Abilità Informatiche

Programma del corso

- Introduzione
- L'informazione digitale
  - Un minimo di basi teoriche
  - Cosa possiamo aspettarci da un computer, e cosa no
- Architettura degli elaboratori
  - Come è fatto un computer
- I sistemi operativi
  - Servizi che un computer offre all'utente, e come funzionano
- Principali applicativi
  - Usare efficacemente alcuni programmi di Office Automation

Obiettivi

- Alla fine del corso, chi ha frequentato con successo:
  - sarà un **utente informato**
  - comprenderà i principi di base dell'informatica, dell'architettura degli elaboratori e dei sistemi operativi
  - avrà una visione d'insieme e aspettative realistiche sull'informatica
  - potrà usare più agilmente gli strumenti informatici nei corsi che lo richiederanno

Informatica e discipline economiche

- Elaborazione delle transazioni
- Contabilità e bilancio
- Sistemi di supporto alle decisioni
- Analisi statistica di serie storiche
- Modelli economico-finanziari
- …

Le parole di oggi

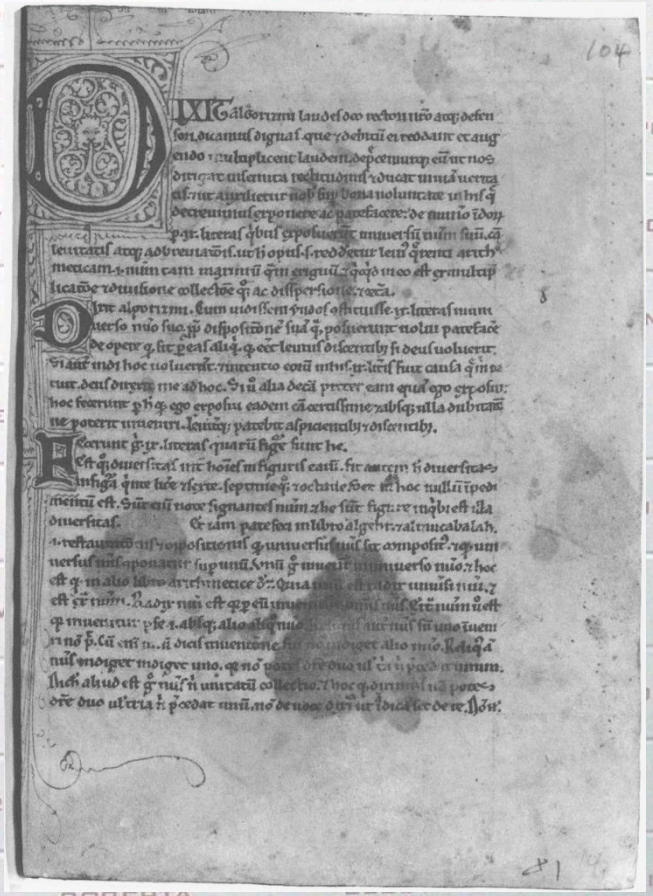
- Algoritmo
- Codifica
- Bit
- Multimedia
- Compressione
- Sicurezza

Algoritmi

- **Algoritmo**: sistema finito di regole e procedure di calcolo definite formalmente che portano alla soluzione di un problema con un numero finito di operazioni, quali che siano i dati di ingresso
- Esiste di un algoritmo per risolvere un problema → la risoluzione del problema può essere automatizzata

Algoritmi

- Il nome deriva dal matematico persiano *al-Khwārizmī*, che operò nella Bagdad nel IX secolo d.C.
  - Il califfo dell'epoca era *al-Ma'mūn,* figlio di *Hārūn al-Rashīd*, reso famoso dalle «Mille e una notte»



Risoluzione di un problema tramite l'elaboratore

1. Formulazione precisa del problema
2. Ricerca dell'algoritmo per la soluzione
3. Formulazione testuale dell'algoritmo in un **linguaggio** accessibile all'elaboratore (linguaggio di programmazione), ottenendo un **programma**
4. Esecuzione del programma

- Raffinamento

Poniamoci alcune domande

- Esiste un solo algoritmo per risolvere un problema?
- Un algoritmo si può implementare in un solo linguaggio?
- Tutti i programmi che implementano un determinato algoritmo in un linguaggio precisato sono equivalenti?
- Confronto e scelta tra algoritmi
- Confronto e scelta tra linguaggi
- Caratteristiche dei programmi

Interpreti e compilatori

- Partiamo dall'ipotesi di poter programmare in un certo linguaggio X
  - Possiamo cioè scrivere programmi in X per calcolare la radice quadrata, per verificare un bilancio, per gestire un magazzino ...
  - e anche un programma (scritto in X) che traduce verso X i programmi scritti in un altro linguaggio di programmazione Y
- A questo punto possiamo programmare anche in Y
  - Poi useremo il traduttore e il gioco è fatto

Interpreti e compilatori

- Partiamo dall'ipotesi di poter programmare in un certo linguaggio X
  - Possiamo cioè scrivere programmi in X per calcolare la radice quadrata, per verificare un bilancio, per gestire un magazzino ...
  - e anche un programma (scritto in X) che traduce verso X i programmi scritti in un altro linguaggio di programmazione Y
- A questo punto possiamo programmare anche in Y
  - Poi useremo il traduttore e il gioco è fatto
- Questi traduttori sono noti come **interpreti**, che traducono un'istruzione alla volta, o come **compilatori**, se traducono in blocco
  - Analogia: traduzione simultanea o traduzione di un libro

Codifica

- Codice Catastale
  - F839 → NAPOLI
  - H703 → SALERNO
  - A638 → BARCELLONA POZZO DI GOTTO
- CAP
  - Castellammare di Stabia → 80053
  - C.MARE DI STABIA → 80053
- Datemi due ragioni sul perché codificare

Codifica

- Assegnazione di un codice **univoco** a **tutti** gli oggetti compresi in un **insieme predefinito**

Codifica – elementi essenziali

- **Alfabeto** dei simboli utilizzabili
  - Esempio (CAP): cifre «0», «1», «2», …, «9»
  - Esempio: simbolo della valuta «€», cifre «0», «1», «2», …, «9», segni più «+» e meno «-», separatore decimale «,» e separatore delle migliaia «.»
    - Q: Che stiamo codificando?

Codifica – elementi essenziali

- Regole di composizione (**sintassi**) che definiscono le sequenze di simboli (parole) ammissibili:
  - Esempio: un CAP ha esattamente 5 cifre
  - Esempio: 1.234,50€ è la rappresentazione di un importo, 1,2€3,4.50 non lo è.
- Significato (**semantica**) delle configurazioni ammissibili

Codifica binaria dell'informazione

- Il calcolatore utilizza un alfabeto binario: usa dispositivi elettronici digitali in grado di assumere due soli stati: acceso/spento, ON/OFF, 1/0, VERO/FALSO
- Q: Avere un alfabeto binario limita le funzionalità? Non sarebbe stato meglio lavorare in decimale?
- *There are only 10 types of people in the world: those who understand binary, and those who don't*

bit

- Il simbolo o cifra binaria si indica con bit (da Binary digIT)
  - quantità di *informazione* che si ottiene selezionando una configurazione da un insieme di due configurazioni equiprobabili

$$\log_2 \frac{1}{p} = -\log_2 p$$

Se $p = \frac{1}{2}$, $\log_2 \frac{1}{p} = \log_2 2 = 1$ bit

Bit

- Quanti valori distinti si possono ottenere con 1 bit?
  - 0
  - 1
- 2 valori distinti

Codifica binaria dell'informazione

- Con 1 bit si possono codificare al più 2 informazioni.

Valore del bit	Codifica A	Codifica B	Codifica C	Codifica D
0	Nero	Non autorizzato	0	1
1	Bianco	Autorizzato	1	2

bit

- Quanti valori distinti si possono ottenere con sequenze (**parole**) di 2 bit?
  - 00
  - 01
  - 10
  - 11
- 4 valori distinti
- Si possono codificare al più 4 oggetti
  - p.es. contare da 0 a 3: {0, 1, 2, 3}
    - oppure da 1 a 4; o magari da 1001 a 1004: il punto di partenza è arbitrario
    - oppure da **-2** a **+1**: {-2, -1, 0, +1}

bit

- Quanti valori distinti si possono ottenere con parole di 3 bit?
  - 000
  - 001
  - 010
  - 011
  - 100
  - 101
  - 110
  - 111
- 8 valori distinti
  - Notate come quattro inizino per 0 e quattro per 1
- Si possono codificare 8 oggetti

Codifica binaria dell'informazione

- Se passiamo da una parola binaria di **k bit** ad una parola di **k+1 bit** si raddoppia il numero di oggetti che si possono rappresentare
  - Si aggiunge il prefisso 0 (e poi il prefisso 1) a ciascuna delle disposizioni dei restanti k bit

Codifica binaria dell'informazione

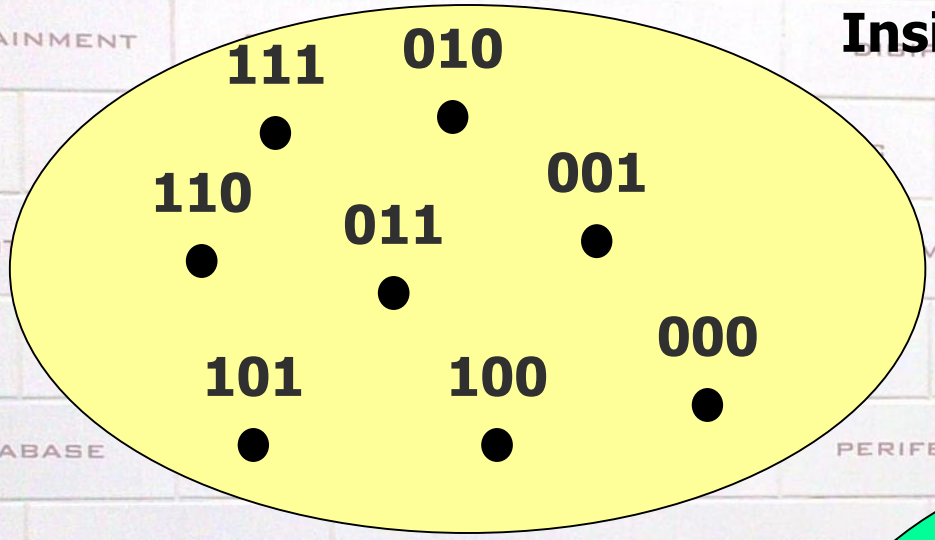
- Quanti oggetti diversi posso codificare con parole binarie composte da **k bit**?
  - 1 bit: $2^1 = 2$ valori (0, 1) $\Rightarrow$ 2 oggetti
  - 2 bit: $2^2 = 4$ valori (00, 01, 10, 11) $\Rightarrow$ 4 oggetti
  - 3 bit: $2^3 = 8$ valori (000, 001, 010, 011, 100, 101, 110, 111) $\Rightarrow$ 8 oggetti
  - …
  - k bit: $\mathbf{2^k}$ valori $\Rightarrow 2^k$ oggetti

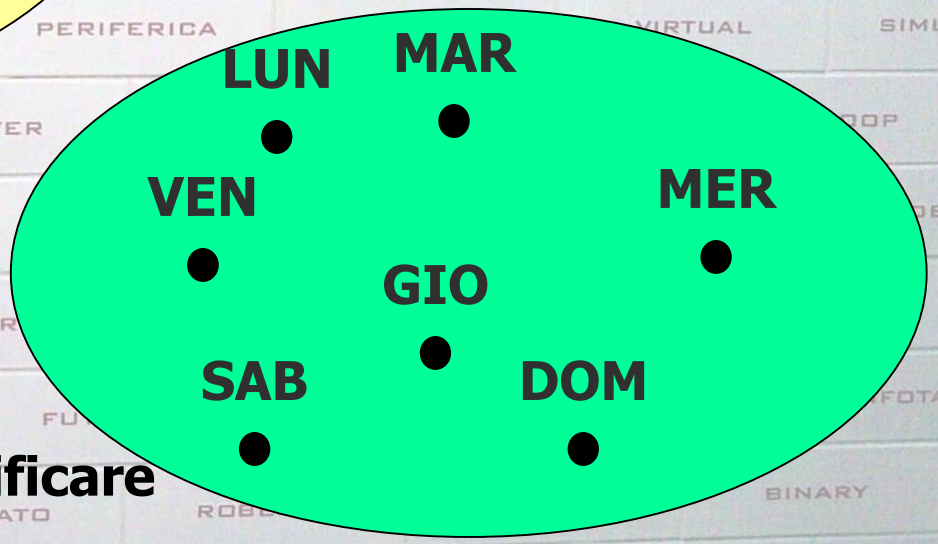
Definire un codice

- Identificare due insiemi:
  - Insieme delle configurazioni ammissibili
  - Insieme degli oggetti da rappresentare
- Associare gli elementi dei due insiemi
- Esempio: associare una codifica binaria ai giorni della settimana
  - 3 bit (che possono rappresentare $2^3 = 8$ oggetti) sono sufficienti

Esempio

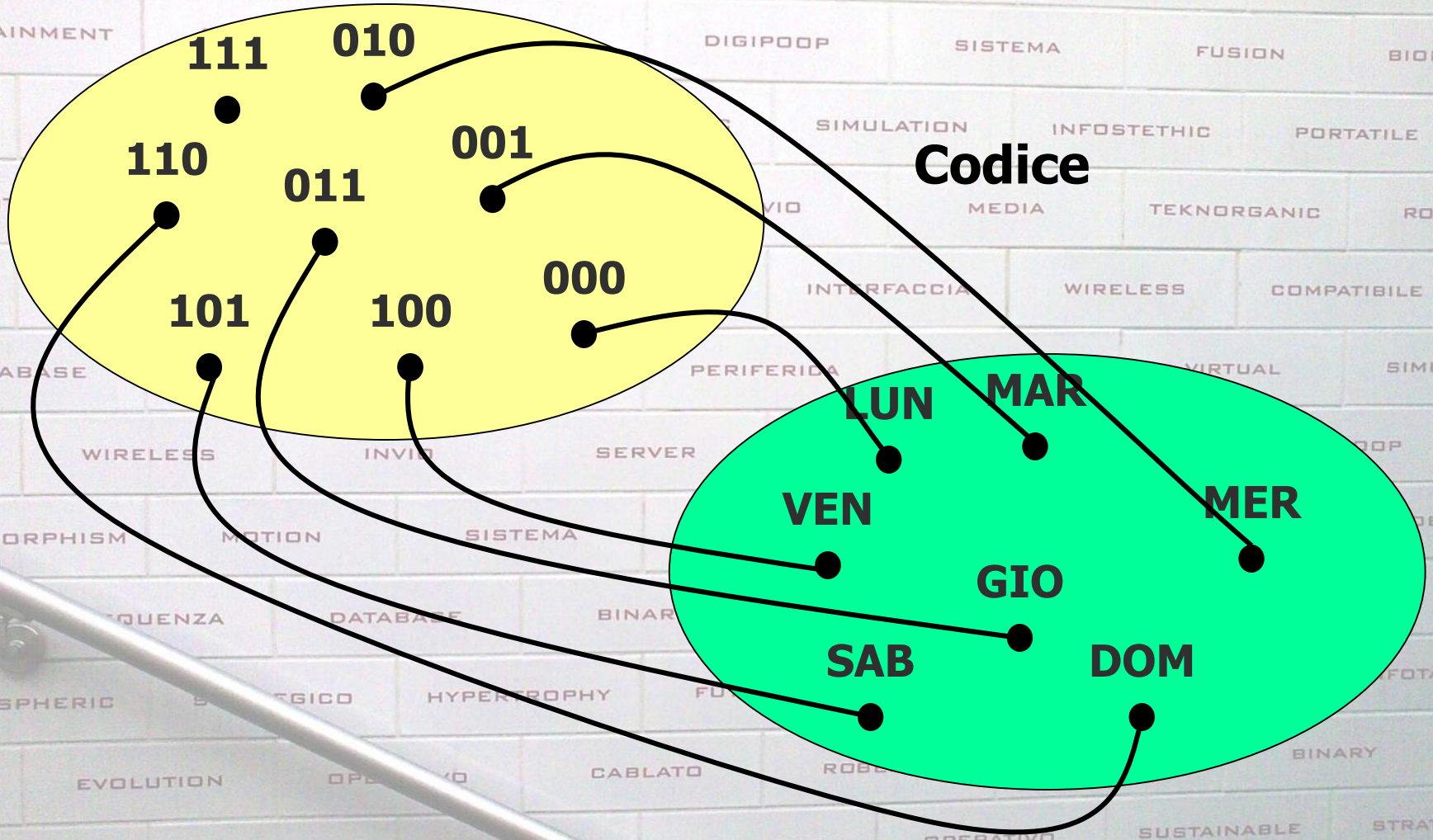
Insieme delle configurazioni ammissibili



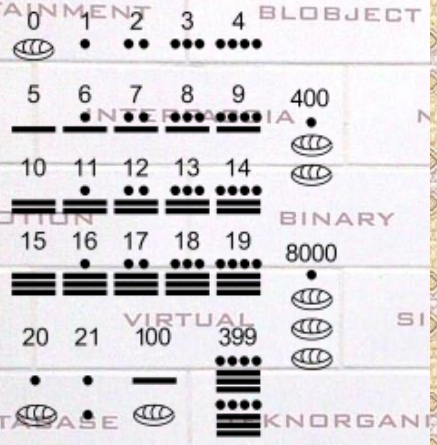


Insieme degli oggetti da codificare

Esempio


111
010
110
001
011
000
101
100
Codice
LUN
MAR
VEN
MER
GIO
SAB
DOM

Come rappresentare i numeri?



Byte

- Un gruppo di 8 bit (*ottetto*) prende il nome di byte
- Con 8 bit si possono codificare $2^8 = 256$ informazioni
- Con un byte si possono ad esempio rappresentare i numeri interi negli intervalli:
  - [0, 255] (da 0 a 255, estremi inclusi)
  - [1, 256] (da 1 a 256, estremi inclusi)
  - [-127, 126] (da -127 a 126, estremi inclusi)
  - Q: perché solo fino a 126 e non 127?

Perché [0,255]?

- 00000000 corrisponde a 0 in decimale
- 11111111 corrisponde a 255 in decimale, cioè $256 - 1 = 2^8 - 1$
- In generale,

$$\sum_{k=0}^{n-1} a^k = \frac{\mathrm{a^n} - 1}{\mathrm{a} - 1}$$

Per $a = 2$

$$\sum_{k=0}^{n-1} 2^k = \frac{2^{\mathrm{n}} - 1}{2 - 1} = 2^n - 1$$

Sistema metrico decimale

- Nel sistema metrico decimale, aggiungere un prefisso ad un'unità di misura indica una moltiplicazione
  - 1 km = 1.000 m
  - 2.5 GHz = 2.500 MHz (banda di frequenza usata dalle reti Wi-Fi … e non solo)

Sistema Metrico Decimale

Prefisso	Denominazione	Moltiplicatore	Significato
k	kilo	10^3 (1.000)	1.000 unità
M	mega	10^6 (1.000.000)	1.000 kilo
G	giga	10^9 (1.000.000.000)	1.000 mega
T	tera	10^{12} (1 seguito da 12 zeri)	1.000 giga
P	peta	10^{15}	1.000 tera
E	exa	10^{18}	1.000 peta
Z	zetta	10^{21}	1.000 exa
Y	yotta	10^{24}	1.000 zetta

Multipli del byte

- Q: Esiste il megabit?
- A: Certo; si usa spesso per esprimere la velocità di trasferimenti di dati in rete (megabit/s)

Codifica dei caratteri

- Vogliamo codificare un testo
  - Supponiamo, per ora, che sia in lingua inglese
- Occorre innanzitutto codificare i caratteri
  - a-z → 26 simboli
- Ma ci sono le maiuscole
  - A-Z → altri 26 simboli
- E i segni di punteggiatura
  - Almeno i più comuni , . ; : ? ! ’ " ( ) [ ]
- Eppure manca ancora un protagonista assoluto. Chi?

Codifica dei caratteri

- Lo spazio bianco!
  - *Altrimentileggeremmosoloun'indefinitasequenzadi caratteri*
- E se c'è qualche numero?
  - 0-9 → 10 simboli
- Le quattro operazioni
  - 4 simboli per + - / *
- Altri alfabeti: caratteri accentati
  - Aggiungiamo à è é ì ò ù (e anche À È É Ì Ò Ù)

Codifica dei caratteri

- E così via
- Insomma, i 256 valori distinti rappresentabili con un byte fanno comodo
- Ciascun carattere è rappresentato da un byte e corrisponde ad un valore numerico
  - Per un libro di 400 pagine con 30 righe, ognuna di 60 battute, quanti byte occorrono?
    - Nell'ordine del kB? Del MB? Del GB?

Codifica ASCII (estesa: 8 bit)

- È una corrispondenza tra i numeri dell'intervallo 0-255 e i simboli
- Esempi:
  - 65 → A 97 → a
  - 66 → B 98 → b
  - 90 → Z 122 → z
  - 58 → : 59 → ;
  - 232 → è 233 → é

Codifica Unicode

- Lingue orientali
  - Ideogrammi
  - Scritture bustrofediche
- Standard Unicode
  - Codifiche
- UTF-8
  - 1 byte per i caratteri ASCII
  - Da due a quattro byte per gli altri

Sicurezza: un nome, tanti significati

- Confidenzialità
- Autenticità
- Integrità
- Non-ripudio
- Disponibità
- Controllo accessi
- Privacy

Sicurezza

- Confidenzialità
  - Solo mittente e destinatario devono avere accesso al contenuto del messaggio
- Autenticità
- Integrità
- Non-ripudio
- Disponibità
- Controllo accessi
- Privacy

Sicurezza

- Confidenzialità
- Autenticità
  - Il messaggio deve effettivamente provenire dal mittente
- Integrità
- Non-ripudio
- Disponibità
- Controllo accessi
- Privacy

Sicurezza

- Confidenzialità
- Autenticità
- Integrità
  - Il messaggio deve raggiungere il destinatario senza alterazioni
- Non-ripudio
- Disponibità
- Controllo accessi
- Privacy

Sicurezza

- Confidenzialità
- Autenticità
- Integrità
- Non-ripudio
  - Il mittente non può negare di aver inviato proprio quel messaggio
- Disponibità
- Controllo accessi
- Privacy

Sicurezza

- Confidenzialità
- Autenticità
- Integrità
- Non-ripudio
- Disponibità
  - Dati e funzionalità di un sistema restano accessibili quando li si desidera
- Controllo accessi
- Privacy

Sicurezza

- Confidenzialità
- Autenticità
- Integrità
- Non-ripudio
- Disponibità
- Controllo accessi
  - Si deve poter regolare l'accesso a dati e funzionalità in base alle credenziali dell'utente
- Privacy

Sicurezza

- Confidenzialità
- Autenticità
- Integrità
- Non-ripudio
- Disponibità
- Controllo accessi
- Privacy
  - Non è possibile ricondurre un'azione a uno specifico utente

Disponibilità

- Ridondanza
  - Più istanze dei sistemi
  - Più copie dei dati
- Backup
  - copia di sicurezza o copia di riserva dei dati
  - Va fatto frequentemente
  - Dove conservare la copia?

Backup

- Si deve controllare che sia utilizzabile
  - restore: procedura che funziona perfettamente finché non ne hai bisogno
- I dati possono talvolta essere recuperati da un disco, ma …

Cifratura

- Modello: Alice, Bob … e Eve
  - mittente, destinatario e intruso in ascolto
- Messaggio in chiaro (plaintext)
- Messaggio cifrato (ciphertext)
- Chiave: segreto condiviso

Cifratura

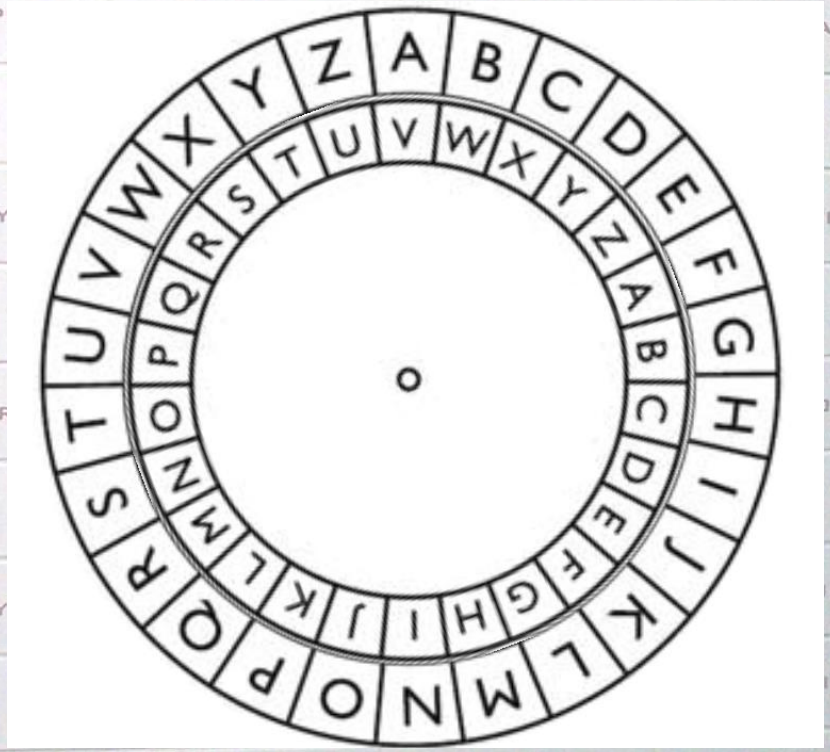
- Testo in chiaro
  - Messaggio originale
- Testo cifrato
  - Messaggio incomprensibile per tutti, tranne che per mittente e destinatario
- Cifrare
  - Trasformare il testo in chiaro in testo cifrato
- Decifrare
  - Recuperare il testo in chiaro dal testo cifrato

Principi di Kerckhoffs

- La sicurezza di un cifrario deve dipendere **solo** dalla segretezza della chiave e **non** dalla segretezza dell'algoritmo usato
  - Il sistema potrebbe cadere in mani nemiche
  - La chiave dev'essere, tuttavia, facile da comunicare e da sostituire
  - Auguste Kerckhoffs, «La cryptographie militaire», *Journal des sciences militaires*, vol. IX, pp. 5–38, Janvier 1883; pp. 161–191, Février 1883.

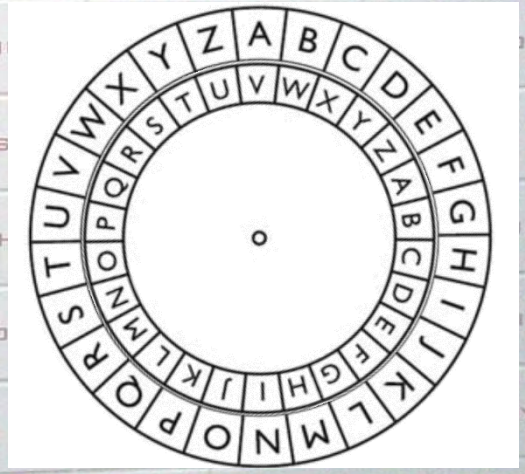
Cifrario a scivolamento

- Giulio Cesare
- Cifrario a sostituzione monoalfabetico
  - Rimpiazziamo ciascun simbolo con un altro
- Esempio:
  - Produciamo un alfabeto modificato facendo scivolare l'alfabeto originale di 5 posizioni a sinistra
  - La chiave è, in questo caso, 5.
  - Quante chiavi sono possibili?



Cifrario a scivolamento

- Alfabeto originale: ABCDEFGHIJKLMNOPQRSTUVXYZ
- Alfabeto modificato: FGHIJKLMNOPQRSTUVXYZABCDE
- Plaintext: DOMANI
- Ciphertext: ITRFSN
- Possiamo decifrare applicando la trasformazione inversa
-

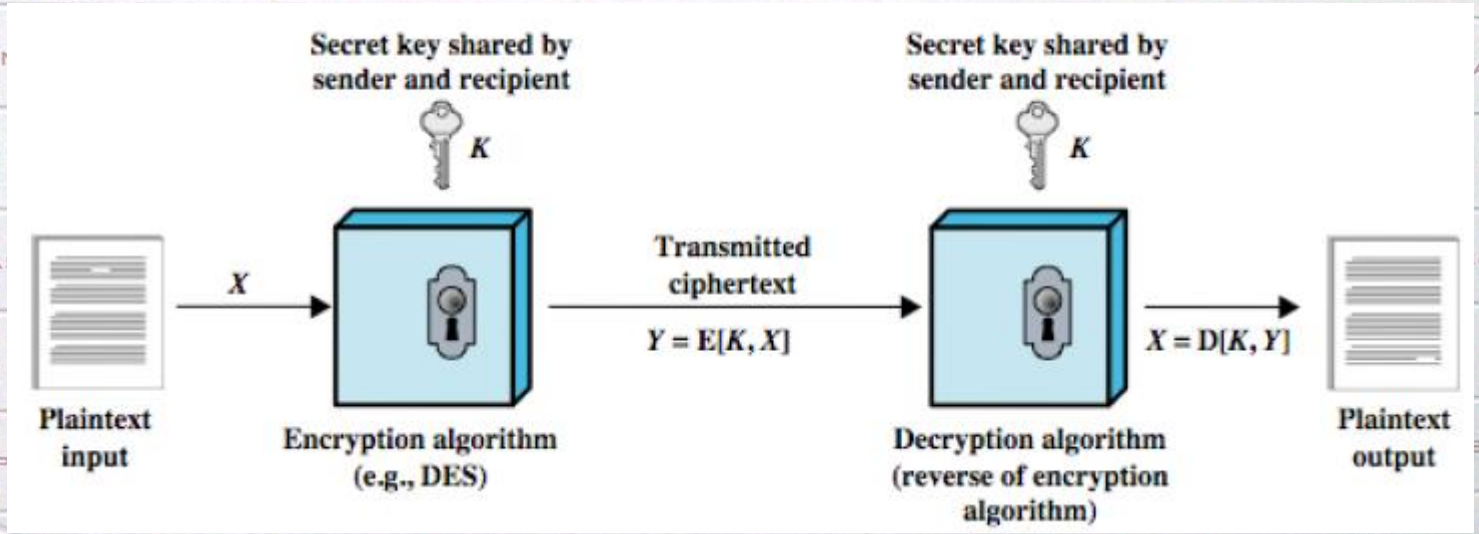


Un cifrario perfetto

- One-time pad (Vernam, 1917)
  - Il messaggio ed il crittogramma sono [dimostrabilmente] indipendenti
- ma
  - La chiave deve avere la stessa lunghezza del testo in chiaro
  - La chiave non deve essere riutilizzata

Crittografia simmetrica

- Si basa su una chiave segreta nota ad entrambe le parti

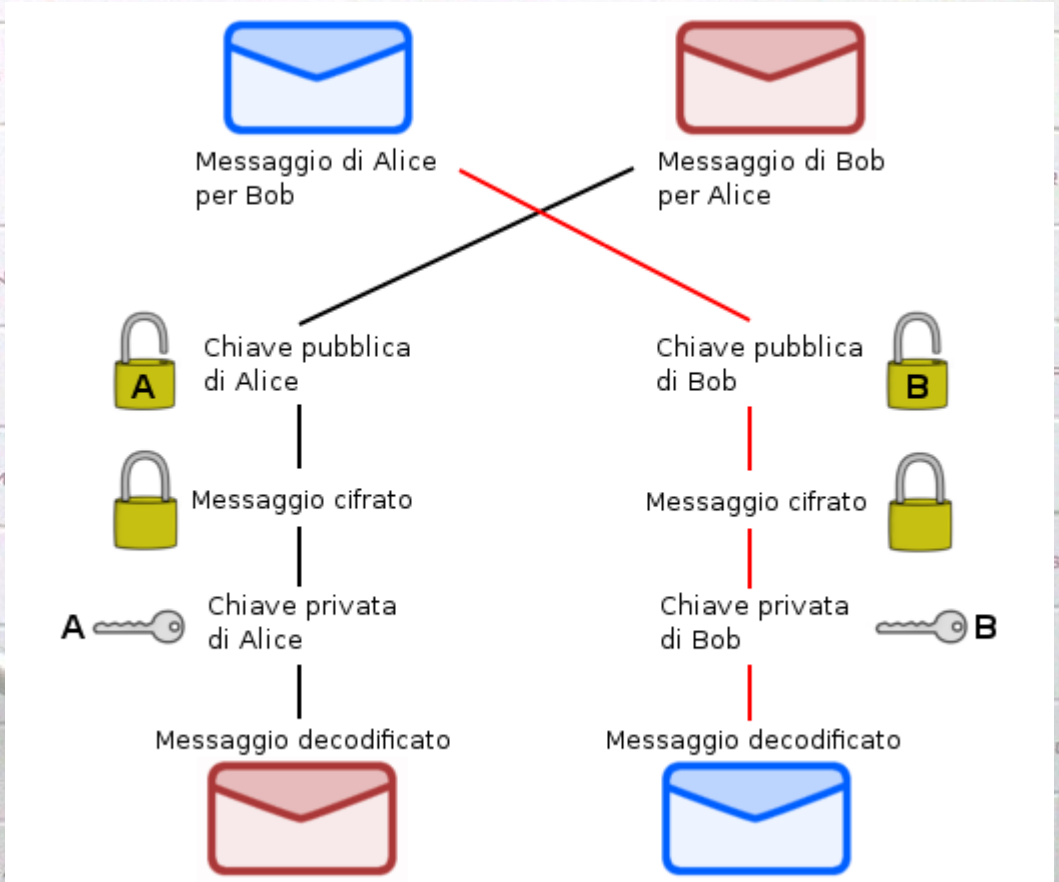


Limiti:

- La chiave deve essere scambiata in maniera sicura
- Non c'è modo di sapere se qualcuno ha corrotto il messaggio

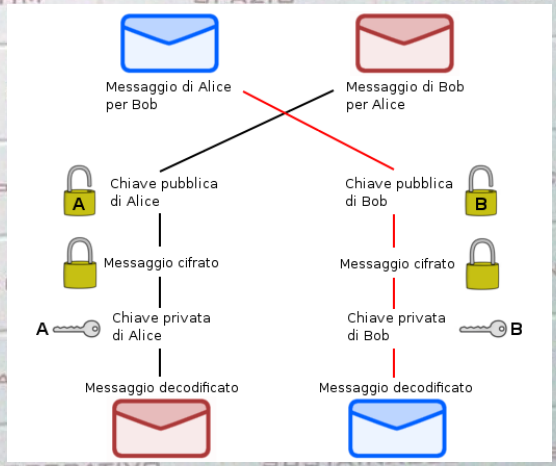
Crittografia asimmetrica

- Si basa su **due chiavi** differenti: una pubblica e una privata



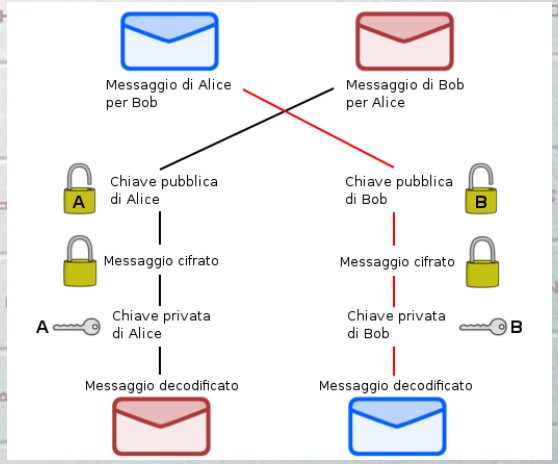
Verso la firma digitale

- Scenario 1 - Alice cifra il messaggio con la chiave **pubblica** di Bob
  - Alice conosce la chiave pubblica di Bob (è pubblica)
    Solo Bob (il solo a conoscere la propria chiave **privata**) può decifrare il messaggio



Verso la firma digitale

- Scenario 2 - Alice cifra il messaggio con la propria chiave **privata** (che conosce solo lei)
  - Per decifrare il messaggio occorre la chiave **pubblica** di Alice
  - Tutti possono quindi decifrare il messaggio
  - Ma chi può averlo prodotto?



Analisi e attacchi

- In teoria, i sistemi crittografici sono robusti
  - Analisi molto approfondite
- "*In theory, theory and practice are the same. In practice, they are not.*" (Einstein)
- Il successo di attacchi contro un sistema crittografico dipende da errori nell'implementazione, p. es.
  - Random Number Generators di bassa qualità
  - riutilizzo

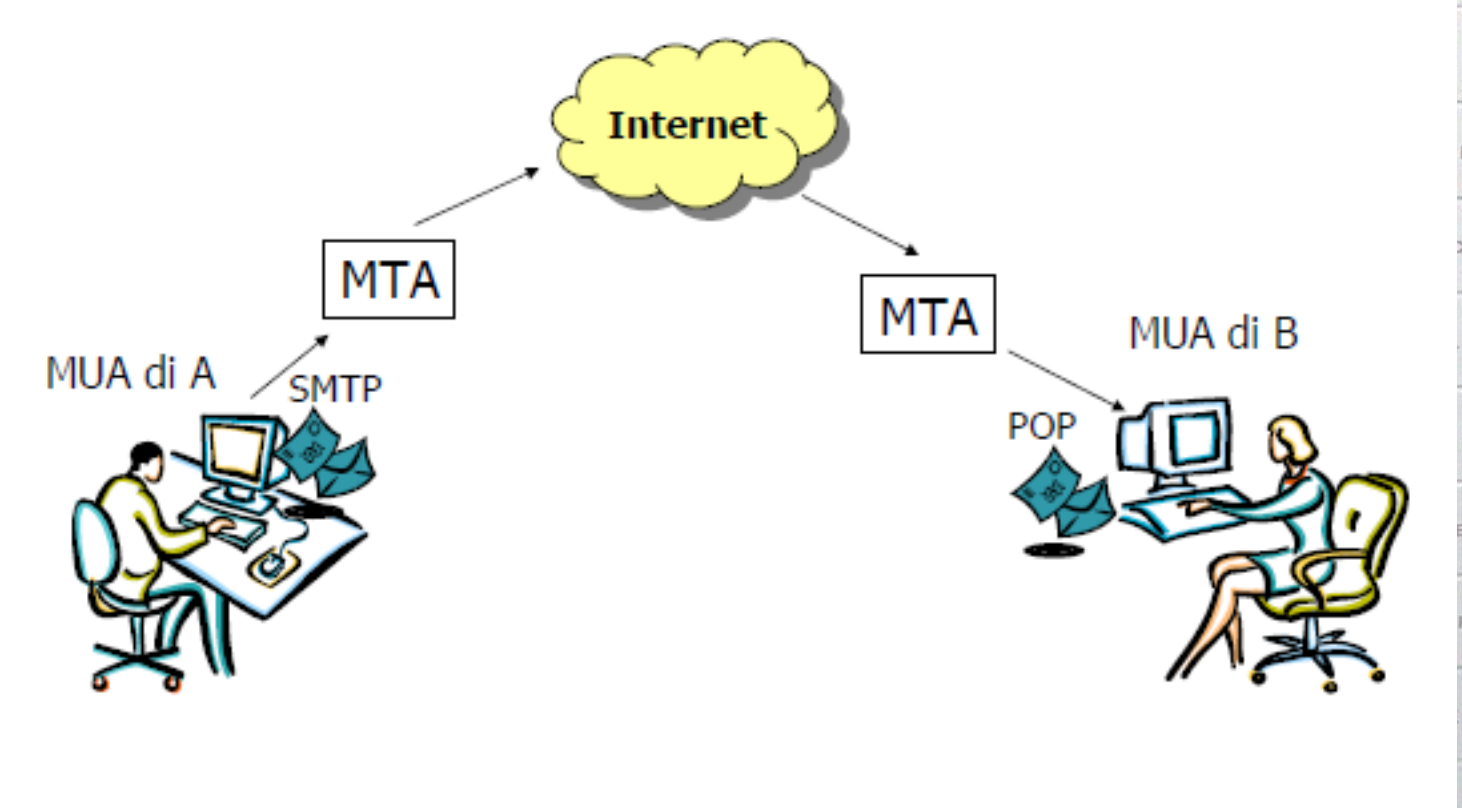
PEC — Definizione

- La Posta Elettronica Certificata è un sistema di posta elettronica nella quale si fornisce al mittente documentazione elettronica, con valore legale, attestante l'invio e la consegna di documenti informatici
- «Certificare» l'invio
  - fornire al mittente una ricevuta che costituisce prova legale dell'avvenuta spedizione del messaggio
- «Certificare» la ricezione
  - inviare al mittente la ricevuta di avvenuta (o mancata) consegna con precisa indicazione temporale

Posta elettronica 'ordinaria'

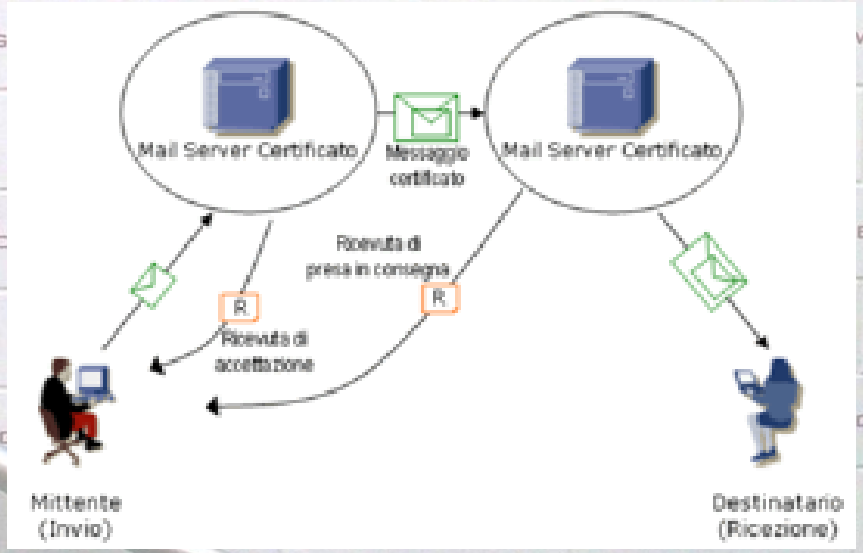
- Errori
  - Indirizzo di posta elettronica errato o inesistente
  - Casella di posta elettronica piena
- La conferma di lettura è un opzione facoltativa per il destinatario

Posta elettronica `ordinaria` Schema generale



PEC—Sistema di invio e ricezione

- Gli MTA (Mail Transfer Agent) sia del mittente che del destinatario sono entrambi certificati secondo procedure e regolamenti stabiliti per legge
- Le varie ricevute (accettazione, presa in consegna e ricezione) sono garantite dal sistema



Malware

- **malware**
  - *malicious software*: in italiano è detto anche *codice maligno*
  - programma creato con l'espresso scopo di causare danni più o meno gravi al computer o un sistema informatico su cui viene eseguito

Virus, trojan e compagnia

- **virus**
  - malware in grado di infettare programmi e file e di clonarsi
- **backdoor** o **bot** o **rootkit**
  - malware in grado di consentire il controllo remoto del computer
- **trojan**
  - contenitore apparentemente innocuo contenente codice malevolo; quasi sempre installa un bot

Virus, trojan e compagnia

- **spyware**
  - malware che raccoglie informazioni dal computer e le trasmette all'esterno
- **keylogger**
  - malware che registra tutto cià che viene digitato sulla tastiera (quindi anche le password)

Antimalware

- **antimalware**
  - software atto a rilevare e, eventualmente, eliminare virus informatici e altri programmi dannosi
  - è sostanzialmente un riconoscitore di sequenze
  - è in grado di combattere solo il malware che conosce
  - meno è aggiornato, meno è utile

Privacy

- Tracking, Profiling, Fingerprinting
  - Pubblicità mirata (nel migliore dei casi)
- Non ci credete?
- https://amiunique.org
- http://whoer.net
- https://www.browserleaks.com

Privacy—contromisure

- Usare più browser
  - O più profili nello stesso browser
  - Uno per e-banking, e-government, etc.
  - Uno per le reti sociali
  - Uno per la navigazione normale
- La compromissione di un profilo non ha impatto sugli altri