Codici segreti: la crittografia nell'era dell'informazione

Agostino Dovier

Dip di Scienze Matematiche, Informatiche e Fisiche
Università degli Studi di Udine

Marzo 2019

Introduzione

- Nella prima lezione abbiamo visto:
- i cifrari monoalfabetici

Introduzione

- Nella prima lezione abbiamo visto:

- i cifrari monoalfabetici
- i cifrari polialfabetici (Vigenére)

P	A	R	I	S	V	A	U	T	B	I	E	N	U	N	E	M	E	...
L	O	U	P	L	O	U	P	L	O	U	P	L	O	U	P	L	O	...
A	O	L	X	D	J	U	J	E	P	C	T	Y	I	H	T	X	S	...

Introduzione

- Nella prima lezione abbiamo visto:

- i cifrari monoalfabetici
- i cifrari polialfabetici (Vigenére)

P	A	R	I	S	V	A	U	T	B	I	E	N	U	N	E	M	E	...
L	O	U	P	L	O	U	P	L	O	U	P	L	O	U	P	L	O	...
A	O	L	X	D	J	U	J	E	P	C	T	Y	I	H	T	X	S	...

- fino al loro “limite” perfetto (one-time-pad)

O	G	G	I	L	A	L	E	Z	I
01110	00110	00110	01000	01011	00000	01011	00100	11001	01000
10100	00011	01000	01101	00100	10100	00011	01000	01101	00100
11010	00101	01110	00101	01111	10100	01000	01100	10100	01100
♣	F	O	F	P	U	Q	M	U	M

⇐ monetina

Introduzione

- Oggi vedremo l'automazione elettromeccanica e informatica della crittografia

Introduzione

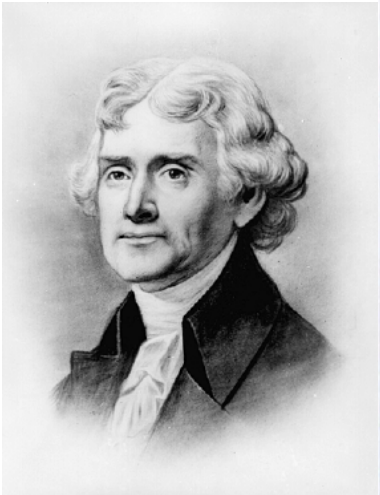
- Oggi vedremo l'automazione elettromeccanica e informatica della crittografia
- Prima studieremo la più famosa "macchina da cifra" ovvero l'ENIGMA

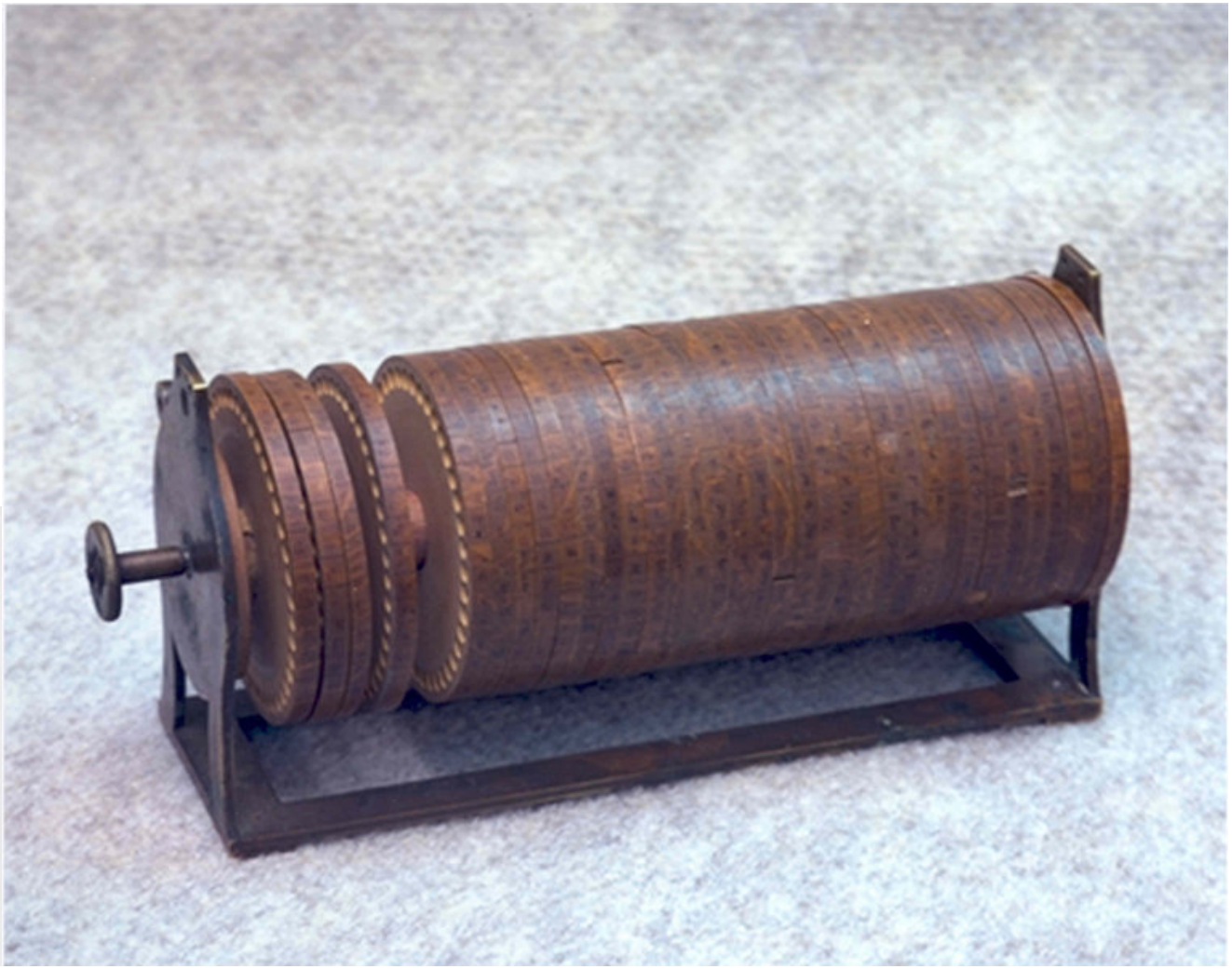
Introduzione

- Oggi vedremo l'automazione elettromeccanica e informatica della crittografia
- Prima studieremo la più famosa "macchina da cifra" ovvero l'ENIGMA
- Poi parleremo della crittografia informatica, sia a chiave privata (tradizionale: DES, AES) che a chiave pubblica (RSA).

Automazione della crittografia

Il rotore di Thomas Jefferson (1743–1826)

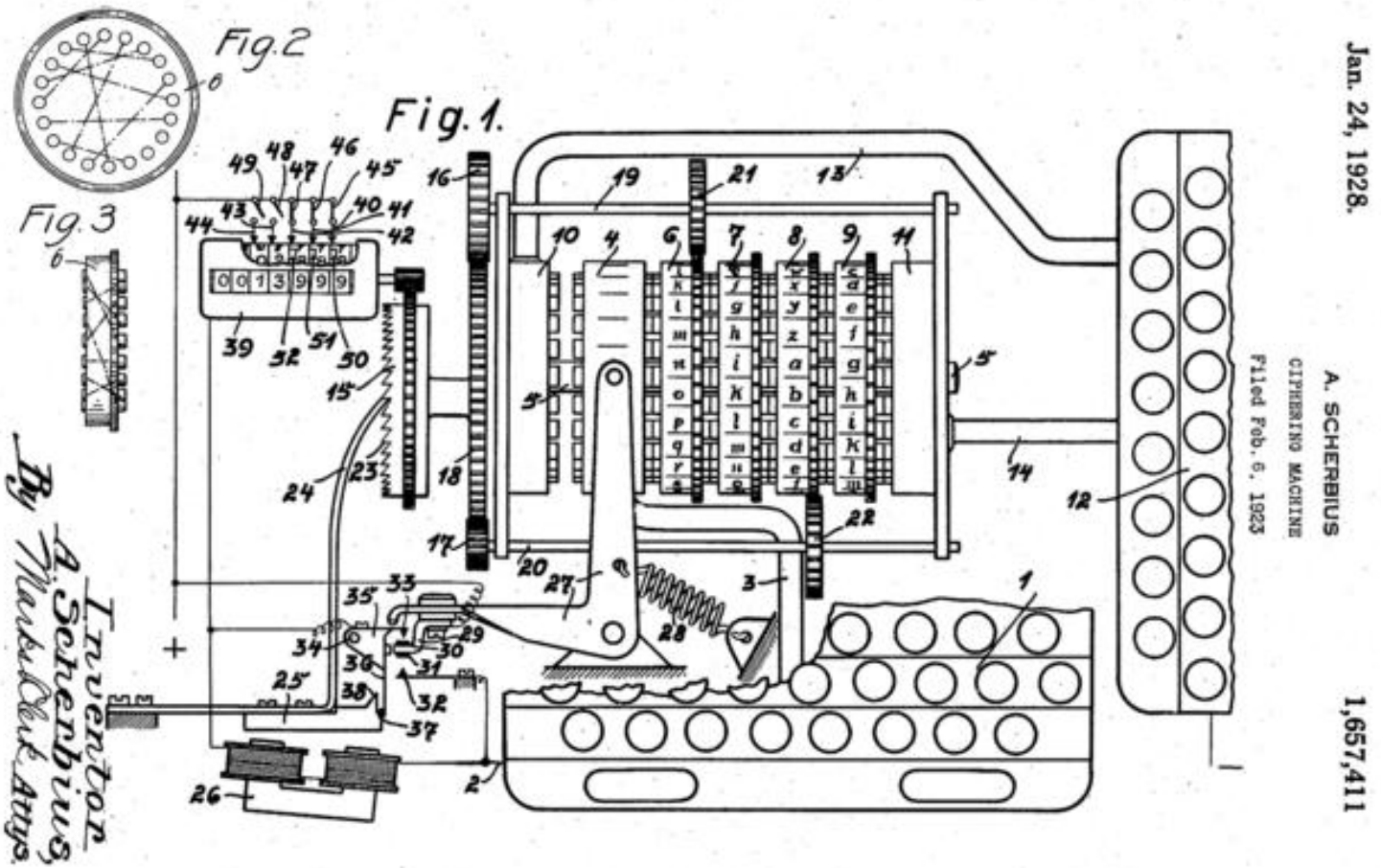




Enigma

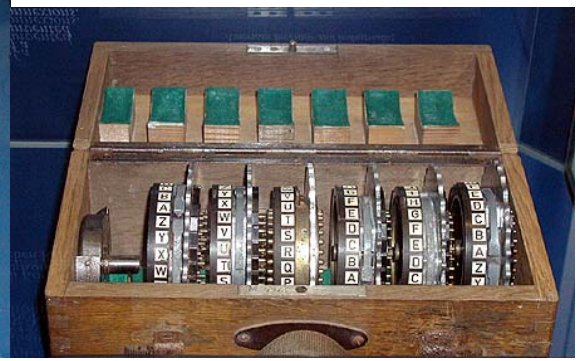
Arthur Scherbius (1878–1929)

Nel 1918 brevetta una macchina da cifra a rotori (multipli)



Enigma

Nel 1923 Scherbius commercializza l'Enigma.



Enigma: funzionamento

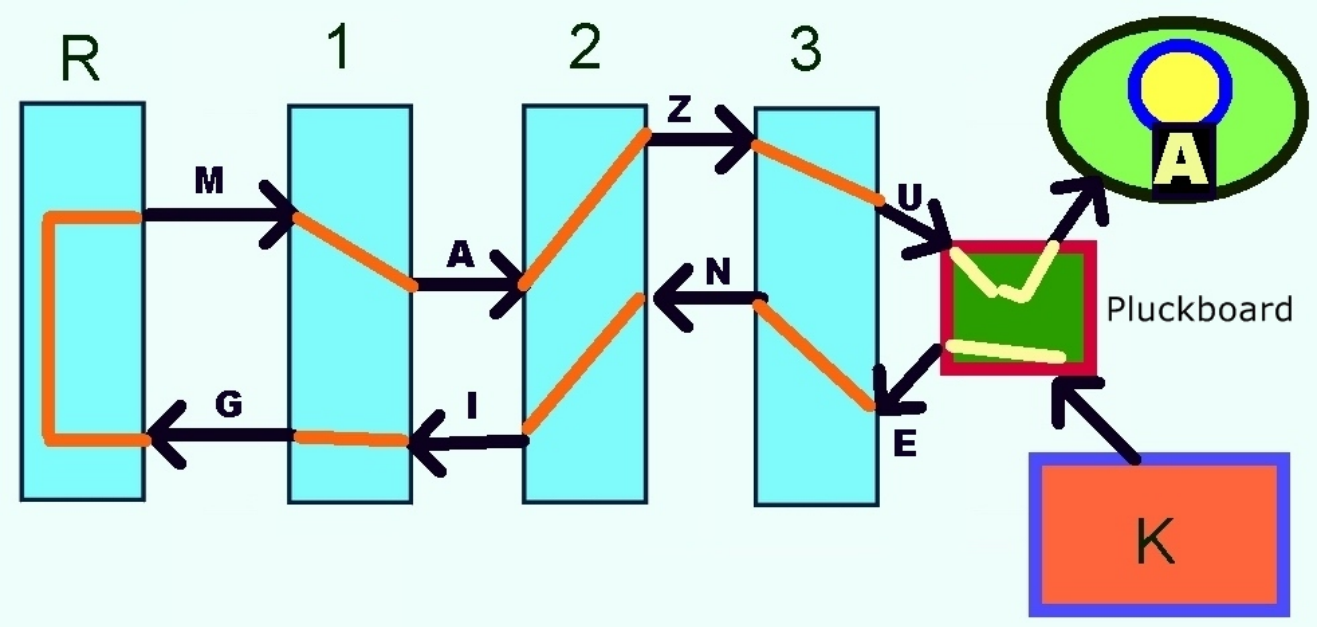
- Si tratta di un cifrario polialfabetico.
- Un punto di forza è che la lunghezza del periodo (o della chiave nel senso di Vigenère) è maggiore della lunghezza dei messaggi, il che rende apparentemente simile al one-time-pad.
- Le tecniche statistiche viste per Vigenère non si possono applicare.

Enigma: funzionamento

- Si tratta di un cifrario polialfabetico.
- Un punto di forza è che la lunghezza del periodo (o della chiave nel senso di Vigenère) è maggiore della lunghezza dei messaggi, il che rende apparentemente simile al one-time-pad.
- Le tecniche statistiche viste per Vigenère non si possono applicare.
- Anche se la chiave che si deve veramente comunicare è corta (vedremo come mai)
- L'altro punto di forza (per l'epoca) era l'automazione elettrica della trasformazione (sia per cifrare che per decifrare) e la relativa semplicità d'uso (anche se mancava la stampante).

Enigma: funzionamento

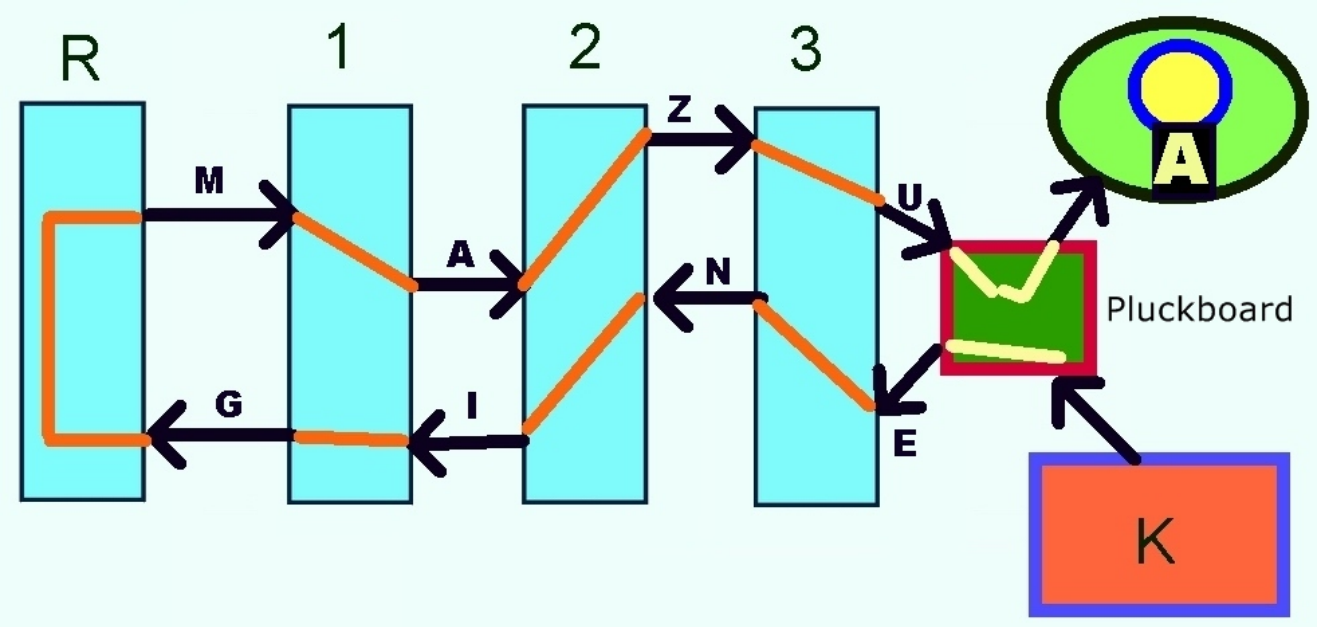
Si tratta di una macchina a rotori non fissi che si muovono con moto odometrico dopo ogni lettera codificata—stepping motion (come i vecchi contachilometri o i contatori dell'acqua o del gas).



Ogni rotore ha 26 contatti su una faccia e 26 sull'altra e implementa una sostituzione monoalfabetica (completa).

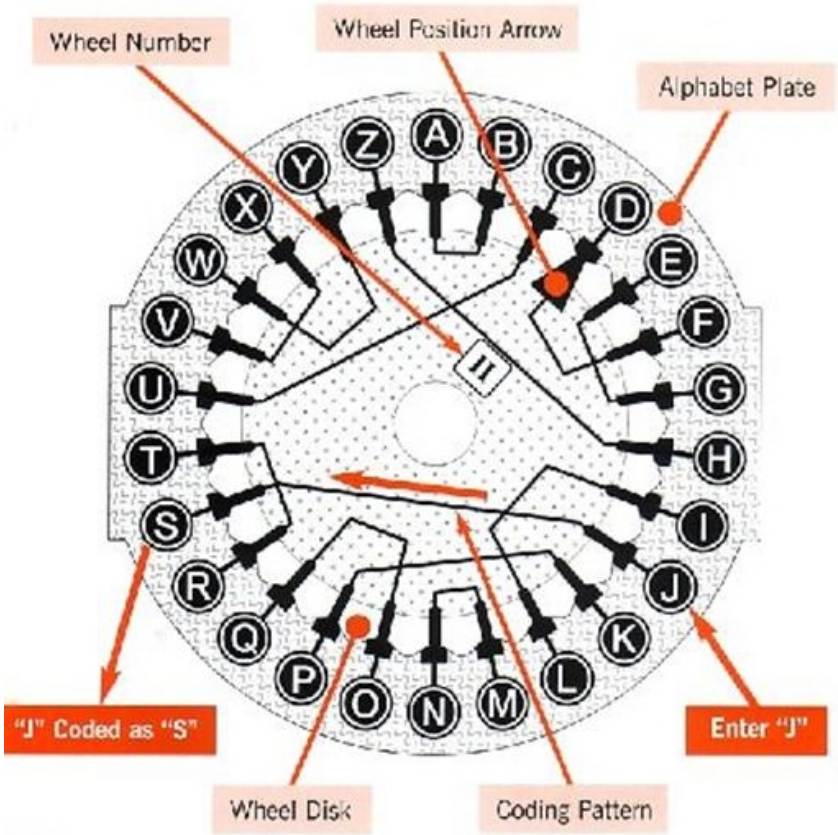
Enigma: funzionamento

Si tratta di una macchina a rotori non fissi che si muovono con moto odometrico dopo ogni lettera codificata—stepping motion (come i vecchi contachilometri o i contatori dell'acqua o del gas).



Ogni rotore ha 26 contatti su una faccia e 26 sull'altra e implementa una sostituzione monoalfabetica (completa).
Il reflector garantisce simmetria. Inoltre (brevetto) mai una lettera era codificata in sè stessa.

Enigma: funzionamento



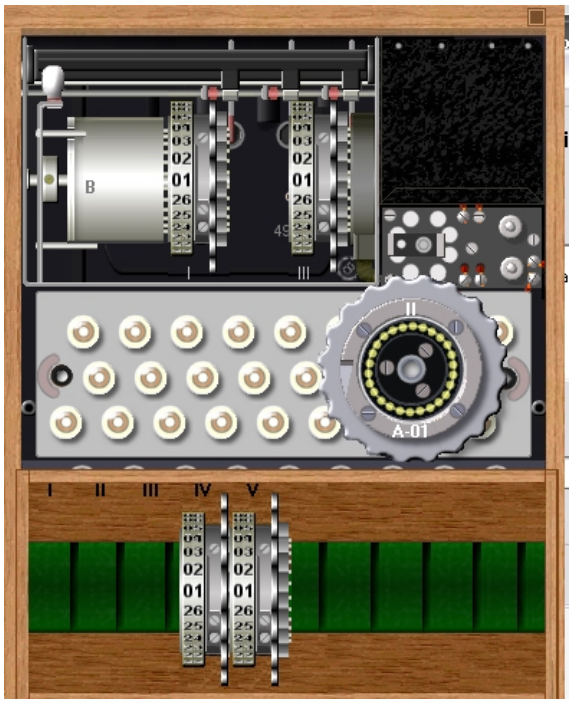
Enigma: funzionamento

Oltre ai 3 (o 4) rotori c'era una pluckboard (pannello elettrico) che permetteva un'ulteriore (e più libera) sostituzione:

Inoltre c'era una sostituzione iniziale tra tastiera e primo rotore (fissa, ma poteva cambiare cambiando il modello della macchina).

Enigma: Un bel simulatore

- DEMO



- Scaricatelo da qui:
  `http://users.telenet.be/d.rijmenants/`
- (ci sono in rete anche simulatori per android e ipad/iphone)

Enigma: funzionamento

- Diverse varianti sono state usate. Concentriamoci su quelle a 3 rotori (per quella a 4, $\times 26$).
- Fissati i rotori, le possibili chiavi iniziali erano $26^3 = 17576$ (456976 per 4 rotori)
- Tale numero è anche la lunghezza del *periodo* (o della chiave nel senso di Vigenère—a dire il vero $26 \cdot 25 \cdot 26$)
- Erano possibili 6 posizioni per i rotori.
- In versioni più evolute veniva fornita una scatola con 5 o, per la marina, 8 rotori da cui sceglierne 3 (o 4). Allora potevano esserci $6 \times \binom{8}{3} = 536$ posizioni.
- Inoltre c'erano i collegamenti sulla plugboard. Con 6 cavi ci sono $\sim 10^{11}$ possibilità.
- In generale, per k cavi ($k = 1, \ldots, 13$) abbiamo:

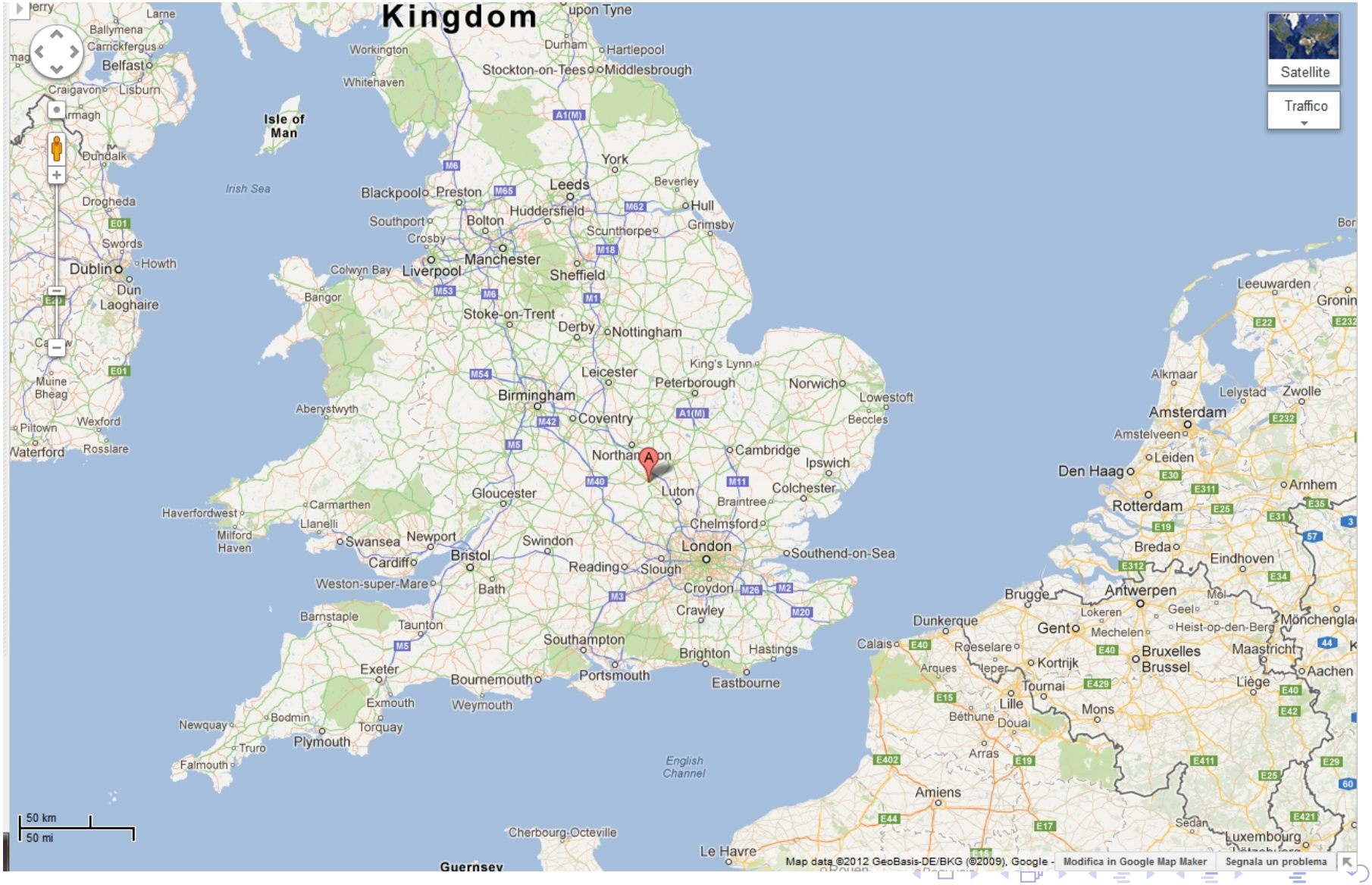
$$\binom{26}{2k} \frac{\binom{2k}{2}\binom{2k-2}{2}\cdots\binom{2}{2}}{k!}$$

Turing

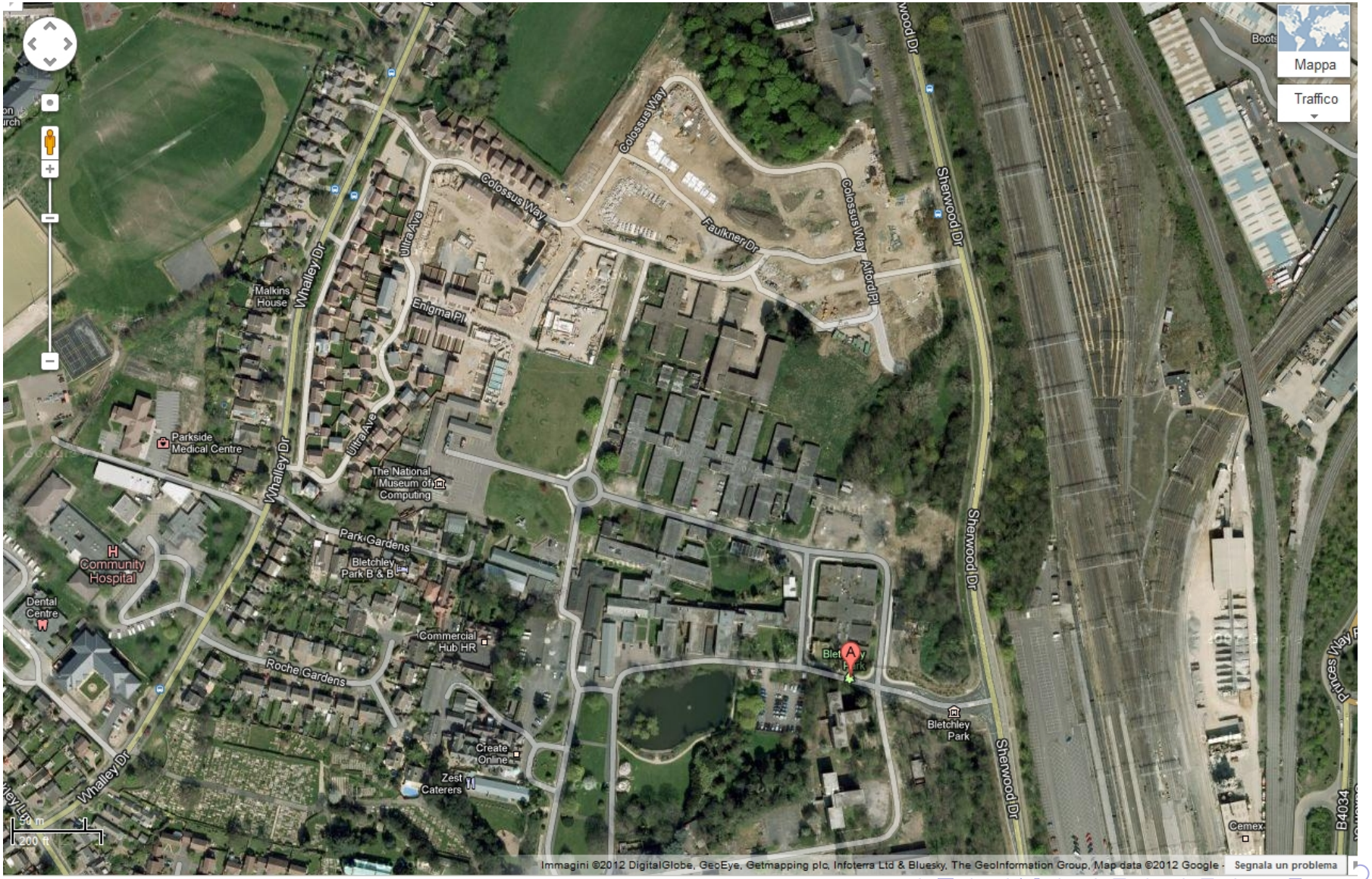
Alan Mathison Turing 23/06/1912–07/06/1954

- Nel 1936 (a 24 anni) Turing introduce un modello astratto per la descrizione degli algoritmi, che usa per dimostrare l'esistenza del calcolatore programmabile (universale), e ne dimostra i limiti teorici: nasce l'informatica come scienza
- Viene arruolato alla guida degli "Hackers" del controspionaggio inglese durante la guerra
- Nel 1946 correva la maratona in 2h46'3". Nel 1948 (Olimpiadi di Londra) Delfo Cabrera vinse in 2h34'51".
- (Durante e) Subito dopo la guerra partecipa alla progettazione e alla realizzazione dei primi calcolatori elettronici: nasce l'informatica come la immaginiamo ora.
- Pone le basi all'Intelligenza Artificiale
- A lui è intitolato il premio per l'Informatica (equivalente al Nobel)

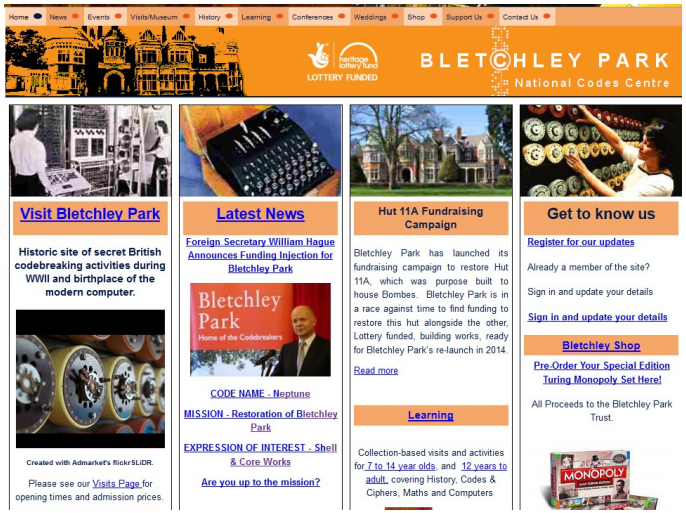
Bletchley Park



Bletchley Park

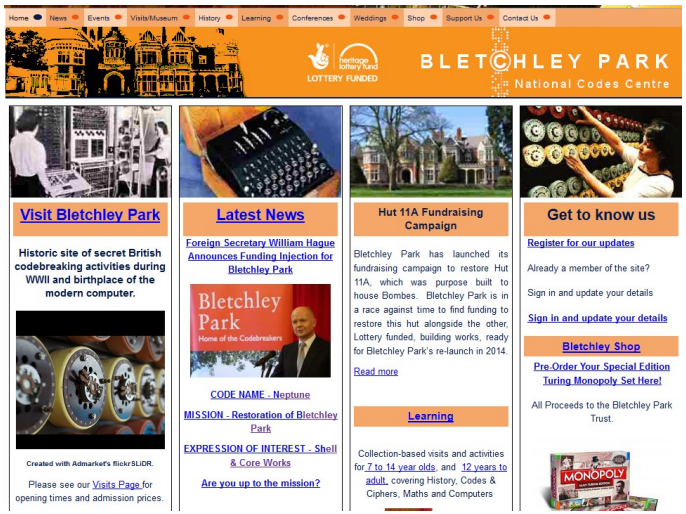


Bletchley Park



"The Government Code and Cypher School" originariamente di stanza a Londra aveva bisogno di un posto più sicuro dove lavorare e nel 1938 decise di installarsi a Bletchley Park. Nell'agosto 1939 divenne il centro operativo del controspionaggio inglese.

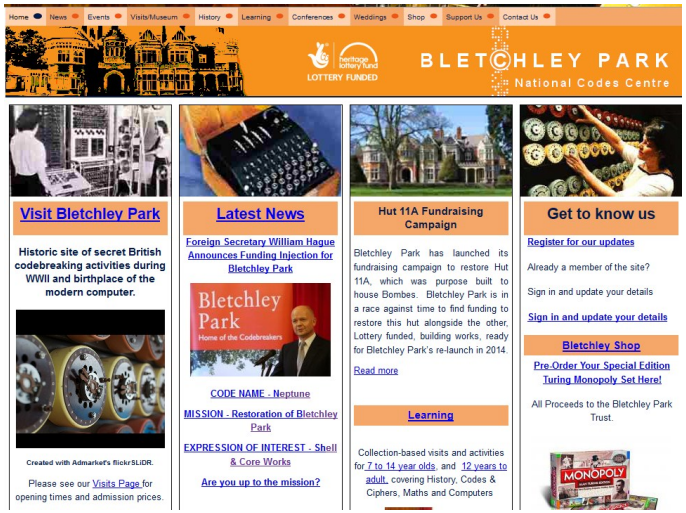
Bletchley Park



"The Government Code and Cypher School" originariamente di stanza a Londra aveva bisogno di un posto più sicuro dove lavorare e nel 1938 decise di installarsi a Bletchley Park. Nell'agosto 1939 divenne il centro operativo del controspionaggio inglese.

Ora è un museo (con qualche problema di finanziamenti)

Bletchley Park



"The Government Code and Cypher School" originariamente di stanza a Londra aveva bisogno di un posto più sicuro dove lavorare e nel 1938 decise di installarsi a Bletchley Park. Nell'agosto 1939 divenne il centro operativo del controspionaggio inglese.

Ora è un museo (con qualche problema di finanziamenti)

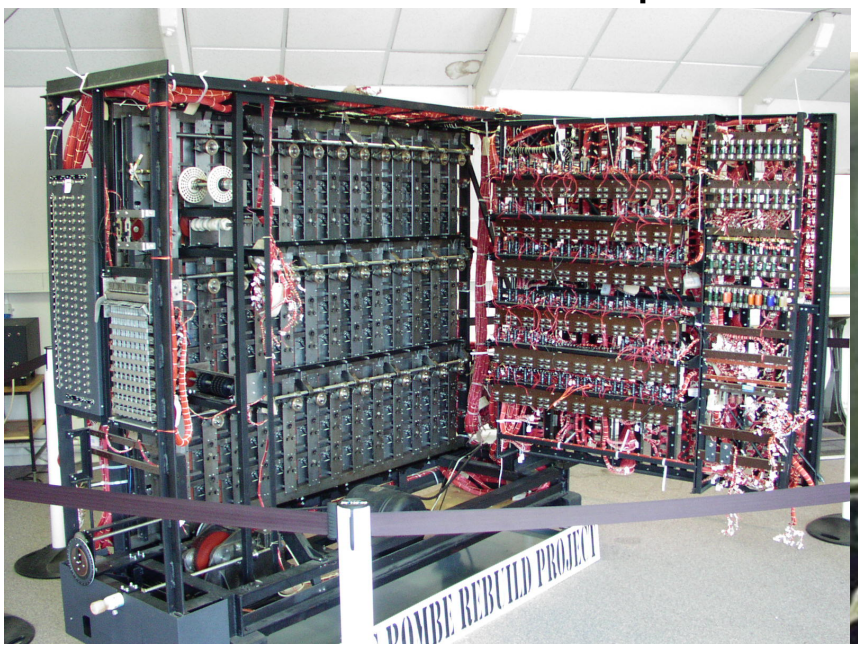
Alan Turing ebbe un ruolo cruciale nel team per la decrittazione dell'Enigma, la "macchina da cifra" impiegata dall'esercito e dalla marina tedesca.

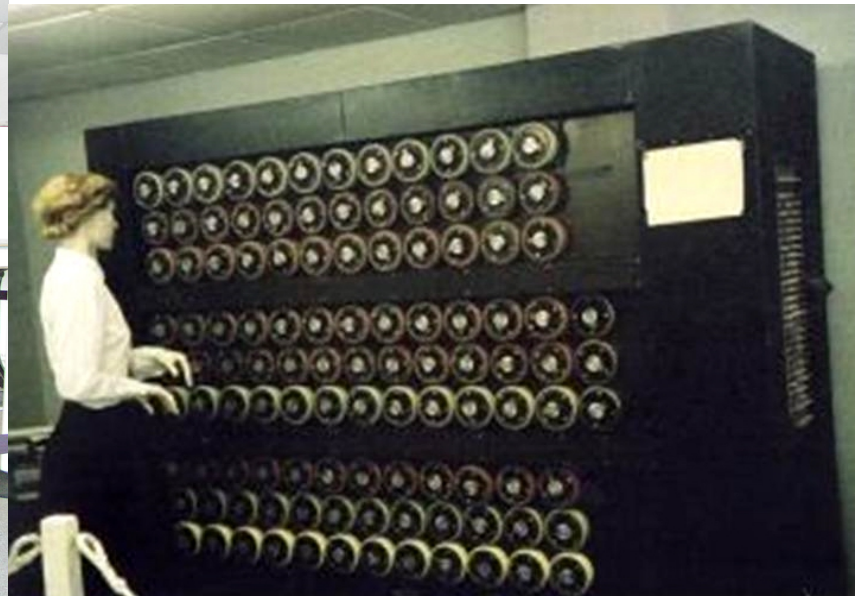
Enigma: attacco

- Le configurazioni iniziali dell'Enigma venivano comunicate segretamente e duravano brevi intervalli di tempo (p.es. plugboard: trimestrale, ordine dei rotori: mensile, carattere di partenza: giornaliero).
- All'inizio del messaggio arrivava codice (in cifra) per modificare ordine rotori.
- L'obiettivo del team di Turing era quello di indovinare rotori/cavi in plugboard/chiave iniziale.
- Furono progettate le bombe che simulavano elettromeccanicamente molti Enigma contemporaneamente.
- La forza bruta, coi numeri visti, non era sufficiente.

Enigma: attacco

Nel 1940 fu costruita la prima Bombe





Ne furono costruite 210 operate da circa 2000 Wrens (Women's Royal naval Service).

Enigma: (poche) debolezze

- Prima debolezza: una lettera non veniva mai crittata in sè stessa: se sospettiamo ci sia un certo testo (l'inizio di una lettera, la frase Keine besonderen Ereignisse—niente da segnalare) eseguiamo un allineamento e vediamo dove è possibile che ci sia. Questo ci fornisce delle informazioni verificabili con simulazione. (crib-based decryption).
- Seconda debolezza: il funzionamento generale è, sempre, simmetrico. Fissata chiave etc, se la A va in L, allora la L va in A. Questo è comodo per usare la stessa macchina per codificare e decodificare, non è una buona proprietà crittografica in quanto dà informazioni alla spia.
- Simile per le connessioni sulla plugboard (e.g., A e L vengono collegate e scambiate tra loro nell'encoding).

Enigma: (poche) debolezze

- Terza debolezza: per correggere eventuali errori di trasmissione, le posizioni iniziali dei rotori (3 caratteri) venivano ripetute due volte.
- Altre particolarità (più che debolezze) costruttive sui rotori.
- Negli anni '30 (prima della guerra) tre giovani matematici polacchi del Cipher Bureau (Marian Rejewski, Henryk Zygalski e Jerzy Różycki) studiarono a fondo le caratteristiche matematico-logiche dell'Enigma.

Enigma: (poche) debolezze

- Supponiamo di aver intercettato (oggi) 4 messaggi:

$$\begin{array}{cccccc} A & B & C & D & E & F & \dots \\ Q & W & E & R & T & Y & \dots \\ E & N & I & G & M & A & \dots \\ M & A & L & I & G & N & \dots \end{array}$$

- All'inizio vengono ripetute le posizioni iniziali dei tre rotori, diversi in ogni messaggio, ma tutti del tipo:

$$\alpha\beta\gamma\alpha\beta\gamma$$

- Dal primo messaggio so che un simbolo (che non è nè *A* nè *D*) va in *A* e in *D* nella prima e quarta posizione, dal secondo so che un simbolo va in *Q* e *R*, dal terzo in *E* e *G*, dal quarto in *M* e *I*)
- Similmente ragionando su II–V e su III–VI.
- Questo permette di avere informazioni su quali configurazioni non provare nemmeno.

Enigma: (poche) debolezze

- Rejewski et al. realizzarono (a mano) un catalogo di 105456 settings iniziali e i corrispondenti parametri quali lunghezza ciclo di permutazioni.
- Alla fine degli anni '30 le comunicarono agli inglesi.

Enigma: (poche) debolezze

- Rejewski et al. realizzarono (a mano) un catalogo di 105456 settings iniziali e i corrispondenti parametri quali lunghezza ciclo di permutazioni.
- Alla fine degli anni '30 le comunicarono agli inglesi.
- Turing mise insieme queste informazioni più altre che derivavano dalla cattura di macchine ENIGMA e di libri di utilizzo destinati agli ufficiali tedeschi, e delle tecniche sviluppate da lui per la crib-based analysis, e se ne servì per ridurre lo spazio di ricerca delle bombe, riuscendo a forzare "shark" (squalo), l'ENIGMA a 4 rotori usato dai sommergibili.

Enigma: (poche) debolezze

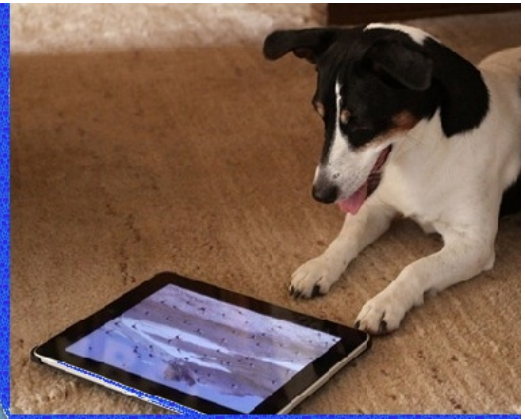
- Rejewski et al. realizzarono (a mano) un catalogo di 105456 settings iniziali e i corrispondenti parametri quali lunghezza ciclo di permutazioni.
- Alla fine degli anni '30 le comunicarono agli inglesi.
- Turing mise insieme queste informazioni più altre che derivavano dalla cattura di macchine ENIGMA e di libri di utilizzo destinati agli ufficiali tedeschi, e delle tecniche sviluppate da lui per la crib-based analysis, e se ne servì per ridurre lo spazio di ricerca delle bombe, riuscendo a forzare "shark" (squalo), l'ENIGMA a 4 rotori usato dai sommergibili.
- Furono progettati e costruiti anche i colossi, i primi calcolatori elettronici (anche se non general-purpose), tenuti segreti per molti anni.

ENIGMA

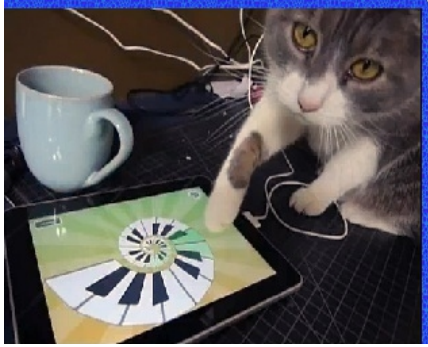
Una cartolina dal Deutsche Museum (Monaco)

Nella prima lezione i puntatori ad alcuni films (basati su libri!)

Crittografia informatica



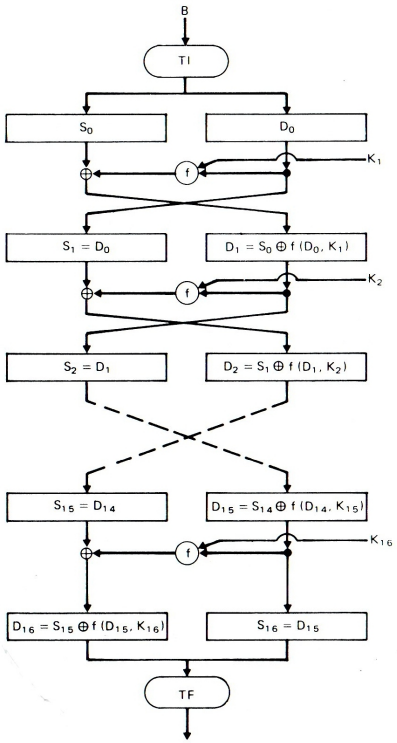



Crittografia informatica

- Si deve assumere il principio di Kerckhoffs nella sua forma più stringente: l'algoritmo per la cifrazione/decifrazione dev'essere pubblico.
- In pratica, è a disposizione di chiunque (in rete) un *codice sorgente* (p.es., un programma C) che implementa la codifica (e uno per la decodifica, eventualmente lo stesso)
- La chiave (meglio se non troppo lunga) va tenuta invece segreta.
- Il codice dev'essere attaccabile solo dalla *forza bruta* e anche mettendo assieme molti calcolatori il tempo necessario ad applicare la forza bruta dev'essere disarmante!
- Dev'essere veloce (comunicazioni riservate anche telefoniche)!

Data Encryption Standard

- 1973 il National Bureau of Standards (ora NIST) richiede un algoritmo standard di cifratura
- 1975 IBM (Horst Feistel et al) definisce il codice che nel 1976/77 diventa DES (Data Encryption Standard)
- Il funzionamento del DES è pubblico (soddisfa il principio di Kerckhoffs).
- Del DES tutto è noto tranne la chiave, costituita da 8 bytes, ove un bit per byte è un bit di controllo (desumibile dalla chiave).
- Dunque la vera lunghezza della chiave è 56 bits (spazio di $2^{56} \approx 7.2 \times 10^{16}$).



Data Encryption Standard

Qualche numero e la fine del DES

- Malgrado anni e anni di studi non si arrivò mai a metodi più intelligenti della forza bruta
- $2^{56} \approx 7.2 \cdot 10^{16}$
- Supponiamo di saper testare una chiave in $1\mu s$
- Dunque sapremmo verificare $10^6 \cdot 60 \cdot 60 \cdot 24 = 8.64 \cdot 10^{10}$ chiavi al giorno.
- Farebbero circa 834 000 giorni (2285 anni)
- Ma se avessimo 100 000 calcolatori, basterebbero 8 giorni!

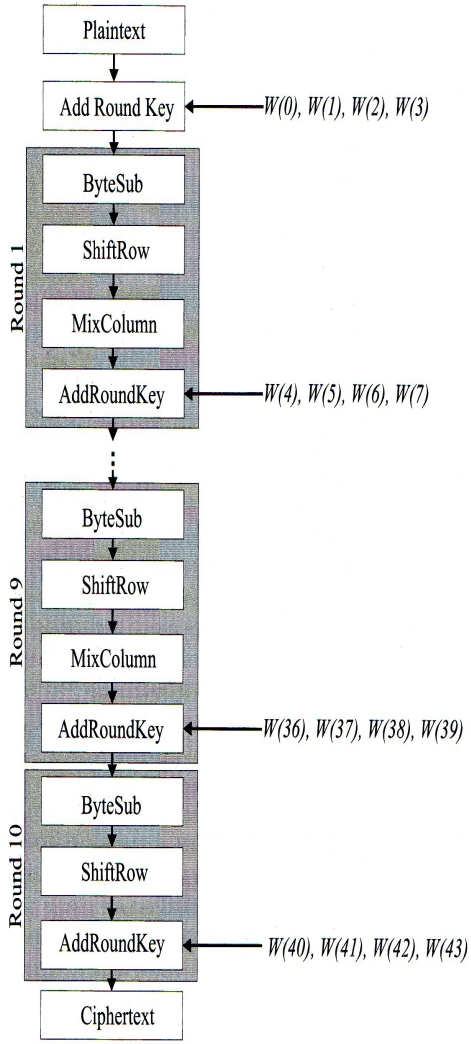
Data Encryption Standard

Qualche numero e la fine del DES

- Malgrado anni e anni di studi non si arrivò mai a metodi più intelligenti della forza bruta
- $2^{56} \approx 7.2 \cdot 10^{16}$
- Supponiamo di saper testare una chiave in $1\mu s$
- Dunque sapremmo verificare $10^6 \cdot 60 \cdot 60 \cdot 24 = 8.64 \cdot 10^{10}$ chiavi al giorno.
- Farebbero circa 834 000 giorni (2285 anni)
- Ma se avessimo 100 000 calcolatori, basterebbero 8 giorni!
- In seguito a una competizione, Rocke Verser preparò un programma in grado di distribuire il lavoro su PC vari collegati in Internet. Nel 1997 furono necessari 5 mesi (ovviamente con carico piuttosto irregolare) a forzare il DES. Nel 1998 bastarono 39 giorni.

Advanced Encryption Standard

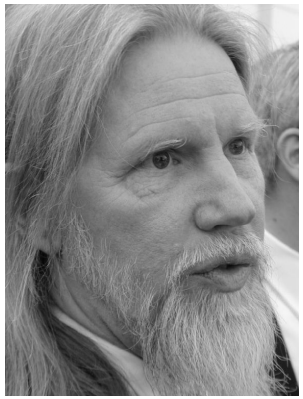
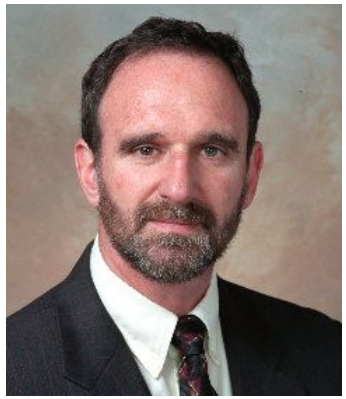
- L'algoritmo Rijndael di Joan Daemen and Vincent Rijmen fu annunciato dalla NIST come standard (AES) nel 2002



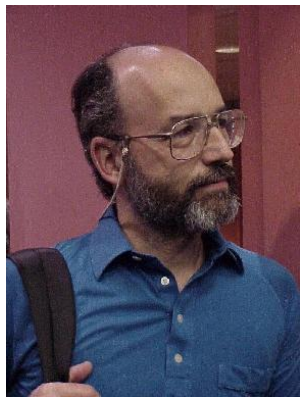
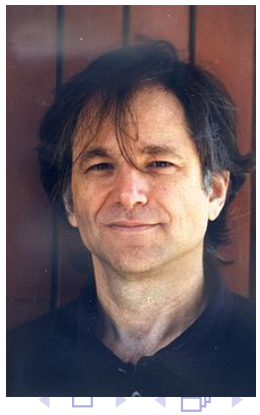
- Come per il DES, il funzionamento è totalmente pubblico.
- AES è in tre versioni differenziate dalla lunghezza della chiave:
  AES-128, AES-192 e AES-256.
- Al momento pare inattaccabile. Ma ha il problema dello scambio delle chiavi.

Crittografia a chiave pubblica

Idea “astratta” di Whitfield Diffie e Martin Hellman nel 1976 (Turing award 2015)

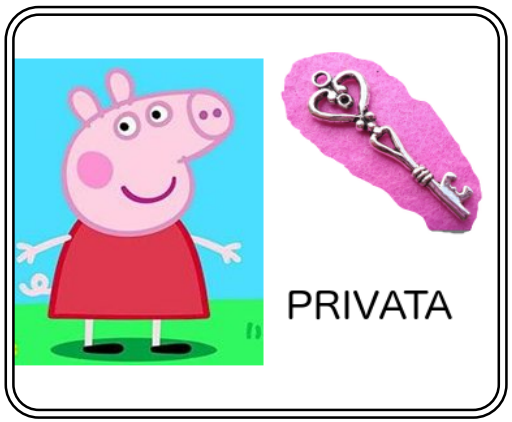



Realizzazione concreta (algoritmo “difficile”) nel 1978 di Ronald Rivest, Adi Shamir e Leonard Adleman (RSA) (Turing award 2002)

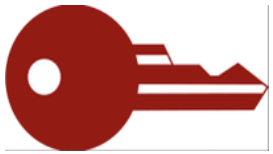
Crittografia a chiave pubblica

Peppa spedisce un messaggio a George

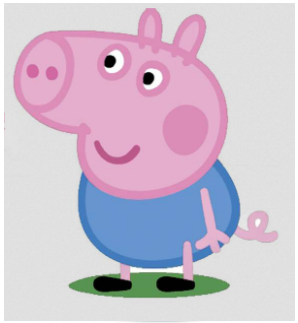


PUBBLICA

PUBBLICA

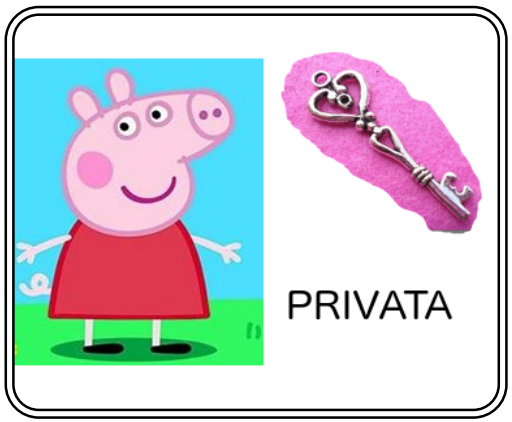


PRIVATA



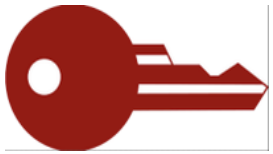
Crittografia a chiave pubblica

Peppa spedisce un messaggio a George



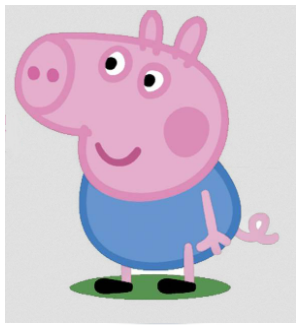
PUBBLICA

PUBBLICA



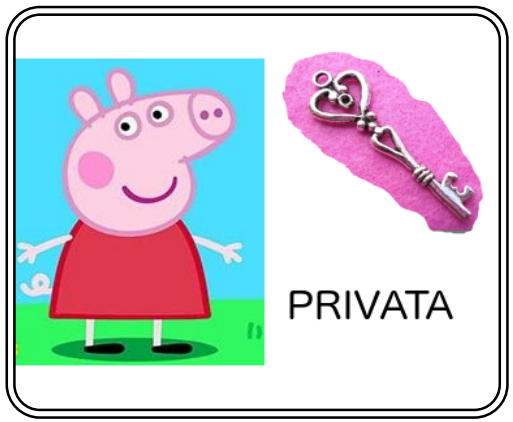
CIAO

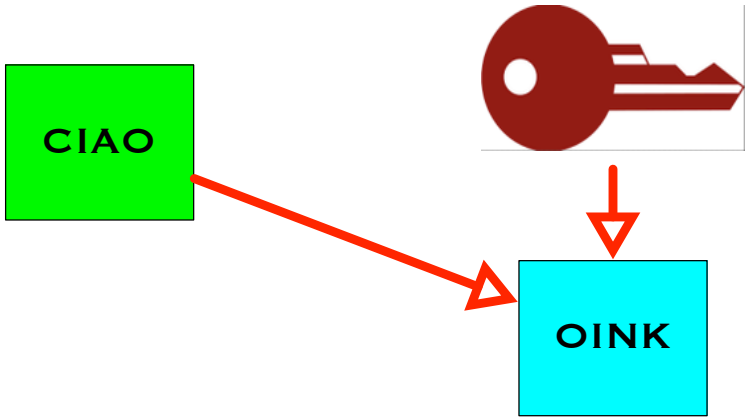
PRIVATA

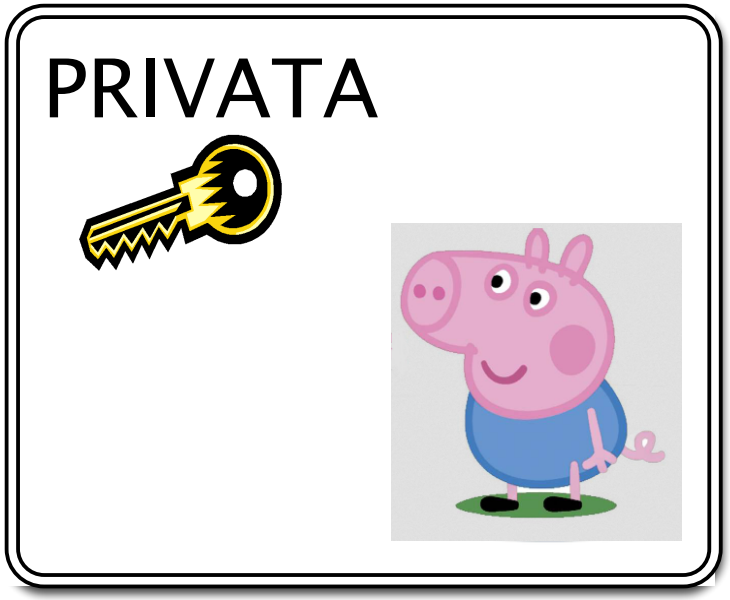


Crittografia a chiave pubblica

Peppa spedisce un messaggio a George

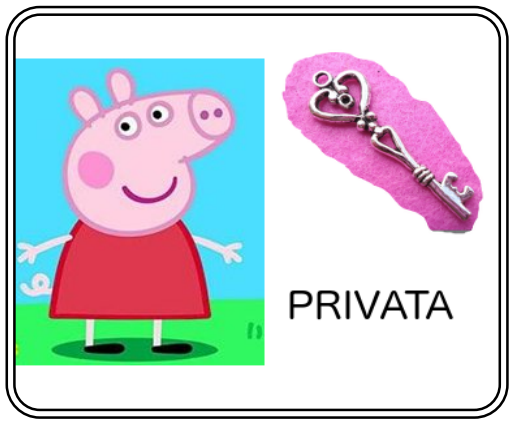


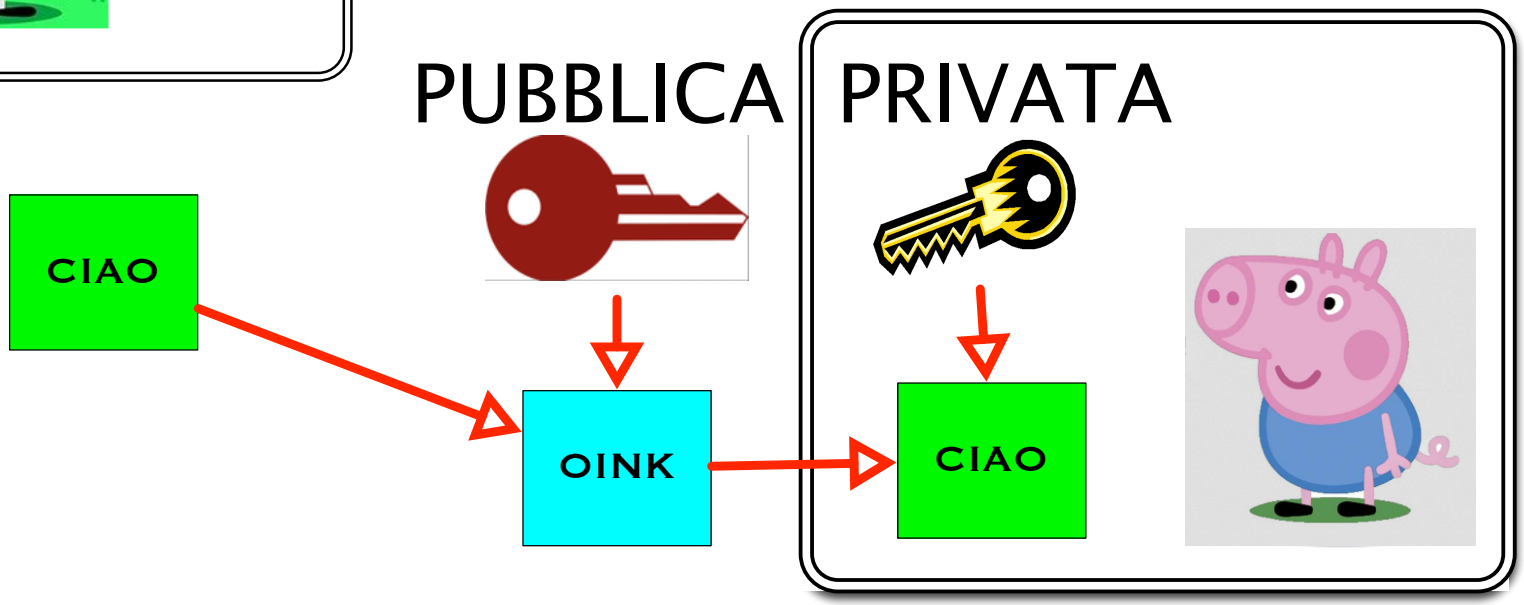




Crittografia a chiave pubblica

Peppa spedisce un messaggio a George





Crittografia a chiave pubblica

Idee generali

- L'operazione di decifrazione, sapendo la chiave privata dev'essere algoritmicamente facile
- L'operazione di decrittazione (per la spia) deve essere algoritmicamente impraticabile.
- Anche se l'impresa è possibile, avendo tempo a sufficienza: un modo è quello di provare tutti i messaggi di una data lunghezza, codificarli con la chiave pubblica di George, quando inseriamo CIAO, troveremo OINK: abbiamo capito che Peppa ha scritto CIAO.
- Così si scopre il messaggio ma non la chiave privata.

Crittografia a chiave pubblica

La sicurezza mondiale è basata su una scommessa

trova un fattore di n:
prova i da 2 a $\sqrt{n}$
se n diviso i ha resto 0
allora stampa i e finisci
(altrimenti aumenta i)

Crittografia a chiave pubblica

La sicurezza mondiale è basata su una scommessa

trova un fattore di n:
prova i da 2 a $\sqrt{n}$
se n diviso i ha resto 0
allora stampa i e finisci
(altrimenti aumenta i)

Nel caso peggiore compie un numero di passi pari a $\sqrt{n}$.
Se n è un numero di 100 cifre, ogni divisione mi costa $10^{-10}s$, e dispongo di 10^9 processori, mi serve tempo

$$\approx \frac{10^{-10}\sqrt{10^{100}}}{3600 \cdot 24 \cdot 365 \cdot 10^9} = 3 \cdot 10^{23}\text{ANNI}$$

Crittografia a chiave pubblica

La sicurezza mondiale è basata su una scommessa

trova un fattore di n:
  prova i da 2 a $\sqrt{n}$
    se n diviso i ha resto 0
    allora stampa i e finisci
    (altrimenti aumenta i)

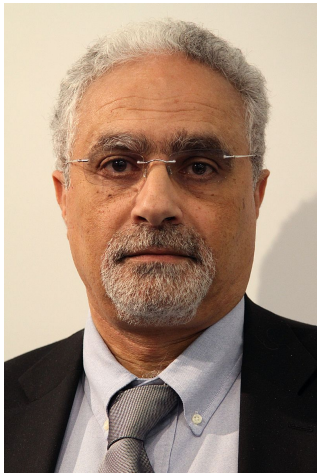
Nel caso peggiore compie un numero di passi pari a $\sqrt{n}$.
Se n è un numero di 100 cifre, ogni divisione mi costa $10^{-10}s$, e dispongo di 10^9 processori, mi serve tempo

$$\approx \frac{10^{-10}\sqrt{10^{100}}}{3600 \cdot 24 \cdot 365 \cdot 10^9} = 3 \cdot 10^{23}\text{ANNI}$$

Ma nessuno ha mai dimostrato che non si può fare in un altro modo!

Elgamal e il logaritmo discreto

- Nel 1985 Taher Elgamal architetta un altro sistema di cifratura a chiave pubblica.
- La forza del sistema di cifratura è basata sulla difficoltà computazionale di calcolare il logaritmo in un campo finito (in breve logaritmo discreto).



Elgamal e il logaritmo discreto

Operazioni nei campi finiti

- Consideriamo la struttura algebrica $\mathbb{Z}_7 = (\{0,1,2,3,4,5,6\}, *, +, 1, 0)$
- Le operazioni si intendono in modulo a 7. Esempio

$$2+3=5, 2+6=8 \mod 7 = 1, 4*4=16 \mod 7 = 2, \ldots$$

- Ogni elemento ha un unico opposto (es. $1+6=0, 3+4=0, 2+5=0$)
- Ogni elemento ha un unico reciproco (es. $1 \cdot 1 = 1, 2 \cdot 4 = 1, 3 \cdot 5 = 1, 6 \cdot 6 = 1$)
- Ora, calcolare a^b è (relativamente) semplice. Anche quando a e b sono molto grandi (numeri di centinaia di cifre).
- Se ora io vi chiedessi chi è $\log_3(5)$, ovvero il b tale che $3^b = 5$?
- Proveremmo una enumerazione di $\mathbb{Z}_7$ basata sulle potenze: $3^1 = 3, 3^2 = 2, 3^3 = 6, 3^4 = 4, 3^5 = 5$ Trovato (è un caso che sia 5)
- In $\mathbb{Z}_7$ si fa, ma in $\mathbb{Z}_p$ dove p è un numero primo di 200 cifre?

Lo scambio delle chiavi

- Cifratura e Decifratura a chiave pubblica sono efficienti (in termini di tempo di computazione per effettuarle lecitamente) ma molto meno di DES o AES
- Idea: usiamo DES/AES e usiamo invece RSA o El Gamal solo all'inizio per passarci segretamente la (piccola) chiave (per esempio PGP — Pretty Good Privacy)
- Oppure usiamo un altro algoritmo per passarci la chiave (di Diffie e Hellman, sempre basato sul logaritmo discreto)
- Oppure usiamo la cosiddetta crittografia quantistica

Conclusioni

- Ovviamente la nostra visita della crittografia è ben lungi dall'essere esaustiva
- I codici ellittici sono una variante geometrica di El Gamal.
- Se avete passione, approfondire un po' sarà facile (magari date un'occhiata ai vari MD4, MD5, WPA, WEP, etc).
- La sicurezza della comunicazione mondiale (nonché delle transizioni on-line, dei dati bancari, medici ecc.) si basa sul fatto che non sappiamo fattorizzare velocemente un numero di un centinaio di cifre o di calcolare il logaritmo discreto per numeri di quelle dimensioni.

Conclusioni

- Ovviamente la nostra visita della crittografia è ben lungi dall'essere esaustiva
- I codici ellittici sono una variante geometrica di El Gamal.
- Se avete passione, approfondire un po' sarà facile (magari date un'occhiata ai vari MD4, MD5, WPA, WEP, etc).
- La sicurezza della comunicazione mondiale (nonché delle transizioni on-line, dei dati bancari, medici ecc.) si basa sul fatto che non sappiamo fattorizzare velocemente un numero di un centinaio di cifre o di calcolare il logaritmo discreto per numeri di quelle dimensioni.
- Ma nessuno ha mai dimostrato che non si può fare!

Conclusioni

E ora tocca a voi!!!

DETTAGLI PER APPROFONDIRE

Crittografia a chiave pubblica

RSA – Generazione chiavi

1. Bob sceglie due numeri primi p e q.

Crittografia a chiave pubblica

RSA – Generazione chiavi

1. Bob sceglie due numeri primi p e q.
   - Per generare un primo:
   - Si prende un numero dispari delle dimensioni desiderate
   - Si vede se è primo
     Manindra Agrawal, Neeraj Kayal, Nitin Saxena, PRIMES is in P. Annals of Mathematics 160(2):781–793, 2004 (RR 2002)
   - Nota bene fin d'ora. Una cosa è stabilire se è primo, un'altra è trovare i suoi fattori se non lo è.
   - Se lo è OK, altrimenti lo si incrementa di 2 e si ripete.
   - I primi sono infiniti (e densi $\frac{\lg n}{n}$). Dopo un po' lo si trova.

Crittografia a chiave pubblica

RSA – Generazione chiavi

1. Bob sceglie due numeri primi p e q.
   - Per generare un primo:
   - Si prende un numero dispari delle dimensioni desiderate
   - Si vede se è primo
     Manindra Agrawal, Neeraj Kayal, Nitin Saxena, PRIMES is in P. Annals of Mathematics 160(2):781–793, 2004 (RR 2002)
   - Nota bene fin d'ora. Una cosa è stabilire se è primo, un'altra è trovare i suoi fattori se non lo è.
   - Se lo è OK, altrimenti lo si incrementa di 2 e si ripete.
   - I primi sono infiniti (e densi $\frac{\lg n}{n}$). Dopo un po' lo si trova.
2. Bob calcola $n = pq$

Crittografia a chiave pubblica

RSA – Generazione chiavi

1. Bob sceglie due numeri primi p e q.
   - Per generare un primo:
   - Si prende un numero dispari delle dimensioni desiderate
   - Si vede se è primo
     Manindra Agrawal, Neeraj Kayal, Nitin Saxena, PRIMES is in P. Annals of Mathematics 160(2):781–793, 2004 (RR 2002)
   - Nota bene fin d'ora. Una cosa è stabilire se è primo, un'altra è trovare i suoi fattori se non lo è.
   - Se lo è OK, altrimenti lo si incrementa di 2 e si ripete.
   - I primi sono infiniti (e densi $\frac{\lg n}{n}$). Dopo un po' lo si trova.
2. Bob calcola $n = pq$
3. Bob calcola $\Phi(n) = (p-1)(q-1)$

Crittografia a chiave pubblica

RSA – Generazione chiavi

4. Bob sceglie un altro numero (esponente) e tale che

$$MCD(e, \Phi(n)) = 1$$

   - Sceglie un numero dispari delle dimensioni opportune
   - Esegue l'algoritmo di Euclide tra e e $\Phi(n)$
   - Se l'output è 1, OK,
   - Altrimenti incrementa di due e ritenta
   - Il processo termina al più al numero primo successivo che non divide né $p-1$ né $q-1$.

5. Bob calcola d tale che $de = 1 \mod \Phi(n)$. (si può fare direttamente al passo precedente usando EE in luogo di Euclide)
6. Bob pubblica la chiave (n, e).

Crittografia a chiave pubblica

RSA – Cifrazione

1. Alice vuole inviare un messaggio a Bob, conoscendo la chiave pubblica di Bob (n, e).
2. Spezza il messaggio in blocchi (stringhe binarie) tali che la loro interpretazione come numero sia quella di un numero $m < n$ (basta fissare blocchi di lunghezza $\lfloor \lg_2 n \rfloor$).
3. Considera dunque un blocco alla volta.

Crittografia a chiave pubblica

RSA – Cifrazione

4. Alice calcola $c = m^e \mod n$
   - L'esponenziale finito è una operazione semplice algoritmicamente.
   - Sia $k = \lfloor lg_2 e \rfloor + 1$.
   - $m = m^{(2^0)}$. Calcolo

     $$m^{(2^1)}, m^{(2^2)}, m^{(2^3)}, m^{(2^4)}, \ldots, m^{(2^k)}$$

     tutti modulo n. Ciò garantisce che i numeri siano tutti $< e$ (e solo temporaneamente tra e e e^2).
   - Scrivo e in base 2 $(e_k, e_{k-1}, \cdots, e_1, e_0)$, e moltiplico tra loro (sempre in modulo e, sempre con numeri "controllati") i vari $m^{(2^i)}$ tali che $e_i = 1$
5. Alice spedisce c a Bob.

Crittografia a chiave pubblica

RSA – Decifrazione

1. Bob riceve $c = m^e \mod n$.
2. Bob conosce d t.c. $de = 1 \mod \Phi(n)$
3. Bob calcola

$$c^d \mod n = (m^e)^d \mod n = m^{ed} \mod n$$

4. Ci possono essere due casi.
   1. $MCD(m, n) = 1$ (essendo n prodotto di due primi, questo fatto è molto probabile).
   2. $MCD(m, n) \neq 1$ (meno probabile, ma va considerato).
5. In entrambi i casi $m^{ed} \mod n = m$

Crittografia a chiave pubblica

RSA – Decrittazione

1. Charlie intercetta $c = m^e \mod n$.
2. Charlie sa che Alice l'ha inviato a Bob e conosce e e n pubblicati da Bob.
3. Al solito, in principio, può generare tutti i messaggi $m_1, m_2, \ldots, m_t$ di lunghezza opportuna (sono in numero finito, dato n), effettuare la codifica e vedere se trova c (forza bruta). È impraticabile.
4. La strada più semplice (in apparenza) è scoprire i *fattori p e q di n*. Ma anche questo non lo sappiamo fare (per ora) con algoritmi polinomiali in $\log n$.
5. Charlie, per ora, non decritta il messaggio.

Elgamal e il logaritmo discreto

- Sia p numero primo grande (100 cifre o più), sia α radice primitiva in $\mathbb{Z}_p$ (ovvero $\alpha^{p-1} = 1$ e $\alpha^j \neq 1$ per $j < p-1$).
- Bob sceglie a caso da $X_B \in \mathbb{Z}_p \setminus \{0\}$ e genera e rende nota la propria chiave pubblica $Y_B = \alpha^{X_B} \mod p$

Elgamal e il logaritmo discreto

- Sia p numero primo grande (100 cifre o più), sia α radice primitiva in $\mathbb{Z}_p$ (ovvero $\alpha^{p-1} = 1$ e $\alpha^j \neq 1$ per $j < p-1$).
- Bob sceglie a caso da $X_B \in \mathbb{Z}_p \setminus \{0\}$ e genera e rende nota la propria chiave pubblica $Y_B = \alpha^{X_B} \mod p$
- Alice vuole inviare m a Bob (ove $m < p$ — se il messaggio è lungo lo spezziamo).
- Alice sceglie a caso $k \in \mathbb{Z}_p \setminus \{0\}$. Calcola:
  1. $K_A = (Y_B)^k \mod p$
  2. $C_1 = \alpha^k \mod p$
  3. $C_2 = m \cdot K_A \mod p$
- Comunica dunque $\langle C_1, C_2 \rangle$ a Bob

Elgamal e il logaritmo discreto

- Bob riceve $\langle C_1, C_2 \rangle$
- Bob calcola $K_B = (C_1)^{X_B} \mod p$ e dunque $\hat{m} = (C_2 \cdot K_B^{-1}) \mod p$.
- Funziona?
- Notate che (tutto $\mod p$):
  $K_B = (C_1)^{X_B} = (\alpha^k)^{X_B} = (\alpha^{k \cdot X_B}) = (\alpha^{X_B})^k = Y_B^k = K_A$.
- E dunque (tutto $\mod p$):
  $\hat{m} = (C_2 \cdot K_B^{-1}) = (m \cdot K_A \cdot K_B^{-1}) = (m \cdot K_A \cdot K_A^{-1}) = m$.

Crittografia a chiave pubblica

Scambio delle chiavi — Diffie e Hellman 1976

- Alice e Bob vogliono concordare una chiave di un DES o di un AES.

Crittografia a chiave pubblica

Scambio delle chiavi — Diffie e Hellman 1976

- Alice e Bob vogliono concordare una chiave di un DES o di un AES.
- Alice e Bob scelgono assieme e pubblicano un numero primo p ed un numero $\alpha \in \{2, \ldots, p-1\}$
- Alice sceglie $A \in \mathbb{N}$ e calcola e comunica $\alpha^A \mod p$
- Bob sceglie $B \in \mathbb{N}$ e calcola e comunica $\alpha^B \mod p$
- La chiave da usarsi è $\alpha^{AB} \mod p$

Crittografia a chiave pubblica

Scambio delle chiavi — Diffie e Hellman 1976

- Alice e Bob vogliono concordare una chiave di un DES o di un AES.
- Alice e Bob scelgono assieme e pubblicano un numero primo p ed un numero $\alpha \in \{2, \ldots, p-1\}$
- Alice sceglie $A \in \mathbb{N}$ e calcola e comunica $\alpha^A \mod p$
- Bob sceglie $B \in \mathbb{N}$ e calcola e comunica $\alpha^B \mod p$
- La chiave da usarsi è $\alpha^{AB} \mod p$
- Alice sa calcolare $\alpha^{AB} \mod p = (\alpha^B)^A \mod p$
- Bob sa calcolare $\alpha^{AB} \mod p = (\alpha^A)^B \mod p$

Crittografia a chiave pubblica

Scambio delle chiavi — Diffie e Hellman 1976

- Alice e Bob vogliono concordare una chiave di un DES o di un AES.
- Alice e Bob scelgono assieme e pubblicano un numero primo p ed un numero $\alpha \in \{2, \ldots, p-1\}$
- Alice sceglie $A \in \mathbb{N}$ e calcola e comunica $\alpha^A \mod p$
- Bob sceglie $B \in \mathbb{N}$ e calcola e comunica $\alpha^B \mod p$
- La chiave da usarsi è $\alpha^{AB} \mod p$
- Alice sa calcolare $\alpha^{AB} \mod p = (\alpha^B)^A \mod p$
- Bob sa calcolare $\alpha^{AB} \mod p = (\alpha^A)^B \mod p$
- Charlie, che intercetta tutto, conosce $\alpha, p, \alpha^A, \alpha^B$, ma NON sa calcolare α^{AB}.
- Di nuovo, un modo per farlo è calcolare il logaritmo discreto.

PGP e scambio delle chiavi

- Nel 1991Phil R. Zimmerman sviluppa PGP (Pretty Good Privacy)
- Cifrari a chiave privata (AES, DES) e Chiave Pubblica (RSA, ElGamal) vengono usati in combinazione.
- Usando RSA ci si passa la chiave (corta) per l'AES, e si usa AES (più veloce) per cifrare il messaggio (lungo).

Zimmerman ebbe una grande idea, senza secondi fini. Fu sottoposto a diversi processi negli USA.