ALBERTO PELENC

La crittografia

Classe V B – Liceo scientifico statale "A. Volta"
2004 – 2005

Introduzione

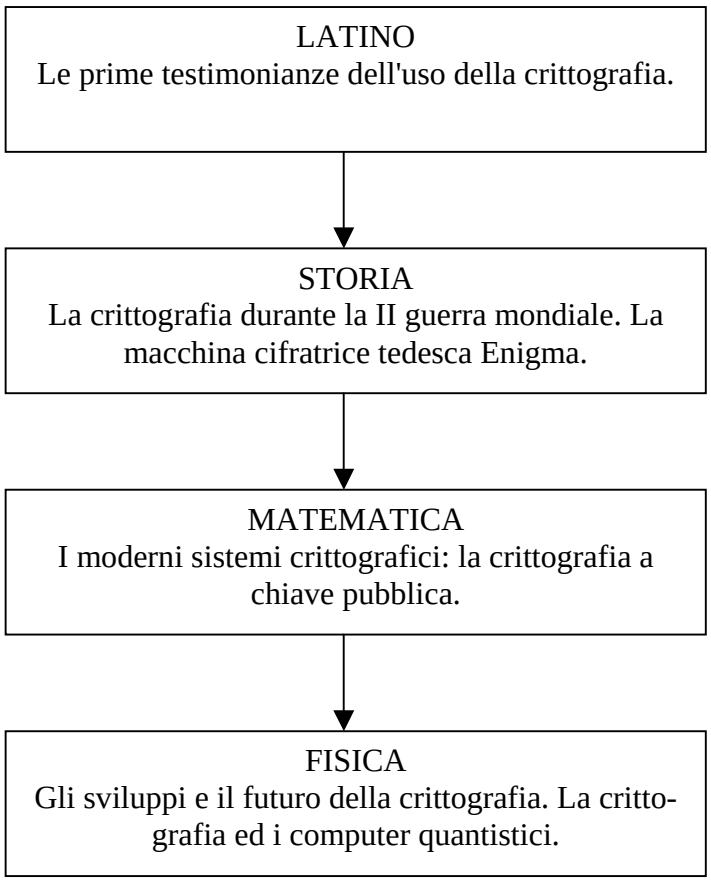
Fin dalle sue prime applicazioni, la crittografia è sempre stata l'espressione di una naturale esigenza dell'uomo: la riservatezza. E' nella natura umana il desiderio di comunicare in maniera sicura; la crittografia è il principale strumento a disposizione dell'uomo per raggiungere questo scopo.

E' affascinante scoprire come nel tempo, parallelamente ad altre discipline e conoscenze, la crittografia si sia evoluta, sia avanzata e abbia raggiunto ogni volta risultati sempre migliori.

Oggigiorno, senza accorgercene, utilizziamo continuamente una grande quantità di oggetti e servizi che non potrebbero esistere senza la crittografia moderna. E' quindi interessante, partendo dalle sue origini, vedere come questa disciplina sia cambiata nel tempo.

Passando in rassegna alcuni dei principali personaggi della storia della crittografia, è bene ricordare che la crittografia non è solo l'arte di creare codici segreti, ma è anche l'arte di violarli. Se non vi fosse sempre stato un costante "scontro" tra i creatori di codici e coloro che tentano di violarli, ora non avremmo a disposizione le avanzate tecniche crittografiche moderne.

Mappa concettuale



LATINO – L'uso per scopi militari della crittografia da parte di Cesare. Le prime testimonianze di sistemi di crittografia.
STORIA – La macchina Enigma: l'invenzione tedesca per comunicare in maniera sicura ed i tentativi degli Alleati di forzare questo sistema.
MATEMATICA – La crittografia a chiave pubblica (RSA), le implicazioni con i numeri primi e la fattorizzazione.
FISICA – I computer quantistici e la crittografia quantistica.

Le origini e i primi sviluppi della crittografia

La crittografia è l'arte di trasformare un testo, o delle informazioni in generale, in qualcosa di incomprensibile e poi ritrasformare il messaggio crittografato nel testo originale. Proprio per questa ragione questa era già conosciuta e utilizzata dai greci e dai romani prevalentemente per scopi militari o diplomatici; già allora, infatti, era fondamentale che le comunicazioni tra i sovrani, o gli ordini di carattere militare potessero essere spediti senza che il nemico fosse in grado di comprendere il contenuto del messaggio, nel caso in cui esso fosse stato intercettato. Nel *De Bello Gallico*, Cesare offre un'importante testimonianza dell'uso della crittografia per scopi militari: egli, infatti, per inviare un messaggio senza che esso potesse essere letto dal nemico, utilizzò un particolare tipo di cifratura per sostituzione, la cosiddetta *cifratura di Cesare*.

Figura 1 Cesare

La cifratura di Cesare è un caso particolare della cifratura per sostituzione, essa consiste nell'assegnare ad ogni lettera dell'alfabeto un'altra lettera ed utilizzare questo secondo alfabeto per cifrare i messaggi.

```
a  b  c  d  e  f  g  h  i  l  m  n  o  p  q  r  s  t  u  v  z
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
g  z  v  e  h  i  d  n  q  m  l  o  b  u  t  a  r  s  i  c  p
```

L'esempio mostra l'alfabeto italiano in chiaro con il suo corrispondente cifrante, ogni volta che nel messaggio originale c'è la lettera `e` nel messaggio cifrato sarà presente la lettera `h` e così via per tutte le lettere dell'alfabeto. Utilizzando questo alfabeto cifrante il messaggio di testo "`messaggio`" è trasformato nel testo "`lhrrgddqb`". Utilizzando una disposizione casuale delle lettere nell'alfabeto cifrante (che è quello utilizzato per cifrare i testi – quello posto più in basso nell'esempio) è possibile realizzare più di 50 miliardi di miliardi di diversi alfabeti; se anche il messaggio fosse intercettato sarebbe necessario provare un numero enorme di combinazioni di alfabeti cifranti per riuscire a decifrarlo; questo grande numero di combinazioni, a quel tempo, offriva una buona sicurezza.
Cesare per inviare messaggi cifrati utilizzava una semplificazione di questo metodo: egli semplicemente usava come alfabeto cifrante il normale alfabeto sfasato di un certo numero di lettere, in questo modo si possono generare un numero di cifrature diverse pari al numero di lettere dell'alfabeto utilizzato (nel caso italiano 21). Questo sistema aveva il vantaggio di rendere praticamente banale la memorizzazione dell'alfabeto cifrante (diversamente sarebbe stato necessario imparare a memoria l'intero alfabeto cifrante) ma comportava lo svantaggio che un eventuale nemico avrebbe dovuto provare un numero limitato di combinazioni (nel caso italiano 21) decisamente minore dei 50miliardi di miliardi possibili con una combinazione completamente casuale dell'alfabeto cifrante.

In crittografia generalmente la comunicazione avviene rispettando il seguente schema generale.

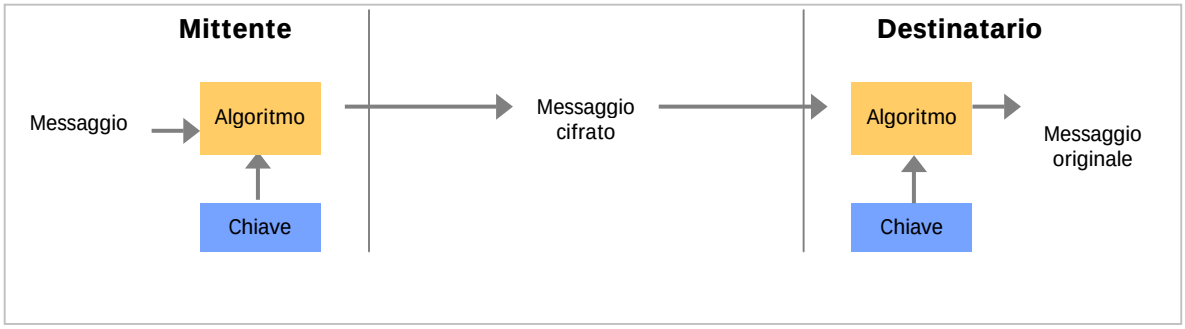


Figura 2 Schema che rappresenta i passaggi di una cifratura in generale.

Il mittente utilizza un algoritmo per cifrare il messaggio, cioè una procedura che trasforma una informazione non cifrata nel suo corrispondente cifrato; oltre ad un algoritmo il mittente ha bisogno di una chiave, che è fornita all'algoritmo affinché possa essere eseguita la cifratura.
Esistono molti algoritmi diversi, in questi casi ad esempio, l'algoritmo utilizzato è quello della cifratura per sostituzione, oltre all'algoritmo è necessaria la chiave, che in questi esempi è uno dei tanti alfabeti cifranti possibili. Affinché la comunicazione sia sicura solo la chiave deve essere mantenuta segreta, l'algoritmo può anche essere conosciuto dal nemico; quest'ultimo infatti, per decifrare il messaggio, dovrebbe provare tutte le chiavi possibili (che per quanto riguarda la cifratura per sostituzione sono tutti i possibili alfabeti cifranti).
Una volta che il messaggio è stato cifrato utilizzando l'algoritmo e la chiave scelta, il messaggio può essere inviato al destinatario che, utilizzando la stessa chiave utilizzata dal mittente, è in grado di decifrare il messaggio e recuperare l'originale.

Visto l'enorme numero di chiavi possibili da utilizzare con la cifratura per sostituzione, si ritenne che essa fosse inviolabile, ed effettivamente per molti secoli questo sistema resistette a qualsiasi tentativo di violazione, ma nel 750 d.C., la civiltà islamica raggiunse un livello culturale e scientifico sufficiente per scoprire che era possibile violare in tempi brevi la crittografia per sostituzione. In quel periodo infatti la civiltà araba fu toccata da un grande sviluppo delle arti matematiche, linguistiche e statistiche, e queste permisero di scoprire che, se si analizza statisticamente un testo, si può notare che alcune lettere sono presenti con una frequenza maggiore rispetto ad altre, o in generale che ogni lettera tende ad avere una propria frequenza che la caratterizza. Ad esempio, nella lingua italiana moderna, la lettera più frequente è la *e*, quindi la *a* e a seguire la *i*; la tabella mostra la frequenza delle lettere nei testi in italiano.

Tabella 1 Questa tabella mostra le frequenze caratteristiche di ogni lettera in un testo in italiano.

Lettera	Frequenza	Lettera	Frequenza	Lettera	Frequenza
A	11.74	H	1.54	Q	0.51
B	0.92	I	11.28	R	6.37
C	4.50	L	6.51	S	4.98
D	3.73	M	2.51	T	5.62
E	11.79	N	6.88	U	3.01
F	0.95	O	9.83	V	2.10
G	1.64	P	3.05	Z	0.49

Partendo da queste considerazioni, gli arabi trovarono un modo per violare le cifrature per sostituzione, dando così origine alla *crittoanalisi*, cioè la scienza dell'interpretazione di un messaggio di cui non si conosce la chiave. Poiché nelle cifrature per sostituzione ad ogni lettera dell'alfabeto originale viene sostituito un simbolo o un'altra lettera dell'alfabeto, la frequenza delle lettere si conserva anche nel messaggio cifrato; nell'esempio precedente la lettera `e` verrà cifrata sempre con la corrispondente `h` e così via per tutte le altre. È quindi possibile eseguire l'analisi delle frequenze anche sui messaggi cifrati e confrontare le frequenze ottenute con quelle di una tabella delle frequenze della stessa lingua del messaggio per ottenere indicazioni affidabili riguardo alla corrispondenza tra le lettere del messaggio cifrato e quelle dell'alfabeto in chiaro e quindi per giungere alla decifrazione del messaggio. Questo fu esattamente il procedimento usato dai crittoanalisti arabi per violare le cifrature per sostituzione monoalfabetiche: bisogna infatti precisare che tutte le cifrature per sostituzione fino ad ora incontrate sono monoalfabetiche, cioè utilizzano un solo alfabeto cifrante.

Quando divenne evidente che i vecchi sistemi di cifratura non erano più sicuri, poiché violabili con il sistema dell'analisi delle frequenze, si cercarono nuovi sistemi in grado di resistere ai metodi della crittoanalisi: le cifrature per sostituzione polialfabetiche. È bene comunque precisare che la più antica testimonianza del procedimento di analisi delle frequenze risale al IX secolo (per opera di Abu al-Kindi) mentre la conoscenza di queste tecniche si diffuse in Occidente attorno al XV secolo, solo sei secoli dopo; ma sebbene in Occidente le arti crittografiche fossero rimaste molto indietro, proprio nel XV secolo si ebbe un grande sviluppo di queste ultime che portò alla nascita delle cifrature per sostituzione polialfabetiche.
Una cifratura per sostituzione polialfabetica si differenzia da quelle fino ad ora considerate per il fatto che anziché utilizzare un solo alfabeto cifrante se ne usano più di uno; con quest'accorgimento il metodo dell'analisi delle frequenze perde di utilità poiché la stessa lettera nel messaggio in chiaro può essere tradotta con lettere diverse nel messaggio cifrato. Ecco un esempio che mostra l'applicazione di questa tecnica; quelli rappresentati sono tre alfabeti: il primo è quello in chiaro, segue il primo alfabeto cifrante e poi il secondo alfabeto cifrante.

```
a  b  c  d  e  f  g  h  i  l  m  n  o  p  q  r  s  t  u  v  z
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
g  z  v  e  h  i  d  n  q  m  l  o  b  u  t  a  r  s  i  c  p
t  m  l  q  c  u  p  d  v  h  f  i  e  r  a  n  z  g  b  o  s
```

Figura 3 Ritratto di Vigenère

Usando un doppio alfabeto cifrante è necessario introdurre una regola che stabilisca quale alfabeto dei due utilizzare per cifrare ogni lettera. Un metodo utilizzato quando gli alfabeti cifranti sono due è quello di alternarli ad ogni lettera nella cifratura del testo. Con questo sistema la prima lettera del messaggio viene codificata usando il primo alfabeto, la seconda lettera usando il secondo alfabeto, la terza lettera nuovamente con il primo alfabeto e così via. Il testo, usato già in un esempio precedente, "`messaggio`" viene ora cifrato in "`lcrzgpdvb`" dove le stesse lettere nel messaggio in chiaro non sono più codificate con la stessa lettera nel messaggio cifrato.

Un’altra cifratura per sostituzione polialfabetica è quella denominata "di Vigenère" basata, anziché su soli due alfabeti, su 21 (nel caso della lingua italiana). Con questa cifratura ogni lettera dell’alfabeto in chiaro può essere cifrata con ogni lettera dell’alfabeto, creando grossi ostacoli al metodo dell’analisi delle frequenze tanto da essere definita "Le chiffre indéchiffrable" (la cifratura indecifrabile), ciò nonostante tale sistema può essere violato con una versione leggermente più sofisticata dell’analisi delle frequenze.

	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
A	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
B	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	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	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
F	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
G	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	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	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
T	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
U	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	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	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

Figura 4 Alfabeti di Vigenère

La situazione continuò a volgere a favore dei crittoanalisti fino agli inizi del 1900 quando, grazie anche ad un miglioramento delle conoscenze tecniche e scientifiche, fu possibile realizzare le prime macchine per cifrare.

L'automatizzazione della crittografia: Enigma e le "bombe"

Agli inizi del XX secolo, nuovi ed efficienti sistemi di comunicazione (basati sul telegrafo e la radio) erano disponibili agli eserciti dei paesi che avrebbero dato successivamente inizio alla prima guerra mondiale. Questo nuovo sistema di comunicazione presentava però un grave problema: le comunicazioni erano facilmente intercettabili, e tutto ciò era evidentemente incompatibile con le necessità di segretezza delle comunicazioni militari. I primi tentativi di inventare dei nuovi sistemi crittografici in grado di resistere alla violazione da parte di stati nemici non ebbero successo; ciò fu dovuto anche al fatto che i principali stati europei si erano già dotati di vere e proprie strutture in cui esperti crittoanalisti si occupavano della decifrazione dei messaggi nemici intercettati.

Prima di trattare effettivamente il funzionamento delle prime macchine per cifrare è necessario accennare ad un sistema crittografico introdotto da G.S.Vernam nel 1917, chiamato anche *cifratura a blocco monouso* (one time pad). Questo sistema prevede l'uso di una chiave completamente casuale lunga quanto l'intero messaggio; con la chiave viene cifrato e decifrato il testo. Se la chiave è casuale, non viene mai riutilizzata ed è lunga quanto il messaggio che viene cifrato, allora il cifrario di Vernam è *inviolabile*. È infatti possibile dimostrare matematicamente che un nemico non ha modo di forzare la cifratura, nemmeno provando tutte le chiavi possibili, poiché, provandole tutte, otterrebbe anche tutti i possibili messaggi di testo, compresi tutti quelli che hanno un senso, e quindi non sarebbe comunque in grado di stabilire quale di esso sia quello effettivamente trasmesso. Sebbene il cifrario di Vernam sia assolutamente sicuro esso in pratica non viene quasi mai utilizzato poiché presenta due grossi problemi: la creazione di lunghe chiavi casuali non è per nulla un compito banale e lo scambio delle chiavi (necessario al destinatario per decifrare il messaggio) richiede un canale sicuro.
Pare che uno dei pochi usi reali di questa cifratura sia stata la comunicazione militare e diplomatica tra USA e URSS durante la guerra fredda, dove la necessità di segretezza era tale da rendere tollerabili i costi di generazione e distribuzione delle chiavi casuali.

Una macchina per cifrare che ebbe una grande importanza durante la seconda guerra mondiale fu la macchina Enigma: essa fu utilizzata dai tedeschi per cifrare le proprie comunicazioni militari. Questo sistema crittografico era un enorme passo avanti rispetto a qualsiasi metodo utilizzato fino ad allora e seppe, inizialmente, resistere agli attacchi dei crittoanalisti degli altri stati europei.
La macchina Enigma fu inventata nel 1918 dal tedesco Arthur Scherbius che la brevettò e decise di proporla come strumento crittografico per i ricchi uomini d'affari tedeschi che avevano bisogno di comunicazioni sicure; in quest'ambito però la macchina non ebbe particolare successo. Quando i vertici militari tedeschi si resero conto che buona parte delle loro comunicazioni durante la prima guerra mondiale erano state intercettate e decifrate, decisero che era necessario adottare un nuovo metodo crittografico più sicuro di quelli fino ad allora utilizzati e la scelta cadde sulla macchina Enigma. Nel 1925 iniziò la produzione di una grande quantità di macchine Enigma, parzialmente diverse dalla versione destinata agli uomini d'affari, che vennero fornite in dotazione alle forze armate tedesche.
La macchina Enigma era formata da diversi componenti

Figura 5 La macchina Enigma

che agivano in modo diverso; innanzi tutto la macchina disponeva di una tastiera con 26 lettere e un corrispondente pannello con 26 lampadine ognuna per ogni lettera dell'alfabeto usate per indicare il risultato della cifratura di una lettera. L'operatore premeva una lettera sulla tastiera e osservava quale lampadina si accendeva stabilendo così la lettera corrispondente nel messaggio cifrato; il procedimento era ripetuto fino a che il messaggio era stato cifrato per intero. La macchina disponeva inoltre, nella prima versione, di tre dischi scambiatori e un pannello a prese multiple. I dischi scambiatori erano dei dischi attraversati da un complicato circuito elettrico che congiungeva 26 contatti da un lato con altri 26 dall'altro lato: ogni disco era l'equivalente elettronico di una cifratura per sostituzione monoalfabetica; infatti ogni contatto da un lato (che rappresentava una lettera) veniva collegato con un altro dall'altra parte del disco tramite un circuito elettrico. La novità introdotta da Scherbius fu di rendere rotanti questi dischi, e fare in modo che la rotazione avvenisse con la pressione di ogni tasto sulla tastiera. La pressione di un tasto faceva quindi avanzare il primo disco di uno scatto; quando il disco aveva compiuto un giro completo un ulteriore scatto provocava anche il movimento del secondo disco, ventisei scatti del secondo disco provocavano uno scatto del terzo disco. Oltre ai tre dischi era presente anche il riflettore che era un disco con i contatti da un solo lato e che provvedeva a far ritornare il segnale elettrico proveniente dai dischi scambiatori nuovamente indietro verso il pannello con le lampadine. I dischi inoltre potevano essere spostati e messi con un ordine a piacere. Questi fattori portano le possibili chiavi date dalla posizione iniziale dei singoli dischi e dall'ordine con cui essi sono disposti a 105456 (26 x 26 x 26 x 6). Sebbene questo fosse già un elevato numero di chiavi possibili, esso non era comunque sufficiente a garantire la resistenza ad un attacco che le provasse tutte, perciò Scherbius introdusse anche il pannello a prese multiple. Questo pannello consentiva, nella prima versione, di scambiare sei coppie di lettere a piacimento tramite degli appositi cavetti aumentando di un fattore 100391791500 le possibili chiavi. Complessivamente il numero di chiavi raggiungeva il valore di circa 10 milioni di miliardi.

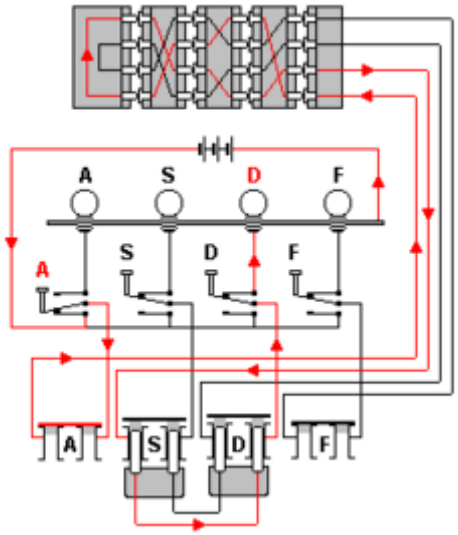


Figura 6 Schema del funzionamento di Enigma, questo esempio mostra la macchina con sole quattro lettere. Il percorso segnato in rosso è quello compiuto dalla corrente elettrica attraverso le varie componenti della macchina.

Figura 7 Il pannello a prese multiple della macchina Enigma; usando due cavetti sono scambiate le lettere S-O e A-J.

Dal 1926 i crittoanalisti inglesi, francesi e statunitensi, che fino ad allora avevano decifrato buona parte dei messaggi tedeschi, si trovarono di fronte a crittogrammi che non erano più in grado di violare, crittogrammi cifrati con la macchina Enigma. Francia, Inghilterra e Stati Uniti però, anziché incrementare i loro sforzi, decisero di non cimentarsi nella violazione del sistema Enigma che era ritenuto inviolabile. Diversamente da questi stati, la Polonia, poiché più preoccupata da una possibile offensiva militare tedesca, continuò i propri sforzi per decifrare i messaggi tedeschi. I crittoanalisti polacchi, con le informazioni ottenute da

un agente segreto francese, furono in grado di costruire una copia della macchina Enigma utilizzata dai militari tedeschi. Avendo ora a disposizione una macchina Enigma uguale a quella usata dai tedeschi per cifrare le proprie informazioni, i matematici dell'ufficio polacco per intercettazione e la decifrazione dei messaggi poterono cimentarsi nell'arduo tentativo di fare breccia nel sistema crittografico nemico.

L'esercito tedesco stabilì alcune regole per l'uso della macchina Enigma: ogni giorno era previsto l'uso di una particolare chiave giornaliera elencata in un apposito blocco distribuito agli operatori Enigma. Per chiave si intende una particolare combinazione di tutti quei parametri che possono essere cambiati nella macchina: il pannello a prese multiple, la disposizione degli scambiatori e l'orientamento degli scambiatori. Questa chiave giornaliera non veniva usata però per cifrare i messaggi veri e propri ma con essa era cifrata una particolare disposizione degli scambiatori (ripetuta due volte) usata per cifrare solo un particolare messaggio; con questo sistema si evitava di cifrare centinaia di messaggi al giorno con la stessa chiave, rendendo quindi più difficile il compito ai crittoanalisti.

Figura 8 Foto di Marian Rejewski.

Tra i matematici polacchi impegnati nel tentativo di violare Enigma, quello che ebbe un ruolo dominante fu Marian Rejewski. Egli sfruttò una debolezza del metodo con cui i tedeschi comunicavano: come detto ogni messaggio era cifrato con una propria chiave (disposizione degli scambiatori) che veniva inviata, ripetendola due volte, prima del messaggio vero e proprio; ad esempio se la chiave del messaggio era VHG, l'operatore Enigma avrebbe cifrato con la chiave giornaliera prima il messaggio VHGVHG e quindi, impostando gli scambiatori secondo la chiave del messaggio, il messaggio vero e proprio. Questa ripetizione della chiave del messaggio permise a Rejevski di fare breccia in Enigma.

Rejevski partì dalla considerazione che poiché ogni messaggio iniziava con la chiave ripetuta due volte, la prima lettera era la stessa della quarta, la seconda lettera la stessa della quinta e la terza lettera la stessa della sesta. Il fatto che queste lettere, sebbene fossero le stesse, erano cifrate diversamente era dovuto al fatto che gli scambiatori ruotavano durante l'uso della macchina. Poiché ogni giorno egli riceveva un grande numero di messaggi cifrati era spesso in grado di formare delle tabelle in cui inseriva le corrispondenze tra le prime lettere e le quarte, le seconde e le quinte e così via; fatto ciò si rese conto che si formavano delle concatenazioni di lettere e che queste, sebbene influenzate dalla disposizione del pannello a prese multiple, non variavano quanto a lunghezza e a numero. In altre parole il numero e la lunghezza di queste sequenze è una specie di impronta digitale di una particolare disposizione degli scambiatori. L'ufficio polacco si cimentò quindi nell'arduo compito di stilare una lista che mettesse in corrispondenza ogni assetto degli scambiatori con una particolare struttura delle sequenze; questo compito richiese un anno, ma una volta terminato era possibile determinare l'assetto giornaliero degli scambiatori in breve tempo. Il pannello a prese multiple era ancora un ostacolo, ma una volta ottenuta la disposizione degli scambiatori, esso poteva essere considerato come una semplice cifratura per sostituzione monoalfabetica. Rejevski fu inoltre in grado di realizzare delle particolari macchine, chiamate "bombe", probabilmente a causa del ticchettio che facevano quando erano in funzione, che riuscivano a determinare la corretta disposizione degli scambiatori in un paio d'ore. Poiché le possibili combinazioni degli scambiatori sono sei, furono realizzate sei bombe.

Figura 9 Struttura degli scambiatori della macchina Enigma.

Nel dicembre 1938 i tedeschi aumentarono la sicurezza di Enigma fornendo a tutti gli operatori altri due nuovi scambiatori, portando quindi da sei a sessanta le possibili combinazioni. I polacchi non potevano però conoscere la struttura dei nuovi scambiatori ne costruire altre cinquantaquattro bombe per verificare tutte le possibili combinazioni degli scambiatori. Dal 1939 quindi i polacchi non furono più in grado di decifrare i messaggi tedeschi; nonostante ciò essi decisero che le loro scoperte, se fosse scoppiata la guerra, non avrebbero dovuto essere perse e quindi contattarono gli inglesi e i francesi. Poco prima dell'invasione della Polonia da parte di Hitler, i polacchi riuscirono a fornire ai loro alleati alcune riproduzioni di Enigma e tutti i dettagli dei metodi usati da Rejewski per violare Enigma, inclusi i dettagli tecnici delle bombe.
In Gran Bretagna, grazie al metodo Rejewski, si formarono presto interi gruppi specializzati nella decrittazione dei messaggi tedeschi cifrati con Enigma. Questi gruppi erano formati da linguisti, umanisti, matematici e scienziati radunati perlopiù a Bletchley Park; tra queste persone vi era anche Alan Turing che ebbe una grande importanza nella violazione del sistema Enigma. Infatti sebbene gli inglesi utilizzassero ampiamente il metodo inventato dal matematico polacco, essi erano molto preoccupati per la possibilità che i tedeschi, rendendosi conto che la ripetizione della chiave messaggio era una pratica nefasta per la sicurezza del sistema Enigma, cambiassero le regole con cui i messaggi erano cifrati, ed in particolare temevano rimuovessero questa ripetizione su cui si basava interamente il sistema Rejewski. Nonostante ciò i crittoanalisti inglesi avevano scoperto altri punti deboli dell'uso della macchina Enigma: ad esempio spesso gli operatori, come chiavi messaggio, anziché scegliere chiavi casuali usavano stringhe di tre lettere consecutive sulla tastiera come HJK o CVB; queste chiavi di messaggio vennero soprannominate dagli inglesi *cillies*. Inoltre altre regole stabilite per l'uso di Enigma prevedevano che, per la disposizione giornaliera degli scambiatori, nessuno di essi potesse occupare lo stesso posto per due giorni consecutivi, riducendo quindi le possibili combinazioni, oppure che, per l'uso del pannello a prese multiple, due lettere consecutive non potessero essere scambiate.

Figura 10 Fotografia di Alan Turing.

Alan Turing, nato nel 1912, entrò a Bletchley nel 1939, a lui fu affidato il compito di trovare un modo di violare Enigma che non si basasse sulla ripetizione delle chiavi messaggio. Egli notò che spesso molti messaggi avevano una struttura rigida sia nei contenuti sia nella struttura, ad esempio i bollettini meteorologici spesso contenevano la parola wetter e essa era situata in alcune precise posizioni nel messaggio, questo consentiva di ipotizzare alcuni abbinamenti tra il testo cifrato e il testo in chiaro. Inoltre, per la stessa natura della macchina Enigma, una lettera non poteva mai essere cifrata come se stessa, fornendo maggiori informazioni sui cosiddetti *cribs*: in sostanza questi indizi riguardano la disposizione degli scambiatori. Una volta individuate alcune regolarità tra testo cifrato e testo in chiaro ipotizzato era necessario calcolare tutte le possibili combinazioni degli scambiatori che generassero tale regolarità; questo compito fu affidato a grosse macchine, progettate inizialmente da Turing, discendenti delle bombe di Rejewski e quindi chiamate anch'esse bombe. La struttura di queste macchine permetteva inoltre di ignorare la disposizione del pannello a prese multiple, riducendo quindi enormemente il numero di calcoli da eseguire; una volta trovata la disposizione degli scambiatori era sufficiente individuare gli scambi di lettere effettuati dal pannello a prese multiple. La prima bomba, fatta costruire a Letchworth, fu disponibile il 14 marzo 1940, essa tuttavia era molto più lenta del previsto e se ne progettò presto una nuova versione. Nel frattempo, il 10 maggio, i tedeschi cambiarono le procedure di cifratura dei messaggi e abolirono la ripetizione della chiave messaggio rendendo ineffi-

cace il metodo Rejewski. Tre mesi dopo fu infine disponibile la nuova bomba a Bletchley Park consentendo nuovamente agli inglesi di decifrare i messaggi Enigma.

Figura 11 Ricostruzione moderna di una bomba di Turing.

La crittografia a chiave pubblica

I crittoanalisti britannici avevano dimostrato che la macchina Enigma era una cifratura violabile ed erano riusciti ad ottenere questo risultato proprio con l'uso di altre macchine: le bombe. Oltre alla macchina Enigma furono inventate anche altri sistemi di codifica dei messaggi basati su macchine sempre più complesse; vi era quindi una competizione tra crittografi e crittoanalisti basata su macchinari sempre più complicati e veloci. Tale situazione diede una forte spinta allo sviluppo dei primi computer (uno di questi fu l'ENIAC) che furono presto usati anche per realizzare cifrature di una resistenza senza precedenti.
Innanzitutto è bene precisare come un messaggio formato da lettere venga gestito da una macchina che notoriamente opera su numeri binari, cioè usa solamente le cifre **0** e **1**, che nei circuiti rappresentano la presenza o l'assenza di un segnale elettrico. Queste cifre sono chiamate *bit* (binary digit) e sono la più piccola unità di informazione che i calcolatori possano elaborare; le lettere sono particolari sequenze di bit che il calcolatore interpreta come numeri e che quindi è in grado di gestire in maniera efficiente e precisa. Uno degli standard più diffusi di conversione delle lettere dell'alfabeto (e non solo, ma anche della punteggiatura) in formato binario è l'ASCII (America Standard Code for Information Interchange). Questo standard assegna ad ogni carattere un particolare codice in forma binaria, poiché i caratteri che era necessario includere nello standard erano minori di 128, si decise che ad ogni carattere sarebbe stato abbinato un numero binario di 7 cifre (infatti 2^7=128), ma per questione di semplicità, ogni carattere verrà convertito in un numero binario di 8 cifre in quanto, per definizione, un numero binario di 8 cifre è chiamato *byte*. E' quindi evidente che, nella tabella, i numeri binari di 8 cifre abbinati ad ogni carattere inizieranno con uno zero che corrisponde all'ottavo bit, aggiunto per rendere ogni cifra codificata con un byte, come in realtà avviene nei moderni computer.

Tabella 2 Tabella dei corrispondenti codici ASCII per le lettere maiuscole. Non è riportata tutta la tabella di conversione ma unicamente la parte riguardante le lettere maiuscole da A a Z (0x41-0x5A).

Lettera	Corrispondente ASCII		Lettera	Corrispondente ASCII	
A	0x41	01000001	N	0x4E	01001110
B	0x42	01000010	O	0x4F	01001111
C	0x43	01000011	P	0x50	01010000
D	0x44	01000100	Q	0x51	01010001
E	0x45	01000101	R	0x52	01010010
F	0x46	01000110	S	0x53	01010011
G	0x47	01000111	T	0x54	01010100
H	0x48	01001000	U	0x55	01010101
I	0x49	01001001	V	0x56	01010110
J	0x4A	01001010	W	0x57	01010111
K	0x4B	01001011	X	0x58	01011000
L	0x4C	01001100	Y	0x59	01011001
M	0x4D	01001101	Z	0x5A	01011010

Il testo "ciao" convertito in codice ASCII in binario diventa:

```
01000011 01001001 01000001 01001111
```

oppure, se espresso in formato esadecimale, diventa:

```
0x43 0x49 0x41 0x4F (0x4F414943)
```

E' importante tenere presente che lo standard ASCII non riguarda la crittografia ma è unicamente un modo di convertire un messaggio scritto con comuni lettere alfabetiche in un

messaggio numerico facilmente gestibile dai computer; sebbene si potrebbe intendere l'ASCII come una cifratura per sostituzione monoalfabetica con alfabeto cifrante fisso, il procedimento di conversione da testo a ASCII verrà definito codifica ASCII e non cifratura ASCII.

Quando gli elaboratori elettronici iniziarono a diffondersi anche in ambiti privati, e furono utilizzati anche per scambiare informazioni cifrate, divenne necessario trovare uno standard con cui crittografare i messaggi che fosse largamente accettato. Il primo algoritmo crittografico ad essere standardizzato fu il DES (acronimo di Data Encryption Standard) sviluppato da Horst Feistel e adottato ufficialmente nel 1976. Il DES era un complesso algoritmo che operava su stringhe di numeri binari (stringhe di bit) e prevedeva sia sostituzioni sia scambi operati su blocchi di 64bit. Esso fu considerato uno dei più sicuri sistemi di cifratura allora esistenti, ma al momento della standardizzazione l'NSA (National Security Agency) statunitense si batté perché il numero di chiavi possibili fosse limitato a 56bit, che in altre parole corrisponde a 2^{56}=72057594037927936. Tale numero era considerato dall'NSA sufficientemente grande per garantire la sicurezza dei privati, ma al contempo permetteva a chi aveva apparecchiature molto avanzate, come l'agenzia stessa, di violare la cifratura provando tutte le chiavi possibili in tempi ragionevoli. Attualmente 56bit non sono più considerati sicuri ma generalmente vengono usati 128bit o più.

Sebbene fosse ormai disponibile un sistema di cifratura sufficientemente sicuro, permaneva un altro problema: *la distribuzione delle chiavi*. Prima di poter comunicare, infatti, è necessario stabilire una chiave comune con cui i messaggi verranno cifrati e decifrati, ma la distribuzione della chiave spesso presenta notevoli problemi. Infatti, se lo scambio della chiave avviene tramite un canale non sicuro, anche i messaggi che verranno cifrati con essa sono soggetti a intercettazione e decifrazione; una soluzione è quella di scambiarsi la chiave di persona incontrandosi appositamente per questo, ma spesso questo metodo non può essere applicato. Ad esempio tutte le volte che si usa una carta di credito è necessario che il numero raggiunga la banca in modo sicuro attraverso linee telefoniche che generalmente non sono sicure, è quindi necessario cifrare la comunicazione, ma non è generalmente possibile gestire i costi e i problemi pratici della distribuzione di chiavi per ogni pagamento con carta di credito. Lo stesso discorso si applica a molti altri campi, e Internet non ha fatto altro che incrementare la necessità di un modo sicuro per lo scambio delle chiavi.
Lo schema rappresenta un normale scambio di informazione tramite una cifratura a chiave privata.

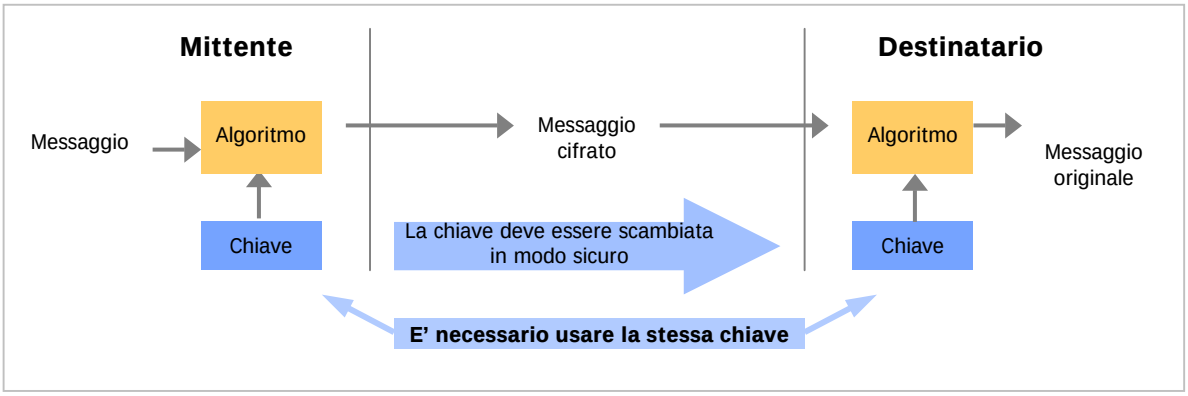


Figura 12 Schema che rappresenta i passaggi richiesti dalle cifrature a chiave privata.

Prima che il problema della distribuzione delle chiavi fosse realmente risolto, era già noto un metodo che, almeno in linea teorica, permette lo scambio di un messaggio in maniera sicura senza la necessità di scambiare delle chiavi. Innanzitutto è necessario precisare che

da qui in poi, per identificare il mittente, il destinatario e un eventuale nemico che tenta di intercettare il messaggio si useranno i nomi Alice, Bob ed Eva; questi tre nomi sono comunemente utilizzati quando si parla di crittografia per identificare appunto le varie parti coinvolte nel processo di comunicazione.
L'esempio citato parte dal presupposto che Alice voglia inviare un messaggio a Bob in maniera sicura senza che essi debbano scambiarsi alcuna chiave. Alice procede quindi mettendo il messaggio in una valigetta e chiudendola con un lucchetto, la chiave del lucchetto resta in mano ad Alice. Bob riceve la valigetta, che ovviamente nessuno ha potuto aprire poiché chiusa con il lucchetto di Alice, nemmeno Bob può aprire la valigetta, ma può aggiungere pure lui un lucchetto e tenere la chiave con se. Quindi Bob rispedisce indietro la valigetta, che ora ha due lucchetti, ad Alice la quale non può rimuovere il lucchetto di Bob ma può togliere quello che lei ha inizialmente messo. La valigetta, che ora ha solo più un lucchetto, viene rispedita a Bob che può agevolmente togliere il lucchetto (il suo) e leggere il messaggio contenuto nella valigetta. Alice e Bob sono quindi riusciti a comunicare in maniera sicura senza avere la necessità di scambiarsi una chiave.

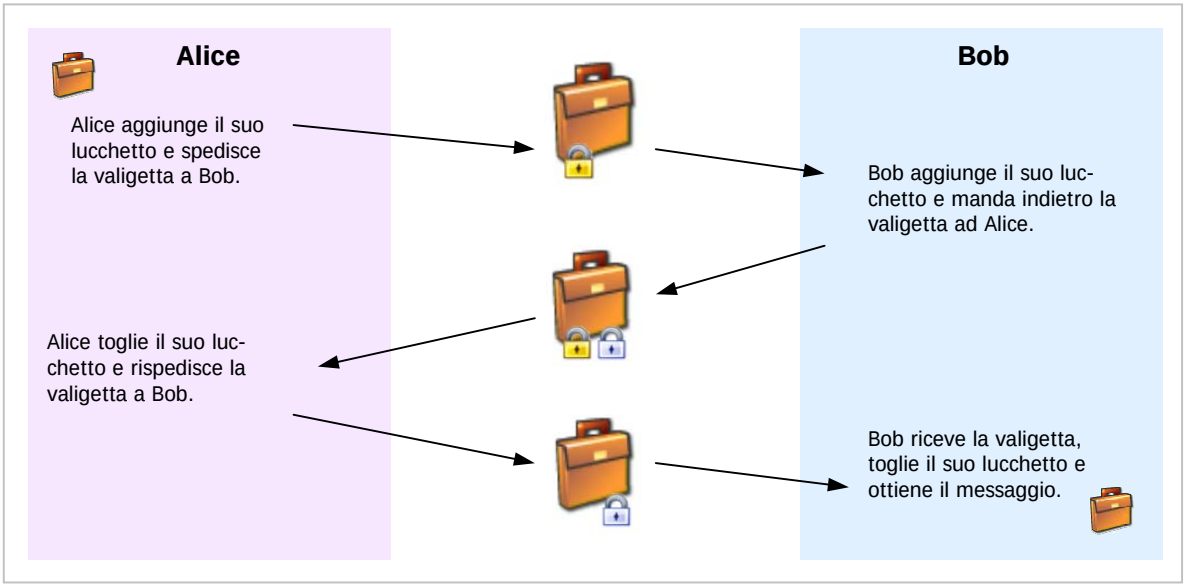


Figura 13 Schema dell'esempio dei "due lucchetti".

Questo esempio stuzzicò a lungo la mente di due ricercatori statunitensi nel campo della crittografia: Whitfield Diffie e Martin Hellman. Essi infatti, a partire dal 1974, cercarono a lungo una versione reale del metodo utilizzato nell'esempio dei due lucchetti, in quanto nella realtà, spesso, preso esattamente come descritto, non è applicabile. Partendo dal presupposto che nella realtà non si usino lucchetti e valigette, ma comunicazioni elettroniche, è necessario trovare un algoritmo di cifratura che si comporti come i due lucchetti dell'esempio, e questo è raro. Infatti se si prende un messaggio e lo si cifra, quindi lo si cifra per una seconda volta (il che corrisponde all'aggiunta del secondo lucchetto), quando si tenta di rimuovere la prima cifratura (il primo lucchetto) ciò non è possibile poiché per rimuoverla è necessario il testo cifrato che essa inizialmente aveva generato, ma è invece disponibile il testo cifrato con due cifrature (prima la prima e poi la seconda). In altre parole è molto importante l'ordine con cui le cifrature e le decifratore sono eseguite; non è possibile cifrare due volte e poi rimuovere la prima cifratura senza aver prima rimosso la seconda, cioè l'ultima che è stata applicata. Vale la regola che "l'ultimo arrivato esce per primo".
Precedentemente si è detto che una cifratura che non sia sensibile all'ordine con cui viene apposta ad un messaggio è rara poiché, infatti, ne esiste una: la cifratura binaria tramite l'operatore logico XOR. Una trattazione più dettagliata degli operatori logici binari è presente nelle appendici.

Diffie e Hellman si resero conto che il primo passo per un sistema che risolvesse il problema dello scambio delle chiavi era di trovare una *funzione matematica unidirezionale*, cioè una funzione matematica che non sia invertibile, o più realisticamente sia computazionalmente molto più difficile da calcolare invertita. Il calcolo di $b=f(a)$ deve essere molto più facile del calcolo di $a=f^{-1}(b)$. Se ad esempio la funzione $f(a)$ è così definita $f(a) = 5a$ la funzione inversa sarà $f^{-1}(b)=b/5$; in questo caso la funzione f richiede un numero di calcoli equivalenti a quelli richiesti dal suo inverso. Nell'esempio la funzione semplicemente moltiplica per cinque il valore che le viene fornito, è quindi semplice eseguire la funzione inversa e riottenere il valore originale: è sufficiente eseguire una divisione per 5. Fortunatamente esistono funzioni il cui inverso è enormemente più impegnativo da calcolare rispetto alla funzione stessa.

Sia il metodo di scambio delle chiavi Diffie-Hellman, sia altri sistemi trattati in seguito, sfruttano le proprietà dell'aritmetica in modulo N. L'aritmetica dei moduli prevede l'uso dei soli numeri minori di N, con N stabilito a piacere. Con queste regole, tutte le operazioni matematiche vengono svolte come nell'aritmetica comune, ma il risultato non può mai essere un numero superiore o uguale a N, quindi il risultato dell'operazione viene diviso per N e il resto è il vero risultato in modulo N. L'esempio mostra la moltiplicazione di due numeri in modulo 6: 3 e 5 vengono moltiplicati dando il risultato 15, ora quest'ultimo viene diviso per 6 e si ottiene un resto di 3 cioè il risultato dell'operazione in modulo 6.

```
3 x 5 = ? (mod 6)   3 x 5 = 15   15 = 2 x 6 + 3   3 x 5 = 3 (mod 6)
```

Il risultato delle operazioni in modulo N è sempre minore di N proprio perché il resto di una divisione non può mai essere maggiore o uguale al divisore. L'aritmetica dei moduli può anche essere compresa con l'immagine di un disco numerato come quello di un orologio, dove i numeri sono solo N e qualsiasi risultato rientra in questi valori. 11 + 4 in modulo 12 è uguale a 3, infatti se sono le 11:00 e passano quattro ore, la lancetta dell'orologio punta verso il numero 3. L'aritmetica dei moduli è spesso utilizzata in crittografia perché è molto ricca di funzioni unidirezionali.

Il metodo Diffie-Hellman si basa sulla presunta difficoltà del *logaritmo discreto* ed opera in questo modo:

- Alice e Bob scelgono di comune accordo due numeri: un numero primo grande p e un'altro numero d (che deve rispettare alcune restrizioni, tra le quali $d < p$).
- Alice sceglie un numero intero a minore di p e trasmette a Bob il valore $d^a \bmod p$.
- Bob sceglie un numero intero b minore di p e trasmette ad Alice il valore $d^b \bmod p$.
- Alice calcola $d^{ab} \bmod p = (d^b \bmod p)^a \bmod p$.
- Bob calcola $d^{ab} \bmod p = (d^a \bmod p)^b \bmod p$.
- La chiave che entrambi devono usare è d^{ab}, in quanto è uguale per entrambi e solo loro hanno questo risultato.

Eva potrebbe aver intercettato le comunicazioni di Alice e Bob, e quindi avere a disposizione $d^a \bmod p$ e $d^b \bmod p$ ma con questi valori non è in grado di calcolare in modo rapido $d^{ab} \bmod p$.

Diffie e Hellman furono quindi in grado di trovare un modo per aggirare il problema delle chiavi, ma il loro metodo presenta ancora una scomodità. Poiché la generazione della chiave è affidata sia ad Alice sia a Bob entrambi devono essere disponibili ogni volta che viene creata una chiave: ciò non sempre è possibile o conveniente. Fu così che Diffie non si accontentò ma continuò la ricerca di un metodo meno macchinoso di quello già trovato. E in effetti egli trovò una soluzione migliore, che prevedeva l'esistenza non di una sola chiave usata per cifrare e per decifrare, ma di due diverse chiavi: una per cifrare (*chiave pubblica*)

e una per decifrare (*chiave privata*). Questo sistema crittografico è chiamato *crittografia a chiave pubblica* o *chiave asimmetrica*.
La crittografia a chiave pubblica prevede che ogni persona disponga di due chiavi, una pubblica che renderà disponibile a ogni persona con cui voglia comunicare, e una privata che custodirà attentamente senza rivelarla a nessuno. Quando Alice vuole mandare un messaggio a Bob, ottiene la sua chiave pubblica, la usa per cifrare il messaggio con un algoritmo di cifratura, quindi invia il messaggio a Bob. Solo Bob sarà in grado di decifrare il messaggio poiché, per invertire la cifratura (che è stata realizzata con la chiave pubblica di Bob) è necessario conoscere la sua chiave privata, che solo lui ha.

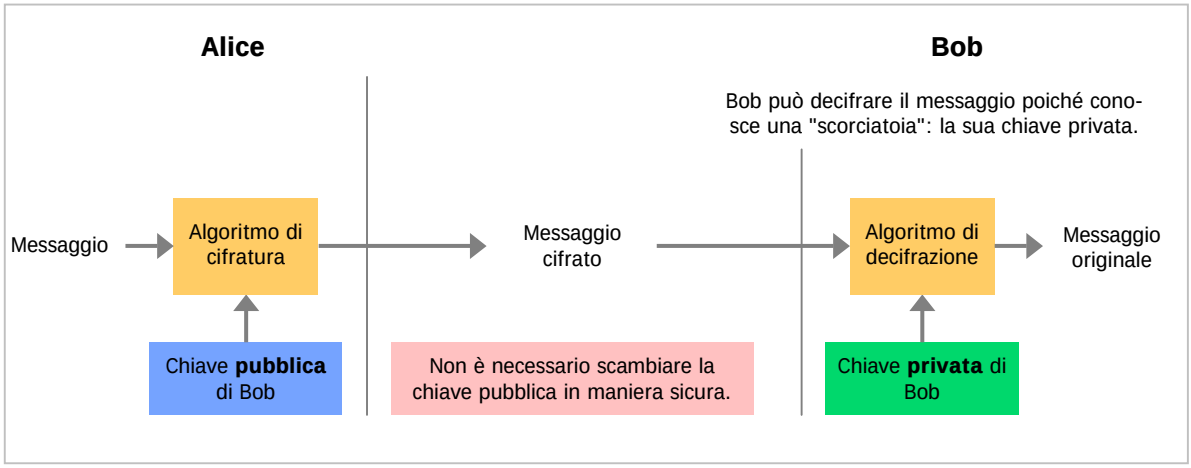


Figura 14 Schema della crittografia a chiave pubblica.

Diffie pensò che questo tipo di cifratura si potesse realizzare con una funzione unidirezionale tale che fosse possibile invertirla solo in circostanze particolari. Tramite la chiave pubblica doveva essere possibile cifrare un messaggio, ma una volta cifrato, questo non poteva più essere decifrato se non conoscendo qualche informazione particolare che rendesse la funzione facilmente invertibile. Tale informazione è esattamente la chiave privata, che è disponibile solo al destinatario del messaggio. Diffie presentò le sue idee nel 1975, ma non riuscì a trovare una funzione che potesse soddisfare le sue necessità.
Furono invece tre ricercatori del MIT (Massachusetts Institute of Technology) a scoprire una funzione che consentiva un uso reale della crittografia a chiave pubblica: Leonard Adleman, Ron Rivest e Adi Shamir. Dalle loro iniziali prende il nome il metodo crittografico a chiave pubblica basato sulla loro funzione unidirezionale, tale sistema è chiamato RSA. La funzione unidirezionale su cui si basa la crittografia RSA sfrutta alcune caratteristiche dei numeri primi e della fattorizzazione di numeri interi. Il procedimento completo usato nella cifratura RSA è esposto più avanti, ora verrà illustrato in maniera semplificata focalizzando l'attenzione sulle implicazioni con i numeri primi e la fattorizzazione.
Bob, affinché Alice possa inviargli dei messaggi, deve provvedere a creare una chiave pubblica, una privata e fornire quella pubblica ad Alice. Il primo passo consiste nello scegliere due numeri primi p e q; un numero è primo se è divisibile unicamente per se stesso e per uno. Ad esempio 5 o 7 sono due numeri primi ma 6 non lo è poiché 6 = 3*2. Una volta determinati i due numeri primi, Bob li moltiplica tra loro ottenendo un terzo numero n che ovviamente sarà divisibile solo per i suoi due fattori primi (oltre ad essere divisibile per uno e per se stesso). A questo punto egli fa avere ad Alice il numero n, che è la sua chiave pubblica; Alice usa n come chiave della funzione unidirezionale cifrante e ottiene il messaggio cifrato da inviare a Bob. Quando Bob riceve il messaggio provvede ad invertire la funzione unidirezionale in maniera rapida poiché egli conosce una "scorciatoia": egli conosce infatti i due fattori primi della sua chiave pubblica, cioè di n, ed è in grado di invertire la funzione in maniera efficiente. Eva, che puntualmente ha intercettato il messaggio, si trova nella condizione di dover invertire la funzione unidirezionale conoscendo solamente

la chiave pubblica di Bob, conoscendo cioè solamente *n*: il prodotto dei due numeri primi *p* e *q*. Ma non esiste alcun metodo rapido per fattorizzare *n* ed ottenere i suoi due fattori primi, l'unico modo è quello di provare a dividere *n* fino a quando non si trova un numero per il quale *n* sia divisibile. In questo senso *n* è la chiave pubblica di Bob e *p* e *q* sono la sua chiave privata con la quale egli è in grado di invertire rapidamente la funzione unidirezionale. La crittografia RSA si basa sul fatto che attualmente non esistono algoritmi di fattorizzazione veloci, e quindi moltiplicare due numeri primi *p* e *q* è davvero molto più rapido che fattorizzare il loro prodotto *n*.
Affinché la fattorizzazione di *n* richieda un tempo sufficientemente grande da garantire la sicurezza della cifratura con RSA, i due numeri primi che lo formano devono essere lunghi centinaia di cifre e devono rispettare alcune regole per evitare di ricadere in alcuni casi particolari per cui esistono algoritmi di fattorizzazioione efficienti. Attualmente la lunghezza minima di *n* ritenuta sicura è di 1024bit che espresso in cifre decimali corrisponde a un numero di 308 cifre. Per raggiungere una maggiore sicurezza si possono usare valori anche di 2048bit o 3096bit; in ogni caso il sistema RSA non impone alcun limite alla lunghezza di *n*.
E' stimato che un numero *n* della lunghezza di 1024bit richiederebbe, con un milione di computer operanti contemporaneamente, un tempo di dieci miliardi di anni: un tempo approssimativamente pari all'età dell'universo.

Per mostrare quanto la fattorizzazione di un numero sia più onerosa della moltiplicazione dei suoi due fattori primi può essere utile verificarlo con una prova. Si prendano per esempio due numeri primi 65413 e 65423, il loro prodotto è 4279514699 e può essere facilmente calcolato poiché richiede una sola operazione (una moltiplicazione). Eseguire l'operazione inversa, cioè fattorizzare 4279514699 richiede però molti più calcoli, approssimativamente richiede radice quadrata di *n* operazioni (dove *n* è il numero da fattorizzare). Tramite il programma NPrime è possibile verificare che effettivamente per fattorizzare 4279514699 è richiesto un grande numero di operazioni (divisioni), esattamente NPrime esegue 32706 divisioni, l'esatto numero può dipendere da come viene eseguita la fattorizzazione o dall'algoritmo utilizzato, ma in generale questo valore non si distacca molto dalla radice quadrata del numero fattorizzato. Maggiori dettagli sul programma NPrime si possono trovare nelle appendici.

Sono ora descritti i vari passaggi della cifratura RSA:

- Bob sceglie due numeri primi *p* e *q* sufficientemente grandi e calcola il loro prodotto $n = p \times q$.
- Calcola $\varphi(n) = (p-1)(q-1)$.
- Sceglie un numero intero *e* tale che $1 < e < \varphi(n)$ e che *e* e $\varphi(n)$ siano primi tra loro.
- Calcola d tale che $d \times e = 1 \pmod{\varphi(n)}$.
- La *chiave pubblica* di Bob sono i due numeri *n* ed *e*.
- La *chiave privata* di Bob sono i numeri *p*, *q* e *d*.

Quando Alice vuole mandare un messaggio a Bob usa la funzione unidirezionale basata sulla sua chiave pubblica. Questa funzione è: $f_B(x) = x^e \bmod n$. Ella non ha però modo di invertire tale funzione, solo Bob può farlo, quindi solo Bob può decifrare il messaggio. Bob è in grado di invertire rapidamente la funzione poiché conosce la sua chiave privata; la funzione che ottiene è: $f_B^{-1}(y) = y^d \bmod n$.

La cifratura con RSA ha un piccolo inconveniente: è computazionalmente molto più dispendiosa di altri algoritmi di cifratura simmetrici. I computer moderni possono tranquillamente affrontare la mole di calcoli richiesti, ma spesso, per risparmiare calcoli, viene adottato uno stratagemma con cifrature miste che permette di ridurre i tempi necessari alla

cifratura. Poiché il tempo impiegato da RSA per cifrare cresce con la lunghezza del messaggio, l'obbiettivo è quello di cifrare il meno possibile; la soluzione consiste nel cifrare per mezzo di RSA solamente la chiave che verrà utilizzata in seguito per cifrare con una cifratura standard simmetrica (ad esempio DES) i messaggi. La cifratura RSA è lenta ma non presenta problemi di distribuzioni delle chiavi, quindi viene usata per cifrare una chiave e comunicarla in maniera sicura, una volta che la chiave è stata distribuita, questa viene usata per cifrare con un algoritmo tradizionale che è molto più rapido di uno a chiave asimmetrica.

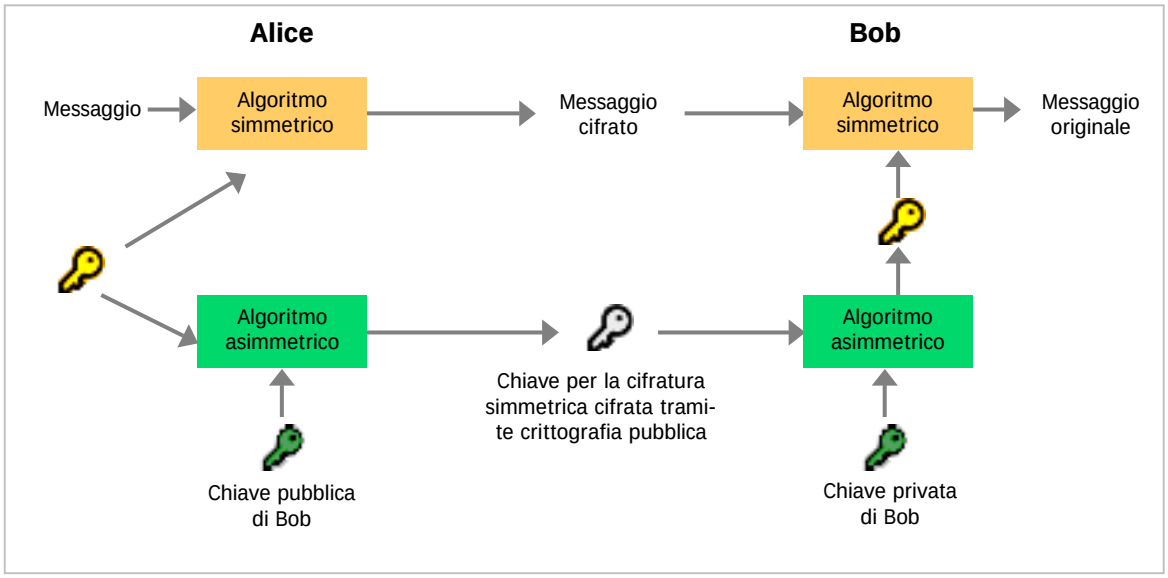


Figura 15 Schema della cifratura combinata simmetrica-asimmetrica

Utilizzando sia la crittografia asimmetrica sia quella simmetrica combinate assieme, come descritto, è possibile ottenere livelli di sicurezza ed efficienza mai realizzati in precedenza. E' possibile inserire circuiti che usano questa crittografia nelle smart-cart (quelle schede con un piccolo processore inserito al loro interno), comunicare i numeri di carta di credito o di bancomat tramite Internet senza che questi possano essere rubati da malintenzionati oppure inviare e-mail senza che esse possano essere decifrate da eventuali intrusi. Molti nuovi servizi basati su Internet, come l'internet-banking o il commercio elettronico, sono possibili solo grazie alla diffusione su grande scala delle tecnologie crittografiche sicure basate sull'accoppiata cifrature asimmetriche-simmetriche. In altre parole, la crittografia moderna si è diffusa enormemente e rende possibili attività ed operazioni che altrimenti, per questioni di sicurezza, non sarebbero in altro modo possibili; fino a non molto fa la crittografia era destinata alle comunicazioni diplomatiche o militari, adesso invece chiunque può sfruttare metodi crittografici in grado di resistere probabilmente anche al tentativo di violazione di un intero stato.

Crittografia quantistica

Come si è già accennato, gli algoritmi crittografici a chiave pubblica si basano sull'esistenza di una funzione unidirezionale e una funzione di questo tipo è per definizione tale se il suo inverso richiede molti più calcoli della funzione diretta. Un esempio, su cui si basa la crittografica RSA, è la fattorizzazione di grossi numeri interi nei loro due fattori primi, tuttavia, sebbene non si conosca alcun algoritmo rapido per effettuare questa operazione, non c'è dimostrazione che esso non esista. Nulla vieta che un giorno effettivamente venga scoperto un modo per fattorizzare in maniera rapida; se questo sistema dovesse essere trovato, la cifratura RSA non sarebbe più sicura. Esistono altre cifrature a chiave pubblica che si basano su altre funzioni unidirezionali, ma anche per queste funzioni si applica il discorso che un giorno potrebbe esistere un metodo per invertirle in maniera rapida senza bisogno di "scorciatoie".

Un altro motivo che non consente di definire veramente sicure le attuali tecniche crittografiche a chiave pubblica, è la possibilità che venga creata una nuova generazione di macchine per calcolare: i computer quantistici.
I computer quantistici si baserebbero sulle proprietà quantistiche della materia che diventano evidenti a livello sub-atomico; sfruttando queste proprietà, un computer quantistico potrebbe svolgere compiti di difficoltà esponenziale in tempi polinomiali. In altre parole potrebbe fattorizzare grossi numeri (come quelli usati nella crittografia RSA) in tempi molto brevi, e, in generale rendere inefficaci tutti i sistemi crittografici fino ad ora trattati.
I computer quantistici sfrutterebbero il principio di indeterminazione di Heisemberg in base al quale misurare lo stato di un sistema quantistico lo altera e rende impossibile conoscere con esattezza lo stato del sistema prima della misurazione.
Un esempio può essere utile per chiarire il concetto: ogni osservazione che compiamo si basa generalmente sulla luce, in altre parole la luce colpisce un oggetto e rimbalza in tutte le direzioni fino a giungere anche ai nostri occhi. Poiché il fascio di luce non ha una quantità di moto sufficiente a spostare in maniera sensibile l'oggetto osservato, questo ci appare come era prima che il fascio di luce lo colpisse. Quando però l'oggetto osservato è molto più piccolo, ad esempio è una particella sub-atomica come un elettrone, il fascio di luce che lo colpisce, cioè i fotoni che lo urtano, sono in grado di deviarne sensibilmente il moto e quindi la nostra osservazione ha comportato un cambiamento dello stato della particella osservata, rendendoci impossibile il compito di stabilire il suo stato precedente all'osservazione.

In base alle leggi della meccanica quantistica, o fisica quantistica, un fotone, ad esempio, fino a quando non ha modo di venire in contatto con un osservatore, che non per forza deve essere considerato un essere vivente, si trova in una *sovrapposizione di stati* per cui, fino a quando non lo si osserva, non è possibile conoscere in che stato esso si trovi. I computer quantistici sfrutterebbero questa caratteristica per eseguire un numero arbitrario di operazioni contemporaneamente. Il computer quantistico imposterebbe lo stato iniziale di alcuni *qbit*, cioè l'equivalente quantistico dei bit dell'informatica comune, dotati della capacità, come i bit, di assumere due valori **0** e **1**, ma in grado anche di trovarsi in uno stato in cui non sono né **0** né **1**: una sovrapposizione di stati. Quando il fotone che immagazzina l'informazione del qbit si trova in una sovrapposizione di stati, si può immaginare che esso assuma entrambi i valori che può avere. Un computer quantistico potrebbe, usando più qbit, trovarsi in una sovrapposizione non di soli due stati, ma di un numero dato solo dalla quantità di qbit utilizzati. Ad esempio, con 8 qbit, potrebbe trovarsi in 256 (2^8) stati contemporaneamente, ed eseguire l'equivalente di 256 calcoli nel tempo necessario a uno solo. Utilizzando un numero sufficiente di qbit, un computer quantistico sarebbe quindi in grado di

violare una cifratura RSA in tempi enormemente minori di quelli necessari ad un normale computer.
Sebbene non si sappia se qualcuno sia già riuscito a creare un computer quantistico, e non si sappia neppure se un simile calcolatore possa essere realizzato, i crittografi hanno già escogitato un metodo crittografico, basato sulle leggi della fisica quantistica, in grado di resistere anche ad un attacco con computer quantistici. Tale sistema è definito *crittografia quantistica*.
La crittografia quantistica sfrutta il principio di indeterminazione di Heisemberg per realizzare un metodo sicuro di scambio delle chiavi. Questo sistema fu ideato da Charles Bennett nei laboratori dell'IBM negli anni ottanta.

La comunicazione avviene emettendo dei fotoni, che possono essere polarizzati in due diversi modi: in rettilineo o in diagonale. Le due diverse polarizzazioni rappresentano rispettivamente uno **0** o un **1**. Alice provvede ad inviare a Bob una serie casuale di bit (fotoni polarizzati); Bob decide casualmente per ogni bit ricevuto quale dei due filtri polarizzatori utilizzare per misurarne il valore. Per il principio di Heisemberg, Bob può misurare i fotoni solo in uno dei due modi possibili; così pure Eva, che ha intercettato la comunicazione, deve ogni volta scegliere quale filtro usare: non può utilizzarli entrambi. Unicamente i fotoni che Bob ha casualmente misurato usando lo stesso filtro usato da Alice per inviarli risulteranno misurati correttamente; a questo punto Bob comunica, anche con un canale non sicuro, ad Alice quale filtro polarizzatore ha usato per ogni singolo fotone (ma non comunica il risultato delle misurazioni); Alice avvisa Bob di escludere tutte le misurazioni che egli ha eseguito con un filtro diverso da quello usato da Alice. Le misurazioni rimanenti sono utilizzate come chiave per una cifratura tradizionale. Con questo sistema, non solo Eva non può in alcun modo ottenere la chiave spiando la comunicazione tra Alice e Bob, ma rischia anche di modificare la polarizzazione dei fotoni mentre li misura, rendendo quindi possibile ad Alice e a Bob sapere che lei sta spiando la loro comunicazione.

Sebbene l'attuale sviluppo di questa tecnologia non consenta ancora di effettuare scambi di chiavi ad una distanza superiore a circa 150 chilometri, questo sistema funziona ed inizia già ad essere commercializzato da alcune aziende specializzate. Non resta che, come insegnano gli eventi passati della storia della crittografia, aspettare e vedere quali stupefacenti novità ha in serbo il futuro in fatto di crittografia.

Appendice A – Aritmetica binaria e operatori logici

Buona parte dei moderni calcolatori elettronici non utilizzano il sistema decimale ma il sistema binario. Nel sistema decimale ogni cifra può assumere dieci valori, da zero a nove, che in base alla loro posizione hanno un valore diverso; in questo caso si dice che il sistema numerico è *posizionale*. Prendendo come esempio il numero 538.47, esso equivale a:

$$5 \cdot 10^2 + 3 \cdot 10^1 + 8 \cdot 10^0 + 4 \cdot 10^{-1} + 7 \cdot 10^{-2}$$

oppure

$$500 + 30 + 8 + 0.4 + 0.07$$

Ogni cifra a sinistra della virgola (o del punto) rappresenta la parte intera del numero e va moltiplicata per una potenza di dieci corrispondente alla posizione che assume la cifra; ogni cifra a destra della virgola va moltiplicata per una potenza negativa di dieci corrispondente di nuovo alla posizione della cifra.
Il sistema binario funziona in maniera simile, le cifre disponibili sono però solo due (lo zero e l'uno) e ogni cifra va moltiplicata per una potenza di due e non di dieci. Il numero binario 101011 può essere facilmente convertito in decimale esprimendolo come esemplificato precedentemente per i numeri decimali:

$$1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

che è equivalente a

$$32 + 8 + 2 + 1 = 43$$

Il numero binario 101011 corrisponde quindi, nel sistema decimale, al numero 43.
Un computer utilizza la numerazione binaria poiché si adatta bene alla struttura stessa di queste macchine che spesso, operando tramite corrente elettrica, assegnano al flusso elettrico il valore 1 e alla sua assenza il valore 0. I calcoli eseguiti spesso si basano sull'uso degli operatori logici. Gli operatori logici principali sono: AND, OR, NOT e XOR. Questi operatori operano sui valori logici *vero* e *falso*, che nei calcolatori elettronici corrispondono rispettivamente a 1 e a 0 (in realtà talvolta 0 corrisponde a falso e qualsiasi altro valore a vero).
L'operatore logico AND accetta due valori e ne restituisce uno:

```
0 and 0 = 0
0 and 1 = 0
1 and 0 = 0
1 and 1 = 1
```

L'operatore AND restituisce vero se il primo operatore *e* il secondo operatore sono vero, altrimenti restituisce falso.
L'operatore logico OR accetta due valori e ne restituisce uno:

```
0 or 0 = 0
0 or 1 = 1
1 or 0 = 1
1 or 1 = 1
```

L'operatore OR restituisce vero se il primo operatore *o* il secondo operatore sono vero, altrimenti restituisce falso.
L'operatore logico NOT accetta un valore e ne restituisce uno:

```
not 0 = 1
not 1 = 0
```

L'operatore NOT restituisce vero se il suo argomento *non* è vero (cioè è falso) e restituisce falso se il suo argomento *non* è falso (cioè è vero). Questo operatore restituisce quindi l'opposto del proprio argomento.
L'operatore logico XOR accetta due valori e ne restituisce uno:

```
0 xor 0 = 0
0 xor 1 = 1
1 xor 0 = 1
1 xor 1 = 0
```

L'operatore XOR restituisce vero se il primo argomento è diverso dal secondo, altrimenti restituisce falso. Effettivamente questo operatore può essere ricavato da una combinazione degli altri:

```
a xor b = (a or b) and not(a and b)
```

La seguente tabella riassume il comportamento dei quattro operatori logici.

a	**b**	**a and b**	**a or b**	**not a**	**a xor b**
0	0	0	0	1	0
0	1	0	1	1	1
1	0	0	1	0	1
1	1	1	1	0	0

Gli operatori logici possono essere usati anche su numeri binari e non solo su singoli valori logici, ad esempio è possibile eseguire un'operazione di AND tra due numeri binari.

```
101011 AND
110110
----------
100010
```

L'operazione viene completata eseguendo l'operazione logica AND tra le due prime cifre dei due numeri, le seconde cifre, e così via. Se uno dei due numeri ha un numero di cifre inferiore all'altro si assume che le cifre mancanti siano zero: il numero binario 11, a seconda delle necessità, può essere considerato come 011, 0011, 0001 e così via.
Le operazioni logiche su numeri binari espandono ampiamente le possibilità di uso degli operatori logici.
Un'ultima nota va riservata all'operatore logico XOR che presenta una particolare proprietà che lo rende particolarmente utile nella crittografia. Il seguente esempio mostra questa particolare proprietà:

```
a xor b = c  ==>  b xor c = a
```

In altre parole se si esegue l'operazione XOR tra due valori *a* e *b* si ottiene il risultato *c*, quindi eseguendo nuovamente l'operazione XOR tra il risultato *c* e uno dei due operatori iniziali (*a* o *b*) si ottiene l'altro operatore iniziale. L'inverso dell'operazione XOR è quindi la stessa operazione XOR. Come già accennato questa caratteristica è molto utile per alcune applicazioni crittografiche.

Appendice B – Il programma Nprime

Il programma NPrime è stato scritto dall'autore della tesina e serve a verificare se un numero è primo o ad individuare tutti i numeri primi presenti in un dato intervallo. Il programma è scritto utilizzando il linguaggio di programmazione C, ed è progettato per funzionare con il sistema operativo Windows. NPrime non è stato scritto apposta per la tesina, ma era già stato scritto circa un anno prima, di conseguenza contiene alcune funzionalità non strettamente legate con gli argomenti trattati nella tesina (ad esempio la possibilità di trovare i numeri primi in un dato intervallo).
In questa appendice viene riportato il codice sorgente del nucleo del programma, la parte che effettivamente esegue il test di primalità; questo codice dovrebbe essere sufficientemente indipendente dal sistema operativo.
Il codice completo di tutto il programma è allegato alla tesina.

```
/*cp-----------------------------------------------------------------------*

  NPrime 1.3 - A tool to test and list prime numbers.
  Copyright (C) 2004 Alberto Pelenc

*-----------------------------------------------------------------------*/

// Include le dichiarazioni delle funzioni matematiche della libreria C.
#include <math.h>


// Dichiarazione delle funzioni.
unsigned long is_prime( unsigned long p, unsigned long *fact);
unsigned long is_prime2( unsigned long p, unsigned long *plist);


/*fd-----------------------------------------------------------------------*

Controlla se il numero p è primo, in tal caso restituisce 1, altrimenti 0.
La funzione restituisce inoltre il più piccolo fattore primo trovato
attraverso il parametro *fact.
Il controllo viene eseguito dividendo il numero p per tutti i numeri dispari
minori della sua radice quadrata, se il resto della divisione è zero allora il
numero p è divisibile, e quindi non è primo.

*-----------------------------------------------------------------------*/
unsigned long is_prime( unsigned long p, unsigned long *fact)
{
	unsigned long i;
	const unsigned long sq_rt = (unsigned long)sqrt( p);

	for (i = 2; i <= sq_rt; i = (i + 1) | 1)
		if (p % i == 0)
		{
			if( fact) *fact = i;
			return 0;
		}

	// Se il numero p è primo, allora *fact viene usato
	// per restituire il numero di divisioni eseguite.
	if( fact) *fact = i;
	return 1;
}


/*fd-----------------------------------------------------------------------*

Controlla se il numero p è primo, in tal caso restituisce 1, altrimenti 0.
Il controllo viene eseguito dividendo il numero p per tutti i numeri primi
minori della sua radice quadrata, se il resto della divisione è zero allora il
numero p è divisibile, e quindi non è primo. La lista dei numeri primi minori
di p deve essere fornita tramite il parametro *plist.

*-----------------------------------------------------------------------*/
unsigned long is_prime2( unsigned long p, unsigned long *plist)
{
	const unsigned long sq_rt = (unsigned long)sqrt( p);
```

```
        while( *plist <= sq_rt)
        {
                if (p % *plist == 0)
                {
                        // Non è primo.
                        return 0;
                }
                plist++;
        }

        return 1;
}
```

Bibliografia

G. Stix, "I segreti meglio custoditi", *Le Scienze*, 438, pagine 92-97, 2005.

A. Languasco, A. Perelli, "Numeri primi e crittografia", *Matematica e Cultura 2000*, pagine 227-233, 2000.

S. Singh, *Codici & Segreti*, Rizzoli, Milano, 1999.

H. Schildt, *La Guida Completa C++, Quarta edizione*, McGraw-Hill, 2003.

A. Languasco, A. Zaccagnini, *Introduzione alla Crittografia*, Hoepli, 2004.

Siti Internet:

http://en.wikipedia.org

www.turing.org.uk/turing

www.codesandciphers.org.uk

http://www.liceofoscarini.it/studenti/crittografia

Indice