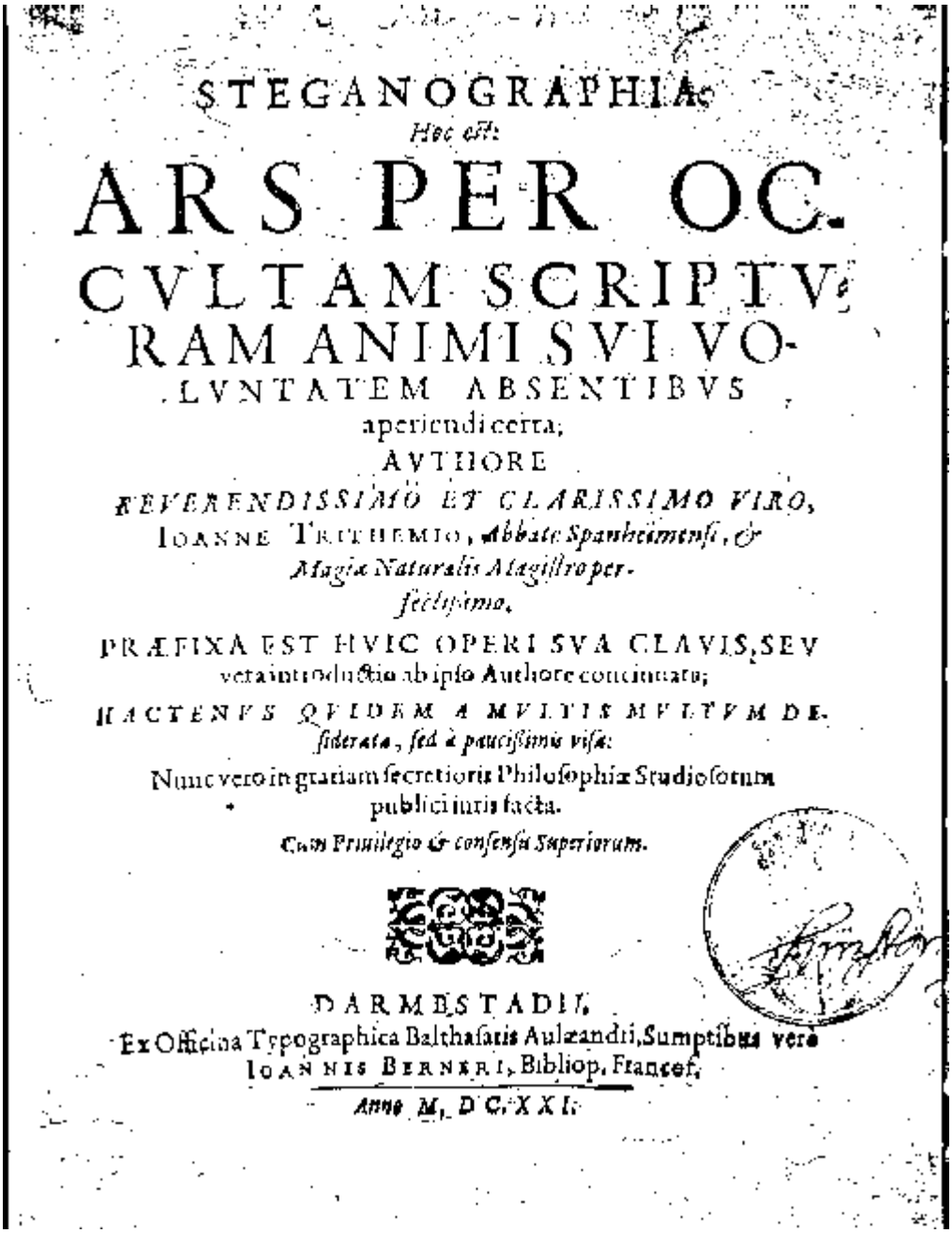
Tecniche di crittografia
Crittografia e firma digitale

La parola crittografia ha origine greca e significa nascosto. Un'altra parola correlata è **steganografia** che significa scrittura nascosta. Un esempio legato all'antichità è di scrivere messaggi segreti non sull'argilla che ricopriva le tavolette, ma sulle stesse tavolette che venivano poi ricoperte d'argilla e sembravano non usate. Della steganografia l'ababe Tritemio (1500 d.C.) è forse uno dei più noti autori.



STEGANOGRAPHIAE
Hoc eſt:
ARS PER OC.
CVLTAM SCRIPTV
RAM ANIMI SVI VO-
LVNTATEM ABSENTIBVS
aperiendi certa;
AVTHORE
REVERENDISSIMO ET CLARISSIMO VIRO,
IOANNE TRITHEMIO, *Abbate Spanheimenſi, &*
Magiæ Naturalis Magiſtro per-
fectiſſimo.
PRÆFIXA EST HVIC OPERI SVA CLAVIS, SEV
vera introductio ab ipſo Authore concinnata;
HACTENVS QVIDEM A MVLTIS MVLTVM DE-
ſiderata, ſed à pauciſſimis viſa:
Nunc vero in gratiam ſecretioris Philoſophiæ Studioſorum
publici iuris facta.
Cum Priuilegio & conſenſu Superiorum.
DARMESTADII,
Ex Officina Typographica Balthaſaris Aulæandri, Sumptibus vero
IOANNIS BERNERI, Bibliop. Francof.
Anno M. DC. XXI.

Quindi la trasformazione del messaggio al fine di renderne incomprensibile il significato, è stato, è e sarà lo stratagemma tramite il quale raggiungere uno degli obbiettivi sella sicurezza: la **confidenzialità**.

In particolare definiamo:

- **Crittologia**: disciplina che tratta delle scritture segrete, dei documenti in cifra.

- **Crittografia**: insieme delle tecniche che consentono di realizzare la cifratura di un testo e la decifratura di un crittogramma.
- **Crittoanalisi**: disciplina che studia come forzare i cifrari.

Sicurezza dell'informazione

Il concetto di informazione viene associato ad una quantità ben definita. Per introdurre la crittografia, è necessario comprendere preliminarmente le caratteristiche relative alla sicurezza dell'informazione.

La sicurezza dell'informazione si manifesta in molti modi a seconda delle situazioni e dei requisiti. In ogni caso, le parti di una transazione devono essere rassicurate che determinati scopi della *information security* siano raggiunti. Abbiamo elencato i principali obbiettivi nella tabella seguente.

privacy o confidenzialità	mantenere segrete le informazioni a tutti tranne ai coloro autorizzati a vederle.
integrità dati	garantire che le informazioni non siano alterate tramite mezzi non autorizzati sconosciuti.
autenticazione o identificazione di entità	verifica dell'identità di una entità (es., persona, *computer*, carta di credito, ...).
autenticazione dei messaggi	verifica della sorgente delle informazioni.
firma	un mezzo per collegare inscindibilmente l'informazione ad una entità.
autorizzazione	convenienza, ad un'altra entità, di poter eseguire una operazione.
validazione	un mezzo per fornire linee temporali di autorizzazione all'utilizzo o manipolazione di informazioni o risorse.
controllo d'accesso	restringere l'accesso alle risorse per le entità privilegiate.
certificazione	certificazione di informazione tramite entità di fiducia.
timestamping	registrazione dell'istante di creazione o esistenza dell'informazione.
witnessing	verificare la creazione o l'esistenza di informazioni tramite una entità differente dal creatore della stessa.
receipt	riscontro che l'informazione è stata ricevuta.
confirmation	riscontro che il servizio è stato fornito.
ownership	un mezzo per fornire ad una entità il diritto legale di utilizzare o trasferire risorse ad altri.
anonimato	nascondere l'identità di una entità coinvolta in un processo.
non ripudio	prevenire la negazione di una precedente azione.
revoca	revoca di un certificato o autorizzazione.

Nel secolo appena trascorso, sono stati sviluppati insiemi di protocolli e meccanismi molto elaborati ai fini della sicurezza delle informazione quando questa era veicolata su documenti fisici.

Spesso l'obiettivo della *information security* non può essere garantito solo da algoritmi matematici e protocolli, ma richiede tecniche procedurali e legali per risaltare il risultato desiderato. La sicurezza fisica del contenitore dell'informazione è, per necessità pratica, limitata e quindi a, in alcuni casi, la sicurezza viene procurata attraverso il documento che ospita l'informazione e non dall'informazione stessa. Basti pensare alla carta-moneta, che richiede inchiostri speciali e materiali per prevenire la

contraffazione.

Obiettivi

Prima di iniziare la reale trattazione dell'argomento crittografia, sarà utile puntualizzare quali siano gli scopi e gli obbiettvi da raggiungere con tale strumenti.

La figura evidenzia gli obiettivi direttamente raggiungibili con gli strumenti crittografici, e dai quali trarre spunto per ottenere la realizzazione degli scopi di protezione dei sistemi, quindi delle applicazioni e dei sistemi.

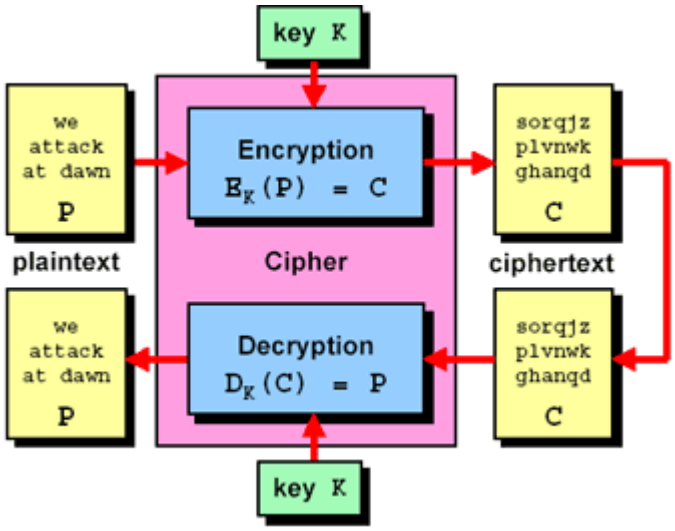
Terminologia

Presentiamo una serie di terminologie attuali rispetto alle tecniche di cifratura:

- Un messaggio è un ***plaintext*** (o ***cleartext***), cioè testo in chiaro. Il processo di alterazione di un messaggio in un qualche modo al fine di nasconderlo, in sostanza è definito cifratura (*encryption*).
- Un messaggio è cifrato o ***ciphertext***. Il processo che restituisce li *plaintext* viene chiamato decifratura (***decryption***).

E riguardo algoritmi e chiavi:

- un algoritmo crittografico (***cipher***), è la funzione matematica utilizzata per la cifratura e decifratura(***encryption*** e ***decryption***).
- La sicurezza di un algoritmo di cifratura moderno è basata sulla chiave segreta. Questa chiave può essere scelta in un insieme di valori. Il campo dei possibili valori delle chiavi viene definito spazio delle chiavi (***keyspace***).
- Sia la cifratura che la decifratura sono operazioni che dipendono dal valore della chiave K e ciò è denotato dal fatto che K è espresso nella funzione EK(P) = C e DK(C) = P.



Considerazioni storiche 1

A volte il confine tra **lecito ed illecito** risulta essere molto sottile. Ci si potrà trovare di fronte alla necessità di scambiare informazioni confidenziali con altri o anche semplicemente di conservare dati in possesso in modo che solo noi si possa accedervi.

Il problema per chi utilizza la crittografia convenzionale è che di fronte all'esigenza di scambiare informazioni con altri in un canale definibile insicuro (come è Internet), non si hanno a disposizione altri canali sicuri (potrebbe essere ad esempio un appuntamento di persona in un luogo riservato) per comunicare quale è la parola chiave o altre informazioni che permettano l'accesso al *file* cifrato.

Grazie a un metodo introdotto pubblicamente (prima era sviluppato soltanto su commissione da società specializzate) da *Philip Zimmermann*, tale inconveniente è stato ovviato.

Zimmermann ha avuto la brillante idea di sviluppare un programma gratuito che permettesse a tutti di utilizzare un metodo nuovo unito ad una elevata sicurezza che la lunghezza delle chiavi offre, usando la crittografia a **chiave pubblica**. è questo una dei fattori che hanno contributo alla diffusione così capillare del noto programma ***P**retty **G**ood **P**rivacy*, chiamato comunemente **PGP**.

Il punto cruciale su cui si basa il pensiero di *Zimmermann* è il seguente:

premesso che la privacy è un diritto dell'uomo che sta alla base delle società tecnologicamente evolute e deve essere tutelato, accade sempre più frequentemente con l'introduzione di nuove tecnologie che tale diritto non venga riconosciuto; è indispensabile quindi utilizzare un metodo per garantire la riservatezza, sia che si utilizzi la crittografia per spedire gli auguri di buon anno, piuttosto che un manuale su come costruire una bomba atomica e PGP è un programma che permette di farlo.

Criminalizzare l'utilizzo di Internet abbinato alla crittografia perchè non controllabile sarebbe come vietare di utilizzare il telefono come mezzo per organizzare attività illecite; sta all'utente in ogni caso deciderne l'uso.

Considerazioni storiche 2

La crittografia come modifica volontaria del testo esisteva già al tempo degli egiziani nel 1900 a.C. (tomba del faraone Knumotete II).

AVANZARE TRA DUE GIORNI
ALL ALBA VERSO IL FIUME

ADRARI ZAOLEF NRILVL ATGAAIEVEEIBOMARUNLSU

Gli Spartani per cifrare un messaggio segreto di tipo militare usavano, 2500 anni fa, una striscia di papiro avvolta a spirale attorno ad un bastone (che costituirà la chiave di decodifica). Una volta scritto il messaggio in verticale sul papiro questo veniva consegnato al destinatario che, con un bastone dello stesso diametro poteva leggere il messaggio in chiaro. Questo metodo è di **trasposizione** perchè il messagggio è in chiaro ma l'ordine delle lettere è da scoprire.

Se ne trovano altre tracce a partire da alcuni scritti storici riguardanti Giulio Cesare (*Vite dei dodici Cesari* di Svetonio) che in luogo di ogni lettera scriveva quella situata tre posizioni oltre nell'alfabeto: **sostituzione**. Una crittografia così semplice si decifra facilmente: si calcola la frequenza delle lettere usate e si confronta con quella delle lettere nella lingua che si suppone usata nel testo in oggetto. Se il testo non è molto breve, si riconoscono subito le equivalenze di tre o quattro lettere più frequenti. Poi si fanno ipotesi e controlli successivi e presto si arriva alla conclusione. Le frequenze delle lettere variano da un testo all'altro, ma si spostano di poco nell'ordine decrescente.

Esempio: pippo con chiave 3 diventa snssr.

Questo metodo è facilmente attaccabile perchè basta confrontare la frequenza delle lettere nella lingua italiana con la frequenza dei simboli usati nel messaggio cifrato. Bisogna inoltre considerare che le chiavi possibili sono solo 26, quindi con un *brute force* si potrebbe scovare la chiave. Data la bassa complessità del metodo usato da Cesare è chiaro che non fosse infallibile, ma dati i risultati militari è stato efficace! Il metodo di Cesare ha ispirato un sistema usato ancora oggi, il ROT-13 dove la chiave è appunto 13, quindi A->N, B->O, etc.

Il metodo che usavano gli ebrei è detto ATBASH. La sostituzione avviene utilizzando questa tabella dove le lettere della seconda riga sono scritte in ordine decrescente:

```
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
z y x w v u t s r q p o n m l k j i h g f e d c b a
```

Messaggio: `Il Libro di Geremia`
Testo Cifrato: `Ro Oryil wr Tvivnrz`

Lo storico greco Polibio sviluppò una tecnica di codifica legando le lettere a una coppia di numeri che ne indicava la posizione in una tabella. La coppia di numeri era comunicata nella notte attraverso delle torce. Ecco un esempio di tabella:

```
   1 2 3 4  5
 1 a b c d  e
```

```
2 f g h ij k
3 l m n o  p
4 q r s t  u
5 v w x y  z
```

pippo diventa (3,5) (2,4) (3,5) (3,5) (3,4)

Se la disposizione delle lettere nella tabella non seguono l'ordine alfabetico si capisce la difficoltà di trovare la chiave che in questo caso è la tabella.

L'imperatore romano Augusto usava invece un altro interessante metodo di sostituzione usando come chiave un'altra parola o frase. La chiave e il testo avevano un corrispettivo numerico, il testo cifrato risultava una sfilza di numeri ottenuti come somma fra testo e chiave. Se la somma (valore cifrato) eccede 21 si ricomincia dalla a; ciò equivale a fare una somma modulo 21. La decifrazione è una semplice sottrazione. La crittoanalisi di questo metodo non beneficia della frequenza delle lettere della lingua usata. Questo metodo può essere attaccato con un *brute force* utilizzando un dizionario di parole. Ovviamente ci si può difendere non usando come chiave parole di senso compiuto, ma un insieme di lettere generate in maniera casuale.

Considerazioni storiche 3

Dopo il fiorire dell'arte della cifratura in Medio Oriente, nel rinascimento si torna in Occidente per dare supporto al rifiorire degli scambi commerciali internazionali e alle innumerevoli tresche politiche, soprattutto in Italia. Lo sforzo propinato nel cifrare i documenti, era bilanciato da quello per intercettare e ricostruire i messaggi di terzi.

Possiamo enumerare il primo trattato noto sulla crittologia di Cicco Simonetta, presso la cancelleria degli Sforza nel XV secolo. Sempre in tale epoca, **Leon Battista Alberti**, architetto letterato e crittologo dilettante, scrisse un trattato dal nome *De Cifris* nel quale introdusse il suo **sistema polialfabetico** che per tre secoli, seppur attribuito ad altri autori, costituì il basamento dei sistemi crittografici. Il suo sistema usa diversi alfabeti spostati rispetto a quello del testo originario saltando dall'uno all'altro ogni due o tre parole. L'alfabeto da usare viene indicato da una lettera in chiaro che, da quel punto in poi, mittente e destinatario impostano in modo da fissare una certa posizione mutua fra due cerchi concentrici ruotanti, che riportano l'intero alfabeto. Lo strumento è costituito da due dischi che ruotano l'uno sull'altro. Sul disco esterno sono riportati numeri e lettere normali, su quello interno i relativi segni cifrati. Per creare una chiave, si ruota il disco e si fa corrispondere una lettera M e un'altra lettera prestabilita.

Altro esempio di cifratura con il sistema del polialfabeto è quello inventato da un tedesco

contemporaneo dell'Alberti: **Johannes Trithemius**. Il sistema fa uso di una tabella che si chiama *tabula recta*, formata da 26 righe (tante quante le lettere dell'alfabeto inglese) riportanti ognuna un alfabeto scalato di una posizione rispetto a quello precedente. La tabella si usa così: la prima lettera da cifrare rimane la stessa, la seconda si cifra con il secondo alfabeto, la terza lettera userà il terzo alfabeto e così via fino a ricominciare dal primo alfabeto dopo la ventiseiesima lettera. Per rendere difficile il lavoro dei crittoanalisti si può usare un alfabeto disordinato o, meglio ancora (è più facile da ricordare e comunicare al destinatario del messaggio) una frase chiave.

Nel 1553, un altro famoso interprete di questa arte era **Giovan Battista Bellaso**, che pubblicò una serie di cifrari, uno dei quali ripreso all'inizio del secolo odierno per l'uso con le telescriventi. Il più noto sistema di Belaso prevede di usare una parola anziché una lettera; questa parola va scritta al di sopra del testo da crittografare. Ogni singola lettera della chiave determina un alfabeto circolare che inizia da quella stessa lettera. Se quindi la prima lettera sarà una V, l'alfabeto corrispondente sarà Vwxyzabcd.... Per cifrare quindi si dovrà vedere quale posizione occupa la lettera nell'alfabeto normale (es. una E occupa il quinto posto) e sommarla nell'alfabeto circolare creato. Questa sarà la lettera da scrivere nel messaggio cifrato.

Se ogni volta si cambia la chiave generandola casualmente, stiamo usando un metodo chiamato *One-Time Pad*, che è difficile da superare senza conoscere la chiave e generando in maniera davvero casuale la chiave (e non è facile perchè si può ottenere solo osservando fenomeni casuali naturali). Occorre notare che non bisogna usare questo metodo per due diversi messaggi con la stessa chiave perchè la differenza tra testo cifrato e testo in chiaro è uguale e, in unione a un *brute force*, aiuta di parecchio la crittoanalisi di questo metodo.

Considerazioni storiche 4

L'uso della crittografia continua intensificandosi sempre di più e migliorandosi con il tempo fino ad avere importanza tale da cambiare il corso della storia durante le due guerre mondiali quando appaiono le prime macchine elettriche per cifrare i messaggi ma, soprattutto, per la crittoanalisi.

I tedeschi usarono per tutta la seconda guerra mondiale una macchia chiamata **Enigma** che avrebbe dovuto cifrare i messaggi in maniera sicura. Così non successe, perchè inglesi e polacchi unendo le loro forze furono in grado di decifrare quasi tutti i messaggi intercettati. L'Enigma era una macchina elettromeccanica con contatti, lampadine, rotori e una tastiera. Ogni lettera veniva cifrata con un alfabeto diverso dando luogo ad un numero così elevato di combinazioni da rendere la decodifica teoricamente impossibile per l'epoca. Ma ciò non fermò gli inglesi che trassero grande beneficio dai messaggi decodificati.

	V.24/V.28	V.35	V36
meccaniche	ISO 2110 o 4902	ISO 2593	ISO 4902
elettriche	V.28	V.10 eV.11.	V.10 e V.11
funzionali	V.24	V.24	V.24
procedurali	V.24, V.25 o V.25 bis		

	X.20	X.20bis	X.21	X.21 bis
meccaniche	ISO 4903	ISO 2110	ISO 4903	ISO 2110
elettriche	X.26 o X.27	V.28	X.26 o X.27	V.28
funzionali	X.24	V.24	X.24	V.24
procedurali	X.20	X.20bis	X.21	X.21 bis

Algoritmi semplici e chiave lunghe ...

Per anni la regola generale è stata di creare algoritmi semplici e di impiegare chiavi molto lunghe per rendere difficile la vita al crittoanalista.

... oppure viceversa

Oggi l'orientamento è opposto vista la potenza di calcolo di cui si può disporre per fare un *brute force*, quindi si creano algoritmi complicatissimi da decifrare in modo che anche se il nostro avversario avesse parecchio materiale su cui condurre un'analisi, gli sarebbe pressocchè inutile.

Oggi la crittografia è utilizzata per il commercio elettronico, l'*autenticazione*, la *riservatezza* delle informazioni, l'*integrità* etc. Uno dei presupposti fondamentali è che si suppone che il crittoanalista di turno conosca in generale il nostro metodo di crittografia, questo perchè sarebbe davvero un disastro cambiare metodo di crittografia ogni qual volta si ha il sospetto che qualcuno sia riuscito a infrangerlo. Da questo presupposto segue che i metodi basano la loro forza sulle chiavi di codifica e decodifica.

Se la chiave è la stessa sia per la codifica che per la decodifica ricadiamo nel caso delle crittografia classica: questi sono i metodi a **chiave simmetrica** o segreta.

Gli algoritmi a **chiave asimmetrica** o pubblica (che risalgono agli anni '70) utilizzano coppie di chiavi complementari. Una delle due chiavi è pubblica e conosciuta da tutti. Le chiavi vanno distribuite a coppie e quindi solo una chiave può decifrare il messaggio generato utilizzando la chiave a lei complementare. In questo modo possiamo trasmettere tranquillamente la nostra chiave pubblica senza paura che venga intercettata. In ogni modo l'argomento verrà approfondito nell'apposito **paragrafo**.

Spesso ci si orienta per metodi ibridi simmetrici-asimmetrici (come succede nel famoso PGP) perchè il solo metodo asimmetrico non è efficiente (è lento!) per grandi moli di dati e, se dovessimo inviare lo stesso messaggio a più persone, dovremmo cifrarlo ogni volta con la giusta chiave.

Cifratura simmetrica

La sicurezza di un crittosistema, o cifrario, deve dipendere solo dalla segretezza della chiave e non dalla segretezza dell'algoritmo usato.

Jean Guillaume Hubert Victor Francois Alexandre Auguste Kerckhoffs von Nieuwenhof (1835-1903), filologo olandese, La Crittographie Militaire [1883]

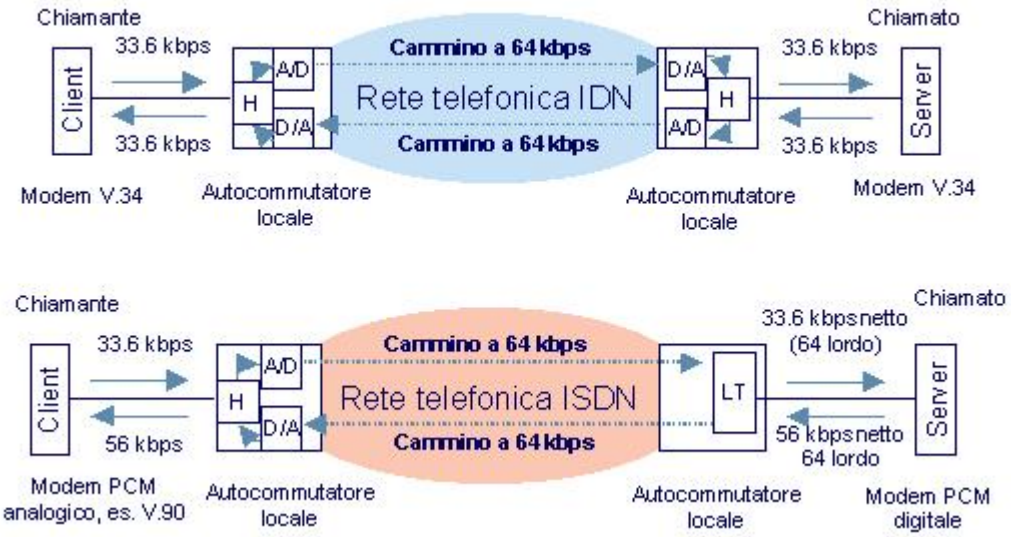
Il crittoanalista conosce sempre il crittosistema che è stato usato e gli algoritmi che il sistema prevede. Solo la chiave è segreta.

La crittografia moderna si incentra prevalentemente su algoritmi basati su sostituzioni, scambi fatti in relazione a tabelle. Tuttavia, durante il passaggio dagli scambi monoalfabetici alla crittografia moderna, è stata ampliamente utilizzata (e lo è ancora) la crittografia mediante l'utilizzo di cifrari polialfabetici. Anche questi tramontarono quando si comprese che, analizzando le statistiche delle ripetizioni di caratteri nel testo ed in seguito a vari tentativi, si poteva poco a poco ottenere la chiave segreta procedendo per tentativi.

Questa tipologia di crittoanalisi ha senso se le chiavi rimangono di lunghezza inferiore al testo da cifrare (diversi decenni fa era inutile e difficile di trasportare chiavi di lunghezza superiore o uguale al testo, ora le cose sono cambiate). Attualmente possono essere ancora usati gli algoritimi polialfabetici se dipendenti da una chiave segreta di lunghezza infinita (matematicamente e intuitivamente è dimostrata la loro inattaccabilità); una **chiave di lunghezza infinita**, magari generate in modo pseudo-casuale in modo dipendente da una *password*, non serve se più lunga del testo da cifrare.

Chiave simmetrica

La cifratura a chiave simmetrica, richiede che mittente e destinatario di una determinata comunicazione utilizzino la stessa parola chiave per decifrare il messaggio in oggetto. Algoritmo e chiave sono le due componenti principali di ogni sistema di crittografia, componenti che permettono il passaggio dal messaggio in chiaro al messaggio cifrato e viceversa. Esistono due tipi di 'crittosistemi' che si basano su chiavi o codici fondamentalmente diversi tra loro. Vengono definiti crittosistemi a **chiave segreta o simmetrica** (**secret-key**) e **chiave pubblica** (***public-key***).



Nel primo sistema viene usata una sola chiave, detta appunto segreta, utilizzata come parametro di una funzione univoca e invertibile permettendo così di elaborare il testo del messaggio da trasmettere rendendolo incomprensibile agli intercettatori. Essendo la funzione invertibile, il destinatario dovrà soltanto elaborare nuovamente il crittogramma richiamando l'inversa della funzione di cifratura avente come parametro la stessa chiave utilizzata dal trasmettitore del messaggio.

La tecnica si basa sulla capacità del mittente e del destinatario di mantenere segreto il codice di cifratura. Tale metodo, noto da secoli, è definito crittografia simmetrica mentre la crittografia a chiave pubblica, relativamente recente in quanto risale agli anni '70, viene anche definita crittografia asimmetrica. Quest'ultima a differenza della precedente, utilizza due chiavi distinte: una per cifrare il messaggio e l'altra per decifrarlo.

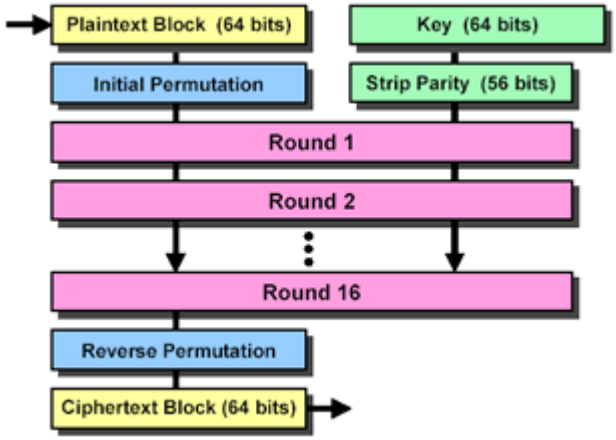
Con il sistema a chiave simmetrica il mittente e il destinatario devono raggiungere un accordo sulla scelta della chiave.
A rendere più complessa l'operazione è il fattore distanza, tipicamente tale da impedire lo scambio di persona della chiave. Come potranno allora scambiarsi la chiave?

Il problema esposto non risulta di semplice soluzione, poiché implica il coinvolgimento di una terza parte che ha il compito di distribuire la chiave accordata. Quest'ultima soluzione diviene presto improponibile nel momento in cui a fare uso di tale metodo è una struttura dalle dimensioni notevoli. Allora si sostituiscono le consegne con terze parti con trasmissioni cifrate con una chiave principale contenenti la chiave da trasmettere (chiave di sessione). Ancora possiamo dire che non vi è sufficiente sicurezza con questo metodo.

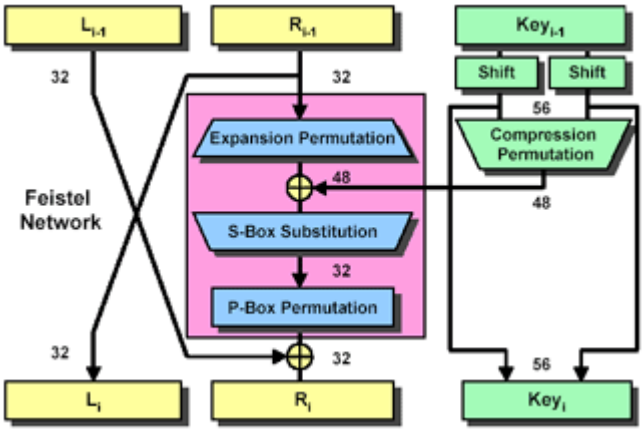
DES, Data Encrytion Standard

Standard federale ancora oggi ufficiale (nella versione triplo-des) per gli USA, è nato nel 1977 per implementazioni per lo più *hardware* come derivazione di *Lucifer*, un algoritmo di IBM nato nel '70, su insistenza del *National Bureau of Standard* per difendere dati riservati ma non segreti militari.

Il DES brevettato nel 1976 da IBM è *royalty-free* dal 1993. Il DES è un codice cifrato a blocchi. Si dice che un codice è cifrato a blocchi quando si applica un codice cifrato a un bit, *byte*, parola o gruppi di parole alla volta. Il blocco che si usa per crittografare è di 64 bits (8 sottoblocchi da 8 bit). Dato che l'ultimo bit di ogni sottoblocco è di controllo, i bit utili sono 56.

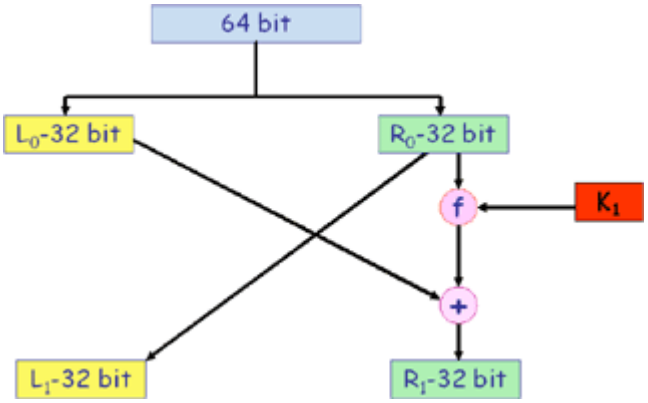


Per cifrare il testo si divide in blocchi da 64 bit che sono cifrati in successione. Se un messaggio non riempie i 64 bit si può completare in diversi modi: si possono aggiungere zeri, si possono aggiungere bit random specificando nell'ultimo quanti se ne aggiungono, etc.



Il nucleo di ogni ciclo DES è la *Feistel network*, così chiamata dal nome dello scienziato della IBM *Horst Feistel*. I 64 bit di *plaintext* vengono divisi (***split***) in destra (**R**) e sinistra (**L**) di 32 bit ciascuno. La parte destra elabora *output* del lato sinistro alla fine del ciclo, ed il lato sinistro entra in una *black box* ove prima viene espanso a 48 bit da una permutazione di espansione e successivamente passato in uno XOR con una chiave da 48 bit.

La somma risultante entra in una matrice di 8 ***S-box*** con 6 linee di *input* e 4 di *output* ognuna, producendo un risultato di 32 bit che verrà permutato dalla ***P-box***. Il risultato della *black box* passa per uno XOR con la metà sinistra dell'*input* iniziale e diviene la metà destra per il seguente ciclo.



Ogniuno dei 16 cicli DES ha una chiave da 48 bit, derivata dallo scorrimento (***shift***) e dalla permutazione dell'intera chiave da 56 bit ciclo per ciclo.

Il DES è molto usato in ambito commerciale perchè, i numerosi passaggi che comprende, sono relativamente semplici (XOR, sostituzioni e permutazioni).

Occorre ricordare che il DES cambia solo la chiave; questo porta vantaggi economici immediati, ma appena verrà scoperto il modo per forzarlo (senza *bruteforce*) occorrerà cambiare radicalmente tutto. Un altro difetto fondamentale è lo spazio limitato delle chiavi pari a 2^56. Per ovviare al problema si tenta di allungare le chiavi o di applicare più volte il DES (triplo-DES o TDES).

Il progetto originale dell' IMB per il DES prevedeva una chiave più lunga dei 56 bits usati di *default*. Probabilmente il progetto originario fu influenzato dall'NSA che impose all'IBM una chiave sicura, ma comunque alla portata dei loro mezzi.

Modalità di utilizzo

Ovviamente non è sufficiente fornire un algoritmo di cifratura e decifratura, ma è anche necessario

fornirne le modalità di utilizzo. Questo si rende necessario in quanto DES è un algoritmo di cifratura che utilizza blocchi di 64 bit e una chiave a 56 bit. In generale si dovranno cifrare messaggi di lunghezza arbitraria.

Il comitato ANSI, nel documento ANSI X3.106-1983 (*Modes of Use*) ha stabilito per DES 4 modi di utilizzo:

- Due di tipo A Blocchi;
- Due di tipo A Flusso.

Electronic Code Book [Block Mode]

In questa modalità di utilizzo ogni messaggio più lungo di 64 bit viene spezzato in blocchi di 64 bit che poi vengono cifrati.
Può accadere che l'ultimo blocco di bit sia più breve di 64 bit: quando questo avviene su questo blocco si applica il *Padding* ossia si completa il blocco introducendo zeri fino a raggiungere la lunghezza corretta.

Questo modo di utilizzo presenta, ovviamente, debolezze:

- Ripetizioni nel messaggio in chiaro possono riflettersi nel testo cifrato:
  - Se allineate con i blocchi del messaggio.
  - Con particolari tipi di dati - ad esempio la grafica.
  - Se il messaggio risulta essere molto costante si presta alla crittoanalisi statistica.
- Una debolezza è data dalla indipendenza reciproca dei blocchi di messaggio cifrato.

Cipher Block Chaining (CBC)[Block Mode]

Questo metodo utilizza il risultato di uno step di cifratura per modificare l'*input* del successivo step. Ciò implica un vantaggio ed uno svantaggio.

Il vantaggio è rappresentato dal fatto che ogni blocco di cifratura è dipendente da tutti quelli che lo precedono. Lo svantaggio è rappresentato dal così detto effetto valanga - *avalanche effect* - ossia la modica di un bit in un blocco si ripercuote su tutti i blocchi a questo susseguenti.
Per cifrare il primo blocco si utilizza un Vettore di Inizializzazione (IV) con il quale si cifra il primo blocco. Ovviamente il vettore di inizializzazione deve essere conosciuto sia da chi cifra che da chi decifra. Quindi il problema è, nel caso di messaggi che viaggiano su rete, l'inviare il vettore di inizializzazione. Ovviamente anche mediante questo modo di utilizzo può essere necessario espandere la dimensione dell'ultimo blocco: il metodo da utilizzarsi è esattamente il medesimo visto per l'utilizzo *Electronic Code Book*.

Cipher FeedBack (CFB) [Stream Mode]

Nel caso in cui si debba operare su dati bit o *byte oriented*, ossia su flussi di dati, tipicamente *file*, è necessario cambiare approccio di cifratura.
L'approccio di cifratura realizzato dal CFB è il seguente:

- Si cifra il Vettore di Inizializzazione IV.
- Del IV cifrato si scartano i 56 bit meno significativi.
- Con il *byte* più significativo dell'IV cifrato si cifra il primo *byte* del *PlainText*.
- Si opera uno *shift* a sinistra di IV di un *byte*.
- Nello spazio così creato si inserisce il primo *byte* cifrato del *PlainText*.
- Si tratta il vettore così ottenuto come se fosse un nuovo vettore di inizializzazione e si reitera

il processo fino ad esaurimento del flusso dati.

La decifratura è molto simile a quella utilizzata nel DES Classico. Per decifrare è sufficiente infatti invertire una parte del processo, avendo cura di utilizzare il DES sempre in modalità cifratura.

Output FeedBack (OFB) [Stream Mode]

Questo tipo di implementazione è superficialmente simile alla implementazione CFB. La differenza risiede nel fatto che il *feedback* non è il risultato della cifratura del *PlainText*, ma è il risultato della cifratura dell'IV. In questo modo si ottengono due cose:

- Indipendenza della cifratura dal messaggio - non sono più possibili attacchi crittoanalitici statistici.
- Si riesce ad eliminare del tutto l'effetto valanga.

Crittoanalisi del DES

A causa della crescente attenzione e dello sviluppo in tema di crittoanalisi, il DES ed il 3DES sono stati abbandonati. Le tecniche di crittoanalisi possono essere raggruppate come segue:

Ciphertext-Only Attack

- L'attaccante conosce il cifrato (*Ciphertext*) di diversi messaggi cifrati con la stessa chiave o con diverse chiavi.
- Recupera il testo chiaro (*plaintext*) di più messaggi o meglio cerca di dedurne la chiave (o le chiavi).

Known-Plaintext Attack

- Si conosce la coppia *ciphertext / plaintext* di diversi messaggi.
- Viene dedotta la chiave o un metodo per decifrare messaggi futuri.

Chosen-Plaintext Attack

- L'attaccante può scegliere il *plaintext* che è stato cifrato del quale si ha maggior possibilità di rintracciare informazioni sulla chiave.

Adaptive Chosen-Plaintext Attack

- L'attaccante può scegliere una serie di *plaintext*, basando la scelta sul risultato di precedenti cifrature (*differential cryptanalysis*).

Differential Cryptanalysis

È stata introdotta nel 1990 da Eli Biham e Adi Shamir, che la usarono per dimostrare che per determinate classi di algoritmi crittografici esiste un *adaptive chosen plaintext attack* che era più efficiente degli attacchi *brute force*. Anche se potenzialmente vulnerabile, il *Data Encryption Standard* (DES) si è dimostrato sorprendentemente resistente alla *differential cryptanalysis*. Perchè le *S-boxes* contengono esattamente valori ottimali da rendere un *differential attack* più difficile possibile? Perchè il DES utilizza esattamente 16 cicli, il minimo richiesto per rendere l'efficienza della *differential cryptanalysis* simile a quella di un attacco *brute force*? Risposta: Perchè nel tardo 1970 gli sviluppatori della IBM conoscevano già la crittoanalisi differenziale!

Don Coppersmith, della IBM, nel 1992 scrisse:

The design took advantage of certain cryptanalytic techiques, most prominently the technique of differential cryptanalysis, which were not known in the published literature. After discussions with the National Security Agency (NSA), it was decided that disclosure of the design considerations would reveal the technique of differential cryptanalysis, a powerful technique that can be used against many ciphers. This in turn would weaken the competitive advantage the United States enjoyed over other countries in the field of cryptography.

Breaking DES

Attack Method	Data Complexity		Storage Complexity	Processing Complexity
	Known	Chosen		
Exhaustive Precomputation	—	1	2^{56}	1 (table lookup)
Exhaustive Search	1	—	negligible	2^{55}
Linear Cryptanalysis	2^{43}	—	for texts	2^{43}
Differential Cryptanalysis	—	2^{47}	for texts	2^{47}
Differential Cryptanalysis	2^{55}	—	for texts	2^{55}

Considerando la tabella, (che proviene p. 259 del testo di ALFRED MENEZES, PAUL VAN OORSCHOT, e SCOTT VANSTONE, *Handbook of Applied Cryptography*, *CRC Press*, 1996), risulta considerevolmente facile eseguire una ricerca esaustiva (*exhaustive search*) con una coppia nota *plaintext-ciphertext* e 2^55 operazioni DES. Inoltre si può tentare una crittoanalisi lineare (*linear cryptanalysis*) che richiede 2^43 coppie note *plaintext-ciphertext*. La crittoanalisi differenziale ha minato la forza degli algoritmi a blocchi, ed un attacco che ha percentuali di successo dello 0,01% è potenzialmente devastante.

Nel 1993 *Michael Wiener* aggiornò il sistema a ricerca esaustiva con le più attuali tecnologie. il risultato fu una macchina da un milione di dollari con 57,000 *chip* DES ed una architettura *pipelined*. *Wiener* stimò la soluzione del problema DES in 3 ore e mezza. Il seguente fervore fu d'obbligo.

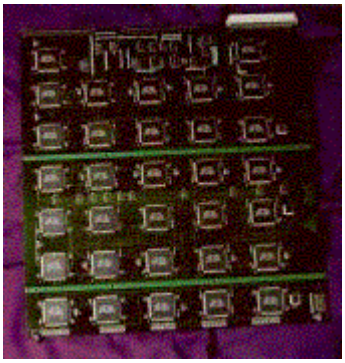
Electronic Frontier Foundation

Nel Luglio 1998, utilizzando un calcolatore appositamente sviluppato, la ***Electronic Frontier Foundation*** costruì il **DES *Cracker***. Il suo costo fu di $250,000 e si impiegò meno di un anno per realizzarlo; il DES *Cracker* riuscì a decifrare il DES in 56 ore. In realtà la chiave fu trovata dopo solo un quarto dello spazio delle chiavi, mentre ci si aspettava di doverne attendere almeno metà. La costruzione del DES *Cracker* ha richiesto l'utilizzo di 1,536 *chip*, i quali sono in grado di cercare 88 miliardi di chiavi al secondo.

Se la **EFF** avesse investito altri $250,000 collegando le due macchine in parallelo si disporrebbe di un DES *Double-Cracker* che impiegherebbe la metà del tempo.

Oggi, disponendo di uno spazio delle chiavi DES pari a 2^56, cioè circa 7,2056 *10^16, si può stimare che un PC, con *clock* a 500 Mhz in grado di provare una chiave per ciclo di *clock* impieghi (ipotesi per assurdo):

in 144115188 secondi (ossia 834 giorni, che equivalgono a 2 anni e 3 mesi) si possono provare 2^55 (3,6 *10^16 chiavi).



È anche interessante osservare quale tipo di attività porta avanti il progetto ***Distributed.net***, degno di nota il loro *Moo client*.

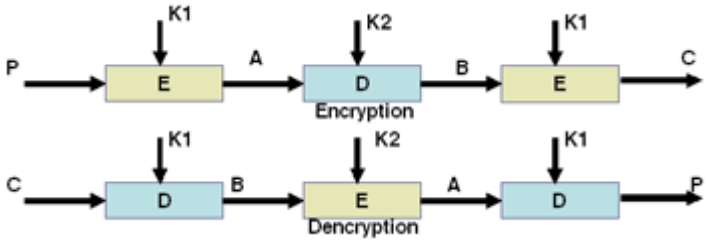
Triplo DES

La brevità della chiave consente attacchi esaustivi di ricerca. Il problema quindi risiede nel riuscire a trovare un metodo che permetta di aumentare la lunghezza della chiave, migliorare la sicurezza dell'algoritmo ed al tempo stesso non aumentare troppo la complessità computazionale dello stesso.

Il primo metodo che può venire in mente è quello di cifrare più volte il medesimo messaggio con chiavi differenti. Ciò che si ottiene è un cifrario a singolo passo dove la chiave, però, è lunga doppia o tripla rispetto il cifrario originale.

TRIPLE-DES (3DES)

Per ovviare alle debolezze della cifratura DES reiterata due volte, si è pensato di applicare la medesima strategia, ma di cifrare tre volte con tre chiavi differenti.



Utilizzando 3 chiavi da 56 bit, si ottiene una chiave da 168 bit, difficile da identificare. Per questo motivo quasi tutte le implementazioni di 3DES utilizzano K1 = K3. Altra interessante caratteristica presentata da quasi tutte le implementazioni di 3DES è conosciuta con il nome di schema **EDE** ***E**ncryption - **D**ecryption - **E**ncryption*. In sostanza in questa implementazione durante la seconda fase, DES non viene utilizzato in cifratura, ma in decifratura. L'effetto collaterale ottenuto mediante questa implementazione è che utilizzando K1 = K3 = K2 si ottiene l'effetto di cifratura che si otterrebbe con un DES classico. La necessità di una simile implementazione è data dalla volontà di mantenere la compatibilità con applicazioni che utilizzano il DES classico invece che il più robusto 3DES.

Altri algoritmi, AES

Name of Algorithm	*Block Size*	*Key Size*
DES (*Data Encryption Standard*, IBM)	64	56
3DES (*Triple* DES)	64	168
IDEA (Lai / Massey, ETH Zurigo)	64	128
RC2 (*Ron Rivest*, RSA)	64	40...1024
CAST (Canada)	64	128
Blowfish (*Bruce Schneier*)	64	128 ... 448
Skipjack (NSA, *clipper chip*)	64	80
RC5 (*Ron Rivest*, RSA)	64...256	64...256

Considerando l'età ed i manifesti difetti di robustezza del DES, diverse sono le soluzioni che nel tempo, fino ad oggi si alternano ad esso. Il già menzionato *Triple* DES con chiave da 168 bit è attualmente un *Federal Information Processing Standard* FIPS 46-3 (rinnovato nell Ottobre 1999).

Ad oggi è terminato il processo ***Evaluation of an Advanced Encryption Standard*** che ha dato risultato (visibile nel sito **http://www.nist.gov/aes**). Il *National Institute of Standards and Technology* (NIST, *U.S. Department of Commerce*) ha avviato un contesto pubblico nel 1997. Cinque dei finalisti hanno ottenuto una menzione d'onore, ma la palma d'oro è spettata al sistema **Rijndael** in Ottobre 2000.

Il nuovo AES ha i seguenti requisiti:

- AES sarà definito pubblicamente.
- AES sarà di tipo *symmetric block cipher*.
- AES sarà implementabile sia tramite *hardware* sia tramite *software*.
- AES sarà progettato in modo tale che la lunghezza della chiave possa essere incrementata come necessario.
- AES avrà blocchi di dimensione n = 128 bit, *key size* k = 128, 192, 256 bit.

Conclusioni

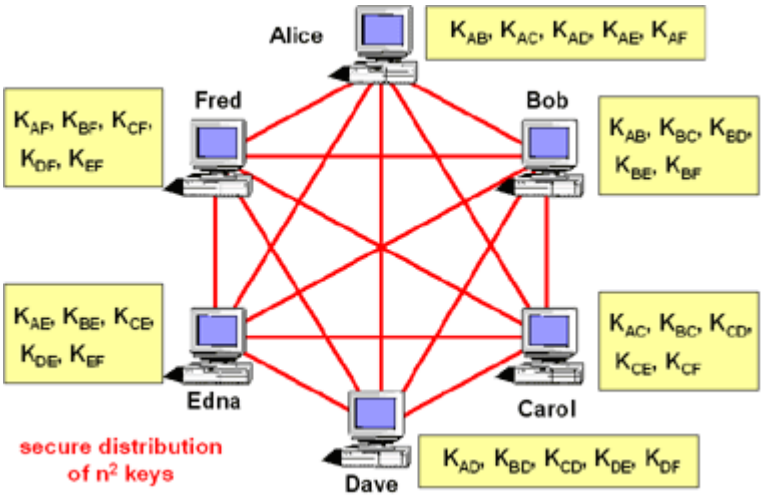
In conclusione va brevemente detto che i sistemi simmetrici lasciano aperti due grandi problemi:

- Metodo per scambio di chiavi assente (OOB).
- Numero di chiavi troppo grande per gestire dei segreti condivisi tra singoli utenti [n(n-1)]/2.

Necessita una soluzione...

Problema della distribuzione delle chiavi

Nelle reti magliate densamente popolate, dove molte parti potrebbero voler comunicare con altre, il numero di chiavi segrete richieste quando si usano algoritmi a chiave simmetrica aumenta quadraticamente con il numero dei partecipanti n ed è dato da [n(n-1)]/2, dal momento che occorre una chiave per ciascuna coppia di terminali.



Si prenda ad esempio una rete di comunicazione *broadband* con 100 nodi completamente magliati ove ogni chiave sia cambiata ad ogni sessione ogni ora. Come risultato avremo la necessità di distribuire 240000 chiavi in tutta sicurezza (OOB) ogni giorno.

La scala di distribuzione delle chiavi segrete peggiora pesantemente la complessità del sistema con un lieve incremento di partecipanti. Si cerca quindi una soluzione al problema della distribuzione delle chiavi tramite canali su connessioni sicure. Una soluzione efficiente è rappresentata dal concetto di ***Public Key Cryptosystem***.

Public Key Distribution System

In un *Public Key Cryptosystem* l'idea applicata è la stessa utilizzata dai cifrari simmetrici, cioè quella di rendere non leggibile un messaggio mediante la sua cifratura a mezzo di una chiave. La differenza risiede nel fatto che nei cifrari asimmetrici ogni utente possiede una coppia di chiavi. Queste sono note con il nome di chiave pubblica e chiave privata.

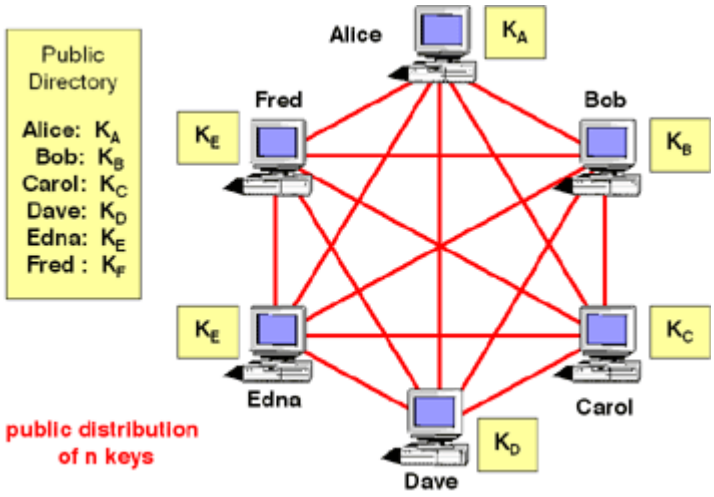
Occorre precisare alcuni requisiti sulle chiavi di un ipotetico sistema asimmetrico:

- Le chiavi sono tra loro indipendenti. Sceglie l'utente quale rendere pubblica e quale mantenere privata.
- Le due chiavi possono essere utilizzate indifferentemente per cifrare o decifrare.
- Entrando in possesso di una chiave, tipicamente quella pubblica, non è possibile in alcun modo risalire all'altra chiave.
- Ogni utente può possedere più di una coppia di chiavi, destinate agli usi più disparati; tali chiavi possono essere disponibili o pubblicate in *public directory* (es. LDAP o HTTP *server*).
- La generazione della coppia di chiavi avviene contemporaneamente.

L'utilizzo di questo tipo di cifrari introduce sostanziali vantaggi:

- Migliora il metodo di gestione delle chiavi.
- A mezzo dei cifrari asimmetrici è possibile realizzare il concetto di Firma Elettronica.

Inoltre, i cifrari asimmetrici permettono di realizzare non solo il concetto di Riservatezza del messaggio, ma anche il concetto di Autenticazione della fonte.



Se Alice vuole mandare un messaggio cifrato a Bob, Alice cifra il suo messaggio con la chiave pubblica di Bob recuperata da una *public directory* e lo invia a Bob. Poichè Bob è l'unico in possesso della chiave privata collimante, solo lui potrà decifrare il messaggio a lui destinato.

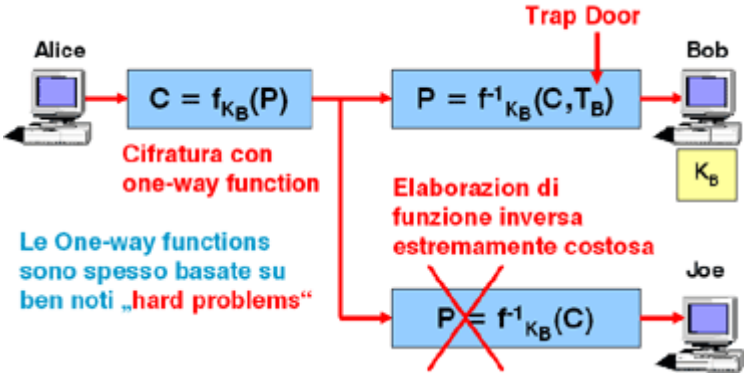
Dato che è necessario disporre solo della chiave pubblica del destinatario, con n utenti serviranno n chiavi distinte. Imponendo l'assioma che ogni utente genererà una propria coppia di chiave pubblica/privata locale, non serviranno canali sicuri per la distribuzione della chiave.

Principi di base della crittografia a chiave pubblica

Il concetto di crittografia a chiave pubblica è stato inventato quasi contemporaneamente da *Whitfield Diffie*, *Martin Hellman* e *Ralph Merkle*. I primi due ricercatori pubblicarono la loro invenzione nel 1976 e disposero di tutta la fama, *Ralph Merkle* ebbe la sfortuna che la stampa del suo lavoro subì il ritardo di oltre un anno e non fu quindi pubblicata prima del 1978. Oggi è generalmente riconosciuto che i tre scienziati sono padri della crittografia a chiave pubblica.

Recentemente è stato reso noto che già nel 1970, *James Ellis*, che al tempo lavorava per il governo Inglese come membro del *Communications-Electronics Security Group* (CESG), formulò l'idea di un *Public Key Cryptosystem*. Sfortunatamente, il governo non consentì la pubblicazione dei risultati della ricerca per ragioni di sicurezza.

Tutti i sistemi di crittografia a chiave pubblica sono basati sulla nozione di ***one-way function***, che, in funzione della chiave pubblica, converte il testo in chiaro in messaggio cifrato, utilizzando una relativamente piccola quantità di potenza computazionale ma la quale **funzione inversa** è estremamente costosa in termini di risorse di calcolo, al punto da rendere impossibile per un attaccante di ricavare il testo chiaro originale dal cifrato trasmesso in tempi ragionevoli.



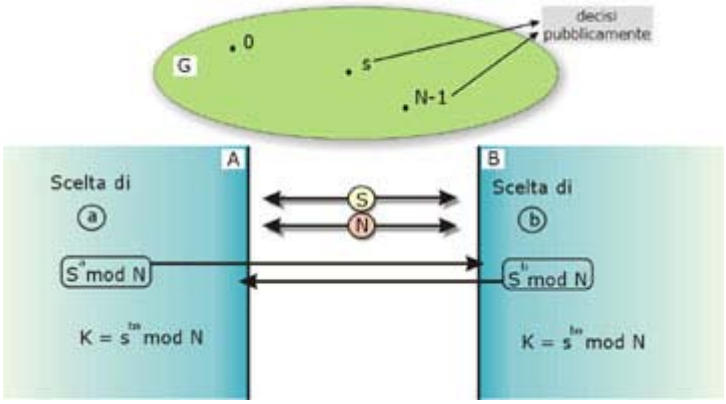
Un'altra nozione utilizzata nei sistemi a chiave pubblica è quello di una ***trap door*** che ogni funzione *one-way* possiede è che può essere attivata solo dal legittimo proprietario tramite la chiave privata. Utilizzando la *trap-door*, la decifrazione si semplifica.

Molti sistemi a chiave pubblica sono basati sulla nozione di ***known hard problem*** (problemi noti e di difficile soluzione) come ad esempio la fattorizzazione di numeri grandi nei loro fattori primi (RSA) o il prelievo di logaritmi discreti su di un campo finito (*Diffie-Hellman*).

Diffie Hellman

Il sistema di *Diffie* ed *Hellman* consiste nello scambio confidenziale di un messaggio, e viene definito: ***Key-Exchange Algorithm***. Ecco i passi necessari, per eseguire lo scambio:

- Alice e Bob scelgono pubblicamente un insieme di interi G=[0,N-1] ed un elemento s dello stesso insieme.
- Alice sceglie in modo casuale un elemento a di G, calcola s^a mod N e lo invia a Bob.
- Bob sceglie in modo casuale un elemento b di G, calcola s^b mod N e lo invia ad Alice.
- Alice, calcolato, come di seguito spiegato, s^b, calcola K= (s^b)a mod N.
- Bob, calcolato, come di seguito spiegato, s^a, calcola K= (s^a)b mod N.



Nella figura la lettera A sostituisce il nome proprio Alice, mentre la lettera B sostituisce il nome Bob.

RSA

Gli inventori *Ron **R**ivest*, Adi **S**hamir e *Leonard **A**dleman* utilizzano come *one-way function* la funzione esponenziale y = f(x) = x^e mod n che può essere elaborata con sforzo ragionevole. La sua inversa x = f^-1(y) è invece estremamente difficoltosa da elaborare.

Il sistema con algoritmo a chiave pubblica RSA è basato sul problema noto e difficile della fattorizzazione di numeri grandi nei loro fattori primi che è stata studiata per molti secoli.

La sfida RSA-155 basata su un numero da 512 bit (155 numeri decimali) ha visto impegnati 301 elaboratori in rete (300 fra *workstation* e *pc pentium*, 1 *Cray supercomputer*) ed ha portato alla

soluzione della fattorizzazione in un tempo di 7 mesi.

```
109417386415705274218097073220403576120
037329454492059909138421314763499842889
347847179972578912673324976257528997818
337970765372440271467435315933543338 97
? =
102639592829741105772054196573991675900
716567808038066803341933521790711307779
*
106603488380168454820927220360012878679
207958575989291522270608237193062808643
```

I passi previsti per eseguire RSA sono i seguenti:

- Step 1: Scegliere casualmente due numeri primi grandi *p* e *q* - Per una maggior sicurezza, scegliere *p* e *q* di lunghezza circa uguale, esempio: 512-1024 bit ciascuno.
- Step 2: Calcolare il prodotto ***n*** = *p*q*.
- Step 3: Scegliere un intero a caso ***e*** < *(p-1)(q-1)* (i numeri ***e*** e *(p-1)(q-1)* devono essere primi fra loro, cioé non devono condividere fattori primi).
- Step 4: Calcolare l'inverso unico ***d***= *e^-1 mod (p-1)(q-1)* (l'equazione *d*e mod (p-1)(q-1) = 1* può essere risolta usando l'algoritmo Euclideo).

RSA esempio 1

Si prenda ad esempio p = 3 e q = 11:

n = p*q = 33

(p-1)*(q-1) = 2 * 10 = 2 * 2 * 5 = 20

l'esponente **e** deve essere relativamente primo a (p-1)*(q-1), non deve quindi contenere i fattori 2 e 5. Ecco le possibili scelte di **e** e **d**.

e	**d**	**(e*d)**	**(e*d) mod 20**
3	7	21	1
7	3	21	1
9	9	81	1
11	11	121	1
13	17	221	1
17	13	221	1
19	19	361	1

Public Key: modulo n ed esponente pubblico e; pubblicare n ed e in una *directory* pubblica, in modo che chiunque voglia mandare un messaggio confidenziale a noi possa venire in possesso di n ed e.

Private Key: modulo n ed esponente privato d; l'esponente privato d è segreto. Deve essere protetto sia memorizzandolo in una *smart card* a prova di intrusione o quando memorizzata in un disco cifrata con algoritmo simmetrico ed una frase di propria scelta. I numeri primi p e q utilizzati per la generazione delle chiavi possono anche essere cancellati dopo aver fatto tale operazione.

RSA esempio 2

Cifratura di un blocco di testo x:

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

Il mittente usa la chiave pubblica del destinatario per cifrare x < n.

Decifratura del blocco y:

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

Il destinatario utilizza la chiave privata per recuperare il blocco in chiaro x.

Senza prova:

$$y^d = (x^e)^d = x^{e \cdot d} = x^{m \cdot (p-1) \cdot (q-1) + 1} = x^1 = x \pmod n$$

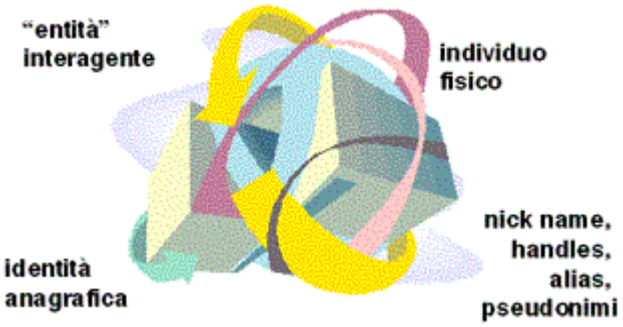
La cifratura e la decifratura sono operazioni simmetriche e l'ordine con il quale si calcolano gli esponenti pubblici e e privati d può essere cambiato.

L'esempio seguente, mostra in dettaglio i passaggi matematici:

- **Cifratura con chiave pubblica** $n = 33, e = 3$
  - Plaintext binario: 01010 01001 00101 10100 11 ...
  - Gruppo di 5 Bit: 01010 01001 00101 10100 ...
  - Plaintext decimale: 10 9 5 20
  - $y = x^3$: 1000 729 125 8000
  - $y = x^3 \bmod 33$: 10 3 26 14
- **Decifra. con chiave Privata** $n = 33, d = 7$
  - Cifrato decimale: 10 3 26 14
  - $x = y^7$: 10^7 2187 26^7 14^7
  - $x = y^7 \bmod 33$: 10 9 5 20

Firma digitale

Le interazioni in rete avvengono tra entità. La difficoltà di risalire con certezza al collegamento biunivoco tra **entità interagente e individuo fisico** che trasferisce una determinata entità costituisce sempre più un problema da un punto di vista giuridico: nonostante in rete possano essere compiuti reati, truffe e illeciti di vario tipo, la responsabilità legale di questi atti non è facilmente attribuibile, in quanto è molto difficile accertare quale persona fisica stia effettivamente operando. Tali problemi sono peculiari della giurisdizione statale (la particolare giurisdizione sviluppata spontaneamente in rete pare non preoccuparsene troppo). è solo in un'ottica repressiva e di controllo che, partendo da questa considerazione, si può arrivare a concludere che la comunicazione in rete è sempre anonima. Ciò che manca in rete non è il nome delle persone, ma semplicemente l'identità anagrafica.



A sostituzione del nome anagrafico, assegnato per legge e immodificabile, in rete prolifera una quantità enorme di altri nomi, *nicks*, *handles*, alias, pseudonimi. L'importanza di questi nomi non è minore di quella del proprio nome anagrafico: è attraverso il loro riconoscimento pubblico che in rete è possibile costruire relazioni sociali significative, relazioni che potranno essere trasferite anche al di fuori della rete.

Se esiste dunque una modalità caratteristica dell'interazione in rete rispetto ai nomi e alle identità individuali, questa non è data principalmente dall'anonimato ma piuttosto dallo **pseudonimato**. Lo pseudonimato comporta un processo di costruzione dell'identità e un suo riconoscimento sociale che perdurano nel tempo ma sono anche mutevoli e continuamente in divenire, mai acquisiti definitivamente; patrimonio fondamentale dello pseudonimo (sia esso corrispondente o no a un nome anagrafico) è la reputazione che riesce a guadagnare attraverso la sua vita in rete o quella che eredita da un'eventuale ragnatela di relazioni sociali avviate in precedenza *off-line*.

La perfetta realizzazione dello pseudonimato si scontra però con gli stessi problemi cui si è accennato a proposito dell'identità anagrafica: via rete è possibile modificare non solo il nome-numero di serie definito dallo Stato, ma anche lo stesso pseudonimo scelto. è possibile scrivere firmandosi con uno pseudonimo altrui. Questa possibilità realizza spesso un'utile opera di decostruzione dei propri pregiudizi. Nonostante questo, ci sono casi in cui, magari a causa della natura molto specifica e concreta della comunicazione, è assolutamente necessario essere certi dell'autore di un dato messaggio. Non tanto essere certi del suo numero di serie statale, quanto piuttosto del fatto che egli è effettivamente la stessa entità (individuale o collettiva) con cui si è comunicato in precedenza, via rete o anche in carne ed ossa. In altre parole, è necessario essere certi del suo pseudonimo.

è in una situazione come questa che la crittografia a chiave pubblica viene in aiuto. Ribaltando l'impiego delle chiavi pubbliche e private, è possibile porre una firma digitale crittografica sui messaggi che immettiamo in rete. La chiave segreta del mittente può infatti essere usata (oltre che per decifrare i messaggi ricevuti) anche per generare una firma da apporre nei messaggi che si spediscono. La firma digitale del messaggio può poi essere verificata dal destinatario (o da chiunque altro) utilizzando la chiave pubblica del mittente.

Questo serve a garantire che il mittente è colui che davvero ha scritto il messaggio e che il messaggio non è stato successivamente manipolato da nessun altro, poiché solo il mittente possiede la chiave segreta per poter firmare. è tecnicamente impossibile falsificare o modificare un messaggio autenticato senza invalidarne la firma e lo stesso mittente non può più revocare la firma una volta apposta.

I rischi della firma digitale

Una delle applicazioni più utili della firma digitale, a parte l'autenticazione dei messaggi veri e propri, riguarda la **conferma delle chiavi pubbliche di terze persone**. La crittografia a chiave pubblica infatti lascia scoperto un possibile punto debole. Nel momento in cui si vuole comunicare

con l'utente A, serve la sua chiave pubblica. Il modo migliore per ottenerla è direttamente dalle sue mani. Talvolta questo non è possibile e si è costretti a farla inviare attraverso la rete. Il sistema a chiave pubblica risolve ogni problema rispetto a un'eventuale intercettazione della chiave lungo il tragitto, ma presta il fianco alla possibilità che l'utente B, conoscendo la volontà di comunicare con l'utente A, si spacci per lui e spedisca una chiave pubblica contraffatta. Se si cade nel tranello e si utilizza quella chiave, i successivi messaggi saranno leggibili non dall'utente A, bensì dall'utente B, titolare della vera corrispondente chiave segreta. L'utente B potrà poi perfezionare il suo inganno rispedendo a sua volta tutti i messaggi all'utente A, che in questo modo non si accorgerà nemmeno dell'esistenza di una tappa in più lungo la strada.

Questo problema (chiamato **problema dell'uomo nel mezzo**) è assolutamente concreto e reale. La soluzione sta nel chiedere e ottenere che ogni nuova chiave pubblica sia firmata da qualcuno che si conosce e di cui si dispone già con certezza della rispettiva chiave pubblica. Se si viene costretti a ottenere la chiave dell'utente via rete, si avrà cura di verificare che essa sia firmata da un utente di cui si ha fiducia, con cui si hanno contatti quotidiani, a patto di possederne con certezza la vera chiave pubblica. Con questa chiave si potrà verificare la firma che l'uomo di mezzo ha apposto sulla chiave pubblica dell'utente A, cioè verificare che l'utente di cui ci si fida garantisce che la chiave pubblica che è appena arrivata è effettivamente la chiave dell'utente A. L'utente B, da solo, non sarebbe mai in grado di inviare una chiave firmata dall'uomo di mezzo, spacciandola per la chiave dell'utente A.

A questo punto è evidente che la certificazione delle chiavi può diventare rapidamente molto complessa consentendo di estendere la rete di contatti a partire da un unico punto iniziale sicuro e comprendendo anche entità che non si incontreranno mai di persona. Con questa caratteristica il cerchio viene chiuso e diventa veramente possibile stabilire un'infrastruttura comunicativa priva di contatti fisici che sia doppiamente sicura, sia dal punto di vista della possibilità di leggere il contenuto della comunicazione, sia da quello di poterne garantire la provenienza.

In conclusione si potrà dire che la firma digitale è una informazione che viene aggiunta ad un documento informatico al fine di garantirne integrità e provenienza. La principale differenza tra firma autografa e firma digitale sta nel fatto che la prima è direttamente riconducibile all'identità di colui che la appone, poiché la calligrafia è un elemento identificativo della persona, mentre la seconda non possiede questa proprietà. Per coprire questa deficienza si ricorre all'autorità di certificazione, il cui compito è quello stabilire, garantire e pubblicare l'assoc+iazione tra firma digitale e soggetto sottoscrittore.

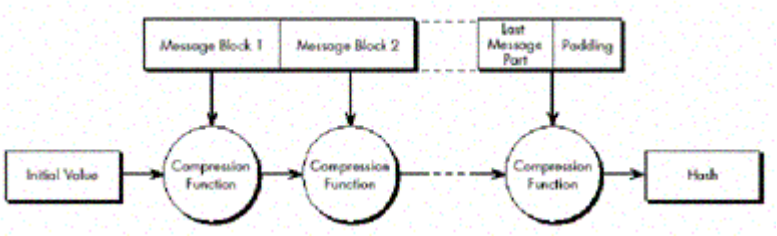
Per contro, mentre l'associazione tra testo di un documento e la firma autografa è ottenuta esclusivamente attraverso il supporto cartaceo, la firma digitale è intrinsecamente legata al testo a cui è apposta, tanto che i due oggetti possono essere fisicamente separati senza che per questo venga meno il legame esistente tra loro. Conseguenza di ciò è l'unicità della firma digitale, nel senso che a testi diversi corrispondono firme diverse; in tal modo, nonostante la sua perfetta replicabilità, è impossibile trasferire la firma digitale da un documento ad un altro.

Processo di firma digitale

La firma elettronica, oltre ad avere la possibilità di autenticare i messaggi ai quali è associata, offre la possibilità di identificare univocamente l'utente firmatario. In sostanza ha applicazioni molto simili a quelle della firma autografa. Ciò è possibile in quanto si utilizza un cifrario asimmetrico del quale si usa la chiave privata per firmare e quella pubblica per autenticare.

Al testo da firmare viene applicata una **funzione di *hash*** appositamente studiata che produce una stringa binaria di lunghezza costante e piccola, normalmente 128 o 160 bit. La funzione di *hash* assicura l'unicità di tale stringa, nel senso che a due testi diversi non corrisponde la medesima impronta. Sono disponibili diversi algoritmi di generazione, quali, ad esempio, **MD2**, **MD4** e **MD5**,

originariamente progettati per operare in combinazione con RSA ma utilizzabili con qualsiasi cifrario. Sono disponibili anche algoritmi di *hash* per i quali è in corso la standardizzazione ufficiale da parte organismi internazionali; ne sono un esempio il RIPEMD a 128 e 160 bit ed il ***S**ecure **H**ash **A**lgorithm* (**SHA-1**).



Affinché una funzione di *hash* operi correttamente è necessario procedere come segue:

- Suddividere il messaggio in blocchi di lunghezza congruente con il *digest* prodotto dalla funzione di *hash* prescelta.
- Sottoporre ogni blocco del messaggio all'operazione di *hashing*.
- Reiterare il procedimento.

Un ***M**essage **D**igest* rappresenta il riassunto di un messaggio. Il suo scopo non è quello di garantire la *privacy*, ma l'**integrità**. Grazie ad esso infatti si può verificare che il contenuto del messaggio non sia stato alterato nel suo tragitto. Lo si applica anche in situazioni che richiedono velocità di calcolo e dove il corpo del messaggio in questione sia di dimensioni troppo elevate.

L'utilità dell'impronta è duplice, in primo luogo consente di evitare che per la generazione della firma sia necessario applicare l'algoritmo di cifratura, che è intrinsecamente inefficiente, all'intero testo che può essere molto lungo. Inoltre consente l'autenticazione, da parte di una terza parte fidata, della sottoscrizione di un documento senza che questa venga a conoscenza del suo contenuto. Una tipica situazione in cui si sfruttano tali caratteristiche dell'impronta è la marcatura temporale che verrà discussa più avanti.

Generazione della firma digitale

La generazione della firma consiste semplicemente nella cifratura dell'impronta digitale generata il precedenza mediante la chiave segreta Ks. In questo modo la firma risulta legata da un lato al soggetto sottoscrittore (attraverso la chiave segreta usata per la generazione) e dall'altro al testo sottoscritto (per il tramite dell'impronta).

In realtà l'operazione di cifratura viene effettuata, anziché sulla sola impronta, su una struttura di dati che la contiene insieme con altre informazioni utili, quali ad esempio l'indicazione della funzione *hash* usata per la sua generazione. Sebbene tali informazioni possano essere fornite separatamente rispetto alla firma, la loro inclusione nell'operazione di codifica ne garantisce l'autenticità.

Apposizione della firma

La firma digitale generata al passo precedente viene aggiunta in una posizione predefinita, normalmente alla fine del testo del documento. Normalmente, insieme con la firma vera e propria, viene allegato al documento anche il valore dell'impronta digitale ed eventualmente il certificato da cui è possibile recuperare il valore della chiave pubblica. è evidente che essendo il legame tra firma e documento stabilito attraverso l'impronta, di natura puramente logica, la firma stessa e le informazioni aggiuntive eventualmente ad essa associate possono essere registrate e gestite in modo del tutto separato rispetto al testo sottoscritto; in particolare possono trovarsi su supporti e sistemi di

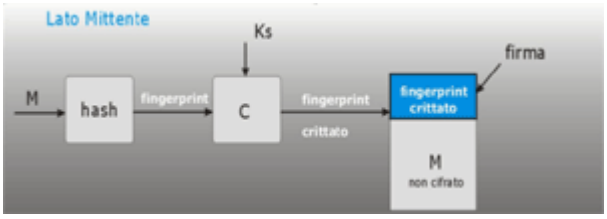
elaborazione del tutto indipendenti tra loro.

Attualmente le tecniche di firma elettronica sono realizzate mediante:

- RSA in unione con MD5 / SHA-1;
- DSA in unione con SHA-1.

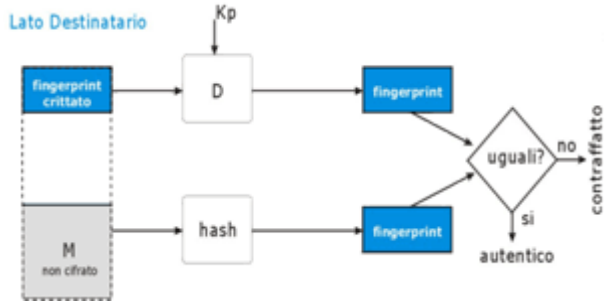
Firma RSA

L'uso dello schema RSA per generare firme elettroniche si basa semplicemente sull'inversione del ruolo delle chiavi rispetto a quello assegnato nell'assicurare la riservatezza; la principale differenza tra le due applicazioni sta nel fatto che per la firma si evita di applicare l'operazione di codifica all'intero testo. Ciò è particolarmente conveniente vista la complessità e la lentezza delle operazioni coinvolte.



- Si applica una funzione di *hash* (SHA-1) al messaggio.
- Il *digest* così ottenuto viene cifrato mediante RSA.
- Il *file* di firma ottenuto è accodato al messaggio oppure inviato parallelamente.

In pratica il testo da firmare viene compresso in una sorta di riassunto, che viene spesso riferito come impronta, mediante una opportuna funzione di *hash* progettata in modo da rendere trascurabile la probabilità che da testi diversi si possa ottenere il medesimo valore. La dimensione del riassunto è fissa ed è molto più piccola di quella del messaggio originale; essa è dell'ordine del centinaio di bit, in modo da rendere estremamente più rapida la generazione della firma effettuata a partire dall'impronta anziché dal testo.



- Si applica SHA - 1 al messaggio, ottenendo un *digest*.
- Si decifra il *digest* allegato al messaggio tramite la chiave pubblica del mittente.
- Si confrontano i *digest*: se corrispondono si ha autenticazione.

Possibili debolezze della firma

La principale osservazione sulla firma elettronica riguarda la conoscenza della chiave pubblica del mittente. Ossia il ricevente il messaggio deve conoscere in modo certo la chiave pubblica del mittente. Altrimenti sarebbe possibile generare una coppia di chiavi facendo credere che queste appartengano a qualcun altro di cui il destinatario conosce l'identità. A questo punto è sufficiente intercettare i messaggi inviati dal mittente, sostituirli, firmarli con la falsa chiave ed inviarli al

destinatario che considererà autentici dei messaggi falsi.

Importanza della protezione Birthday Attack

In un contesto di firma elettronica è molto importante proteggere i *Birthday Attack*, quindi utilizzare una funzione di *hash* resistente alle collisioni.Supponiamo di procedere in questo modo:

- Si generano una quantità di varianti di messaggi che il potenziale mittente si sente pronto a firmare.
- Parallelamente si generano una quantità di messaggi alterati, ma che producono i medesimi risultati di *Hash* (collisioni).
- A questo punto è sufficiente staccare la firma dal messaggio originale ed attaccarla al messaggio alterato. Essendo questi messaggi una collisione per la funzione di *hash* risulteranno indistinguibili per il destinatario.

Firma DSS

Il metodo di firma **DSS** (***D**igital **S**ignature **S**tandard*) ed il relativo algoritmo **DSA** (***D**igital **S**egnature **A**lgorithm*) sono metodi di firma alternativi allo standard RSA, e sono una variante degli algoritmi di firma e cifratura Schnorr (brevettato con scadenza nel 2008) e ElGamel.

Pubblicato dal **NIST** (***N**ational **I**nstitute of **S**tandard and **T**echnology*) il 30 agosto 1991, è sostanzialmente una funzione di *hash* H che ha come unico argomento il messaggio; in tal modo il suo valore non dipende dalla chiave di cifratura. Il suo attuale nome è **FIPS 186**, poiché una pubblicazione del ***F**ederal **I**nformation **P**rocessing **S**tandard* ne descrive il DSA. Nel documento, l'algoritmo chiave della codifica è il ***S**ecure **H**ash **S**tandard* (**SHS**).

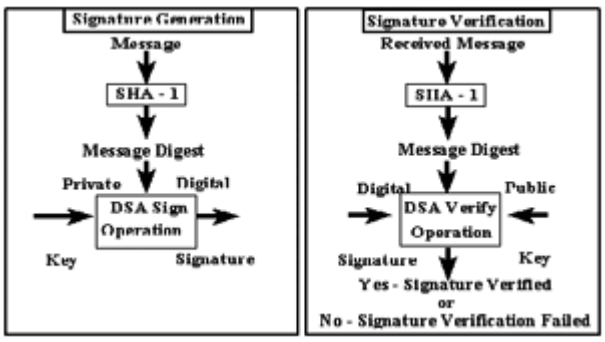
La sicurezza dell'algoritmo è basata sulla difficoltà di calcolare il logaritmo discreto in un gruppo finito. In sostanza occorre calcolare:

a^x mod z (che è considerato computazionalmente facile), essendo

x tale che a^x = b mod z (che è considerato computazionalmente difficile).

Ricordiamo, invece, che la forza di RSA è basata sulla difficoltà di fattorizzare in primi. è possibile dimostrare che il calcolo di un logaritmo discreto in un gruppo finito equivale dal punto di vista computazionale a fattorizzare un numero ottenuto come prodotto di due primi.

Il processo di firma è graficamente rappresentabile con la seguente figura:



Certificati

Il processo di firma digitale richiede che l'utente effettui una serie di azioni preliminari necessarie alla predisposizione delle chiavi utilizzate dal sistema di crittografia su cui il meccanismo di firma

si basa; in particolare occorre effettuare le seguenti operazioni:

- registrazione dell'utente presso un'autorità di certificazione;
- generazione di una coppia di chiavi Ks e Kp;
- certificazione della chiave pubblica Kp;
- registrazione della chiave pubblica Kp.

Per motivi di confidenzialità, la chiave privata non dovrà mai essere inviata con mezzi di comunicazione non sicuri. Quindi non si avrà la certificazione della chiave privata (DPCM 08 febbraio 1999 - Art. 28 - Generazione dei certificati).

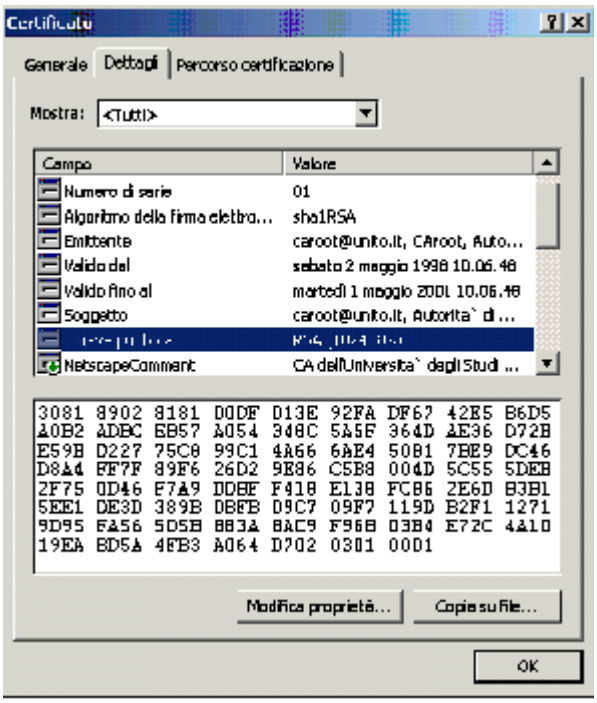
Un certificato è costituito dalla coppia [nome utente, chiave pubblica utente] certificata, ossia firmata da una autorità di certificazione. Le autorità di certificazione possono essere di due tipi:

- autorità di certificazione create da una realtà, aziendale o amministrativa, il cui compito è quello di certificare ed autenticare lo scambio di informazioni all'interno della rete dell'organizzazione;
- autorità di certificazione il cui scopo è quello di certificare realtà commerciali che si occupano di *e-commerce*.

Si può ottenere un elenco aggiornato delle autorità di certificazione legalmente riconosciute presso il sito **AIPA**. La richiesta di un certificato è una operazione tradizionale: infatti ci si deve recare dall'autorita di certificazione personalmente e muniti di un documento di identità, in questo modo l'autorità potrà essere sicura dell'identità di colui il quale richiede il certificato. In seguito a questa operazione e successivamente nel tempo, l'utente riceverà dall'autorità il proprio certificato il quale contiene:

- I dati identificativi dell'utente.
- La chiave pubblica dell'utente.
- La firma dell'autorità di certificazione.

Da questo momento l'utente sarà in grado di utilizzare il certificato per gli scopi previsti. Notiamo che un certificato non può essere manomesso: infatti ogni certificato reca con sé un *fingerprint* univoco il quale altro non è che un *digest* ricavato dal messaggio stesso. Ovviamente essendo un *digest* ottenuto con una funzione di *hash* ad una via e *collision resistant* difficile è che due certificati differenti abbiano il medesimo *fingerprint*. Di solito il *fingerprint* è una caratteristica del certificato al quale si cerca di dare la maggiore pubblicità possibile in modo che sia facilmente verificabile.



Autorità di certificazione

Una autorità di certificazione è un organismo che si occupa di verificare ed assicurare la corrispondenza chiave pubblica - utente. Inizialmente, si era pensato di organizzare le autorità in maniera gerarchica, per alcuni motivi:

- Tramite una gerarchia è possibile creare autorità locali che vengono certificate da autorità di livello superiore.
- Tramite una gerarchia è possibile generare CA indipendenti e specializzate.
- è possibile, inoltre, fornire maggiore sicurezza all'autorità stessa. Infatti se per qualche motivo dovesse venire meno l'affidabilità di una autorità intermedia, è possibile sostituire l'entità intermedia con una nuova autorità senza dovere invalidare i certificati emessi dall'autorità di più alto livello.
- Tramite una gerarchia si può garantire interoperabilità ai certificati: è quindi possibile considerare attendibile un certificato emesso, ad esempi, alle isole Hawaii.

La gerarchia delle autorità produce un effetto immediato: le autorità al di sotto della *root* ereditano le proprietà di quest'ultima, ossia se la *root* è abilitata a certificare la posta ed i siti web, ma non le applicazioni anche i certificatori che da questa dipendono avranno le medesime caratteristiche.

Il meccanismo di gerarchia non ha però avuto successo. L'insuccesso è dovuto sostanzialmente al fatto che le CA corporate non hanno bisogno di essere inserite in una gerarchia in quanto il loro scopo è quello di fornire sicurezza all'interno dell'organizzazione alla quale appartengono. A questo è necessario aggiungere che alcune CA corporate, in particolare quelle bancaria, mal tollerano il fatto di avere un'autorità a loro superiore che, di fatto, le costringerebbe a considerare attendibili certificati emessi da altri.

Certificate Revocation List

Il ***C**ertificate **R**evocation **L**ist* (**CRL**) è l'elenco dei certificati che sono stati revocati dall'autorità che li ha emessi. I motivi di revoca possono essere diversi: ad esempio per qualche motivo viene violata la chiave privata dell'utente e diventa quindi possibile falsificarne la firma, oppure vengono meno i motivi per cui l'utente ha necessità di mantenere un certificato. Si potrebbe anche lasciare scadere il certificato: la situazione che si verificherebbe tra il momento della violazione e quello della

scadenza sarebbe ingestibile.