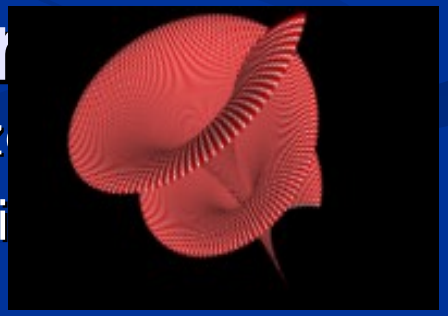
Liceo Michelangelo, 5 novembre 2010

Matematica nascosta

Qualche esempio di matematica che usiamo tutti i giorni senza saper

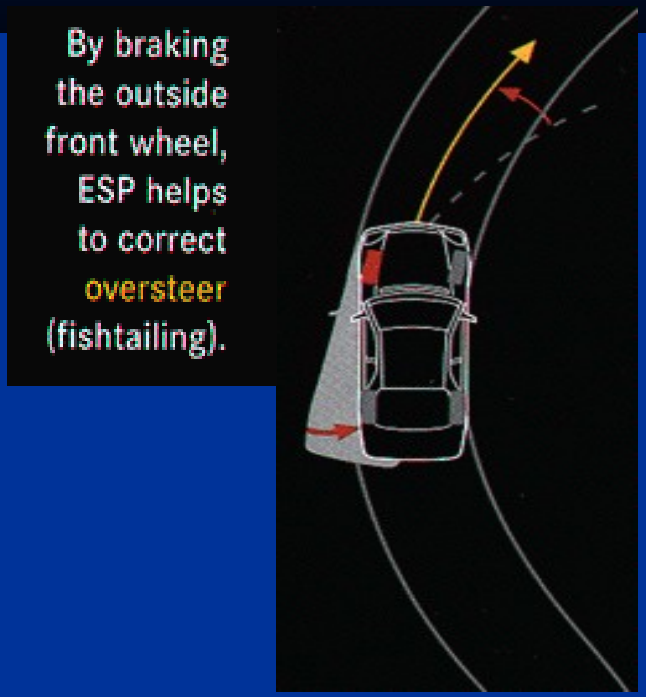
Riccardo Ricci, Università di Firenz
Dipartimento di Matematica “U. Dini



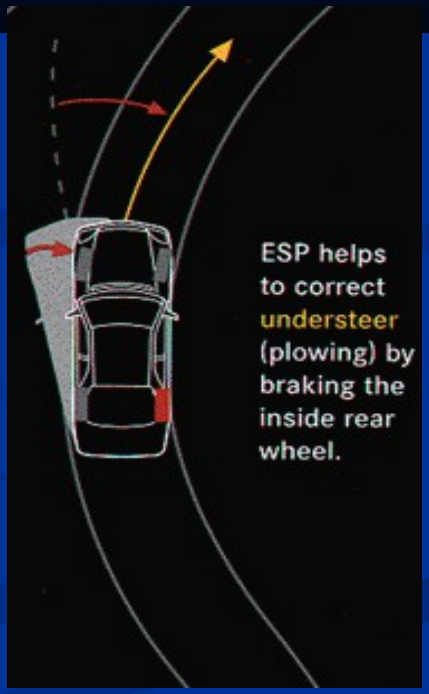
Un copilota intelligente

ESP acronimo per Electronic Stability Program

1997 M. Classe A

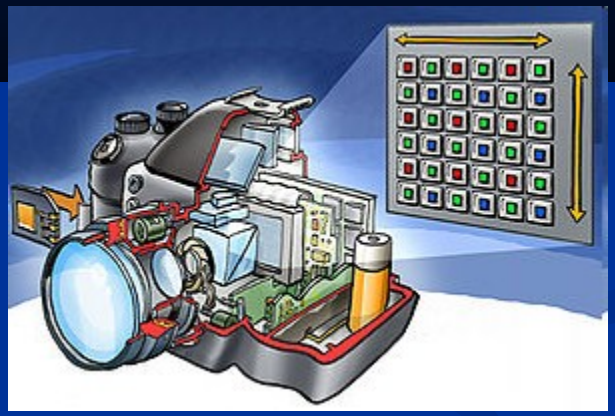


Il segreto dell'ESP è un computer che controlla costantemente dei sensori (accelerometri). Quando rileva il pericolo di ribaltamento reagisce agendo selettivamente su uno o più freni delle ruote anteriori o posteriori, riducento o aumentendo la spinta del motore.



Fotografia digitale

Al posto della pellicola c’è un SENSORE che reagisce alla luce



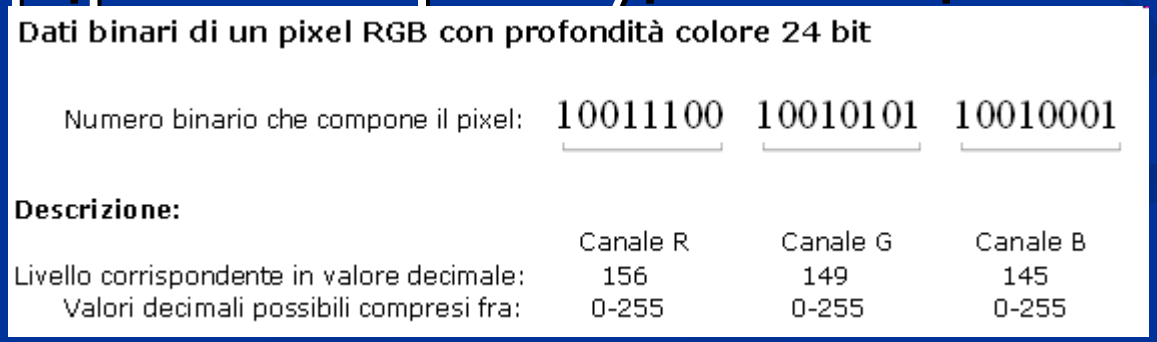
e trasforma il segnale luminoso in segnale elettrico

che viene poi convertito in un formato “digitale”

(una serie di 1 e 0, detti bit)

PIXEL

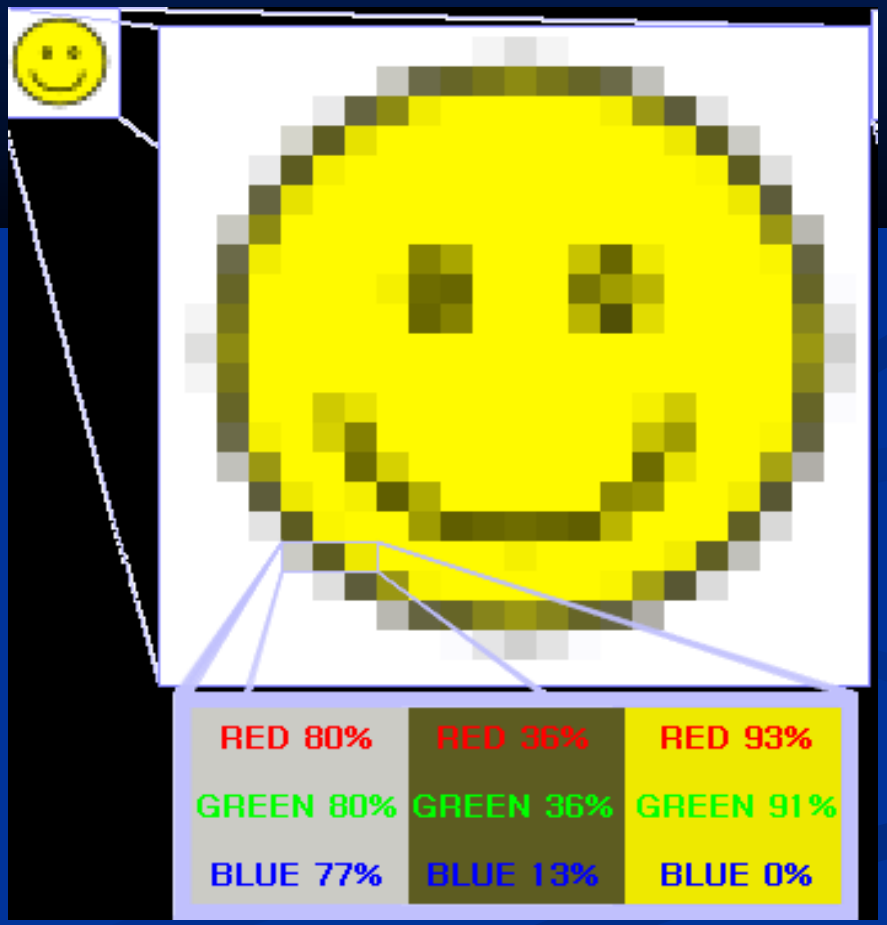
- alla base della codifica digitale c’è il pixel che registra le informazioni provenienti da tre sensori per i tre colori fondamentali (rosso verde e giallo)
- ogni colore viene registrato con una certa intensità
- p.e. 8 bit per colore (in questo caso 24 bit)



Dati binari di un pixel RGB con profondità colore 24 bit

Numero binario che compone il pixel:	10011100	10010101	10010001
Descrizione:			
	Canale R	Canale G	Canale B
Livello corrispondente in valore decimale:	156	149	145
Valori decimali possibili compresi fra:	0-255	0-255	0-255

RGB (red-green-blue)



Compressione dei dati

i dati raccolti nei pixel (immagine raw) occupano una enorme quantità di memoria e sono difficilmente trasferibili (immagini in Internet)

(una fotocamera media odierna associa 10 Megapixel, ovvero 1048578 pixel, a ogni foto)

occorre COMPRIMERE questi dati

la matematica in azione

il formato più noto di compressione per le foto è il formato JPEG, da **J**oint **P**hotographic **E**xperts **G**roup, nome del comitato che definì lo standard

- sfrutta una tecnica di rappresentazione matematica delle “funzioni” in termini di funzioni fondamentali (**Analisi di Fourier**, in questo caso le funzioni sono dei **coseni**) che rileva le variazioni spaziali delle informazione e permette di eliminare le “ripetizioni”

inoltre si “buttano via” le informazioni a cui l’organo della vista è meno sensibile.

JPEG

si ha quindi un doppio processo:

Immagazzinamento dell'immagine tramite la compressione

immagine raw → file .jpeg

(trasformata discreta del coseno, riordinamento dei dati per "importanza" ,

eliminazione dei dati "poco importanti")

Visualizzazione dell'immagine tramite decompressione dei dati

file .jpeg → immagine sullo schermo

(anti-trasformata discreta del coseno)

livelli di compressione

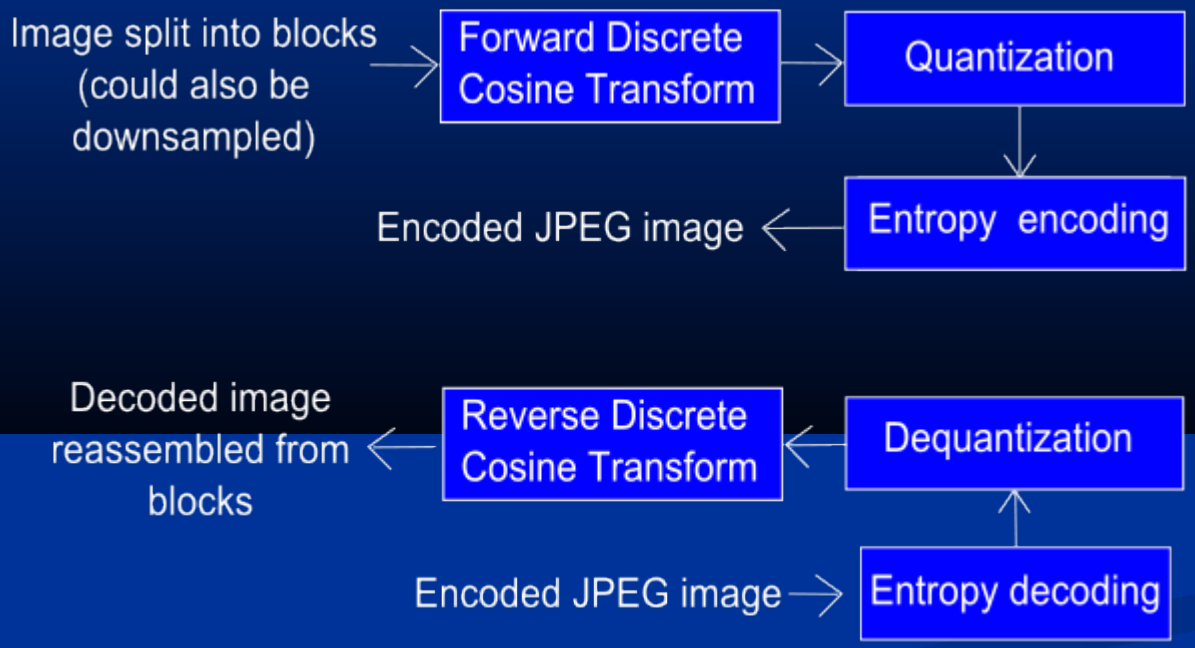
10% 3,2 Kb

50% 6,7 Kb

90% 30,2 Kb

100% 87,7 Kb

c’è poca differenza visiva tra queste due (ma molta di dimensione)



i terrorismo matematico! un esempio di trasformata discreta del coseno bidimen

$$X_{k_1,k_2} = \sum_{n_1=0}^{N_1-1}\left(\sum_{n_2=0}^{N_2-1} x_{n_1,n_2}\cos\left[\frac{\pi}{N_2}\left(n_2+\frac{1}{2}\right)k_2\right]\right)\cos\left[\frac{\pi}{N_1}\left(n_1+\frac{1}{2}\right)k_1\right]$$

$$= \sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} x_{n_1,n_2}\cos\left[\frac{\pi}{N_1}\left(n_1+\frac{1}{2}\right)k_1\right]\cos\left[\frac{\pi}{N_2}\left(n_2+\frac{1}{2}\right)k_2\right].$$

confronto tra formati

jpeg

$$X_{k_1,k_2} = \sum_{n_1=0}^{N_1-1}\left(\sum_{n_2=0}^{N_2-1} x_{n_1,n_2}\cos\left[\frac{\pi}{N_2}\left(n_2+\frac{1}{2}\right)k_2\right]\right)\cos\left[\frac{\pi}{N_1}\left(n_1+\frac{1}{2}\right)k_1\right]$$

$$= \sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} x_{n_1,n_2}\cos\left[\frac{\pi}{N_1}\left(n_1+\frac{1}{2}\right)k_1\right]\cos\left[\frac{\pi}{N_2}\left(n_2+\frac{1}{2}\right)k_2\right].$$

png (un formato con minor perdita di informazioni, specialmente nei contorni)

$$X_{k_1,k_2} = \sum_{n_1=0}^{N_1-1}\left(\sum_{n_2=0}^{N_2-1} x_{n_1,n_2}\cos\left[\frac{\pi}{N_2}\left(n_2+\frac{1}{2}\right)k_2\right]\right)\cos\left[\frac{\pi}{N_1}\left(n_1+\frac{1}{2}\right)k_1\right]$$

$$= \sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} x_{n_1,n_2}\cos\left[\frac{\pi}{N_1}\left(n_1+\frac{1}{2}\right)k_1\right]\cos\left[\frac{\pi}{N_2}\left(n_2+\frac{1}{2}\right)k_2\right].$$

MP3 e dintorni

Anche la trasformazione della musica in formato digitale è basata sull'Analisi di Fourier (o qualche sua variante moderna, wavelet, ..)

Il processo è più complesso in quanto il suono è un fenomeno intrinsecamente “analogico” (vibrazioni dell'aria, della membrana dell'orecchio, etc) mentre la vista è “quasi” digitale (nell'occhio c'è una sorta di matrice di ricettori simili ai pixel)

All'origine della moderna musica digitale c'è la codifica del suono per lo standard dei CD (campionamento 44.1 Khz, ovvero 44100 “saggi” al secondo, con una risoluzione di 16-bit per canale).
Anche qui abbiamo una quantità enorme di dati (un CD registra 80 minuti di musica in 700Mb di dati) e il formato è poco adatto alla circolazione della musica in Internet.

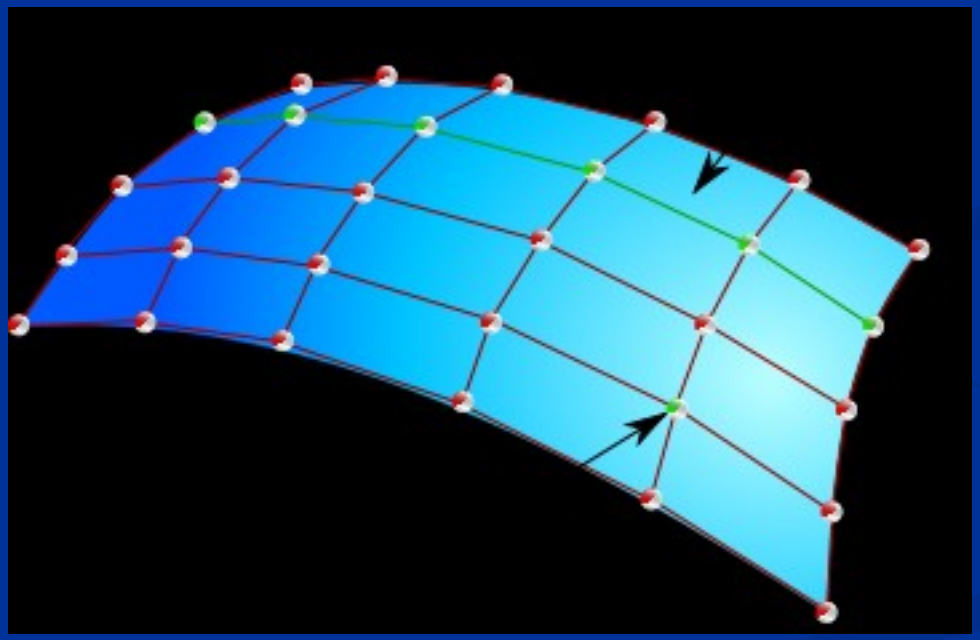
Sono così nati gli standard MP3, Ogg, etc. per la compressione di questi dati, basata principalmente su studi di “psicoacustica” per individuare quali componenti del segnale possono essere eliminate.

La computer graphics

(ovvero geometria proiettiva + analisi numerica + equazioni integrali)

Costruzione delle superfici

Una griglia di punti e delle funzioni che definiscono le superfici sopra la griglia in modo da garantire certe proprietà geometriche il cui equivalente visivo è la creazione di una superficie “liscia”



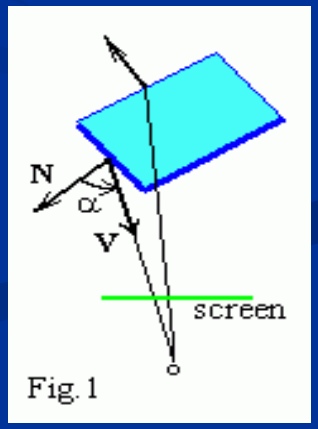
Le funzioni sono note con il nome di NURSB

Non Uniform Rational B-Splines

Nascondere le parti che non si vedono

Questo è relativamente semplice per figure “convesse” (l'algoritmo si chiama *backface culling*)

La superficie è rimossa se la normale (esterna) e la direzione dello sguardo formano un angolo acuto.



Rimozione di altre parti

Per oggetti non convessi si pone il problema delle parti che vengono nascoste da altre parti dell'oggetto

Lo stesso problema si ha se una parte dell'oggetto è nascosta da un altro oggetto

In questo caso si usa l'algoritmo detto “del pittore” che consiste nel rappresentare prima gli oggetti più lontani e poi “dipingendoci sopra” quelli più vicini (*Z-sorting*)

Si deve decidere “matematicamente” quali sono gli oggetti “più lontani”!

Ora bisogna “fare luce”

Trasmissione dati e numeri primi

- Un problema fondamentale oggi è la trasmissione (tramite Internet) di informazioni che devono restare segrete a tutti tranne che al destinatario
- Ovviamente è un vecchio problema, una volta di quasi esclusivo interesse militare e diplomatico.
- La disciplina che se ne occupa è nota con il nome di CRITTOGRAFIA

Nell'approccio classico chi invia il messaggio e chi lo riceve possiedono entrambi un codice comune (chiave) che funziona sia per la codifica che per la decodifica del messaggio (crittografia simmetrica)

La “chiave” può essere sia “astratta” (p.e. un algoritmo matematico) che “fisica” (una macchina) o un combinazione delle due

Cifrario di Cesare

Alfabeto

abcdefghilmnopqrstuvz

Una trasposizione ciclica delle lettere dell'alfabeto (di tre posizioni)

Alfabeto “cifrato”

defghilmnopqrstuvzabc

La macchina ENIGMA usata dai tedeschi durante
La seconda guerra mondiale e “decodificata”
dall'Intelligence inglese (tra cui Alan Turing)

La crittografia simmetrica non basta!

La crittografia simmetrica ha un difetto fondamentale: per farla funzionare si deve PRIMA condividere la chiave.

Questo va bene per situazioni “istituzionalizzate” (p.e. lo scambio di messaggio tra uno stato e le proprie ambasciate) me non funziona per i “rapporti occasionali”, come può essere il commercio elettronico: non è pensabile che chi fa acquisti in Internet debba prima procurarsi “segretamente” una chiave per cifrare lo scambio di informazioni con il venditore.

Abbiamo quindi bisogno di un sistema cifratura che permetta di comunicare la chiave di cifratura e che sia sicuro anche se questa chiave viene

La crittografia a chiave pubblica

Il problema viene risolto negli anni '70 con l'introduzione della la crittografia asimettrica, dove la chiave di codifica e di decodifica sono diverse.

Si arriva a definire il concetto di chiave “pubblica” : la chiave di codifica può essere comunicata pubblicamente, ma la decodifica del messaggio resta possibile solo per chi possiede la chiave di decodifica.

Da questo principio nasce la **RSA** (dalle iniziali di Rivest, Shamir e Adleman)

Il principio base è una semplice osservazione sulla pratica matematica:

è “facile” moltiplicare due numeri (anche grandi)

è difficile trovare i divisori (fattorizzare) di un numero grande (soprattutto se anche loro sono grandi)

Ingredienti della RSA

L'ingrediente matematico fondamentale è l'**algebra modulare,** cioè quella dei “resti” delle divisioni

Vediamo quelle per i resti “modulo” 3

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

*	1	2
1	1	2
2	2	1

3 è un caso “speciale”, la moltiplicazione è invertibile (lo è per

+	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

*	1	2	3
1	1	2	3
2	2	0	2
3	3	2	1

Tabellina del 4: notare che c'è un “divisore dello zero”

Come funziona la RSA:

Immaginiamo che A debba spedire un messaggio segreto a B. Occorrono i seguenti passaggi:

B sceglie due numeri primi molto grandi (per esempio da 300 cifre) e li moltiplica con il suo computer (impiegando meno di un secondo).

B invia il numero che ha ottenuto ad A. Chiunque può vedere questo numero.

A usa questo numero per cifrare il messaggio

A manda il messaggio cifrato a B, **chiunque può vederlo ma non decifrarlo**

B riceve il messaggio e **utilizzando i due fattori primi** che solo lui conosceva lo decifra.

A e B hanno impiegato pochi secondi a cifrare e decifrare, ma chiunque avesse intercettato le loro comunicazioni impiegherebbe troppo tempo per scoprire i due fattori primi, con cui si può decifrare il messaggio.

In realtà questo sistema non è così semplice e per trasmettere grandi quantità di dati occorre molto tempo, quindi A e B si scambieranno con questo sistema una chiave segreta (che

L'RSA in matematica

1) Si scelgono due numeri *p* e *q* primi, diversi tra loro e grandi.
2) si calcola $n=p*q$, detto modulo
3) si sceglie un numero *e* (esponente pubblico) coprimo e più piccolo del prodotto $(p-1)*(q-1)$
4) si calcola il numero d tale che $e*d \equiv 1 \mod (p-1)*(q-1)$ (cioè il resto di $e*d$ diviso $(p-1)*(q-1)$ fa *1*)

La coppia di numeri (n,e) sarà la chiave pubblica, mentre la coppia (n,d) la chiave privata

La forza di questa cifratura sta nel fatto che per calcolare *d* conoscendo *e* NON basta conoscere *n*, dobbiamo conoscere $(p-1)$ e $(q-1)$

Il messaggio (sotto forma di un numero *m*), viene codificato tramite l'operazione

$$c = m^e \bmod(n)$$

il simbolo ^ sta a indicare l'elevamento a potenza

e decodificato tramite l'operazione

$$c^d = m^{(ed)} = m^1 \bmod(n)$$

nota che se *m* è minore di *n* allora *m* mod *(n)* è proprio *m*

Detto così sembra un mistero! In realtà si può dimostrare tutto abbastanza facilmente usando due risultati classici di algebra modulare:

il piccolo teorema di Fermat e il teorema cinese del resto.

Esempio

p=3, q=11 (p-1)(q-1)=20*

*n=p*q=33*

Possiamo scegliere *e =7* che non ha fattori comuni con *20*, avremo

d = 3 (infatti *d*e =1* mod*(20)*)

(33,7) chiave pubblica, *(33,3)* chiave privata

Messaggio *m =15*

c = m^e mod *(n) = 15^7* mod *(33) = 27* messaggio codificato

m = c^d mod *(n) = 27^3* mod *(33) = 15* messaggio decodificato