di Mozilla si può trovare il flag:

**“Interroga risponditori OCSP per confermare la validità attuale dei certificati”**

**OCSP: Online Certificate Status Protocol (*OCSP*)**  
è un protocollo che permette di verificare la validità di un certificato senza ricorrere alle liste di revoca dei certificati.

# Legittimità di un certificato

Per poter verificare la legittimità di un certificato che ci viene presentato bisogna risalire la catena di certificazioni.

Se, ad esempio, si possiede un certificato di Tizio, emesso dalla CA D, per verificarne l'autenticità occorre:

- procurarsi il certificato dell'autorità D (per estrarne la chiave pubblica ed utilizzarla per verificare la firma sul certificato di Tizio)
- il certificato dell'autorità D è stato emesso dall'autorità B per cui o si dispone di detto certificato altrimenti occorre procurarselo e risalire alla Root CA per poter verificare l'autenticità di B
- una volta ottenuto (ed avendone verificato l'autenticità) il certificato di B può essere usato per verificare l'autenticità del certificato di D
- una volta verificata l'autenticità del certificato di D, si può con esso verificare l'autenticità del certificato di Tizio.

Generalizzando, per verificare la validità di un generico certificato di x:

- si decifra l'hash del certificato di x con la chiave pubblica della CA di x (ricavata dal certificato di quest'ultima)
- si calcola l'hash del certificato di x
- si controlla che i due hash, quello calcolato e quello decifrato, siano uguali

Ovviamente, per verificare che il certificato sia valido, occorre anche controllare la scadenza del certificato e controllare presso la base di dati della CA che ha emesso il certificato che questo non sia presente nelle liste di revoca.



# Uso dei certificati digitali

## Uso dei certificati digitali

Gli usi più importanti dei certificati digitali sono i seguenti:

- certificati SSL (per comunicazioni client/server) → utilizzati per stabilire l'identità del client e del server che comunicano tra loro; il server “presenta” il suo certificato al client per autenticarsi, successivamente il client “presenta” il suo certificato al server per autenticarsi (il processo di autenticazione fa uso della crittografia a chiave pubblica e della firma digitale): una volta che client e server si sono autenticati usano la crittografia a chiave simmetrica (più veloce rispetto a quella asimmetrica) per criptare le informazioni che si scambiano;
- certificati S/MIME → utilizzati per firmare e criptare e-mail ; una e-mail che include una firma digitale assicura il ricevente sulla esatta identità del mittente e sull'integrità del messaggio trasmesso; infatti nel caso vi sia una qualche discordanza tra il messaggio di partenza e quello ricevuto la firma allegata a quest'ultimo non può essere validata;
- certificati object-signing → utilizzati per firmare codice Java, JavaScript, o altri tipi di file; una ditta produttrice di software, ad esempio, può, firmando i suoi prodotti, garantire i suoi clienti della autenticità del prodotto.
- commercio elettronico → si avvale dei certificati digitali nell'esecuzione delle transazioni commerciali in quanto permette ai due (o più) contraenti di identificarsi in modo inequivocabile

Quando compare questa immagine?



**Questa connessione non è affidabile**

È stata richiesta a Firefox una connessione sicura con [certificati SSL](#), ma non è possibile confermare la sicurezza del collegamento.

Normalmente, quando si cerca di attivare un collegamento in modalità sicura, il sito web fornisce un'identificazione affidabile per garantire all'utente che sta visitando il sito corretto. Tuttavia l'identità di questo sito non può essere verificata.

## Che cosa dovrei fare?

Se generalmente è possibile collegarsi a questo sito senza problemi, è possibile che questo errore sia causato dal tentativo da parte di qualcuno di sostituirsi al sito originale. Il consiglio è di non proseguire la navigazione.

[Allontanarsi da questo sito](#)

- ▶ **Dettagli tecnici**
- ▶ **Sono consapevole dei rischi**

# Certificati SSL

I **certificati digitali** vengono utilizzati per **attestare l'identità di un sito web** (così come di un soggetto, di una società, di un sistema e così via). Sul web, l'uso più comune dei certificati digitali è per l'accesso ai siti attraverso il protocollo **HTTPS** (ossia **HTTP** con l'aggiunta del protocollo crittografico **SSL**). **Ricorrendo all'impiego dei certificati digitali**, il browser web può accertarsi che il server a cui si è connessi sia autentico ossia che corrisponda effettivamente a quello che dichiara di essere. Se il certificato è stato firmato da un'autorità di certificazione riconosciuta, **il browser provvede ad utilizzare la chiave pubblica indicata nel documento digitale per scambiare dati in modo sicuro, senza la possibilità che vengano in qualche modo intercettati (VERY IMPORTANT)**

# Certificati SSL

Il protocollo **HTTPS**, insieme con un certificato digitale valido, viene quindi utilizzato da tutti i siti web che permettono la gestione di dati particolarmente importanti od informazioni sensibili (si pensi ai siti di e-commerce ed ai servizi online dei vari istituti di credito...).

Più di recente, anche i principali social network hanno iniziato ad abbracciare HTTPS: Facebook e Twitter su tutti mentre Google ha esteso l'utilizzo del medesimo protocollo a tutti i suoi servizi.

Abilitando l'utilizzo di **HTTPS**, ***i dati non viaggiano più "in chiaro" ma sono crittografati*** così da impedire l'"intercettazione" da parte di terzi di tutti i contenuti inviati e ricevuti.



# Certificati SSL

Le più recenti versioni dei vari browser web ben evidenziano quando si sta utilizzando una connessione HTTP e quando, invece, ci si è collegati ad una pagina web che fa uso del protocollo HTTPS. Sia Internet Explorer, sia Chrome, sia Firefox, ad esempio, espongono un lucchetto nella barra degli indirizzi insieme con l'indicazione https: ogni qualvolta si utilizzi una pagina che utilizza un certificato digitale e che quindi provvede a crittografare i dati scambiati tra client e server (e viceversa)

# Certificati SSL

Quando il certificato viene ritenuto valido e rilasciato da un'autorità riconosciuta (i certificati digitali possono essere creati anche in modo autonomo ma in questo caso sono sprovvisti delle firme che ne attestano la veridicità e l'affidabilità), tutti i vari browser espongono – nella barra degli URL – l'icona del lucchetto.

I certificati generati in modo autonomo seguono il formato PGP/GPG.

# PKI

L'insieme di utenti e authority, nonché delle tecnologie coinvolte che queste utilizzano prendono il nome di **PKI** (Public Key Infrastructure).

La **PKI** può essere pubblica o privata. I servizi offerti sono regolati da accordi commerciali

# Posta elettronica certificata

La **Posta Elettronica Certificata** (PEC) è il sistema che consente di inviare e-mail con **valore legale equiparato ad una raccomandata con ricevuta di ritorno**, come stabilito dalla vigente normativa (DPR 11 Febbraio 2005 n.68).

Benché il servizio PEC presenti forti similitudini con la tradizionale Posta Elettronica, è doveroso dare risalto alle **caratteristiche aggiuntive**, tali da fornire agli utenti la certezza – a valore legale - dell'invio e della consegna (o della mancata consegna) delle e-mail al destinatario.

La Posta Elettronica Certificata ha il medesimo valore legale della raccomandata con ricevuta di ritorno con **attestazione dell'orario esatto di spedizione**.

Inoltre, il sistema di Posta Certificata, grazie ai **protocolli di sicurezza** utilizzati, è in grado di **garantire la certezza del contenuto** non rendendo possibili modifiche al messaggio, sia per quanto riguarda i contenuti che eventuali allegati.

# Posta elettronica certificata

**La Posta Elettronica Certificata garantisce - in caso di contenzioso - l'opponibilità a terzi del messaggio.**

Il termine "Certificata" si riferisce al fatto che il gestore del servizio rilascia al mittente una **ricevuta** che costituisce **prova legale** dell'avvenuta spedizione del messaggio ed eventuali allegati. Allo stesso modo, il gestore della casella PEC del destinatario invia al mittente la **ricevuta di avvenuta consegna**.

I gestori certificano quindi con le proprie "ricevute" che il messaggio:

- E' stato spedito
- E' stato consegnato
- Non è stato alterato

# Differenza tra la raccomandata con ricevuta di ritorno ed il servizio di PEC

Il servizio di **PEC** consente di effettuare l'invio di documenti informatici avendo la garanzia di "certificazione" dell'invio e dell'avvenuta (o mancata) consegna. Il servizio ha, pertanto, tutti i requisiti della **raccomandata con A/R** cui si aggiungono notevoli vantaggi sia in termini di tempo che di costi. In particolare, nella PEC si riscontra:

- semplicità ed economicità di trasmissione, inoltro e riproduzione;
- semplicità ed economicità di archiviazione e ricerca;
- facilità di invio multiplo, cioè a più destinatari contemporaneamente, con costi estremamente più bassi rispetto a quelli dei mezzi tradizionali;
- velocità della comunicazione ed inoltre non è necessaria la presenza del destinatario per completare la consegna;
- possibilità di consultazione ed uso anche da postazioni diverse da quella del proprio ufficio o abitazione (basta un qualsiasi PC connesso ad Internet e un normale browser web), ed in qualunque momento grazie alla persistenza del messaggio nella casella di posta elettronica;
- presenza nella ricevuta di avvenuta consegna, diversamente dalla raccomandata, anche dei contenuti del messaggio originale.

# Le modalità di utilizzo della PEC sono diverse da quelle di una normale posta elettronica?

Le modalità di accesso sono sostanzialmente le stesse. Si può accedere alla propria casella di PEC, infatti, sia attraverso un client di posta elettronica che attraverso un browser Internet.

Nel primo caso, prima di poter utilizzare la propria casella sarà necessario configurare il proprio client con i parametri forniti dal Gestore di PEC scelto.

Quali caratteristiche ha in più la PEC rispetto all'e-mail tradizionale? La PEC, per quanto in apparenza simile al servizio di posta elettronica "tradizionale", offre un servizio più completo e sicuro, prevedendo:

- livelli minimi di qualità del servizio e di sicurezza stabiliti dalla legge;
- certificazione dell'invio e della consegna del messaggio;
- l'opponibilità a terzi delle evidenze relative alle operazioni di invio e ricezione di un messaggio.

## **In quali casi è preferibile inviare messaggi di PEC?**

La casella di PEC è indicata soprattutto per effettuare comunicazioni “ufficiali” per le quali il mittente vuole avere delle evidenze con valore legale dell’invio e della consegna del messaggio.

## **In che modo si ha la certezza della consegna di un messaggio di PEC?**

Nel momento in cui l’utente invia il messaggio, riceve, da parte del proprio Gestore di PEC, una ricevuta di accettazione con relativa attestazione temporale. Tale ricevuta costituisce prova legale dell’avvenuta spedizione del messaggio. Allo stesso modo, quando il messaggio perviene nella casella del destinatario, il suo gestore di PEC invia al mittente la ricevuta di avvenuta (o mancata) consegna, con l’indicazione di data ed orario, a prescindere dalla visualizzazione del messaggio da parte del destinatario.



## **La PEC certifica la lettura del messaggio da parte del destinatario?**

No, la certificazione è relativa ai soli eventi di invio del messaggio e di consegna dello stesso nella casella di PEC del destinatario.

## **La PEC è in grado di garantire l'identità della casella mittente?**

Sì, in quanto è assicurata l'inalterabilità dell'indirizzo associato alla casella dalla quale si effettua l'invio del messaggio.

Inoltre il soggetto che intende richiedere un servizio di PEC deve presentare al Gestore, oltre alla richiesta di attivazione del servizio, anche un documento che attesti la sua identità diventando quindi titolare del servizio.

**Da una casella di PEC è possibile inviare un messaggio certificato a chiunque abbia una casella di posta elettronica?**

Sì, in questo caso il mittente disporrà delle attestazioni circa l'invio del messaggio. Nel caso in cui anche il destinatario sia dotato di una casella di Posta Elettronica Certificata, oltre alle garanzie sull'invio del messaggio, il mittente disporrà delle attestazioni di avvenuta consegna.

**E' possibile inviare messaggi di Posta Elettronica Certificata tra utenti che utilizzano Gestori di PEC differenti?**

Sì, la normativa impone ai differenti gestori di PEC di garantire la piena interoperabilità dei servizi offerti.

**Il destinatario di un messaggio di Posta Elettronica Certificata può negare di averlo ricevuto?**

Nel caso in cui il messaggio sia stato effettivamente consegnato, il destinatario non può negare l'avvenuta ricezione, dal momento che la ricevuta di avvenuta consegna del messaggio, firmata ed inviata al mittente dal Gestore di PEC scelto dal destinatario, riporta la data e l'ora in cui il messaggio è stato consegnato nella casella di PEC del destinatario, certificandone l'avvenuta consegna.

# Da fare

<https://www.youtube.com/watch?v=Y3waHkSqAf4>

Smartcard

https

Posta certificata