Crittografia: aspetti storici e matematici.

di Marco Triverio

INDICE

INTRODUZIONE ALLA CRITTOGRAFIA

Che cosa è la crittografia

La **crittografia** è l'arte e la scienza di mantenere riservata una certa informazione.

Supponiamo che due persone, che chiameremo Alice e Bob, vogliano scambiare un messaggio che deve rimanere riservato a loro due soltanto: in prima battuta potrebbero cercare un canale di comunicazione solo a loro accessibile (ovvero un canale "sicuro"), come il telefono oppure internet; ma presto si accorgerebbero di non avere la certezza assoluta che nessuno "ascolti" la loro comunicazione: infatti le informazioni trasmesse via telefono o via internet possono essere facilmente intercettate e registrate all'insaputa di mittente e ricevente.

Potrebbero allora decidere di incontrarsi in un vicolo buio a notte tarda, ma anche in questo caso qualcuno potrebbe ascoltarli, magari attraverso una "cimice" nascosta: come possono fare, allora, Alice e Bob per scambiarsi un'informazione attraverso un canale insicuro (telefono, internet o vicolo buio) con la certezza che essa non venga carpita da terzi? Possono "crittografare" (o "cifrare") l'informazione, ovvero renderla illeggibile ad altri se non a loro due, gli unici che conoscono la modalità con cui questa è stata resa illeggibile (e quindi sanno anche come renderla nuovamente leggibile): chiunque venga in possesso dell'informazione "crittografata" non può dunque capirla, a meno che non riesca a risalire alla modalità per renderla nuovamente leggibile.

In quanto tale, la crittografia ha giocato un ruolo molto importante nella storia, diventando un'arma indispensabile: da Giulio Cesare, al quale garantiva comunicazioni sicure, fino ad oggi, in cui la crittografia ha lo scopo principale di rendere sicure le trasmissioni telematiche.

Il termine "crittografia" deriva dal greco *kripto* e *graphos* che letteralmente significa "scrittura segreta".

Se la crittografia si propone di trovare metodi sempre più efficaci e sicuri per mantenere un'informazione riservata, la crittoanalisi cerca i punti deboli del lavoro dei crittografi e li sfrutta per riuscire a scavalcare la cifratura (e ottenere l'informazione scambiata, anche se non si è né mittente né ricevente).

La crittografia, assieme alla *crittoanalisi*, forma la **crittologia**, una branca della matematica.

Le sue finalità

Anche se la crittografia in passato ha riguardato principalmente l'ambito militare, attualmente è alla base del mondo moderno: il mondo delle comunicazioni.

Mai come ora infatti si sente il bisogno di scambiare messaggi di qualsiasi natura (posta elettronica, transazioni finanziarie,...): e cresce anche il bisogno di mantenere parte dei messaggi scambiati segreti e integri.

Infatti la crittografia si propone di salvaguardare:

- la confidenzialità: il messaggio deve rimanere riservato, ovvero garantirne l'accesso soltanto a chi è autorizzato;
- l'integrità: il messaggio inviato non deve poter essere modificato o falsificato durante il percorso tra il mittente e il destinatario;
- l'autenticazione e il non-ripudio attraverso la *firma elettronica*: l'origine del messaggio deve poter essere garantita e non falsificata; il destinatario quindi deve poter essere certo che il mittente non ripudierà ciò che, apparentemente, ha scritto.

Perchè è così importante

Se fino a pochi decenni fa la crittografia era praticata soprattutto in ambito governativo e militare, ora è utilizzata massicciamente nelle comunicazioni telematiche: se in passato era utilizzata principalmente in guerra, ora è impiegata nella "guerra per la sicurezza" e per la difesa dai criminali informatici.

Tutte le volte che usiamo il *telefono cellulare* o apriamo con il *comando a distanza* la nostra automobile stiamo scambiando con l'esterno delle informazioni che sono soggette ad essere captate e sfruttate a nostro svantaggio. Se vogliamo evitare che questo accada dobbiamo pretendere che, anche se un potenziale terzo dovesse captare le informazioni da noi inviate, queste siano di fatto inutilizzabili: e questo può avvenire solo cifrando le informazioni; in questo modo esse saranno fruibili solo dal mittente e dal ricevente e questi saranno certi che non siano state "*forgiate*", ovvero modificate da terzi o inviate da qualcuno che si vuole spacciare per il reale mittente.

La crittografia insomma opera affinchè nessuno possa ascoltare le nostre telefonate e affinchè nessuno possa aprire la nostra automobile anche senza avere la chiave, ma anche affinchè nessuno possa utilizzare il Bancomat al posto nostro o possa accedere a dati riservati: la crittografia, quando ben utilizzata, elimina la possibilità di nuocere alla persona attraveso un utilizzo improprio della tecnologia.

Da chi difendersi

Il mondo sta evolvendo in senso tecnologico e, se da un lato le nuove tecnologie hanno ampliato enormemente lo spettro di possibilità, dall'altro hanno portato ad una informatizzazione della figura del criminale e del concetto di crimine: la tecnologia risulta un'arma a doppio taglio che può essere usata in diversi modi positivi ma che, in mano ad un malintenzionato, può risultare davvero pericolosa.

La tecnologia attuale ci permette di scambiare facilmente tantissime informazioni con altre persone, ma siamo davvero sicuri che queste informazioni giungeranno solo al destinatario? Cosa succederebbe se qualcun altro le carpisse? Potrebbe utilizzarle per farsi un'idea di chi noi siamo, cosa facciamo e quale tipo di vita conduciamo: la nostra privacy verrebbe così gravemente violata.

Ma non solo: se le informazioni scambiate sono molto sensibili (numeri di carte di credito, ad esempio) il danno può diventare, oltre che personale, anche economico o sociale. E' importante capire che, se non vogliamo rinunciare alle grandi possibiltà che il mondo delle comunicazioni ci offre, dobbiamo difendere le nostre informazioni e non permettere che diventino pubbliche: il primo passo è sicuramente l'adozione della crittografia.

Ad esempio, quando compriamo attraverso un sito internet è nostro dovere verificare che lo scambio di informazioni (tra cui anche il numero di carta di credito) sia crittografato (è sempre scritto nella pagina di acquisto) per impedire ad un eventuale terzo che "origli" la comunicazione di potere impossessarsi del nostro numero di carta di credito.

TERMINOLOGIA E CONCETTI FONDAMENTALI

Immaginiamo che due persone, chiamate convenzionalmente Alice e Bob, vogliano *comunicare in maniera sicura*: dovranno adottare un linguaggio o un codice conosciuto solo da loro in modo che, se qualcuno dovesse "origliare" la loro comunicazione, non possa capire cosa stiano dicendo.

Il processo attraverso il quale il *messaggio originale M* (detto *testo-in-chiaro* o plaintext) viene reso "incomprensibile" a terzi è detto "**cifratura**"; il processo inverso, che trasforma il *messaggio risultante C* (detto *testo cifrato* o ciphertext) nel messaggio originale M, è detto "**decifratura**".

La funzione matematica che permette il processo di cifratura e/o decifratura è detta **algoritmo crittografico** o cifrario. A livello matematico possiamo dunque descrivere la cifratura come una funzione E che agisce sul testo M restituendo C. Dunque:

$$E(M)=C$$

La funzione D invece agisce sul testo cifrato C e restituisce M. Dunque:

$$D(C)=M$$

Di conseguenza sarà di sicuro vero che:

$$D(E(M))=M$$

Gli algoritmi crittografici rappresentano soltanto le modalità "generiche" attraverso cui un messaggio M viene crittato in C.

La **chiave** è ciò che invece definisce le *modalità "specifiche"*. Mentre spesso l'algoritmo è pubblico (ovvero si conoscono le operazioni "generiche" che svolge per crittografare un'informazione) e analizzabile da tutti, la chiave è personale e *deve rimanere segreta*.

Gli algoritmi che utilizzano la stessa chiave per cifrare e decifrare il messaggio sono detti **algoritmi simmetrici**, per cui:

$$E_K(M)=C$$
$$D_K(C)=M$$

Dunque:

$$D_K(E_K(M))=C$$

Diversi algoritmi moderni (più avanti analizzeremo perchè) solitamente utilizzano una chiave diversa per la cifratura e la decifratura: questi sono detti **algoritmi asimmetrici**. Quindi:

$$E_{K1}(M)=C$$
$$D_{K2}(C)=M$$
$$\text{e: } D_{K2}(E_{K1}(M))=C$$

Prima dell'avvento dei computer, le tecniche crittografiche si basavano principalmente su due concetti elementari: la *trasposizione* e la *sostituzione*.

La prima consiste nel modificare all'interno del messaggio originale l'ordine delle lettere: il testo risultante è dunque un'anagramma dell'originale e la chiave è la "regola" in base alla quale gli elementi vengono trasposti.

La seconda tecnica invece va a modificare le lettere stesse, che vengono sostituite con altre. Anche in questo caso la chiave descrive la "regola" in base alla quale le lettere sono state sostituite ed è quindi ciò che bisogna conoscere per poter comprendere il testo cifrato.

Ciò che è cambiato con l'avvento dei computer è che gli algoritmi, invece di lavorare sulle lettere, lavorano su bit (che possono valere 1 oppure 0) oppure su gruppi di bit (ad esempio su "*blocchi*" di 32 o 64 bit).

La **sicurezza** di una comunicazione crittata è determinata da due fattori fondamentali:

- il *tempo T(n)* medio necessario per "rompere" l'algoritmo, ovvero per risalire all'informazione senza di fatto conoscere la chiave ;
- la *segretezza* della chiave.

Il tempo T(n) necessario per riuscire ad ottenere il testo-in-chiaro (ad esempio attraverso la tecnica del *brute-force*, che consiste nel provare tutte le chiavi possibili fino a trovare quella esatta) è sufficiente se è maggiore del tempo durante il quale l'informazione crittata deve rimanere segreta.

Il tempo T(n) è in funzione del *numero n di operazioni elementari* necessarie per rompere la cifratura: questo vuol dire che più i computer si evolvono e diventano veloci, più il tempo necessario diventa sempre minore. Dunque una cifratura che oggi è considerata sicura, tra qualche anno potrebbe essere facilmente superata: il tempo T quindi dipende anche dagli strumenti utilizzati per eseguire un brute-force e dalle risorse economiche del potenziale attaccante.

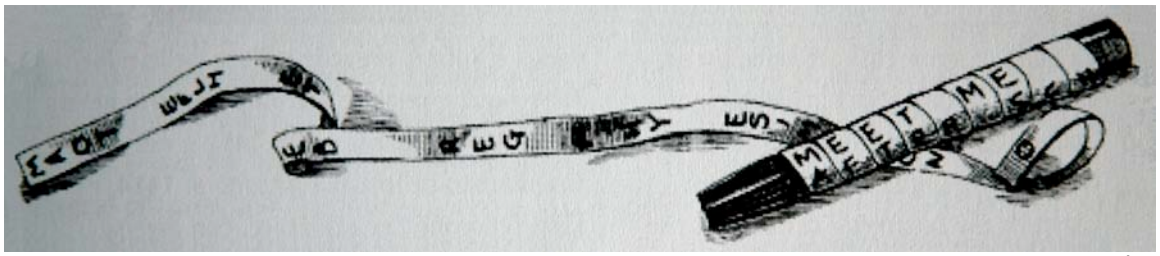
I Primi Metodi Crittografici e i Cifrari Monoalfabetici

La crittografia è stata coltivata come un'arte per diversi secoli ed è stata sempre segno dell' inventiva delle persone che, nei modi più ingegnosi, riuscivano a

mantenere segreta una certa informazione. Solo negli ultimi decenni si è sviluppata la necessità di una crittografia come *vera e propria scienza* che desse garanzie sull'inviolabilità delle informazioni crittate: la crittografia ha dunque iniziato a fondersi con la *matematica*, raggiungendo un livello di complicatezza e di astrattezza davvero impensabile.

Il primo esempio di crittografia risale all'antico Egitto (1900 a.C. circa), quando gli scribi utilizzarono geroglifici non "standard" nelle loro iscrizioni.
Un'altro esempio è rappresentato da alcune tavole mesopotamiche (risalenti al 1500 a.C.) che contengono delle informazioni cifrate.

Decisamente astuto è invece il metodo (descritto da *Plutarco nelle "Vite Parallele"*) attraverso cui il re spartano Agide comunicava con i generali: questi, infatti, avvolgeva a spirale una striscia di cuoio o di pergamena attorno ad un bastoncino cilindrico di legno (con un diametro ben specificato) detto "**scitale**" e poi vi scriveva longitudinalmente: una volta srotolata, la striscia di cuoio o di pergamena risultava un insieme di lettere senza senso e poteva essere riordinata solo se veniva avvolta attorno ad un bastoncino dello stesso diametro. Questo è un ottimo esempio di *trasposizione*.



Scitale

In passato, inoltre, la crittografia era spesso mescolata alla **steganografia**, ovvero l'arte e la scienza di nascondere messaggi dentro a messaggi apparentemente innocui. Una forma di steganografia piuttosto moderna e abbastanza efficace è l'*inchiostro invisibile.*

Attualmente la steganografia non può essere però considerata parte della crittografia in quanto basa la propria sicurezza sulla segretezza del metodo e non su una chiave e quindi non può assicurare trasmissioni sicure: non è dunque una scienza, ma, piuttosto, è rimasta un'arte.

Certamente il più importante esempio di crittografia antica è il **Cifrario di Giulio Cesare**, che racchiude in sè tutti i principi della crittografia moderna.

Il Cifrario di Cesare

Il Cifrario di Cesare risale all'epoca romana e ne abbiamo documentazione nel "*De Bello Gallico*" dello stesso Cesare e nella "*Vita del Divo Giulio*" di Svetonio.
Il funzionamento del Cifrario di Cesare è molto semplice: ciascuna lettera del testo-in-chiaro viene sostituita con quella che la segue N posizioni più avanti nell'alfabeto; il numero N costituisce dunque la *chiave segreta di cifratura*.
Per chiarezza:

- Algoritmo, ovvero modalità generiche con cui viene cifrato il testo: "sostituire ciascuna lettera del testo-in-chiaro con quella che la segue N posizioni più avanti nell'alfabeto";
- Chiave, ovvero modalità specifiche (da mantenere segrete) che definiscono il funzionamento dell'algoritmo stesso: "sostituire ciascuna lettera del testo-in-chiaro con quella che la segue 3 posizioni più avanti nell'alfabeto. La chiave segreta è dunque 3".

Cesare solitamente utilizzava il numero 3, quindi:

Alfabeto utilizzato nel testo-in-chiaro:																								
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
Ad ogni lettera dell'alfabeto utilizzato nel testo-in-chiaro viene fatta corrispondere la lettera che segue 3 posizioni più avanti e viene così creato l'alfabeto del testo cifrato:																								

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

Il cifrario di Cesare è detto **monoalfabetico** (oppure a sostituzione semplice) in quanto ad ogni carattere del testo-in-chiaro corrisponde un carattere del testo cifrato e perchè l'alfabeto usato per cifrare è una *permutazione molto semplice* dell'alfabeto normale.

Aspetti matematici del Cifrario di Cesare

L'aritmetica modulare

I Cifrario di Cesare è molto semplice da utilizzare ma può anche essere descritto dal punto di vista matematico, introducendo molti concetti che sono presenti nella crittografia moderna.
Il primo è **l'aritmetica modulare**.

Nel cifrario di Cesare, come nella maggior parte degli algoritmi moderni si fa uso dell'aritmetica modulo *n*, ovvero un'aritmetica che lavora con i numeri interi minori di *n*.
Anche se forse non ce ne siamo mai accorti facciamo quotidianamente uso dell'aritmetica modulare.
Gli orologi sono un esempio di come l'aritmetica modulare possa essere applicata alla realtà: prendendo ad esempio un orologio che segna l'ora nel formato 12 ore, ci rendiamo conto che non ha senso parlare delle ore 20.00 nè, tanto meno, delle ore 35.00 o delle ore 42.00. Ha piuttosto senso parlare delle 8.00, delle 11.00 e delle 6.00.

Ma come abbiamo fatto ad arrivare a questi risultati? *Abbiamo fatto uso dell'aritmetica modulare*, infatti abbiamo considerato l'ora (ad esempio le 35.00) abbiamo diviso per 12 e il resto della divisione ha rappresentato la nuova ora.

20 / 12 = 1 con resto 8
35 / 12 = 2 con resto 11
42 / 12 = 3 con resto 6

Scriveremo dunque che:

20 = 8 mod12
35 = 11 mod12
42 = 6 mod12

che sta ad indicare che dividendo 20 (oppure 35 o 42) per 12 otteniamo come resto 8 (oppure 11 o 6).

Un altro esempio di applicazione dell'aritmetica modulare sono i giorni della settimana, i quali implicano dei calcoli modulo 7.
Ammettendo che oggi sia Lunedì che giorno sarà tra 30 giorni? E tra 91?
Per rispondere alla prima domanda:
Considerando che al Lunedì sia associato il valore 1, al Martedì il valore 2, etc...:

Al giorno di oggi (Lunedì = 1) dovrò sommare il numero di giorni (= 30); otterrò 31 e, dividendolo per 7, il resto mi indicherà che giorno sarà.

Lunedì + 30 giorni =
= (1 + 30) mod 7 = 31 mod7

E sapendo che 31 / 7 = 4 con resto 3, posso dire che:

31 mod7 = 3 = Mercoledì

Quindi tra 30 giorni sarà un Mercoledì.

Per rispondere alla seconda domanda agiremo allo stesso modo:

$$\text{Lunedì} + 91 \text{ giorni} =$$
$$= (1 + 91) \bmod 7 = 92 \bmod 7$$

Sapendo che 92 / 7 = 13 con resto 1, posso dire che:

$$92 \bmod 7 = 1 = \text{Lunedì}$$

Tra 91 giorni sarà di nuovo un Lunedì.

Dunque potremo dire che, *dato un qualsiasi intero n > 1 esistono n resti possibili dati dalla divisione di un qualsiasi intero per n.*
L'insieme dei resti viene indicato con Z_n:

$$Z_n = \{ 0; 1; 2; \ldots n-1 \}$$
$$Z_7 = \{ 0; 1; 2; 3; 4; 5; 6; \}$$

Su questo insieme possiamo definire le usuali operazioni di somma, sottrazione, moltiplicazione e divisione.

Proprietà dell'aritmetica modulo n

- Dati tre numeri interi a, b, n (con $n \neq 0$) tali che:

  $$a = b \bmod n$$

  Possiamo dire che il resto dato dalla divisione di a/n e di b/n è lo stesso, per cui:

  $$b - a = s \quad \text{con s divisibile per n (ovvero } s|n).$$

- Come fatto notare in precedenza l'insieme dei resti comprende tutti i valori interi compresi tra 0 e n-1, per cui il numero degli elemeni contenuti in esso è uguale ad n.

- a è sempre minore o uguale di b;

- se $b < n$ allora $a = b$;

- se $b = n$ allora $a = n \bmod n = 0$;

- quindi se $b = n + 1$ allora $a = (n+1) \bmod n = 1$;

- E' sempre vero che: $b \bmod 1 = 0$;

- e che: $0 \bmod n = 0$;

- Il resto di una somma è equivalente alla somma dei resti:
  $$(a+b) \bmod n = a \bmod n + b \bmod n$$

- Il resto di un prodotto è pari al prodotto dei resti:
  $$(a*b) \bmod n = a \bmod n * b \bmod n$$

- Di conseguenza il resto di un quadrato equivale al quadrato del resto R:
  $$a^2 \bmod n = (a*a) \bmod n = a \bmod n * a \bmod n = R * R = R^2$$

L'aritmetica modulo n e il Cifrario di Cesare

E' dunque chiaro che quando impieghiamo il cifrario di Cesare utilizziamo, probabilmente senza accorgercene, l'aritmetica modulare.
Infatti associando ad ogni lettera dell'alfabeto convenzionale un numero:

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
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

L'alfabeto trasposto risulterà:

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
4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	1	2	3

Cesare usava come *chiave il numero 3* quindi in realtà non faceva altro che considerare ogni lettera come un numero e aggiungere a questo 3 e trattare il risultato in modulo 26.

Ad esempio:

$$A = 1$$
$$A + 3 = (1 + 3) \bmod 26 = 4$$
$$4 = D$$

"A" infatti veniva cifrata in "D".
Ancora un esempio:

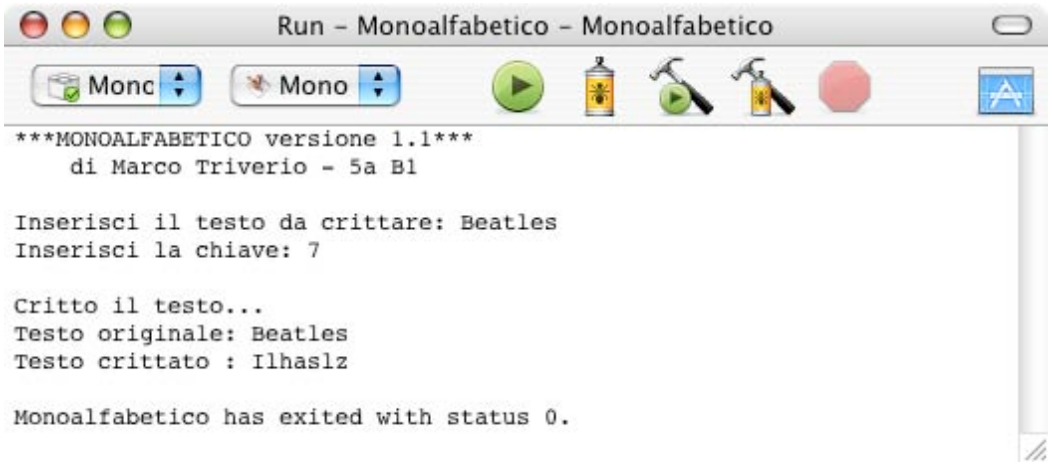
$$Y = 25$$
$$Y + 3 = (25+3) \bmod 26 = 28 \bmod 26 = 2$$
$$2 = B$$

Ed infatti "Y" veniva cifrata in "B".

Monoalfabetico: un programma in C++

Descrizione versione 1.1

Monoalfabetico è un programma scritto in C++ che critta un testo dato da tastiera secondo il cifrario di Cesare. La chiave può essere impostata dall'utente e può anche essere un numero negativo.



L'output di Monoalfabetico

Codice

```
// * Monoalfabetico *
// di Marco Triverio
// classe: Quinta B1

#include <iostream.h>

#define VERSIONE "1.1"
#define NUMCARATTERI 25

int main () {
    // dichiaro due array di caratteri
    char testo_in_chiaro[NUMCARATTERI+1];
    char testo_crittato[NUMCARATTERI+1];
    // dichiaro le variabili
    int chiave;
    int len;
    int buffer;

    cout << "***MONOALFABETICO versione " << VERSIONE
        << "***\n    di Marco Triverio - 5a B1\n\n";

    cout << "Inserisci il testo da crittare: ";
    cin >> testo_in_chiaro;
    len = strlen(testo_in_chiaro); //calcola lunghezza della stringa

    cout << "Inserisci la chiave: "; //l'utente può scegliere la chiave
    cin >> chiave;

    cout << "\nCritto il testo...\n";
    // ciclo FOR per il calcolo del testo cifrato
     for (int dat = 0; dat < len; dat++) {
          //caratteri minuscoli
          if (testo_in_chiaro[dat] > 95) {
               buffer = ((testo_in_chiaro[dat]+chiave)%96)%26;
               if (buffer == 0) { buffer = 26; }
               testo_crittato[dat]= 96 + buffer;
                                                }
          //caratteri maiuscoli
          else {
               buffer = ((testo_in_chiaro[dat]+chiave)%64)%26;
               if (buffer == 0) { buffer = 26; }
               testo_crittato[dat]= 64 + buffer;
            }
                                                }

    // "chiudo l'array"
    testo_crittato[len]='\0';
    // mostro i due array
    cout << "Testo originale: " << testo_in_chiaro << "\n"
         << "Testo crittato : " << testo_crittato << "\n";
    return 0;
}
```

Evoluzioni del Cifrario di Cesare

Il Cifrario di Cesare è davvero molto semplice ed è per questo facilmente violabile. Altri Cifrari simili furono usati dagli Ebrei ma questi risultavano addirittura più facili da violare in quanto solo l'algoritmo era segreto (*le chiavi possibili erano zero*).

Il libro di Geremia nella Bibbia fa uso di un semplicissimo codice per cifrare la parola "Babele"; la prima lettera dell'alfabeto ebraico (Aleph) viene cifrata con l'ultima

(Taw), la seconda (Beth) viene cifrata con la penultima (Shin) e così via; da queste quattro lettere è derivato il nome di "**Atbash**" (A con T, B con SH) per questo codice. Quindi, per l'alfabeto a 26 lettere:

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

Ad esempio la frase:

"Cifrario Atbash"
diventa: "Xruiziro Zgyzhs"

Un'altra evoluzione del Cifrario di Cesare è rappresentata dai **Cifrari Monoalfabetici a Parola-Chiave**, in cui la chiave è proprio una parola, ad esempio "segreto". La prima cosa da fare è eliminare le lettere doppie (se ce ne dovessero essere): in questo caso si dovrà eliminare una delle due "e" e la parola chiave diventerà "segrto". A questo punto costruiamo un alfabeto a partire da questa parola.

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

Come si può notare, non conoscendo la parola chiave, è difficile ricostruire l'alfabeto cifrante. In più è evidente che alcune lettere rimangano invariate.

L'evoluzione più moderna del Cifrario di Cesare è chiamata **ROT n** (in cui n indica la chiave, ovvero il numero di posizioni di cui bisogna spostarsi per cifrare/decifrare il messaggio) ed è semplicemente una versione informatizzata del Cifrario di Cesare: ROT 13 è presente nella maggior parte dei sistemi *Unix*.

Punti deboli del Cifrario di Cesare

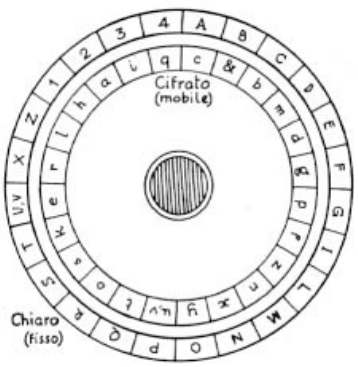
Come detto la semplicità del Cifrario di Cesare e delle sue evoluzioni è anche il loro punto debole. Attualmente un testo crittato con questi metodi viene risolto da un computer in meno di qualche frazione di secondo (e da una persona in qualche minuto): infatti ad ogni lettera dell'alfabeto corrisponde univocamente una sola lettera nell'alfabeto cifrante. Questo vuol dire che in un testo cifrato ad ogni (ad esempio) "U" corrisponderà sempre la stessa lettera. Si può ricorrere dunque all'*analisi statistica*: infatti ogni lettera (in ogni determinata lingua) ha una determinata frequenza caratteristica, per cui determinate lettere si presentano con più probabilità di altre. Per questo, analizzando la **frequenza** con cui una lettera nel testo cifrato si presenta e confrontandola con la *frequenza* media con cui si presentano le lettere in un testo italiano, si può risalire all'alfabeto cifrante e quindi ottenere il testo-in-chiaro.

La scoperta della fragilità dei cifrari monoalfabetici fatta dai crittoanalisti Arabi ed Europei portò a non considerare più questi sistemi di cifratura sicuri: venne stimolata la volontà di inventarne di nuovi e più complessi.

I Cifrari Polialfabetici e Le Macchine Cifranti

L'intuizione dell'Alberti: i Cifrari Polialfabetici

Leon Battista Alberti (Genova, 1404 – Roma, 1472) è architetto e teorico a cui non solo dobbiamo diversi splendidi edifici (quali il Tempio Malatestiano e Palazzo Rucellai) ma anche un importante scritto in materia crittografica: il "*De Componendis Cyfris*" (1466). In questo l'artista genovese teorizza il primo **disco cifrante**, pensato per l'utilizzo in Vaticano. Il disco presentava due ghiere concentriche di cui quella esterna è fissa mentre quella interna è mobile: la prima presentava tutte le lettere dell'alfabeto in ordine (a parte quelle a bassa frequenza, come H, J, K, Q, W e Y,



le quali avrebbero indebolito enormemente il cifrario); il secondo presentava le lettere in un ordine casuale.
Dopo aver scelto una lettera sul disco esterno (ovvero l'indice di cifratura) si può iniziare il processo di cifratura.
La differenza sostanziale rispetto ad una cifratura monoalfabetica è che l'Alberti proponeva di *cambiare via via l'indice di cifratura*. In questo modo l'analisi delle frequenze sarebbe stata falsata: era stato scoperto un nuovo metodo sicuro per cifrare i messaggi.

L'Alberti non riuscì però a trasformare questa geniale intuizione in una tecnica ben definita: per questo bisognerà aspettare il "*Traitè des Chiffres ou secrethes manières d'écrire"* (1585) di Blaise de Vigenere (1523-1596), il quale, partendo dalle teorie dell'Alberti, inventò una tecnica ben definita e potentissima. La **Cifratura di Vigenere** consiste nell'utilizzare non un solo ma più alfabeti cifranti (tra i 26 possibili) per crittare uno stesso testo, variando ad ogni lettera l'alfabeto da usare.
Ecco i 26 alfabeti possibili:

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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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

Se per esempio vogliamo crittare la frase "Distruggete la fortezza" con la chiave (7, 22, 2) otterremo la frase "Keuanwncgaa nh bqypggvc", infatti, associando ad ogni lettera un numero e lavorando con l'aritmetica modulare:

D	I	S	T	R	U	G	G	E	T	E	L	A	F	O	R	T	E	Z	Z	A	
4	9	19	20	18	21	7	7	5	20	5	12	1	6	15	18	20	5	26	26	1	
																					+
7	**22**	**2**	7	22	2	**7**	**22**	**2**	7	22	2	**7**	**22**	**2**	7	22	2	**7**	**22**	**2**	
																					=
11	31	21	27	40	23	14	29	7	27	27	14	8	28	17	25	42	7	33	48	3	mod26
																					=

11	5	21	1	14	23	14	3	7	1	1	14	8	2	17	25	16	7	7	22	3
K	E	U	A	N	W	N	C	G	A	A	N	H	B	Q	Y	P	G	G	V	C

Il numero di volte in cui viene cambiata la chiave è detto "**lunghezza della chiave**".
Perchè la chiave sia più facile da ricordare si potrebbe associare ad ogni numero n nella chiave la *n*esima lettera nell'alfabeto; per cui, invece di ricordare numericamente una chiave del tipo (15, 21, 19, 9, 3, 1), è più facile ricordare la parola "musica".
Per cui crittando la frase "Suonate la chitarra" con la chiave "musica":

S	U	O	N	A	T	E	L	A	C	H	I	T	A	R	R	A	
19	21	15	14	1	20	5	12	1	3	8	9	20	1	18	18	1	
																	+
M	U	S	I	C	A	M	U	S	I	C	A	M	U	S	I	C	
13	**21**	**19**	**9**	**3**	**1**	13	21	19	9	3	1	**13**	**21**	**19**	**9**	**3**	
																	=

32	42	34	23	4	21	18	33	20	12	11	10	33	22	37	27	4	mod26
																	=
6	16	8	23	4	21	18	7	20	12	11	10	7	22	11	1	4	
F	P	H	W	D	U	R	G	T	L	K	J	G	V	K	A	D	

La frase crittata sarà: "Fphwdur gt lkjgvkad".

I Punti Deboli dei Cifrari Polialfabetici

Nonostante quanto detto in precedenza i testi crittati con cifrari polialfabetici possono essere decrittati usando *l'analisi delle frequenze*.
Infatti l'unica vera difficoltà in più rispetto ai cifrari monoalfabetici è trovare la *lunghezza della chiave*: ciò è possibile applicando il teorema elaborato nel 1925 da William **Freedman**.

Fatto questo non resta che suddividere il testo in n colonne (ove n rappresenta la lunghezza della chiave) e studiare l'analisi delle frequenze sulle singole colonne: infatti ogni lettera di una stessa colonna è stata crittata allo stesso modo.

Con testi sufficientemente lunghi e chiavi abbastanza corte la difficoltà di crittoanalizzare un testo cifrato con più alfabeti è pressochè la stessa che si aveva con il cifrario di Cesare.

Uno schema di Crittografia perfetto: One-Time Pad

A questo punto ci si potrebbe domandare: esiste uno schema di crittografia immune dalle debolezze dei cifrari polialfabetici? La risposta è, almeno in teoria, si. Lo schema in questione si chiama **One-Time Pad** ed è stato teorizzato nel 1917 dal Maggiore Joseph Mauborgne e da Gilbert Vernam.
Questo tipo di schema prevede che *il numero N di caratteri da crittografare sia uguale alla lunghezza L della chiave*. In questo modo sono *impossibili* attacchi basati sull'analisi delle frequenze, infatti ogni carattere viene crittato in maniera differente.

Ma vediamo un esempio di come potrebbe relizzarsi una comunicazione basata sullo schema One-Time Pad.

- assumiamo che Alice voglia comunicare a Bob l'ingrediente segreto di una ricetta ("miele");
- Alice utilizza una chiave casuale (non deve essere in alcun modo prevedibile) o pseudo-casuale della stessa lunghezza del messaggio codificando così quest'ultimo. Un chiave pseudo-casuale potrebbe essere ad esempio "nharz";
- Il messaggio allora diventerà, se la chiave è "nharz":

M	I	E	L	E
13	9	5	12	5

					+
N	H	A	R	Z	
14	8	1	18	26	
					=

27	17	6	30	31	mod26
					=
1	17	6	4	5	
A	Q	F	D	E	

- Bob riceverà e decifrerà il messaggio con la chiave concordata con Alice;
- chi dovesse interecettare il messaggio non potrebbe in alcun modo capire che il testo in chiaro è "miele"; o meglio, *non potrebbe dire con certezza che questo non sia "latte", "panna" o "burro"* (tutte parole da 5 caratteri);

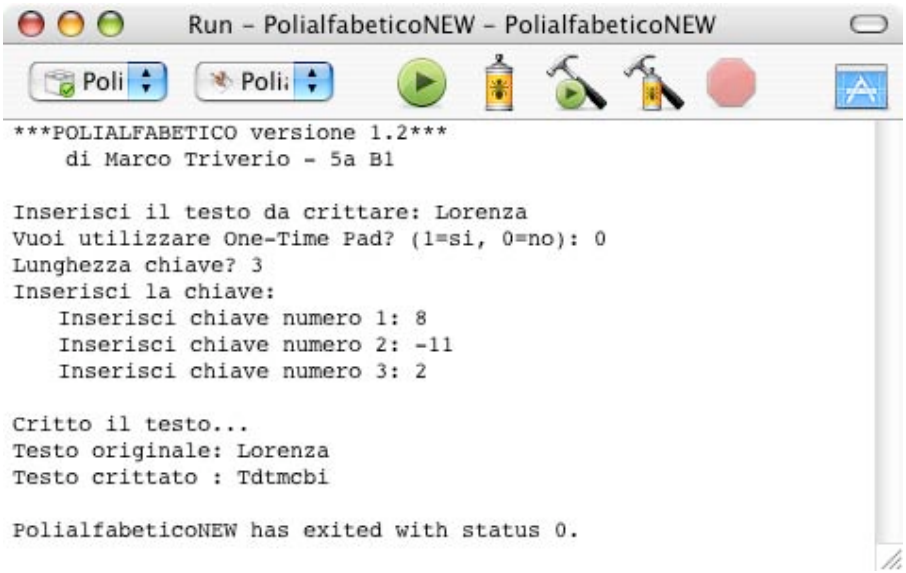
Purtroppo questo schema, che in teoria è perfetto, in pratica è *difficilmente realizzabile*.
Infatti ci sono alcuni problemi non risolvibili facilmente:

- prima di tutto, la chiave deve essere *generata in maniera davvero casuale*: il che è (anche se non sembrerebbe), per il momento, praticamente impossibile; infatti i generatori di numeri casuali (ad esempio) sono detti in realtà "pseudo-casuali" in quanto generano numeri con proprietà non del tutto casuali.
- in secondo luogo, *ogni messaggio dovrà essere crittografato con una nuova chiave* (e ogni vecchia chiave dovrà essere distrutta irrimediabilmente): infatti, nell'eventualità in cui qualcuno entri in possesso di UN SOLO testo-in-chiaro (oltre che del corrispondente testo cifrato) potrebbe facilmente risalire alla chiave e decifrare OGNI messaggio che è stato e che sarà cifrato con quella chiave;
- poi, una comunicazione piuttosto massiccia utilizzando one-time pad comporterebbe *chiavi di dimensioni spropositate*;
- inoltre, in che modo possono Alice e Bob scambiarsi la chiave in maniera sicura? Rimane dunque il rischio che la *segretezza della chiave possa essere violata*.

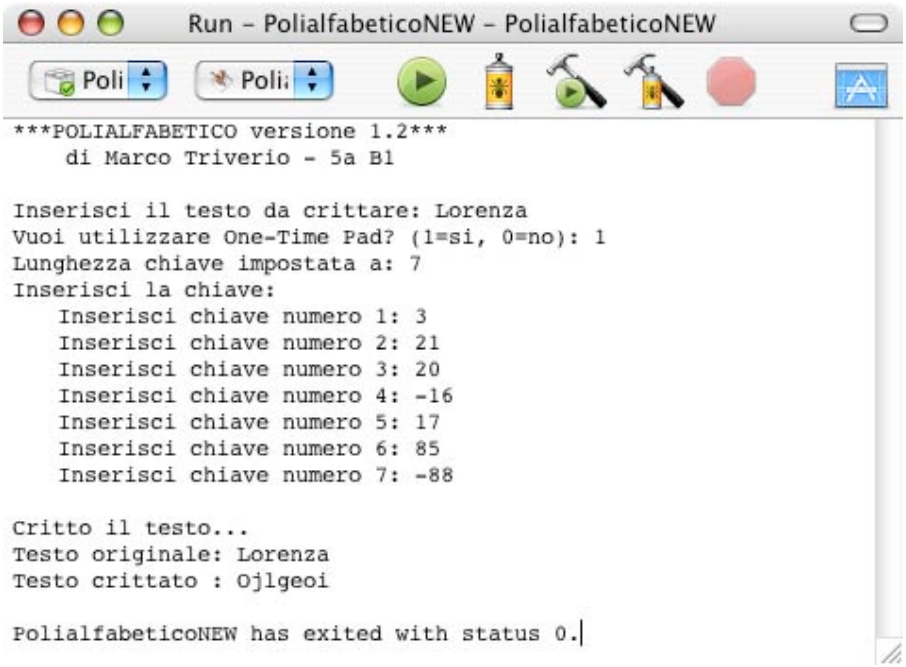
Polialfabetico: un programma in C++

Descrizione versione 1.2

Polialfabetico è un programma scritto in C++ che permette di crittare un testo secondo la tecnica di Vigenere, cioè cambiando alfabeto cifrante periodicamente. In particolare Polialfabetico permette di scegliere la lunghezza della chiave e specificarne ogni dettaglio; permette inoltre di utilizzare One-Time-Pad, ovvero il sistema in cui lunghezza del testo e lunghezza della chiave sono identici.



Funzionamento di Polialfabetico rinunciando a One-Time-Pad



Funzionamento di Polialfabetico avvalendosi di One-Time Pad

Codice

```
// * Polialfabetico *
// di Marco Triverio
// classe: Quinta B1

#include <iostream.h>

#define VERSIONE "1.2"
#define NUMCARATTERI 25

int main () {
    // dichiaro i due array per testo in chiaro e testo cifrato
    char testo_in_chiaro[NUMCARATTERI+1];
    char testo_crittato[NUMCARATTERI+1];
    // dichiaro variabili
    int numeri;
    int buffer;
```

```
    int len;
    int par;
     int one_time_pad;

    cout << "***POLIALFABETICO versione " << VERSIONE
        << "***\n    di Marco Triverio - 5a B1\n\n";

    INIZIO:
    cout << "Inserisci il testo da crittare: ";
    cin >> testo_in_chiaro;
    len = strlen(testo_in_chiaro); //calcolo lunghezza stringa
        // verifica array
        if (len > NUMCARATTERI) {
            cout << "\n\n     ***** ERRORE *****     "
                << "\nHai inserito troppi caratteri.\n"
                << "Il massimo è " << NUMCARATTERI << " caratteri\n\n";
            goto INIZIO;
                }
     //ONE TIME PAD
     ONETIMEPAD:
     cout << "Vuoi utilizzare One-Time Pad? (1=si, 0=no): ";
     cin >> one_time_pad;
     switch(one_time_pad) {
          case 0:
               cout << "Lunghezza chiave? ";
               cin >> numeri;
               break;
          case 1:
               numeri = strlen(testo_in_chiaro);
               cout << "Lunghezza chiave impostata a: " << numeri << "\n";
               break;
          default:
               cout << "\n\n Valore non valido \n";
               goto ONETIMEPAD;
                              }
     int chiavi[numeri];
    cout << "Inserisci la chiave:\n";
    // ciclo FOR per l'inserimento delle chiavi
    for (par = 0; par < numeri; par++) {
        cout << "\tInserisci chiave numero " << par+1 << ": ";
        cin >> chiavi[par];
    }
    cout << "\nCritto il testo...\n";
    par = 0;
    // ciclo FOR per crittare il testo
    for (int dat = 0; dat < len; dat++) {
            //caratteri minuscoli
            if (testo_in_chiaro[dat] > 95) { //ERA 95
                buffer = ((testo_in_chiaro[dat]+chiavi[par])%96)%26;
                    if (buffer == 0) { buffer = 26; }
                testo_crittato[dat]= 96 + buffer;
                                        }
            //caratteri maiuscoli
            else {
                buffer = ((testo_in_chiaro[dat]+chiavi[par])%64)%26;
                    if (buffer == 0) { buffer = 26; }
                    testo_crittato[dat]= 64 + buffer;
            }
        par=(par+1)%numeri;
    }
    testo_crittato[len]='\0';
    cout << "Testo originale: " << testo_in_chiaro << "\n"
         << "Testo crittato : " << testo_crittato << "\n";
    return 0;
}
```

LE MACCHINE CIFRANTI: UNO SGUARDO AD ENIGMA

Le Macchine Cifranti

All'inizio del 20° secolo stava nascendo la necessità di potere usufruire di una crittografia sicura ma anche e soprattutto veloce e facilmente utilizzabile: per questo nacquero le prime macchine cifranti.

La prima fu quella inventata nel 1891 dal comandante Bazèries, seguita a ruota dallo "*scotografo*" realizzata dal colonnello italiano Ducros; la più famosa di queste macchine è sicuramente **Enigma**, inventata nel 1918 dall'ingegnere elettrotecnico Arthur Scherbius e utilizzata dai tedeschi durante la Seconda Guerra Mondiale; *è inoltre interessante notare come queste macchine siano state principalmente inventate ed utilizzate per scopi militari*.

Il loro funzionamento si basa sull'utilizzo di cilindri (*rotori*): ognuno presenta una permutazione dell'alfabeto a 26 lettere. I rotori girano attorno ad un asse: questo permette che ogni lettera immessa venga cifrata con un alfabeto diverso; possiamo dunque dire che le macchine cifranti sono una versione meccanica del cifrario di Vigenère.

Una macchina cifrantre ad un rotore solo ripete il suo schema di crittografia ogni 26 lettere e si dice dunque che il suo periodo T sarà uguale a 26. Se però i rotori sono due il periodo diventerà 26*26 ovvero $26^2 = 676$ lettere prima che il testo venga cifrato con lo stesso schema. Possiamo dunque dire che la sicurezza aumenta esponenzialmente con l'aumentare dei rotori e il periodo T di una macchina a *n* rotori è dato da 26^n.

Enigma: descrizione del suo funzionamento

La versione originale di Enigma, la più famosa delle macchine cifranti, consisteva in diverse parti (si vedano le fotografie nelle pagine successive):

- una *tastiera*, da cui si inseriva il testo da cifrare o da decifrare;
- *un'unità scambiatrice*, in cui sono si trovano i *rotori*, grazie ai quali le lettere venivano cifrate;
- un *pannello a prese multiple*, per rendere ancora più sicuro il sistema di decifrazione;
- un *riflessore*, che permetteva di chiudere il circuito e di decifrare i messaggi inviati da altri tedeschi;
- un *visore* fornito di tante lampadine quanto sono le lettere; ogni volta che si premeva un tasto sulla tastiera (veniva quindi inserita una lettera del testo-in-chiaro) il segnale elettrico veniva elaborato dalla macchina e faceva accendere una lampadina sul visore (corrispondente alla lettera cifrata).

Ogni giorno la macchina Enigma doveva essere impostata in maniera uniforme in tutti i centri tedeschi per far sì che i messaggi potessero essere inviati e letti in pochissimo tempo. Per fare questo i ogni operatore di Enigma era in possesso di un "libretto dei codici" (*Code Book*) che stabiliva le impostazioni (non tutte però) giorno per giorno: Enigma richiedeva l'utilizzo di 3 rotori, ovvero di tre ruote su cui erano stampate le 26 lettere dell'alfabeto, disposte in un ordine diverso da quello alfabetico; i tedeschi (in generale, perché ad esempio la Marina utilizzava una versione di Enigma molto più complicata) potevano scegliere 3 rotori tra i 5 disponibili da inserire nei meccanismi di Enigma: le possibili combinazioni dei 5 rotori presi 3 alla volta erano in totale 60. Nel libretto dei codici veniva indicato quali rotori usare ed in quale ordine (ad esempio 1-3-2 oppure 5-1-3), ma non, ad esempio, la lettera da cui far partire il rotore.

I parametri non definiti nel Code Book erano modificati ogni mattina: questi avrebbero dovuto essere ogni volta diversi ma in realtà si ripetevano spesso e questo non faceva che aiutare il lavoro di decifrazione degli Alleati: doveva ad esempio essere scelta la lettera iniziale per ogni rotore che, per pigrizia, spesso coincideva con le prime tre lettere sulla tastiera (Q, W, E). Si dovevano anche definire le impostazioni del pannello a prese multiple: questo aveva il compito di scambiare delle lettere prima che queste fossero processate dai rotori.

Stabilite le impostazioni, l'operatore digitava il messaggio sulla tastiera: ogni lettera premuta veniva crittata attraverso i tre rotori e il circuito veniva chiuso grazie al riflettore; sul visore (una specie di tastiera con, al posto dei tasti, una serie di lettere illuminate) compariva la lettera crittata. Ogni volta che si premeva un tasto il primo rotore avanzava di una tacca: passavano dunque 26 digitazioni prima che questo tornasse alla posizione di partenza. Ma, accaduto questo, il secondo rotore scalava di una tacca e il primo ricominciava un nuovo giro: quando questo veniva completato il secondo rotore scalava di un'altra tacca e così via. I rotori in totale erano tre per cui, prima che tutti questi tornassero nelle rispettive posizioni di partenza erano necessarie $26*26*26 = 26^3 = 17576$ digitazioni.

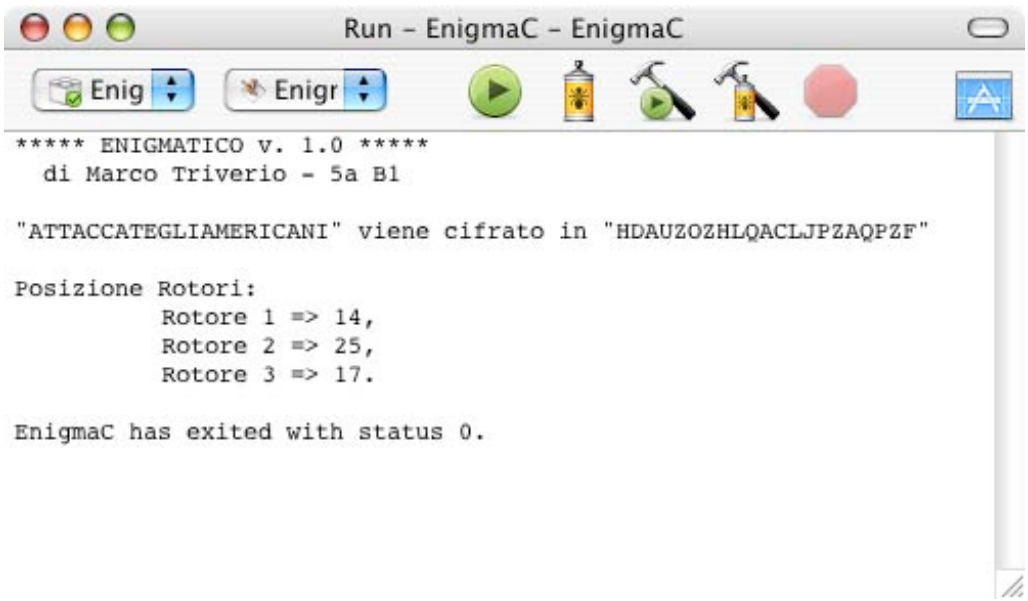
La versione di Enigma della Marina Tedesca disponeva di ben 4 rotori e questo aumentava ulteriormente la sicurezza: erano infatti necessarie $26^4 = 456976$ digitazioni prima che una lettera venisse crittata alla stessa maniera.

Tutto questo non faceva che avvicinare, almeno nella teoria, Enigma a un One-Time Pad; in realtà i diversi errori dei tedeschi (come si vedrà nel capitolo successivo) e la loro ingenuità permisero agli Alleati di arrivare a decifrare una piccola ma importantissima parte dei loro messaggi.

Enigmatico: un programma in C/C++

Descrizione versione 1.0

Enigmatico è un programma in C/C++ che simula in maniera piuttosto semplificata il funzionamento di Enigma a tre rotori. I tre rotori presentano permutazioni dell'alfabeto simili a quelle utilizzate nei rotori tedeschi.



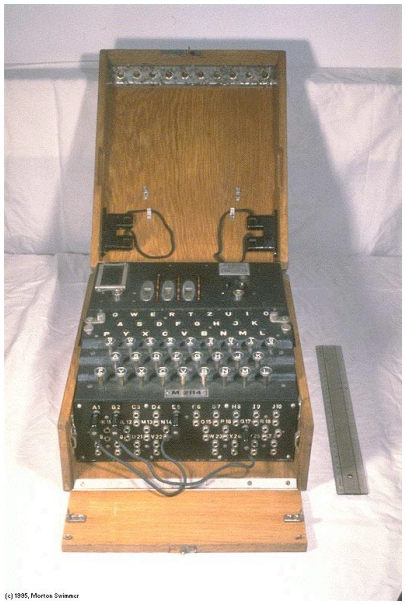
Schermata di Enigmatico

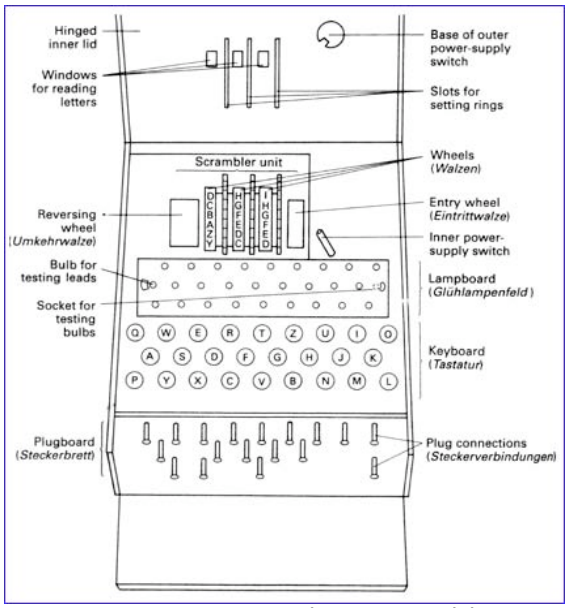
Codice

```
// * Enigmatico *
// di Marco Triverio
// Quinta B1
//
// grazie a Mike per l'aiuto

#include <stdio.h>
#include <iostream.h>
#define VERSIONE "1.0"

char Codifica1(char Input);
```

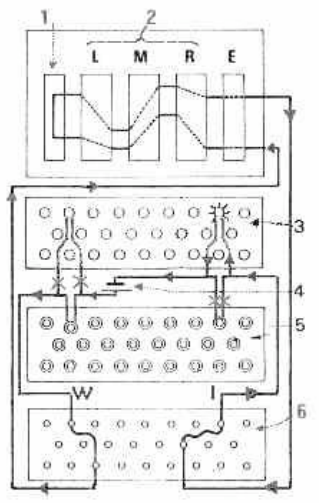
```
char Codifica2(char Input);
char Codifica3(char Input);
char LeggiDecodifica(char Input);
void EnigmaInc();

int i, j, k, n;

// Definisci i rotori
char Rotore_1[27] = "ZYXWVUTSRQPOMNLKJIHGFEDCBA";  //  Alfabeto al contrario
char Rotore_2[27] = "BADCFEHGJILKNMPORQTSVUXWZY";  //  Alfabeto scambiato
char Rotore_3[27] = "FGHIJABCDEPQRSTKLMNOXYZUVW";  //  Blocchi di 5


// Messaggio da decifrare
char Testo_In_Chiaro[22]    = "ATTACCATEGLIAMERICANI";
char Codifica[22] = "attaccategliamericani";

//  Definisci gli operatori per i rotori
char Codifica1(char Input)
{                                 //  Conversione attraverso "i"
  return Rotore_1[(i + Input - 'A') - ((i + Input - 'A') / 26 * 26 )];
}

char Codifica2(char Input)
{                                 // Conversione attraverso "j"
  return Rotore_2[(j + Input - 'A') - ((j + Input - 'A') / 26 * 26 )];
}

char Codifica3(char Input) {              //  Conversione attraverso "k"
  return Rotore_3[(k + Input - 'A') - ((k + Input - 'A') / 26 * 26 )];
}

char LeggiDecodifica(char Input) {      // Cerca di capire il carattere
char retvalue;
int  count;

  for (count = 0; Rotore_3[ count ] != Input; count++);
  if ((count = count - k) < 0) count += 26;       //  Trova una valida posizione per il
carattere
  retvalue = count + 'A';       //  Converti a ASCII
  for (count = 0; Rotore_2[ count ] != retvalue; count++);
  if ((count = count - j) < 0) count += 26;
  retvalue = count + 'A';       //  Converti
  for (count = 0; Rotore_1[ count ] != retvalue; count++);
  if ((count = count - i) < 0) count += 26;
  return count + 'A';
                                          }

void EnigmaInc() {
    if (++i > 25) {           //  Ora ruota i rotori
      i = 0;                    //  "i" fa un Rolling Over
      if (++j > 25) {
        j = 0;
        if (++k > 25)
          k = 0;
                        }
                      }
                    }

int main() {

int Rotore_1Start = 14;          //  Stabilisci la posizione iniziale dei rotori
int Rotore_2Start = 25;
int Rotore_3Start = 17;

i = Rotore_1Start;             // Stabilisci i parametri di controllo
```

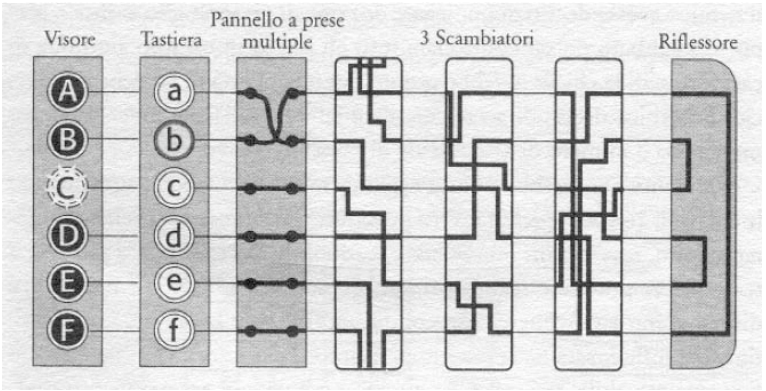
```
j = Rotore_2Start;
k = Rotore_3Start;

cout << "***** ENIGMATICO v. " << VERSIONE << " *****\n";
cout << "  di Marco Triverio - 5a B1\n\n";

for (n = 0; n < strlen(Testo_In_Chiaro); n++) {
    Codifica[n] = Codifica3(Codifica2(Codifica1(Testo_In_Chiaro[n])));
    EnigmaInc();               //  Trova il carattere crittato
                                                          }

printf("\"%s\" viene cifrato in \"%s\"\n", Testo_In_Chiaro, Codifica);

for (i = 0; i < 26; i++)    //  Ora, trova "i", "j" e "k"
   for (j = 0; j < 26; j++)
     for (k = 0; k < 26; k++) {
        Rotore_1Start = i;         //  Salva i valori iniziali per il Looping
        Rotore_2Start = j;
        Rotore_3Start = k;
        for (n = 0; (n < strlen(Testo_In_Chiaro)) && (Testo_In_Chiaro[ n ] ==
LeggiDecodifica(Codifica[ n ])); n++) {
               EnigmaInc();

        if (n == strlen(Testo_In_Chiaro))
          printf("\nPosizione Rotori:
               Rotore 1 => %i,
               Rotore 2 => %i,
               Rotore 3 => %i.\n", Rotore_1Start, Rotore_2Start, Rotore_3Start);
      i = Rotore_1Start;         // Ripristina i valori dei rotori
      j = Rotore_2Start;
      k = Rotore_3Start;
    }

}
```



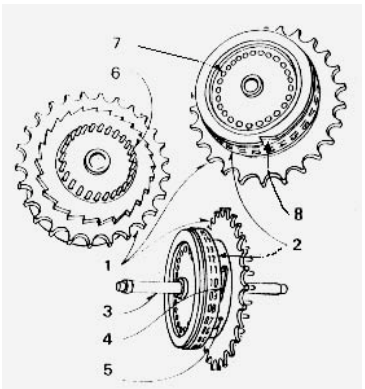


La macchina Enigma dal vero e schematizzata





Schema del funzionamento di Enigma



I rotori di Enigma

Enigma: la Guerra Segreta

1928-1939

La Seconda Guerra Mondiale non venne solamente combattuta con le armi, le bombe e i soldati ma anche con la crittografia, la matematica e i crittoanalisti: è questo un aspetto della guerra che è rimasto segreto e taciuto per molti anni; ma neanche oggi siamo a conoscenza di cosa esattamente avvenne e di quali e quanti uomini vennero impiegati per questa guerra lontana dai campi di battaglia. Ed è evidente che probabilmente mai lo sapremo in quanto non c'è volontà di raccontare da parte dei governi "protagonisti": si pensi che i luoghi dove vennero concentrati gli sforzi delle forze Alleate (alcuni Hut di Bletchley Park, in Inghilterra) sono stati rasi al suolo da pochi anni.

La macchina Enigma venne presentata ad alcune mostre commerciali nel 1923 ma, voluta dal governo tedesco per proteggere le proprie comunicazioni, venne presto ritirata dal commercio: le sue dimensioni ridotte, che ne garantivano la portabilità, e la sua facilità d'uso la resero lo strumento ideale per relizzare il progetto tedesco detto **Blitzkrieg** che mirava a rendere più efficiente e coordinata l'azione militare tedesca.

Enigma venne usata inizialmente dalla Marina Militare, successivamente venne modificata (e resa ulteriormente più sicura) e fornita a tutto l'esercito.

La guerra segreta iniziò molto prima della Seconda Guerra Mondiale: in particolare già nel 1928 i polacchi vennero a conoscenza di Enigma; i tedeschi, infatti, inviarono erroneamente un esemplare di questa macchina a Varsavia per posta ordinaria. Capito il loro sbaglio, fecero pressioni sulle autorità polacche per avere urgentemente il pacco indietro: ma queste, insospettite, decisero di esaminarlo attentamente prima di rispedirlo due giorni dopo.

Questo naturalmente diede piena conoscenza di Enigma (almeno in questa sua prima versione) ai polacchi: avrebbero dunque potuto studiarne i meccanismi, che però da soli non permettevano di decifrare i messaggi; permettevano invece di capire i meccanismi generali di funzionamento del metodo crittografico, ma non quelli specifici: erano dunque in possesso dell'algoritmo, ma non della chiave.

I polacchi però riuscirono ad ottenere anche il "libro dei codici" (*codebook*); in questo erano indicate le modalità specifiche di funzionamento di Enigma, ovvero come dovevano essere disposte le varie componenti a seconda dei giorni.

Infatti per questioni di sicurezza ogni giorno i tedeschi variavano il modo di disporre i rotori e i collegamenti della plug-board, la quale rendeva il lavoro dei crittoanalisti ancora più difficile. In particolare venivano modificati:

- i rotori; ce n'erano tre da disporre, in un ordine a propria scelta, nella macchina cifrante;
- di ogni rotore disposto bisognava scegliere la lettera da cui partire (ricordo che i rotori erano dei cilindri con 26 lettere ciascuno);
- qualsiasi lettera poteva essere scambiata con un'altra attraverso la plug-board.

Alcune di queste impostazioni erano definite attraverso il codebook, mentre altre (come la lettera da cui partire per ogni rotore) venivano definite per ogni messaggio (*indicator system*): teoricamente queste ultime impostazioni dovevano essere casuali, ma di fatto si ripetevano con costanza aiutando così i crittoanalisti a decifrare i messaggi.

I polacchi intercettarono e lessero i messaggi tedeschi dal 31 dicembre 1932.

L'indicator system venne cambiato nel 1938, ma i polacchi riuscirono nuovamente a leggere i messaggi tedeschi attraverso l'invenzione delle "bombe" (create da Rejewski e chiamate così per il ticchettio che producevano) ovvero macchine dai circuiti molto simili a quelli di Enigma in grado di analizzare e trovare le impostazioni mutevoli per ogni messaggio: le "bombe" possono essere considerate dei veri e propri computer degli anni Quaranta anche se non sostituivano assolutamente i crittoanalisti, i quali dovevano comunque programmare queste macchine e valutare criticamente i risultati che queste davano.

Nel Dicembre del 1938 tutti gli sforzi fatti fin qui dai polacchi vennero vanificati: Enigma venne resa molto più sicura grazie all'introduzione di due ulteriori rotori. Si usavano ancora tre rotori alla volta, che dovevano essere però scelti tra i cinque

disponibili: questo vuol dire che se prima le disposizioni possibili dei rotori erano 6 (1-2-3, 1-3-2, 2-1-3, 2-3-1, 3-1-2 e 3-2-1; infatti 3!=3*2*1=6) adesso diventavano sessanta. I polacchi non avevano più le risorse per decifrare i messaggi tedeschi.

Il 25 luglio 1939 l'intelligence polacca consegnò, in un incontro a Parigi, a francesci e inglesi i risultati del lavoro dei crittoanalisti polacchi: la descrizione dettagliata di Enigma (nella sua prima versione) e alcune sue repliche, della Bomba e tutte le informazioni che avevano raccolto in questi primi anni di "guerra segreta".

1939-1945

Inizia dunque così il lavoro a Bletchley Park che porterà gli alleati, grazie alle geniali intuizioni di Alan Turing e di molti altri matematici, a vincere prima la guerra segreta e poi la Seconda Guerra Mondiale: il materiale fornito dai polacchi fu fondamentale ma non sarebbe certamente bastato agli Alleati; c'era bisogno di nuove idee che permettessero di decifrare i messaggi tedeschi.

Il 3 settembre 1939 la Germania dichiarò guerra all'Inghilterra e proprio in quei giorni il centro per la crittoanalisi inglese (*British Government Code & Cypher School* ovvero **GCCS**; dopo la guerra venne rinominato GCHQ) venne spostato da Londra a Bletchley Park (circa a metà strada tra Oxford e Cambridge): il governo inglese fu fermo nella decisione di investire molto; naturalmente non si conosce la cifra stanziata, ma è ormai più che provato che le speranze di Churchill risiedessero in gran parte nell'operato di decodifica dei messaggi tedeschi. L'intenzione era quella di anticipare il nemico e di rendere vana la sua superiorità militare.

Si è calcolato che circa 10'000 tra matematici, linguisti, geni degli scacchi e anche fanatici di cruciverba vennero arruolati e fatti vivere nei 58 acri di Bletchley Park (sotto stretta sicurezza: nessuno di loro avrebbe mai dovuto rivelare al nemico l'esistenza di un tale apparato crittoanalistico). L'area era organizzata in diversi edifici minori (detti Hut) e il personale era organizzato gerarchicamente e in maniera efficiente: in questo modo tutti contribuivano costantemente a creare un quadro generale della situazione e le informazioni erano facilmente accessibili. Tra gli uomini più importanti e più geniali di Bletchley Park ricordo: John Tiltman, Dilly Knox, Hugh Foss, Harry Hinsley, Frank Birch e Alan Turing.

Tutti i messaggi intercettati venivano pazientemente registrati in attesa di essere analizzati: infatti più materiale si ha più è facile trovare la *chiave* con cui sono stati decifrati i messaggi di un determinato giorno (infatti la chiave cambiava quotidianamente, a volte anche da messaggio a messaggio). Negli ultimi mesi del 1939 il lavoro fu molto difficile (perché tutto manuale) e pochissimi messaggi riuscivano ad essere decifrati; nel 1940 Alan Turing ultimò la sua "Bomba" (detta *British Bombe* oppure *Turing Bombe*), molto più veloce ed efficiente di quella polacca: faceva infatti uso della teoria probabilistica per cui non tentava di decodificare il messaggio provando tutte le possibili combinazioni (il che avrebbe richiesto, supponendo che la macchina potesse eseguire un tentativo ogni secondo, circa qualche miliardo di anni) ma supponendo che il messaggio iniziasse in un determinato modo; l'intuizione fu geniale, perché le comunicazioni tedesche erano poco varie, soprattutto nei primi caratteri. Ad esempio ogni mattina gli Alleati intercettavano il bollettino meteo tedesco, la cui prima parola era quasi sempre "wetter" ("tempo" in tedesco); in altri casi il messaggio era semplicemente "Niente da comunicare": questo spesso permetteva di arrivare alla chiave e capire molte delle comunicazioni tedesche.

```
TYINP KPQOT YENSO IYOBO YRAKK SLSPP ZCDOA YSLPO
MXMNP PNXPT YCITT YQYBO ZRBIG MPLSE ZKCTX RCRQG
LEIKC RDMPP RTBNX WYWQG MDAYT GFVMF XEYSL LQNII
GIWRQ IGFEV NGDNN IOBDT MDPTT YXNKB UXEMW PPKPW
ST
```

I messaggi cifrati si presentavano in questa forma

Sempre nel 1940 accade un episodio simbolico per far capire l'importanza che aveva per gli Alleati poter decifrare i messaggi del nemico: vennero infatti intercettate

e decrittate le comunicazioni di preparazione di un bombardamento sulla cittadina di **Coventry**: era questa una mossa militare o solo un tentativo di verificare l'effettiva capacità inglese di decifrare i messaggi tedeschi? Il presidente inglese Churchill, consapevole che se non fosse stata evacuata la cittadina entro breve sarebbero morte migliaia di persone, decise di agire come se quei messaggi non fossero mai stati decifrati. Il 14 novembre 1940 Coventry venne devastata dai bombardamenti della Luftwaffe, 6000 persone circa morirono e ci fu un numero imprecisato di feriti.

Va anche detto che esiste una interpretazione alternativa del fatto, meno accreditata dagli studiosi: alcuni sostengono che il bombardamento fosse in realtà previsto su Londra e che un prigioniero tedesco (un pilota della Luftwaffe) disse che il vero obiettivo era Coventry; questi venne però ignorato e ciò costò 6000 vite.

Altre vite (questa volta di militari) furono certamente sacrificate per cercare di recuperare informazioni ulteriori su Enigma; è questo il caso (raccontato abbastanza fedelmente nel film U-571) di tutti coloro morti nel tentativo di "rompere" le comunicazioni della marina tedesca, che usava una versione di Enigma più complicata (i rotori possibili erano 8, anche se ne venivano utilizzati tre alla volta): si cercò di fare irruzione negli U-boat tedeschi per recuperare la macchina cifrante e il relativo *codebook*.

Questi sforzi non furono vani: dalla primavera del 1941 le comunicazioni degli U-boat poterono essere nuovamente decifrate; ma l'1 febbraio 1942 i Tedeschi optarono per una versione di Enigma più sicura (a quattro rotori) e gli Inglesi dovettero ricominciare da capo. E' questo un periodo sfavorevole per le navi alleate: infatti, non conoscendo più la posizione dei sottomarini tedeschi, queste venivano affondate in numero maggiore, con ovvie ripercussioni sui rifornimenti agli Inglesi.

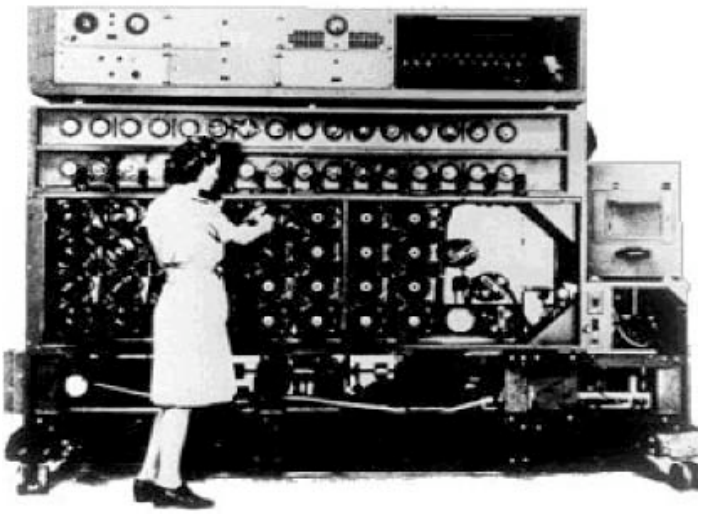
Con la creazione di macchine sempre più efficienti e "intelligenti" (facevano infatti uso dei teoremi probabilistici in maniera più oculata) gli Alleati riuscirono, dal 1943 a decifrare nuovamente i messaggi della marina tedesca.

Il lavoro fatto a Bletchley Park può essere riassunto nei seguenti punti:

- si contribuì in maniera decisiva alla sconfitta degli U-boat nella Battaglia dell'Atlantico;
- si contribuì a rafforzare la difesa aerea (potendo di fatto prevedere le mosse del nemico) e l'offensiva aerea (capendo il momento migliore per attaccare);
- si contribuì a vincere le campagne del Mediterraneo e del Nord Africa (inclusa El Alamein);
- si contribuì al successo delle Operazioni Overlord e Double Cross e ad una più facile invasione della Francia;
- si contribuì a capire il potenziale e ad identificare le nuove armi tedesche (quelle legate alla ricerca atomica, le armi V, i nuovi U-boat e jet);
- si contribuì a capire i reali effetti (economici e militari) degli attacchi alleati sulla Germania, ad esempio quelli mirati alle riserve di petrolio (fine del 1944);
- si contribuì inoltre a decifrare i messaggi giapponesi, fondamentali per capire la disposizione della difesa tedesca.

Si calcola (ma non se ne ha ovviamente la certezza) che i messaggi decifrati tra il 1938 e il 1945 furono circa 50'000: questa cifra può sembrare alta ma va detto che circa 1000 messaggi venivano intercettati ogni giorno, per un totale di circa 2,5 milioni di messaggi nell'arco dei 7 anni di guerra. Dunque gli Inglesi riuscirono a decifrare meno del 2% delle comunicazioni tedesche (anche perché molti messaggi venivano scartati prima di essere letti perché non ritenuti importanti): ma questo bastò a capire le intenzioni del nemico, ad anticiparlo e a sorprenderlo.

Anche per quanto riguarda lo sbarco in Normandia il lavoro dell'Hut 8 (il "reparto" dove lavorava Turing) fu fondamentale: i tedeschi non prendevano in considerazione un'invasione da quel tratto di costa e le intercettazioni lo resero noto agli americani. Gli alleati hanno saputo dunque vincere la guerra segreta, forse sottovalutata dai tedeschi, e questo ha aperto loro la strada verso la vittoria della Seconda Guerra Mondiale.



La bomba di Turing

UN CENNO ALLA CRITTOGRAFIA MODERNA

La grossa novità della crittografia moderna è che, per la prima volta nella storia, questa sia "uscita dai campi di battaglia" per essere di fatto disponibile a chiunque possegga un computer: è dunque diventata uno strumento di massa, atto a proteggere i segreti di stato tanto quanto i dati che noi vogliamo, o almeno vorremmo, rimanessero privati.

Con l'utilizzo di applicazioni come PGP (www.pgp.com) è possibile cifrare qualsiasi documento elettronico che non vogliamo venga visualizzato da altri. Ma la crittografia non è uno strumento per soli paranoici: è piuttosto l'unica difesa che abbiamo per salvaguardare un bene spesso trascurato: la nostra privacy.

Infatti ogni volta che (ad esempio) utilizziamo il nostro telefonino per inviare un SMS oppure per effettuare e ricevere chiamate pretendiamo che:

1) la nostra comunicazione non possa essere ascoltata da nessuno che non abbia un'autorizzazione;
2) il mittente di un messaggio SMS non possa essere falsificato;
3) il messaggio SMS inviato non possa essere in alcun modo modificato o intercettato durante il tragitto.

In poche parole pretendiamo autenticità, integrità e riservatezza; **questi sono esattamente** (si veda anche l'introduzione) **le finalità della crittografia**.

La crittografia non vuole solo aiutarci a mantenere segreta un'informazione ma vuole anche impedire che qualcuno possa modificare le informazioni a noi indirizzate, facendoci credere una cosa piuttosto che un'altra. Facciamo un esempio molto semplice: quando riceviamo un SMS sul telefonino, ci fidiamo ciecamente di ciò che

leggiamo; non ci chiediamo mai se la persona da cui riceviamo il messaggio lo abbia scritto effettivamente o se qualcun altro voglia ingannarci in qualche modo.

Non teniamo conto del fatto che, con appositi strumenti informatici, *non è difficile falsificare il mittente di un messaggio SMS*; i casi di falsificazione del mittente ma anche i casi di intercettazioni non autorizzate sono ancora moltissimi nonostante l'adozione di standard di sicurezza più alti. Ma perché ciò è possibile? Ed è possibile evitare che questo accada? In che modo?

La risposta a queste domande è la crittografia moderna che, usata in maniera opportuna, può garantire standard di sicurezza altissimi e la pressoché totale certezza in quanto ad autenticità, integrità e riservatezza.

Gli algoritmi simmetrici

Fino ad ora abbiamo parlato solamente di algoritmi simmetrici ovvero processi di cifratura (quali il cifrario di Cesare e quello di Vigenere) in cui *la chiave per crittare il messaggio è anche la chiave di decrittazione*.

Facciamo un esempio: supponiamo di avere la parola "chitarra" da cifrare attraverso il cifrario di Cesare; stabiliamo che la chiave sia 3, dunque, utilizzando l'aritmetica modulare:

C	H	I	T	A	R	R	A	
3	8	9	20	1	18	18	1	
								+
				3				
								=

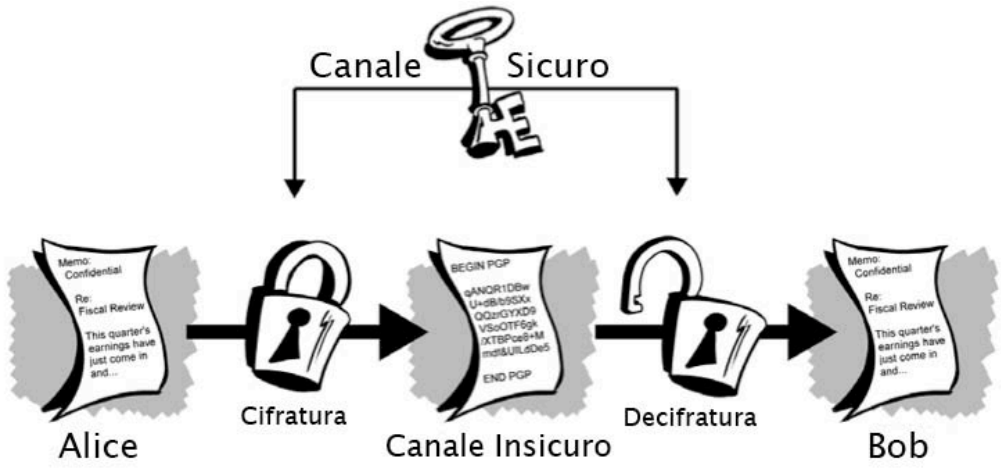
6	11	12	23	4	21	21	4	mod26
								=
6	11	12	23	4	21	21	4	
F	K	L	W	D	U	U	D	

Il destinatario dovrà sapere preventivamente che il messaggio è stato crittato con chiave 3 e quindi procederà alla decodifica facendo il processo inverso, ovvero facendo arretrare di 3 posizioni ogni lettera dell'alfabeto:

F	K	L	W	D	U	U	D	
6	11	12	23	4	21	21	4	
								-
				3				
								=

3	8	9	20	1	18	18	1	mod26
								=
3	8	9	20	1	18	18	1	
C	H	I	T	A	R	R	A	

Questo sistema presuppone quindi **due canali di comunicazione**; infatti prima i due interlocutori (che chiamiamo Alice e Bob) dovranno scambiarsi la chiave attraverso un canale considerato "sicuro" (in cui sono ritenute impossibili o comunque difficili le intercettazioni) e solo dopo potranno comunicare su un canale "insicuro" ovvero attraverso il quale è calcolata la possibilità che il messaggio venga intercettato (ma non che venga decifrato). Schematizzando:



In conclusione:

a) la stessa chiave è usata per cifrare e decifrare; quindi chi entri in possesso della chiave non solo sarà in grado di leggere la comunicazione tra Alice e Bob ma potrà falsificare i messaggi (e cifrarli nuovamente) e dunque modificare sostanzialmente la conversazione;
b) *esiste davvero un canale sicuro?* Come fanno ad essere sicuri Alice e Bob che nel momento in cui si scambieranno la chiave non ci sia una microspia? Dunque: la crittografia simmetrica può essere ritenuta valida?

La risposta è no, almeno in prima analisi. Gli algoritmi simmetrici hanno bisogno di essere applicati più volte oppure di essere combinati con algoritmi asimmetrici per garantire una comunicazione sicura.

Un esempio di algoritmo simmetrico usato ancora oggi (ed è stato utilizzato per 20 anni dal governo americano) è il **Data Encryption Standard (DES):** questo è nato negli anni 70 dai laboratori IBM e divide il messaggio in blocchi da 64 bit (56 bit utili, gli altri sono di controllo) crittandoli con chiave che è considerata sicura solo se di dimensioni maggiori di 1024 bit.

Piuttosto che il DES ora si usa il TDES, ovvero il **Triple-DES**, che consiste nell'applicazione ripetuta (per tre volte) del DES.

Gli algoritmi asimmetrici

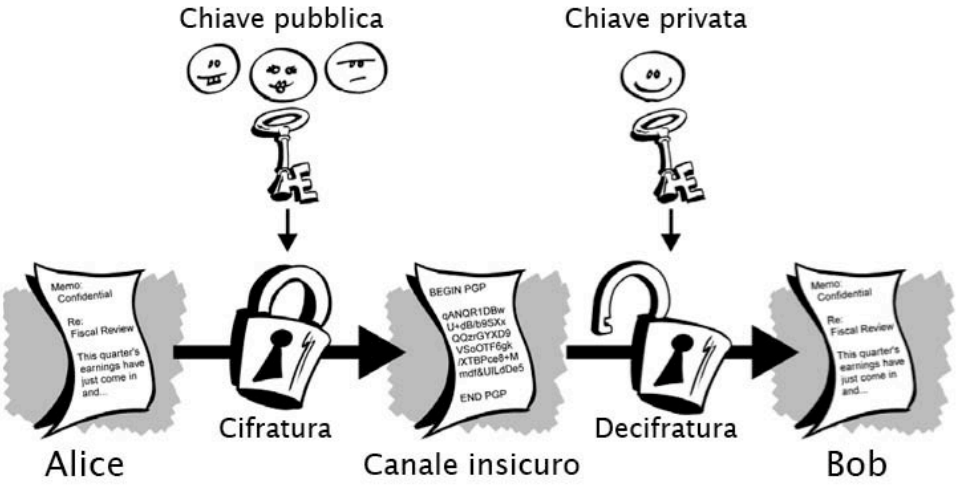
Gli algoritmi asimmetrici fanno uso di una chiave diversa per la cifratura e la decifratura: questo permette diversi vantaggi, tra cui l'abolizione della necessità di un canale "sicuro".

Come è dunque possibile (almeno in teoria) comunicare sicuramente senza istituire di un canale "sicuro"? Facciamo un esempio: Alice deve mandare dei documenti top-secret a Bob e, per farlo, vuole utilizzare soltanto un canale pubblico (ad esempio il servizio di posta) quindi insicuro. Alice può dunque mandare a Bob una valigetta (contenente i documenti top-secret) chiusa con il lucchetto di Alice; Bob la riceverà, ma non potrà aprirla: attaccherà dunque il suo lucchetto e la rispedirà ad Alice. Questa, ricevuta nuovamente la valigetta, toglierà il suo lucchetto (di cui solo lei, ovviamente, ha la chiave) e la rispedirà ancora una volta a Bob; a questo punto Bob riceverà la valigetta più solo chiusa con il suo lucchetto: utilizzando la chiave che solo lui possiede avrà finalmente accesso ai documenti top-secret.

Ma come si può applicare questo modello teorico (e anche un po' scomodo) alla realtà? Si può pensare che chiunque possa avere un lucchetto come quello di Bob (che diventerà la *chiave pubblica* di Bob) ma ovviamente che solo Bob abbia la chiave (detta *chiave privata* di Bob, in quanto nessun'altro oltre a lui deve conoscerla) per aprirlo. In questo modo chiunque voglia mandare un messaggio segreto a Bob può "chiuderlo" con il lucchetto di Bob e da quel momento nessuno (neanche il mittente) potrà più "riaprirlo": solo chi avrà la chiave privata di Bob (e si suppone sia *solo* Bob ad averla) potrà "riaprire" il messaggio.

E' dunque chiaro che la crittografia asimmetrica faccia uso di una chiave diversa per cifrare (**chiave pubblica**) e decifrare (**chiave privata**): naturalmente esse sono

collegate ma possedendo la chiave pubblica *non* è possibile risalire alla chiave privata. Schematizzando una potenziale comunicazione:



A livello puramente matematico all'algoritmo della chiave pubblica deve corrispondere una funzione molto facile da calcolare in un verso ma pressoché impossibile da invertire: questo tipo di funzioni sono dette "*one-way*". Un esempio molto pratico di funzione one-way è prendere un piatto e spaccarlo in molti e piccoli pezzi: è facile spaccare il piatto, è invece un po' più difficile (quasi impossibile) ricomporlo. Sfortunatamente non possiamo usare una qualsiasi funzione one-way per cifrare un messaggio: infatti rompere un piatto in tanti pezzi lo rende irrecuperabile e quindi un messaggio crittato con una funzione one-way simile sarebbe illeggibile a chiunque; dobbiamo dunque utilizzare soltanto le funzioni "**trapdoor one-way**". Queste sono facili da calcolare in un verso e pressoché impossibili nell'altro: ma se si conosce il segreto (appunto "*trapdoor*") è possibile invertirle facilmente.

Un esempio molto pratico di funzione trapdoor one-way può essere smontare un orologio: smontarlo è di per sé abbastanza facile, ma rimettere assieme i pezzi è molto difficile, a meno che non si abbiano le istruzioni. Un altro esempio è la cassetta della posta: è facile inserire le lettere, ma, se non si ha la chiave per aprirla, è molto difficile recuperarle.

Un esempio di algoritmo asimmetrico: RSA

RSA è il più conosciuto tra gli algoritmi asimmetrici: prende il nome dai suoi inventori Ron Rivest, Adi Shamir e Leonard Adleman e si basa completamente sulla difficoltà di fattorizzare numeri grossi; preso ad esempio il numero 33 è facile "fattorizzarlo" ovvero trovare i numeri primi che moltiplicati mi danno proprio 33 (che saranno 3 e 11). Ma dato il numero 50235641 è più difficile trovare i suoi fattori primi, ovvero 6397 e 7853: più le cifre diventano alte e più la situazione si complica, anche per i moderni e velocissimi computer.

RSA lavora con numeri primi di 200 cifre e questo, oltre a renderlo un algoritmo piuttosto complesso, lo rende anche lento nell'utilizzo.

Prima di mostrare il funzionamento di RSA è necessario chiarire alcuni concetti matematici.

Teorema di Fermat

Il teorema di Fermat (1601-1655) afferma che *in un'aritmetica finita in ordine* ***p*** *(con* ***p*** *numero primo) per ogni numero intero positivo* ***a****, diverso da zero e di cui esiste l'inverso* ***x****, è sempre valida la relazione*:

$$a^{(p-1)} = 1 \bmod p$$

Sia **x** l'inverso di a tale che:

$$a*x = 1 \bmod p$$

Consideriamo Z_p^* l'insieme dei resti possibili ottenuti dalla divisione di un qualsiasi intero per p, escluso lo zero, che conterrà (p−1) elementi:

$$Z_p^* = \{ 1; 2; \ldots p-1 \}$$

Dunque Z_p^* conterrà (p−1) elementi *distinti*, che potrò rinominare:

$$z_1, z_2, z_3, \ldots z_{p-1}$$

Considerando un numero qualsiasi $a \in Z_p^*$, i seguenti saranno (p−1) prodotti ancora distinti:

$$a*z_1, a*z_2, a*z_3, \ldots a*z_{p-1}$$

Possiamo dire che:

1) Questi prodotti saranno sicuramente *diversi da zero* in quanto, essendo p = 0 mod p, se $a*z_i=0$, allora o a=0 oppure $z_i=0$. Ciò non è però possibile perché a è per ipotesi diverso da zero, mentre z_i non può essere uguale a zero. Infatti, sapendo che è sempre vero che p = 0 mod p e, posto (per assurdo) che $a* z_i=0$, abbiamo che $p = a* z_i$ e questo contraddice l'ipotesi che p sia un numero primo.
2) Questi prodotti saranno sicuramente *distinti*. Supponiamo per assurdo che esistano due termini distinti (per cui $z_i \neq z_j$) $a*z_i$ e $a*z_j$ per cui: $a*z_i = a*z_j$.
   Dunque: $a*z_i - a*z_j = 0 \bmod p$
   $a*(z_i - z_j) = 0 \bmod p$
   Sapendo per ipotesi che a sia invertibile (e minore di p), questa espressione è verificata solo quando $z_i = z_j$. Ne deduco che tutti i prodotti devono essere distinti.

Ma questi prodotti, siccome stiamo lavorando in modulo p, essendo tutti distinti, saranno in realtà una permutazione degli elementi di Z_p^*, dunque sapendo che:

$$a*z_1, a*z_2, a*z_3, \ldots a*z_{p-1} = a^{(p-1)}*z_1*z_2*\ldots*z_{p-1}$$

Posso eguagliare il prodotto dei prodotti e il prodotto degli elementi di Z_p^*.

$$a^{(p-1)}*z_1*z_2*\ldots*z_{p-1} = z_1*z_2*\ldots*z_{p-1}$$

A questo punto posso "semplificare" il prodotto $z_1*z_2*\ldots*z_{p-1}$ perché è sicuramente diverso da zero (nessun elemento è uguale a zero) e perchè è invertibile. Dunque:

$$a^{(p-1)} = 1 \bmod p$$

Questo vale per qualsiasi $a \in Z_p^*$.

Per chiarire il concetto facciamo un esempio: prendiamo il numero 5. Sarà:

$$Z_p = \{ 0; 1; 2; 3; 4; \}$$
$$Z_p^* = \{ 1; 2; 3; 4; \}$$

Scegliendo un numero qualsiasi $a \in Z_p^*$ (ad esempio 3) e moltiplicandolo per ogni elemento di Z_p^*:

3*1 = 3 mod 5 = 3	3*2 = 6 mod 5 = 1	3*3 = 9 mod 5 = 4	3*4 = 12 mod 5 = 2
3	1	4	2

Abbiamo ottenuto 3, 1, 4 e 2, ovvero una permutazione di Z_p^*.

Facciamo un altro esempio: prendendo un numero primo, ad esempio (p=) 97 e un numero $a \in Z_p^*$, ad esempio 19. *Senza svolgere calcoli* possiamo sapere che 19^{96} diviso per 97 mi dà un resto pari ad 1.

Il teorema di Eulero-Fermat

Il teorema precedentemente enunciato fu successivamente (nel 1736, per la precisione) generalizzato da Eulero nella forma, dimostrabile in modo simile a quanto fatto in precedenza:

$$a^{\Phi(n)} = 1 \bmod n$$

Nel prossimo spiegherò cosa sia la funzione Φ e perché se questa agisce su numeri primi (supponiamo quindi *n primo*) vale il teorema di Fermat:

$$a^{\Phi(n)} = a^{n-1} = 1 \bmod n$$

La funzione Φ

Tenendo conto di tutte le proprietà dell'aritmetica modulare (elencate precedentemente), prendiamo tre numeri a, b ed n interi e tali che:

$$a = b \bmod n$$

Questa scrittura mi dice che a/n mi dà lo stesso resto di b/n.
Quindi:

$$\frac{a}{n} = Q + R \quad e \quad \frac{b}{n} = Q_1 + R$$

$$\frac{a}{n} - \frac{b}{n} = Q + R - Q_1 - R = Q - Q_1$$

Da cui:

$$\frac{a-b}{n} = Q - Q_1 \qquad e \qquad a - b = n * (Q - Q_1)$$

Essendo Q e Q1 sicuramente interi abbiamo dimostrato che a-b è divisibile per n.

Se a e b sono primi tra loro (ovvero non hanno divisori comuni all'infuori di 1) allora è facile dimostrare che esistono sicuramente due numeri interi x e y tali che:

$$ax + by = 1$$

Per assurdo possiamo dimostrare che a e b devono per forza essere primi tra loro: ammettendo che a e b non siano primi tra loro equivale a dire che essi hanno almeno un divisore k in comune, dunque:

$$a = k*c \qquad e \qquad b = k*d$$
$$kc*x + kd*y = 1$$
$$k*(cx + dy) = 1$$

Il termine c*x, che sicuramente è intero, risulta equivalere a:

$$cx = \frac{1}{n} - dy$$

Questo vuol dire che il termine c*x è in realtà intero solo se n = 1, cosa che succede solo quando a e b sono primi tra loro e, in poche parole a = c e b = d.
Da questo posso arrivare a dire:

$$by = 1 - ax$$
$$\frac{by}{a} = \frac{1}{a} - x$$

Il che mi dice che dividendo by per a questo mi dà come risultato –x e resto 1. Dunque:

$$by = 1 \bmod a$$

In conclusione: se a e b sono primi tra loro è sempre possibile determinare un valore di y per cui valga la relazione appena scritta.

Definiamo ora, nell'insieme dei numeri naturali, la funzione *Φ(n)*, nota con il nome di *funzione di Eulero*, che restituisce il numero di interi positivi minori o al più uguali ad n che sono primi con n.

$$\Phi(n) = \{\text{numero di interi positivi} \leq n \text{ primi con } n\}$$

Facendo qualche esempio:

$$\Phi(4) = \{ 1; 3; \} = 2$$
$$\Phi(7) = \{ 1; 2; 3; 4; 5; 6; \} = 6$$
$$\Phi(11) = \{ 1; 2; 3; 4; 5; 6; 7; 8; 9; 10; \} = 10$$
$$\Phi(15) = \{ 1; 2; 4; 7; 8; 11; 13; 14; \} = 8$$

Quando il numero preso in considerazione è primo, è facile notare che:

$$\Phi(n) = n - 1$$

Inoltre presi due numeri p e q primi tra loro (ad esempio 3 e 5) si può dimostrare che:

$$\Phi(p*q) = (p-1)(q-1)$$
Esempio: $\Phi(3*5) = \Phi(15) = (3-1)(5-1) = 2*4 = 8$

Riprendendo il teorema di Eulero-Fermat, secondo cui $a^{\Phi(n)} = 1 \bmod n$, diventa chiaro che, nel caso particolare in cui n sia primo, valga:

$$a^{\Phi(n)} = a^{n-1} = 1 \bmod n$$

Il funzionamento di RSA

Ora possediamo tutti gli elementi per capire il funzionamento dell'algoritmo asimmetrico RSA.

Ipotizziamo di voler scambiare un messaggio con un nostro conoscente: per semplificare il ragionamento, supponiamo di averlo trasformato in un numero *m*.

Scegliamo casualmente due numeri *primi* ***p*** e ***q*** molto grandi (come ho detto prima di più di 100 cifre) e sia ***n*** = p*q. L'unico vero vincolo è che *n sia minore o al più uguale a m*.

Sappiamo che:

$$\Phi(n) = \Phi(p*q) = (p-1)(q-1)$$

Si scelga a caso un numero ***e*** che sia primo con (p–1)(q–1) ovvero con Φ(n). *I numeri n ed e formeranno dunque la chiave pubblica, che può essere conosciuta da chiunque*. Possiamo dunque calcolare un numero ***d*** tale che:

$$d*e + x*\Phi(n) = 1 \bmod n$$

Questa è formalmente identica a: *ax + by = 1 mod n*.
Possiamo dunque dire i numeri *d* ed *x* esisteranno sempre visto che *e* e *Φ(n)* sono primi tra loro.
Il numero d costituirà la chiave privata, che deve essere nota solo al decrittatore.
Per crittare il messaggio m e trasformarlo in testo cifrato c, agirò nel seguente modo:

$$\boldsymbol{c = m^e \bmod n}$$

Si nota che per cifrare il messaggio è necessario conoscere solo la chiave pubblica e non la chiave privata. Per decrittare il messaggio invece dovrà calcolare nuovamente m:

$$\boldsymbol{m = c^d \bmod n}$$

Come faccio però ad essere certo che cd sia proprio uguale al messaggio originale? Lo posso dimostrare:

$$\mathbf{c^d} = (m^e)^d = m^{e*d} = m^{[1-x*\Phi(n)]} = \\ = m*[m^{\Phi(n)}]^{-x}$$

Per il teorema di Eulero-Fermat so che: $m^{\Phi(n)} = 1 \bmod n$.
Dunque:

$$m*[m^{\Phi(n)}]^{-x} = m*(1)^{-x} = \mathbf{m}$$

Riassumendo quanto detto:

Chiave pubblica	
n	Prodotto di due numeri primi **p** e **q** che devono rimanere segreti.
e	Relativamente primo a (p-1)(q-1)

Chiave privata	
d	Ricavata dalla relazione: $d*e + x*\Phi(n) = 1 \bmod n$

Per cifrare	
c	Calcolo (*con la sola chiave pubblica*) il testo cifrato $c = m^e \bmod n$

Per decifrare	
m	Ricalcolo il testo-in-chiaro (*con la sola chiave privata*) $m = c^d \bmod n$

Come faccio invece ad essere sicuro che, dato c, sia praticamente impossibile risalire a m, cioè al testo in chiaro?

La sicurezza di RSA si basa sulla difficoltà di calcolare Φ(n), che è equivalente a fattorizzarlo (ovvero trovare p e q); fino a quando non si scopriranno nuovi algoritmi o i computer raggiungeranno una potenza di calcolo miliardi di volte maggiore dell'attuale, sarà incredibilmente lungo (miliardi e miliardi di anni) risalire, tentando tutte le combinazioni possibili, da c a m.

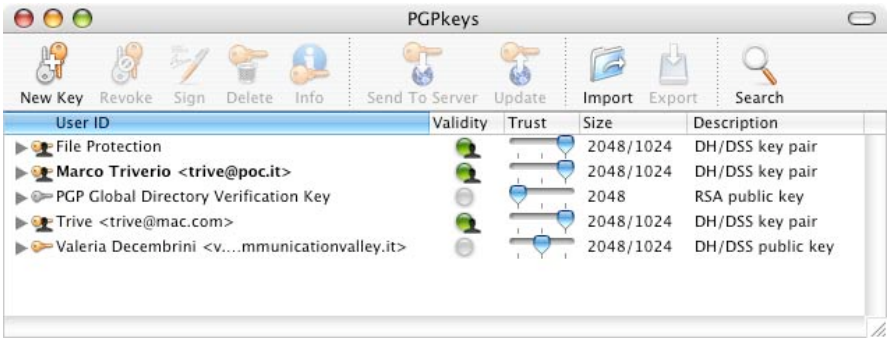
Un sistema ibrido: PGP

Gli algoritmi asimmetrici hanno un problema molto serio: anche se danno una sicurezza molto alta, sono davvero lentissimi. Per questo essi sono stati "incrociati" con quelli simmetrici per ottenere una crittografia veloce, efficiente e sicura.

E' questo il principio su cui si basa PGP, noto programma per il computer, che permette di crittografare i propri documenti e la propria

posta elettronica in un tempo ragionevole e in maniera assolutamente trasparente.

La finestra principale di PGP



PGP permette di gestire facilmente la propria chiave privata e le chiavi pubbliche di altri

Cosa ci riserva il futuro: la crittografia quantistica

Cosa succederà allora quando i computer saranno troppo potenti e nessun sistema crittografico sarà più intrinsecamente sicuro?

La risposta si conosce già ora; si abbraccerà una nuova crittografia: la **crittografia quantistica**.

Questa, in sintesi, non permette alcun tipo di intercettazione o modifica, da parte di terzi, alla comunicazione: infatti il mondo dei quanti è facilmente perturbabile; quindi un intruso che si intrometta nella comunicazione *in realtà blocca* la comunicazione in quanto i fotoni vengono assorbiti da lui e non arrivano al destinatario.

Questo ci fa capire che un intruso non può non essere scovato: anche se questi provasse a spedire fotoni "falsi" al destinatario, fare una "copia perfetta" della comunicazione è impossibile (per il teorema di non-clonazione dimostrato nel 1982 da Wooters e Zurek e, indipendentemente da Dieks); la comunicazione sarebbe visibilmente corrotta e verrebbe così facilmente scartata.

La crittografia quantistica è tuttora in via di sviluppo ed è oggetto di un progetto europeo (*SECOQC, Integrated Project Development of a Global Network for Secure Communication based on Quantum Cryptography*).

BIBLIOGRAFIA

i. Bruce Schneier, "*Applied Cryptography*", New York, John Wiley & Sons Inc., 1996;

ii. Bruce Schneier, "*Secret and Lies*", New York, John Wiley & Sons Inc., 2000;

iii. Michele Elia (docente del Politecnico di Torino), "*Note di Crittografia*", materiale didattico da www.polito.it;

iv. Robert Harris, "*Enigma*", Milano, Arnoldi Mondadori Editore, 1ª Edizione 1996;

v. Pierluigi Perri, "*Pevggbtensvn = Crittografia*", tratto da Hackers&C n.1, 2002, Piscopo Editore S.r.l., Roma;

vi. Pierluigi Perri, "*Storia della crittografia: come Hitler perse la guerra*", tratto da Hackers&C. n.2, 2002, Piscopo Editore S.r.l., Roma;

vii. Herbert Schildt, "*C++, guida completa*", Milano, McGraw-Hill, 1995;

viii. Namir Clement Shammas, "*A scuola di C++ for dummies*", Milano, Apogeo, 1998;

ix. Enrico Zimuel, "*Crittografia OpenSource: algoritmi asimmetrici*", per gentile concessione dell'autore;

x. Diego Casadei, "*Crittografia Quantistica*", tratto da Hackers&C. n.9, 2005, Piscopo Editore S.r.l., Roma.

SITI WEB CONSULTATI

(sono stati inclusi solo i più significativi)

i. Sito del Politecnico di Torino, http://www2.polito.it/didattica/polymath/htmlS/Archivio/Mappa/Mappa.htm e http://www2.polito.it_80/didattica/polymath/htmlS/argoment/APPUNTI/TESTI/Ott_02/APPUNTI.HTM

ii. SSH Communication Security, http://www.ssh.com/support/cryptography

iii. Sito ufficiale di PGP, www.pgp.com

iv. Sito del Liceo "M. Foscarini" di Venezia, http://www.liceofoscarini.it/studenti/crittografia/storia/storia.html

v. http://www-math.science.unitn.it/~caranti/Didattica/Algebra2/2000-01/Note/Algebra_2000-01/node9.html

vi. Teoria dei Numeri, http://math.usask.ca_80/encryption/index.html

vii. Introduzione alla crittografia, http://www.corsolinux.it/corsi/linux_networking/ldap/gnupg-crittografia.jsp

viii. Sito del National Security Agency, http://www.nsa.gov/

ix. Storia di Bletchley Park, www.english-heritage.org.uk/bletchleypark

x. Cryptography World, http/www.cryptographyworld.com_80/index.htm

ALTRO MATERIALE CONSULTATO

i. "*Enigma*", regia di Michael Apted, 2001, prodotto da Broadway Pictures, Jagged Films, Meespierson Film, Mulholland Pictures, Senator Film Produktion GmbH e distribuito in Italia da Istituto Luce;

ii. "*U-571*", regia di Jonathan Mostow, 2000, distribuito in Italia da Uip.