Codici segreti: la crittografia nell'era dell'informazione

Agostino Dovier

Dip di Matematica e Informatica, Univ. di Udine

Gennaio 2013

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	⇐ monetina
11010	00101	01110	00101	01111	10100	01000	01100	10100	01100	
♣	F	O	F	P	U	Q	M	U	M	

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	⇐ monetina
11010	00101	01110	00101	01111	10100	01000	01100	10100	01100	
♣	F	O	F	P	U	Q	M	U	M	

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
- 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).

Introduzione

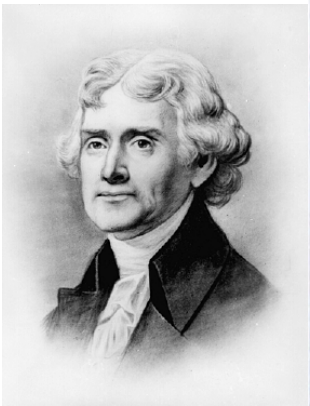
- Oggi vedremo l'automazione elettromeccanica e informatica della crittografia
- Prima studieremo la più famosa "macchina da cifra" ovvero l'ENIGMA
- Poi parleremo della crittografia informatica, sia a chiave privata (tradizionale: DES, AES) che a chiave pubblica (RSA).

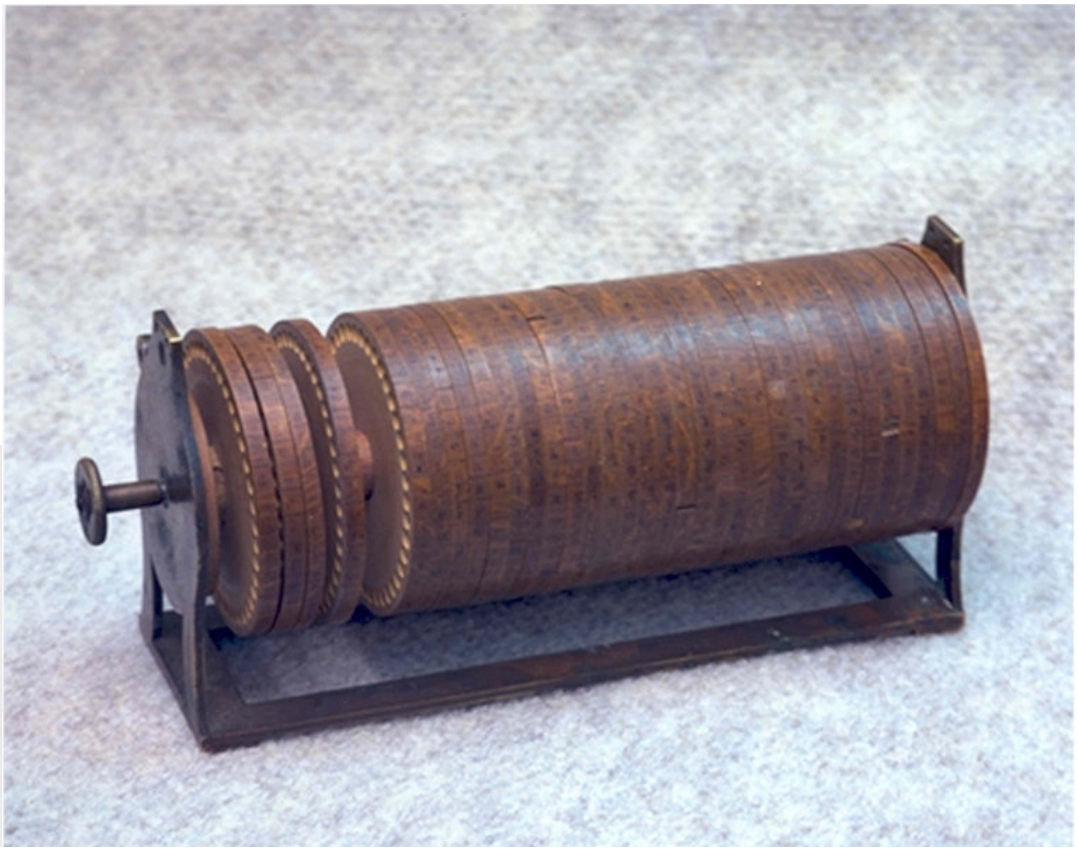
Introduzione

- 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)

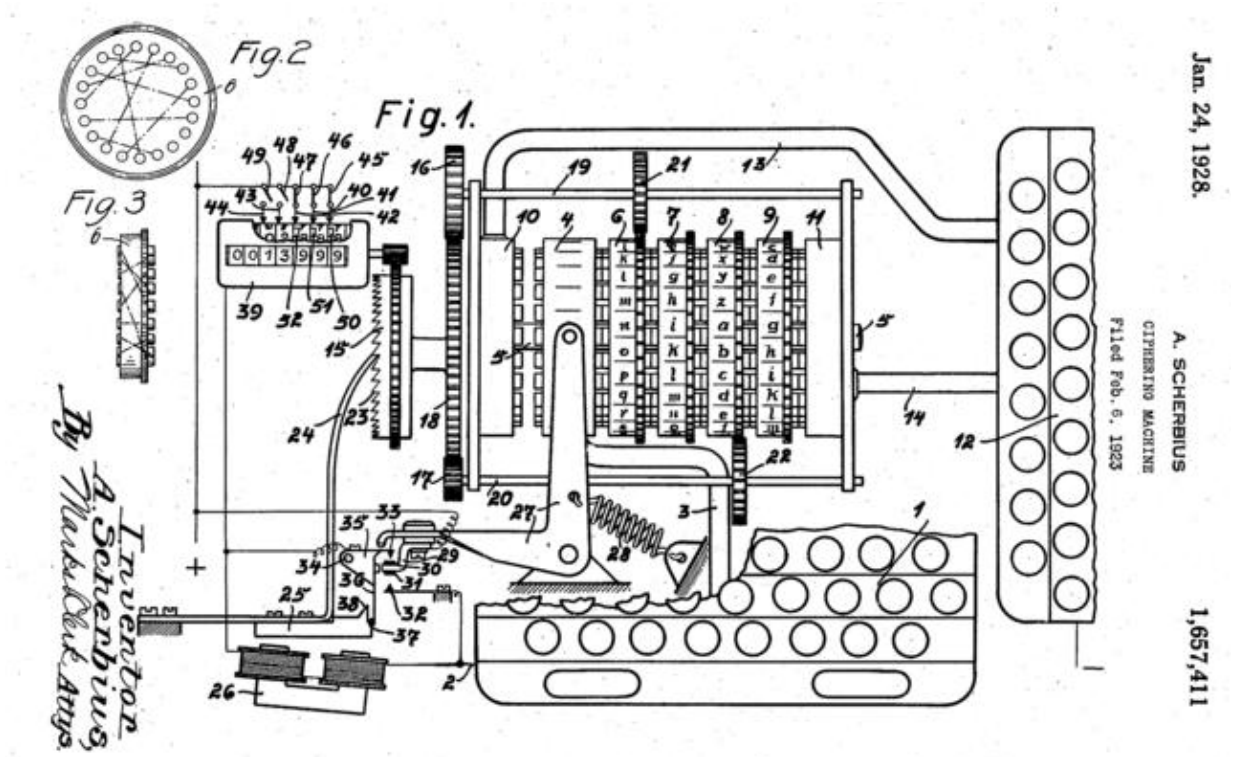




Enigma

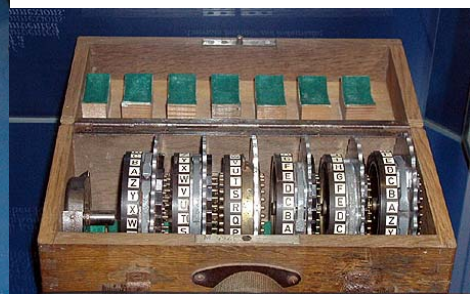
Arthur Scherbius (1878–1929)

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



Enigma

Nel 1923 Scherbius commercializza l'Enigma.



Enigma: funzionamento

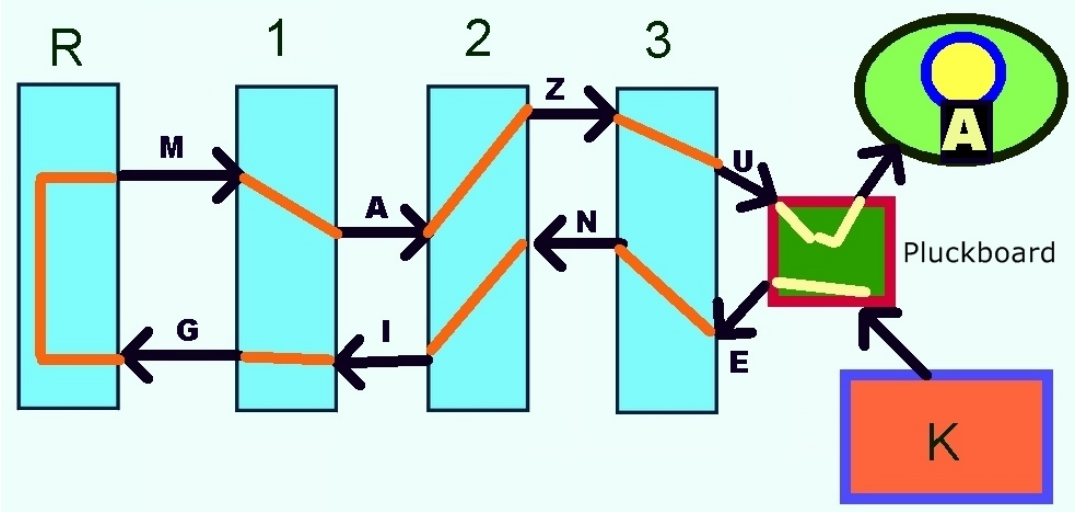
- Si tratta di un cifrario polialfabetico.
- Un punto di forza è che la lunghezza del periodo (o della chiave nel senso di Vigenère) è maggiore della lunghezza dei messaggi, il che rende apparentemente simile al one-time-pad.
- Le tecniche statistiche viste per Vigenère non si possono applicare.
- Anche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

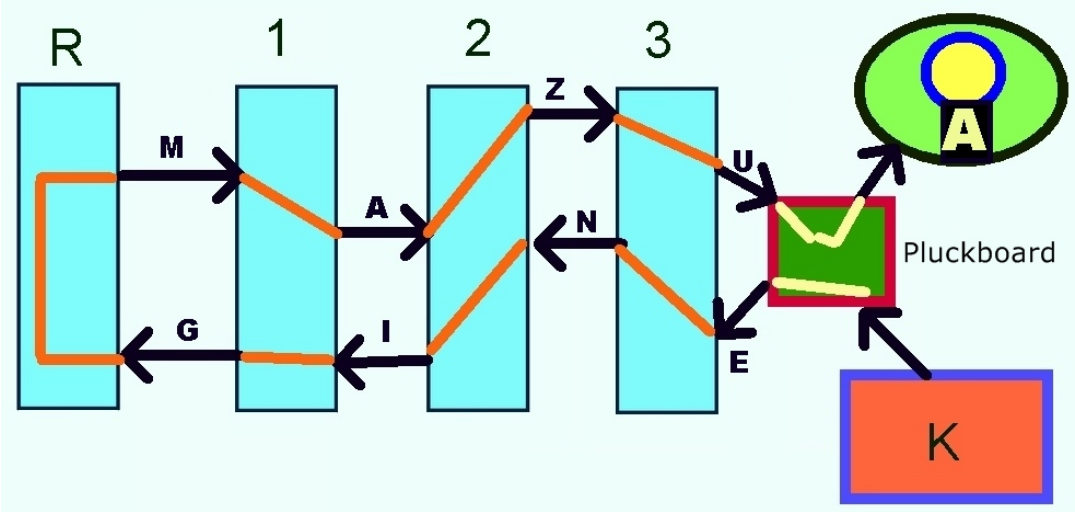


Ogni rotore ha 26 contatti su una faccia e 26 sull'altra e implementa una sostituzione monoalfabetica (completa).

Il reflector garantisce simmetria. Inoltre (brevetto) mai una lettera era codificata in sè stessa.

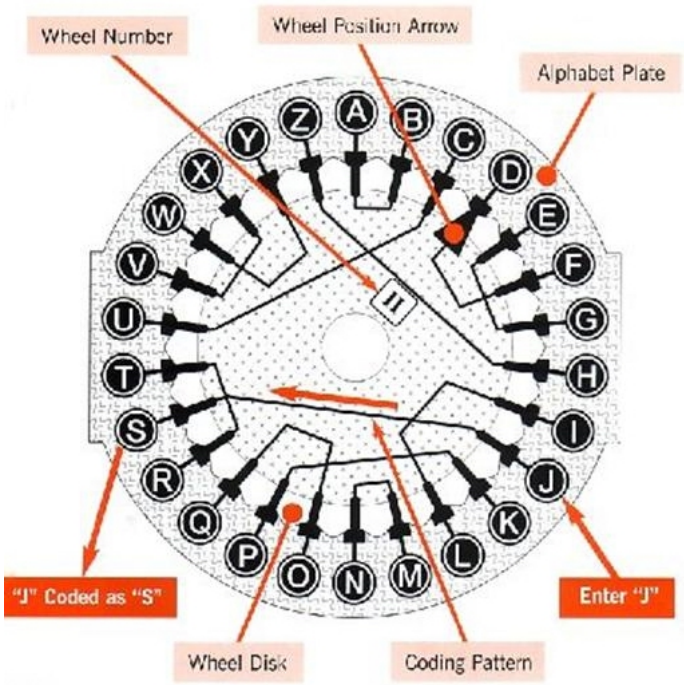
Enigma: funzionamento

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).



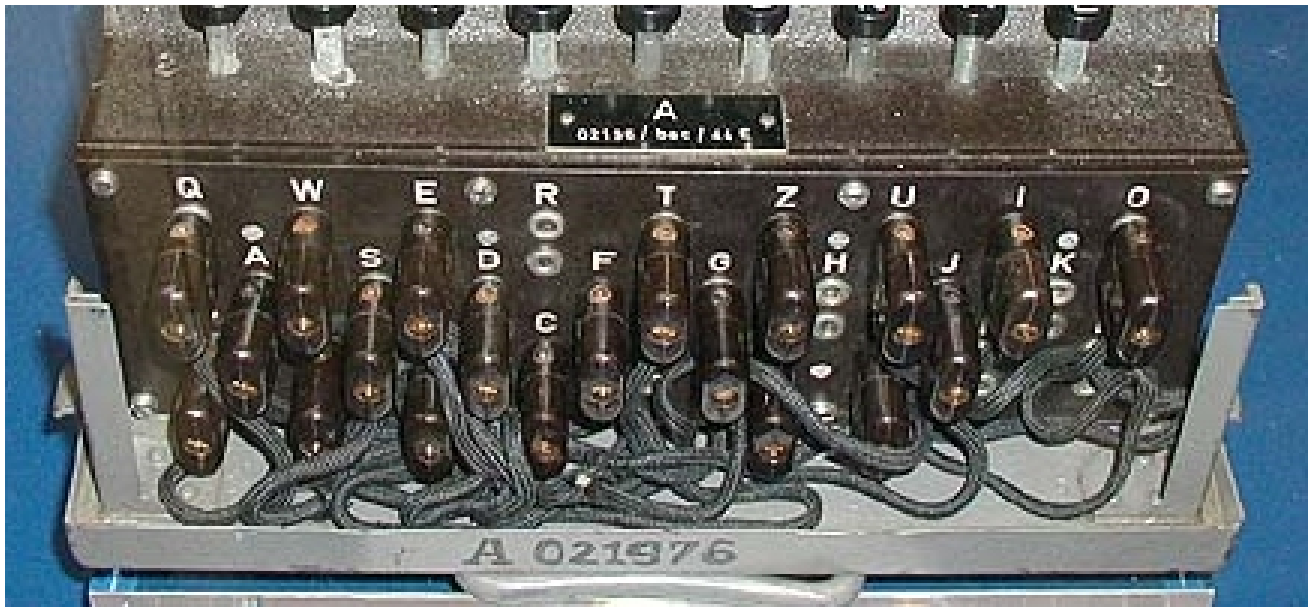
Ogni rotore ha 26 contatti su una faccia e 26 sull'altra e implementa una sostituzione monoalfabetica (completa). Il reflector garantisce simmetria. Inoltre (brevetto) mai una lettera era codificata in sè stessa.

Enigma: funzionamento



ENIGMA: FUNZIONAMENTO

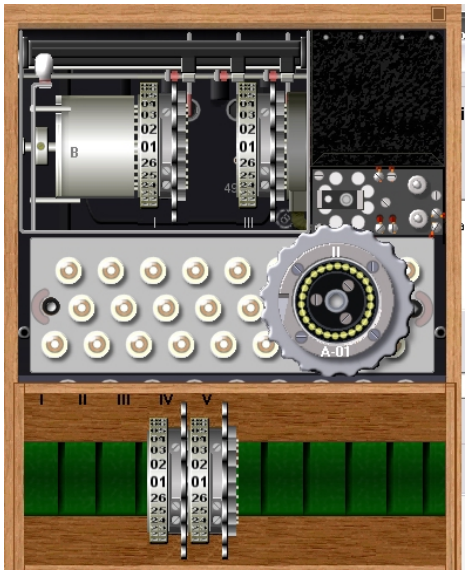
Oltre ai 3 (o 4) rotori c'era una pluckboard (pannello elettrico) che permetteva un'ulteriore (e più libera) sostituzione:



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

ATHLETICS

MARATHON AND DECATHLON CHAMPIONSHIPS

The Amateur Athletic Association championships for this year were concluded at Loughborough College Stadium, Leicestershire, on Saturday, with the second, and last, day of the Decathlon and the decision of the Marathon championship.

MARATHON CHAMPIONSHIP (26 miles 385 yds.) (record: 2hrs. 30min. 57.6sec., by H. W. Payne, Windsor to Stamford Bridge, on July 5, 1929; standard time: 3hrs. 5min.).—J. T. Holden (Tipton Harriers), 2hrs. 33min. 20 1-5sec., 1; T. Richards (South London Harriers), 2hrs. 36mni. 7sec., 2; D. McNab Robertson (Maryhill Harriers, Glasgow), 2hrs. 37min. 54 3-5sec., 3; J. E. Farrell (Maryhill Harriers), 2hrs. 39min. 46 2-5sec., 4; Dr. A. M. Turing (Walton A.C.), 2hrs. 46min. 3sec., 5; L. H. Griffiths (Reading A.C.), 2hrs. 47min. 50 2-5sec., 6.

DECATHLON CHAMPIONSHIP.—H. J. Moesgaard-Kjeldsen (Polytechnic Harriers, London), 5,965 points, 1; Captain H. Whittle (Army and Reading A.C.), 5,650, 2;

Nel 1948 (Olimpiadi di Londra) Delfo Cabrera vinse in 2h34’51”

TURING

ALAN MATHISON TURING 23/06/1912–07/06/1954

ATHLETICS

MARATHON AND DECATHLON CHAMPIONSHIPS

The Amateur Athletic Association championships for this year were concluded at Loughborough College Stadium, Leicestershire, on Saturday, with the second, and last, day of the Decathlon and the decision of the Marathon championship.

MARATHON CHAMPIONSHIP (26 miles 385 yds.) (record: 2hrs. 30min. 57.6sec., by H. W. Payne, Windsor to Stamford Bridge, on July 5, 1929; standard time: 3hrs. 5min.).—J. T. Holden (Tipton Harriers), 2hrs. 33min. 20 1-5sec., 1; T. Richards (South London Harriers), 2hrs. 36mni. 7sec., 2; D. McNab Robertson (Maryhill Harriers, Glasgow), 2hrs. 37min. 54 3-5sec., 3; J. E. Farrell (Maryhill Harriers), 2hrs. 39min. 46 2-5sec., 4; Dr. A. M. Turing (Walton A.C.), 2hrs. 46min. 3sec., 5; L. H. Griffiths (Reading A.C.), 2hrs. 47min. 50 2-5sec., 6.

DECATHLON CHAMPIONSHIP.—H. J. Moesgaard-Kjeldsen (Polytechnic Harriers, London), 5,965 points, 1; Captain H. Whittle (Army and Reading A.C.), 5,650, 2;

Nel 1948 (Olimpiadi di Londra) Delfo Cabrera vinse in 2h34’51”

TURING

ALAN MATHISON TURING 23/06/1912–07/06/1954

ATHLETICS

MARATHON AND DECATHLON CHAMPIONSHIPS

The Amateur Athletic Association championships for this year were concluded at Loughborough College Stadium, Leicestershire, on Saturday, with the second, and last, day of the Decathlon and the decision of the Marathon championship.

MARATHON CHAMPIONSHIP (26 miles 385 yds.) (record: 2hrs. 30min. 57.6sec., by H. W. Payne, Windsor to Stamford Bridge, on July 5, 1929; standard time: 3hrs. 5min.).—J. T. Holden (Tipton Harriers), 2hrs. 33min. 20 1-5sec., 1; T. Richards (South London Harriers), 2hrs. 36mni. 7sec., 2; D. McNab Robertson (Maryhill Harriers, Glasgow), 2hrs. 37min. 54 3-5sec., 3; J. F. Farrell (Maryhill Harriers), 2hrs. 39min. 46 2-5sec., 4; Dr. A. M. Turing (Walton A.C.), 2hrs. 46min. 3sec., 5; L. H. Griffiths (Reading A.C.), 2hrs. 47min. 50 2-5sec., 6.

DECATHLON CHAMPIONSHIP.—H. J. Moesgaard-Kjeldsen (Polytechnic Harriers, London), 5,965 points, 1; Captain H. Whittle (Army and Reading A.C.), 5,650, 2;

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.

TURING

ALAN MATHISON TURING 23/06/1912–07/06/1954

ATHLETICS

MARATHON AND DECATHLON CHAMPIONSHIPS

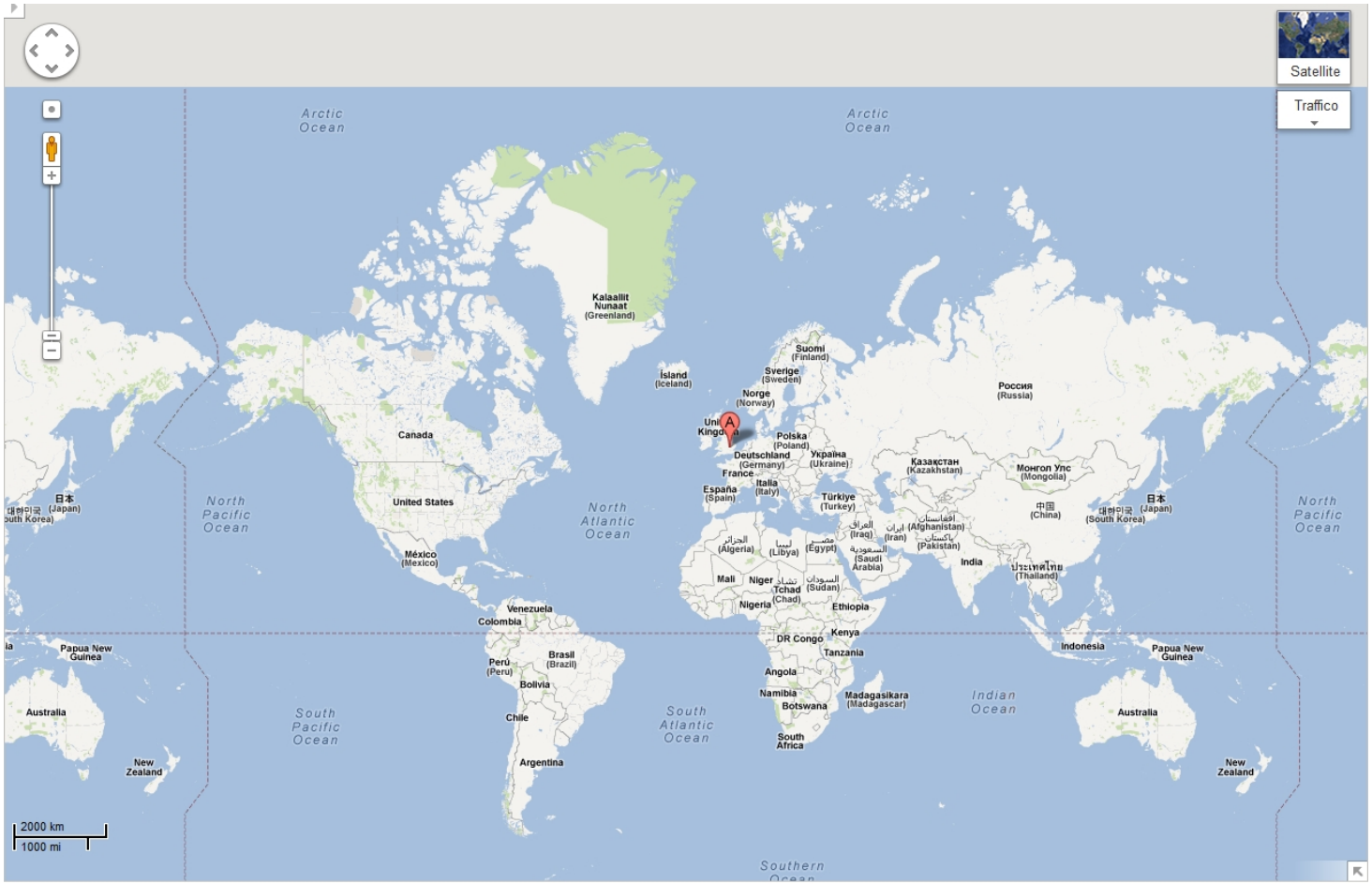
The Amateur Athletic Association championships for this year were concluded at Loughborough College Stadium, Leicestershire, on Saturday, with the second, and last, day of the Decathlon and the decision of the Marathon championship.

MARATHON CHAMPIONSHIP (26 miles 385 yds.) (record: 2hrs. 30min. 57.6sec., by H. W. Payne, Windsor to Stamford Bridge, on July 5, 1929; standard time: 3hrs. 5min.).—J. T. Holden (Tipton Harriers), 2hrs. 33min. 20 1-5sec., 1; T. Richards (South London Harriers), 2hrs. 36mni. 7sec., 2; D. McNab Robertson (Maryhill Harriers, Glasgow), 2hrs. 37min. 54 3-5sec., 3; J. E. Farrell (Maryhill Harriers), 2hrs. 39min. 46 2-5sec., 4; Dr. A. M. Turing (Walton A.C.), 2hrs. 46min. 3sec., 5; L. H. Griffiths (Reading A.C.), 2hrs. 47min. 50 2-5sec., 6.

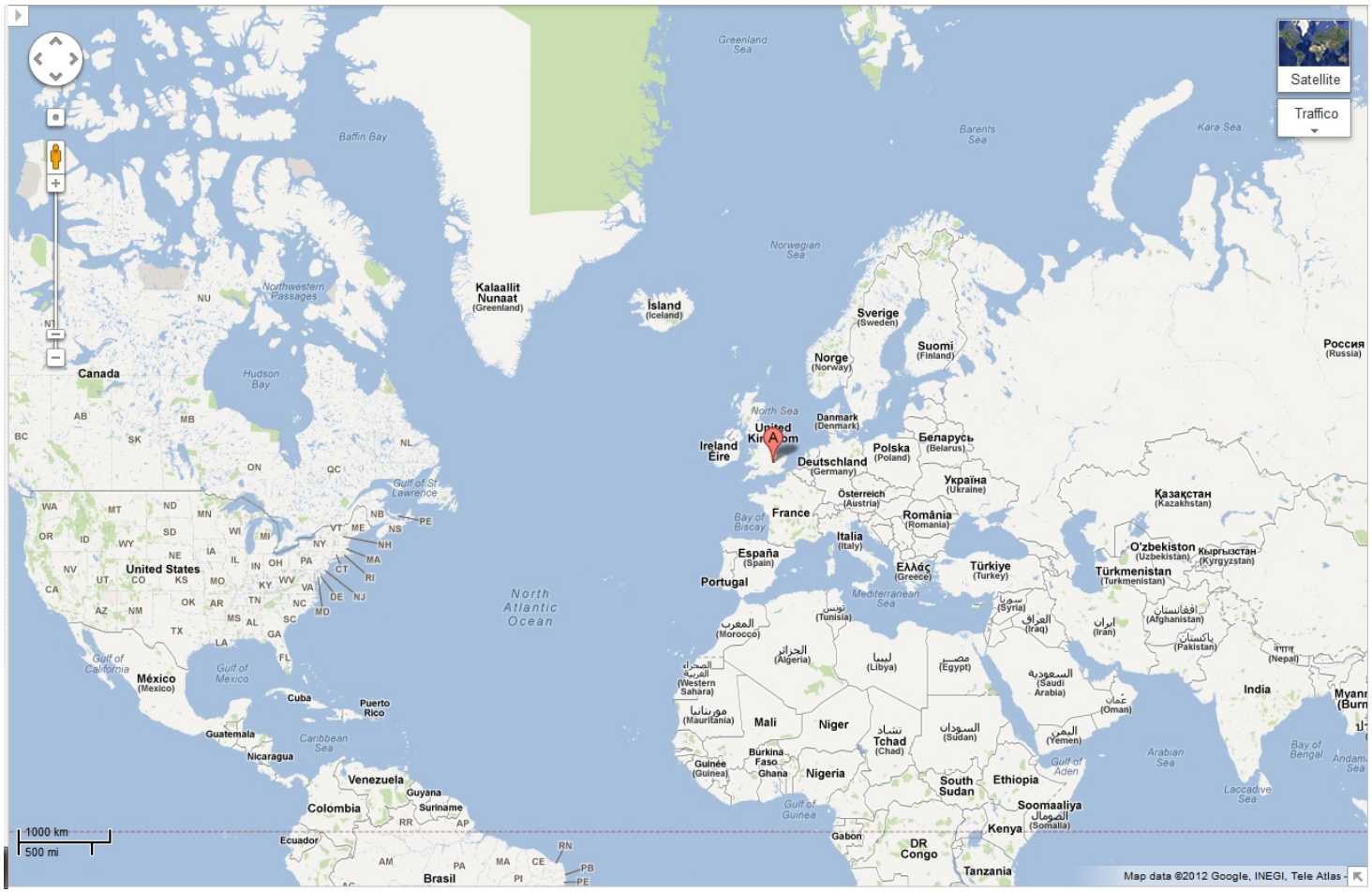
DECATHLON CHAMPIONSHIP.—H. J. Moesgaard-Kjeldsen (Polytechnic Harriers, London), 5,965 points, 1; Captain H. Whittle (Army and Reading A.C.), 5,650, 2;

(Durante e) Subito dopo la guerra partecipa alla progettazione e alla realizzazione dei primi calcolatori elettronici: nasce l'informatica come la immaginiamo ora.

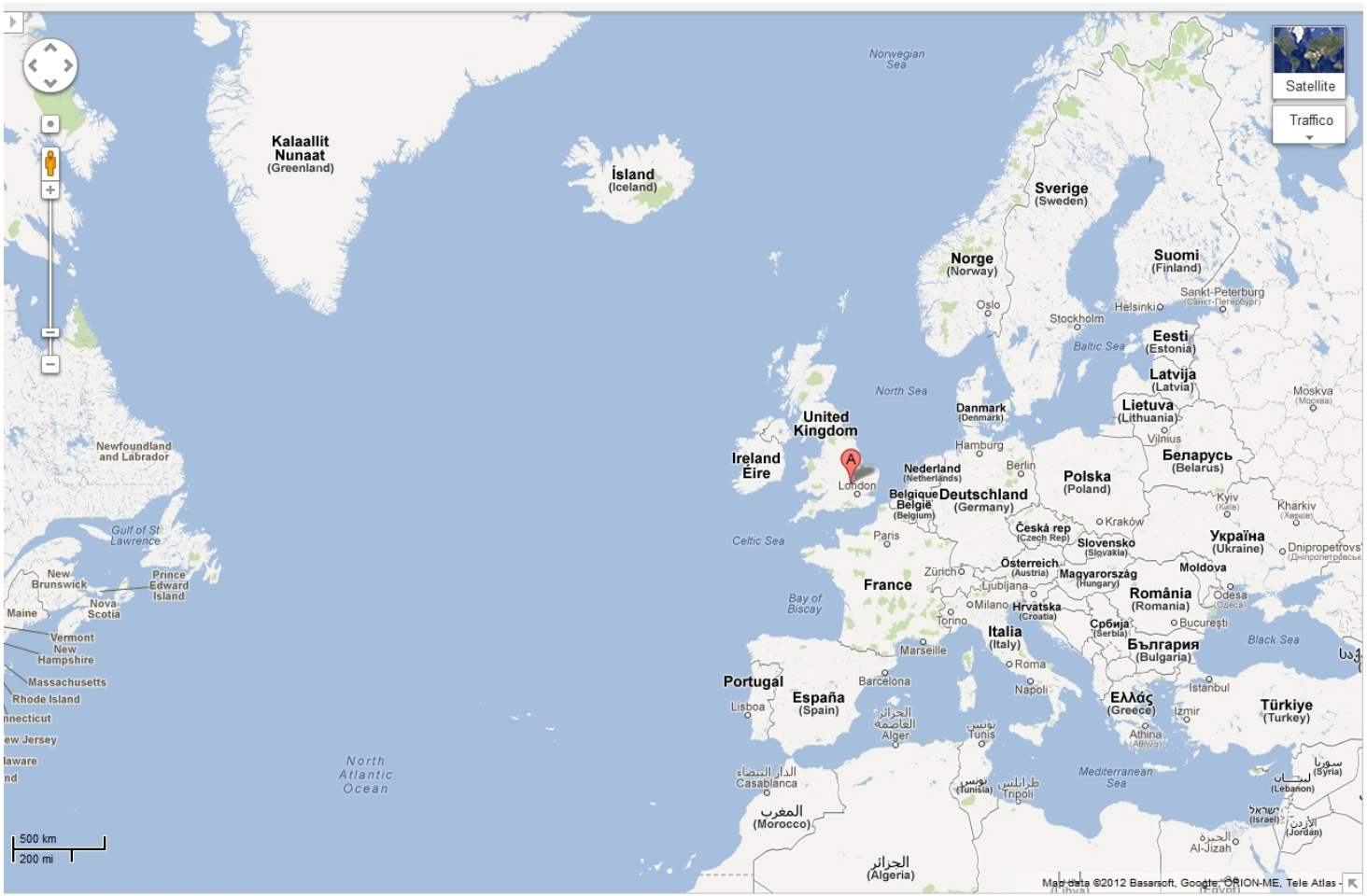
Bletchley Park



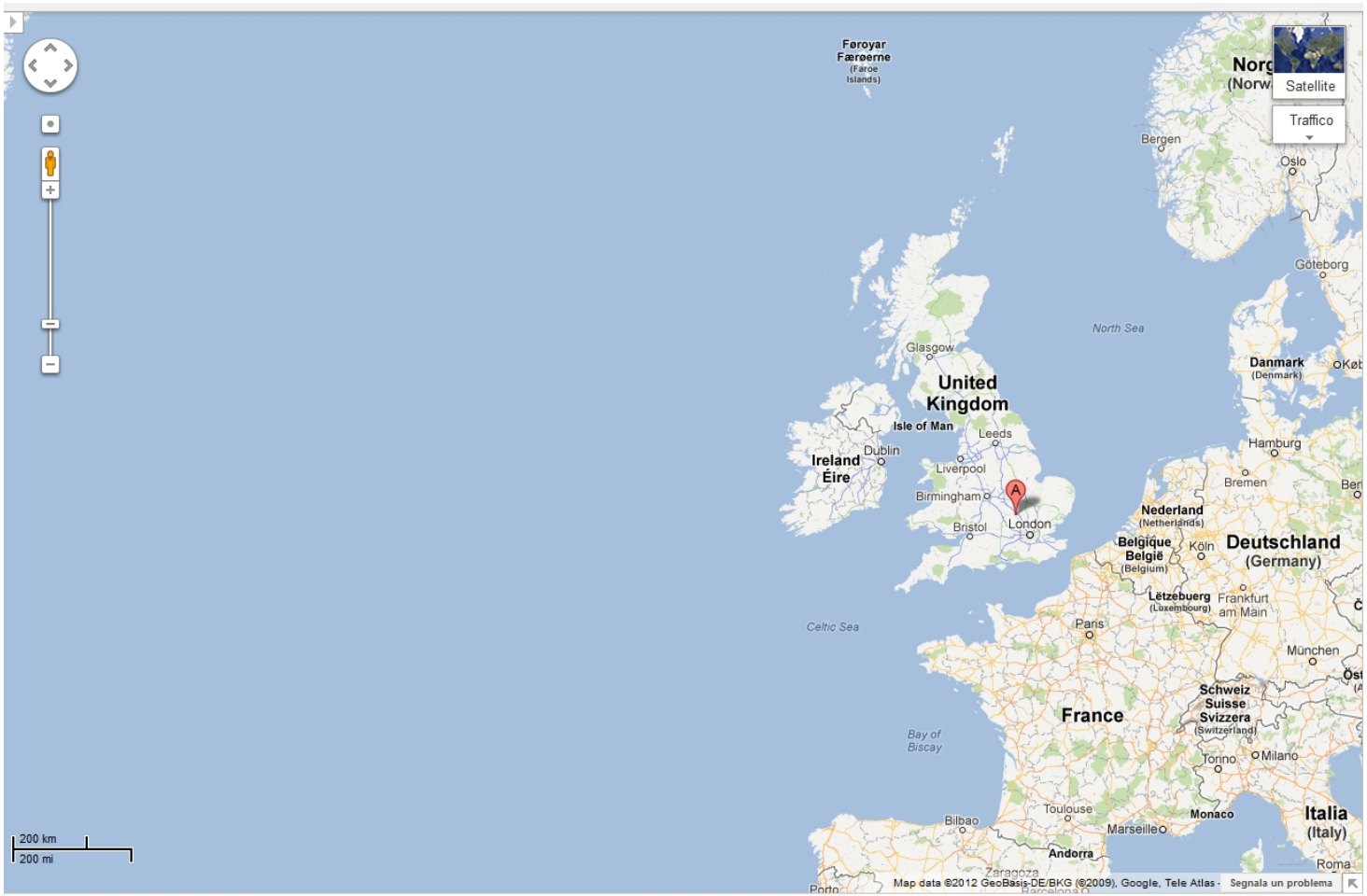
Bletchley Park



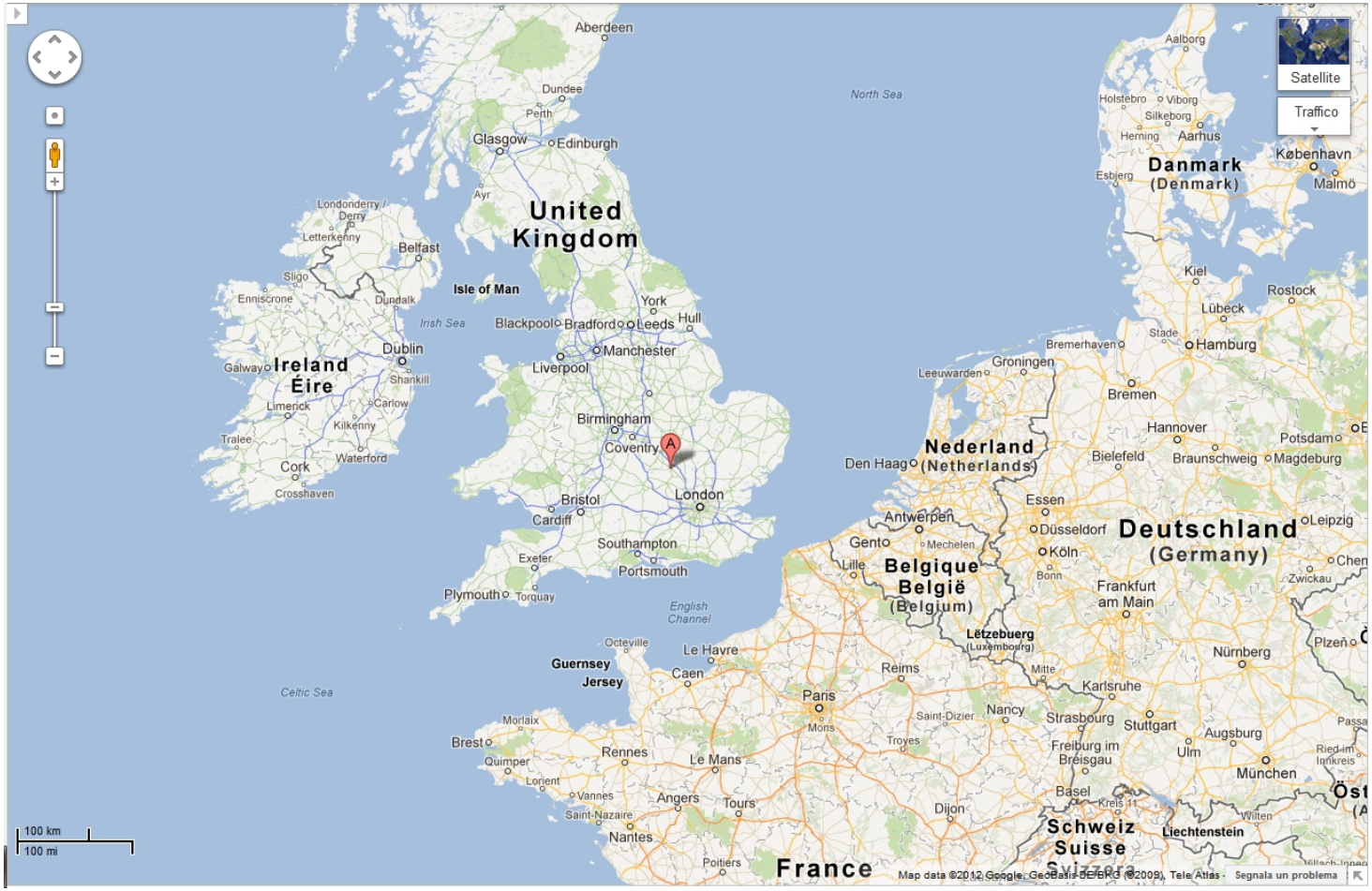
Bletchley Park



Bletchley Park

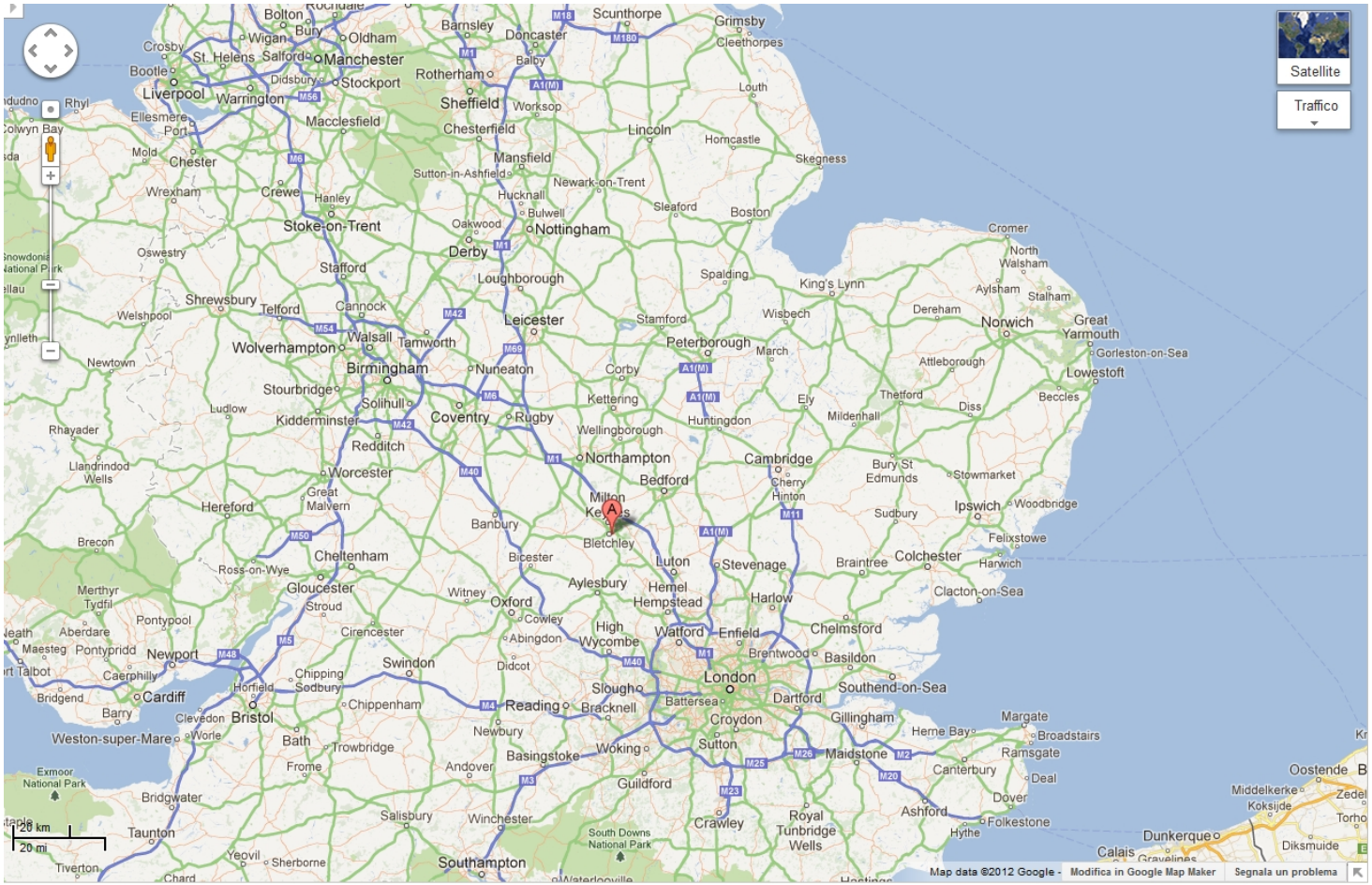


Bletchley Park

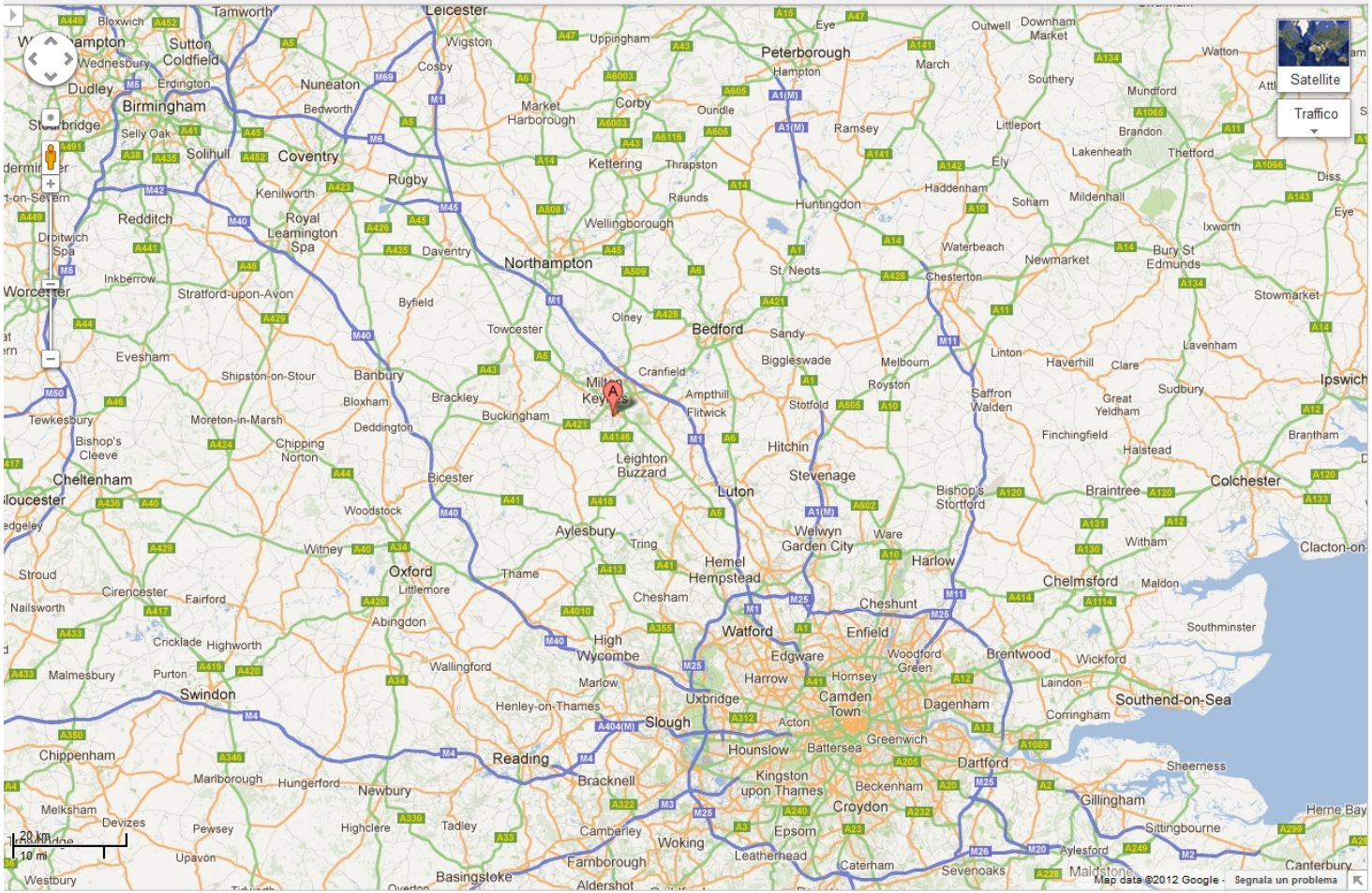


Bletchley Park

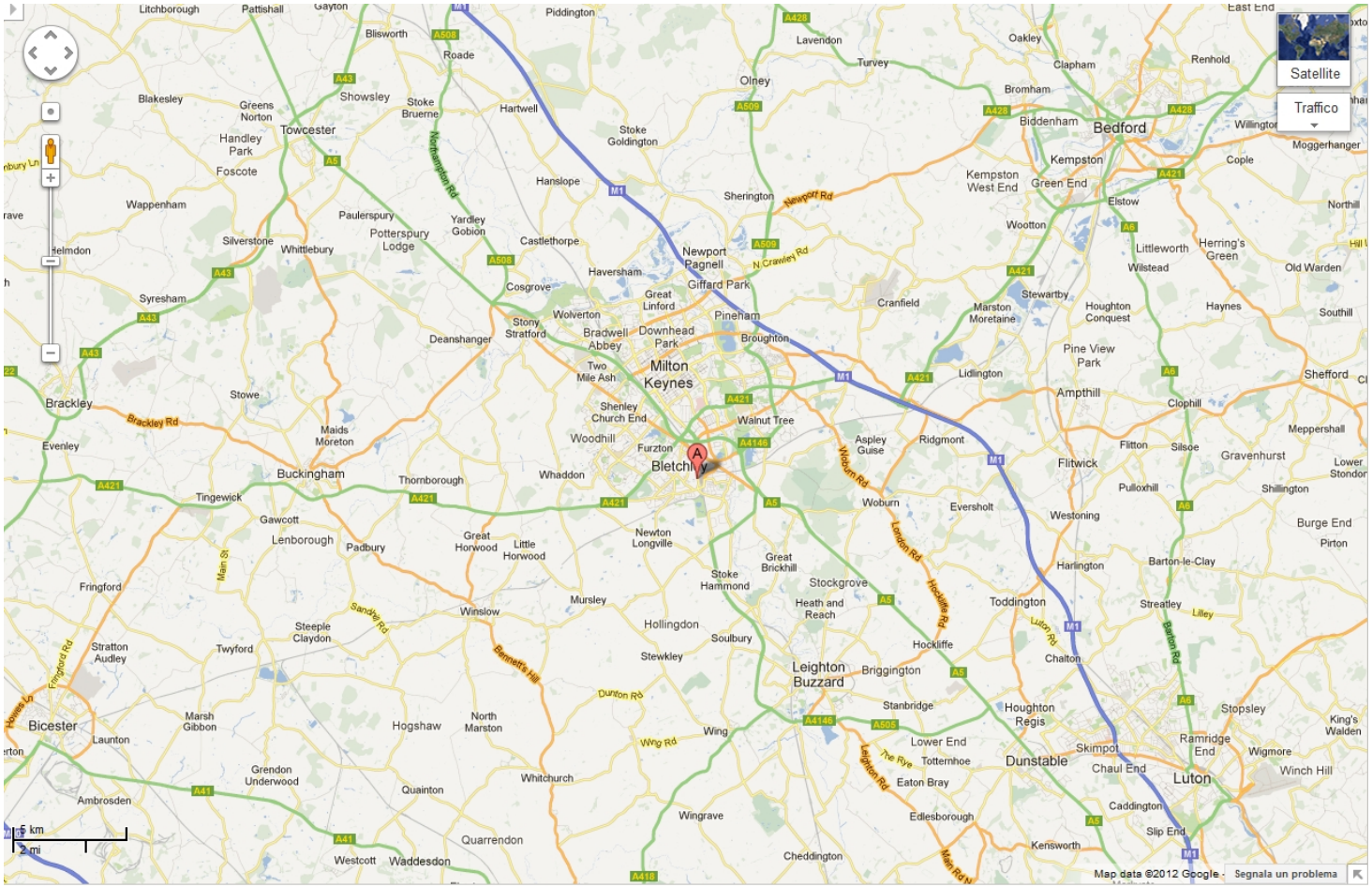
Bletchley Park



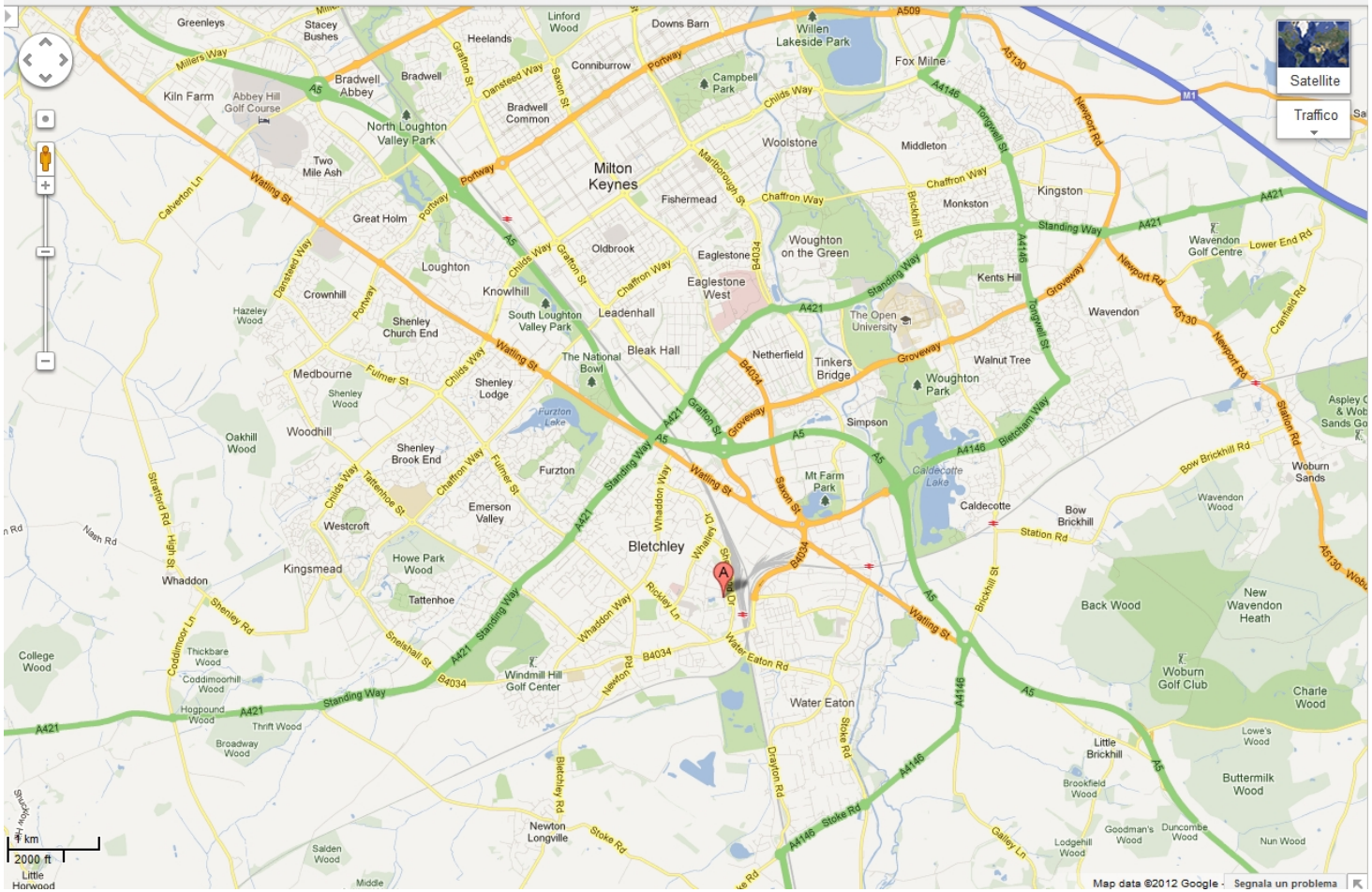
Bletchley Park



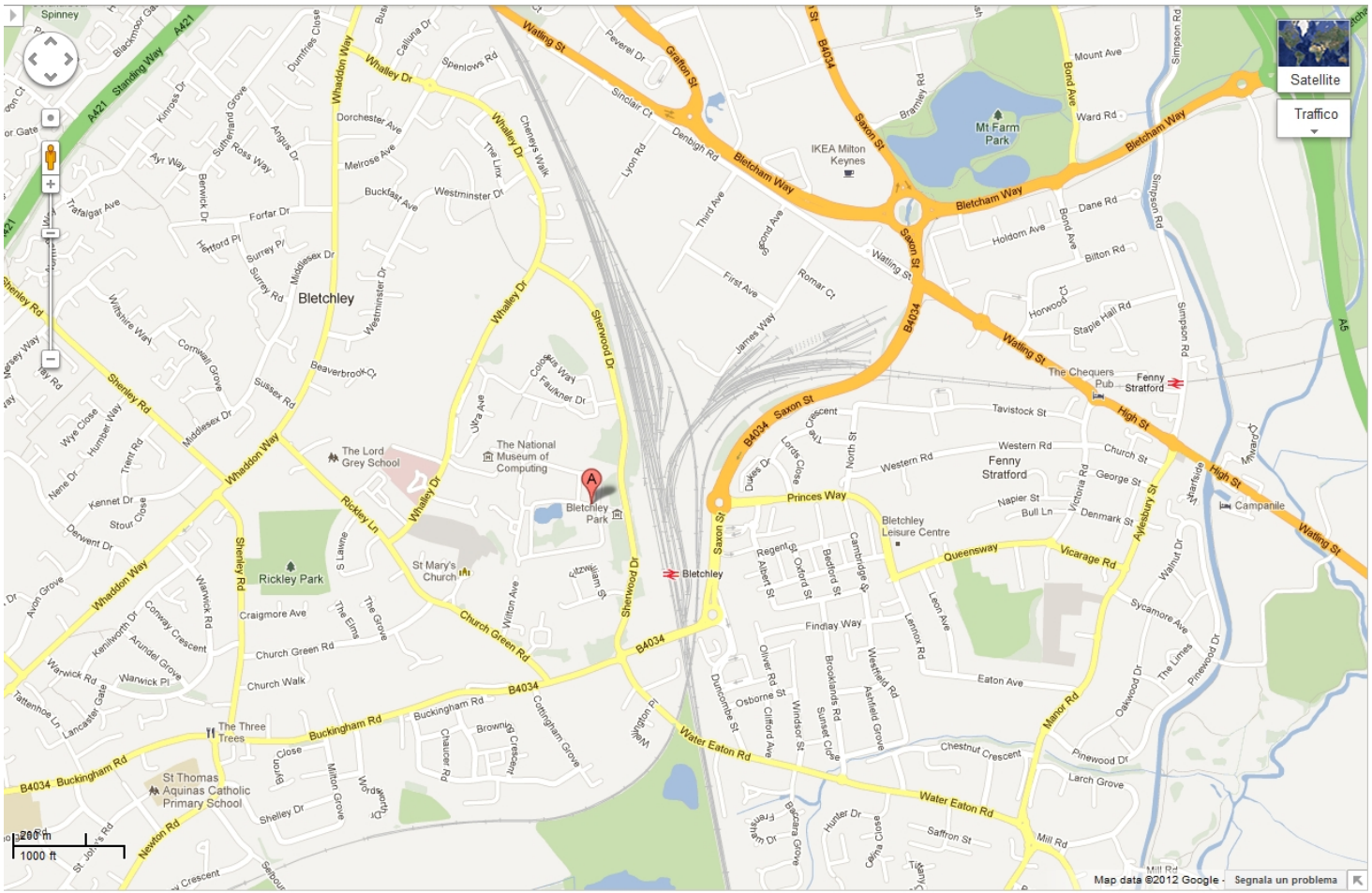
Bletchley Park



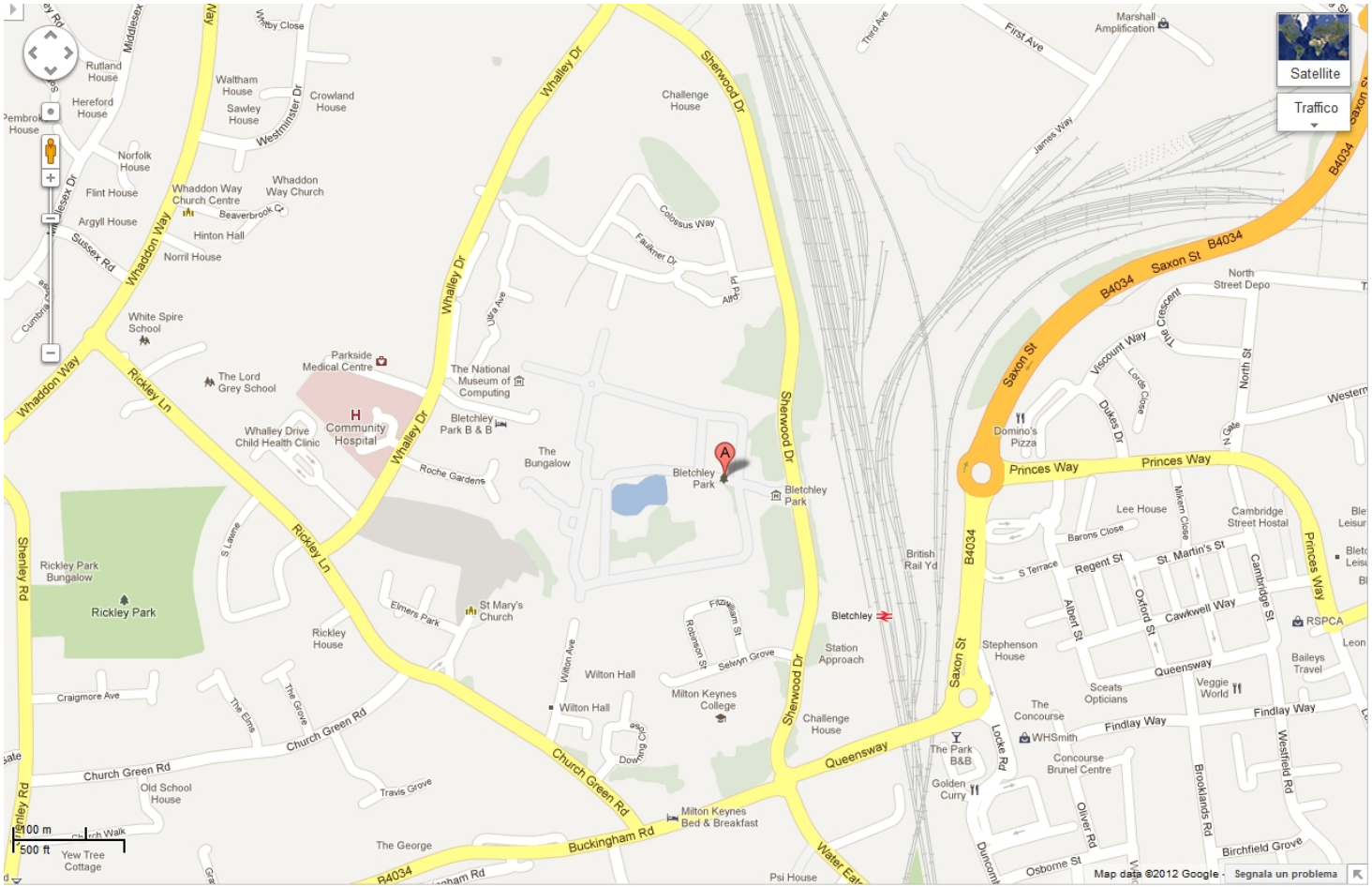
Bletchley Park



Bletchley Park

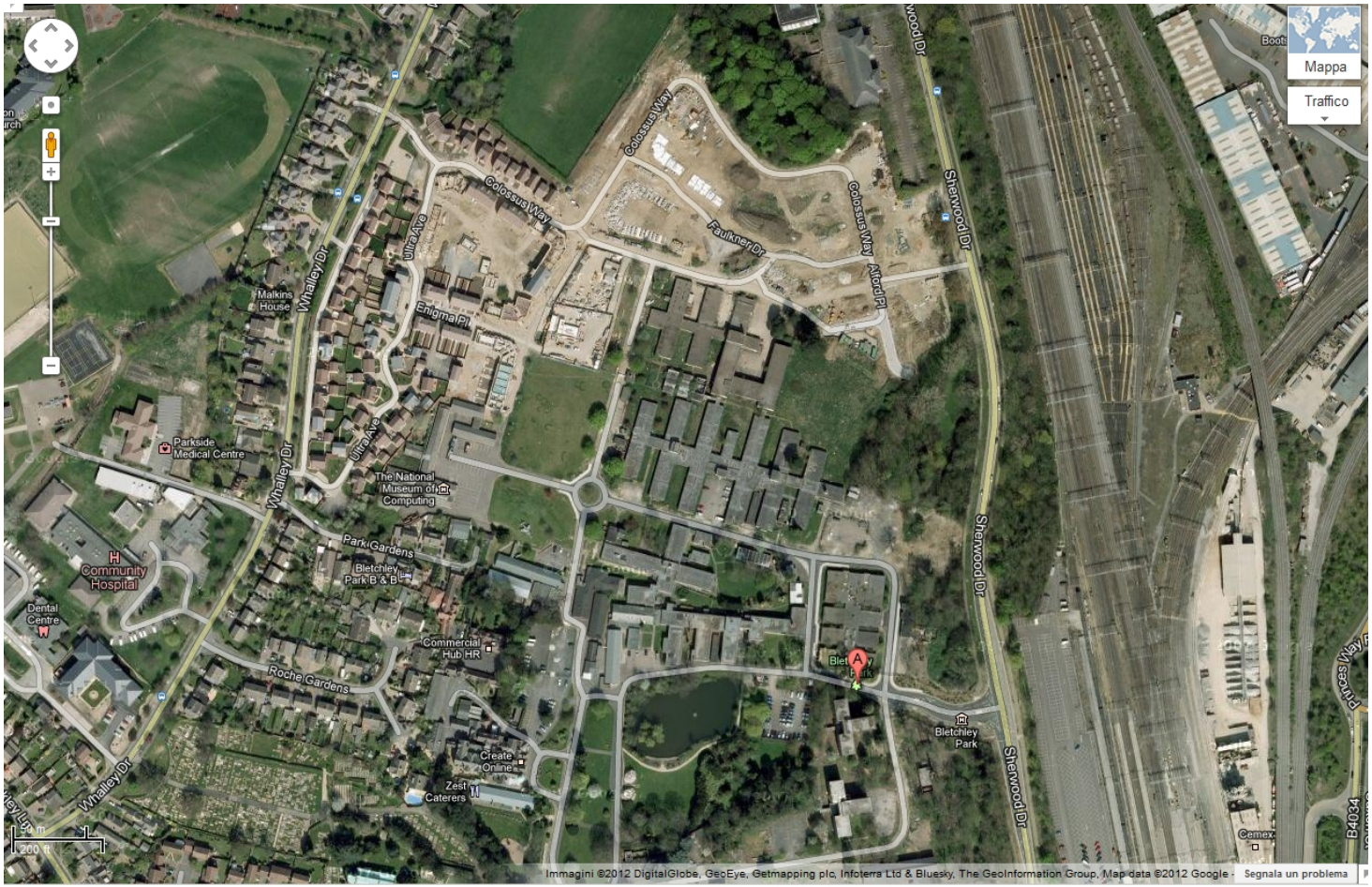


Bletchley Park

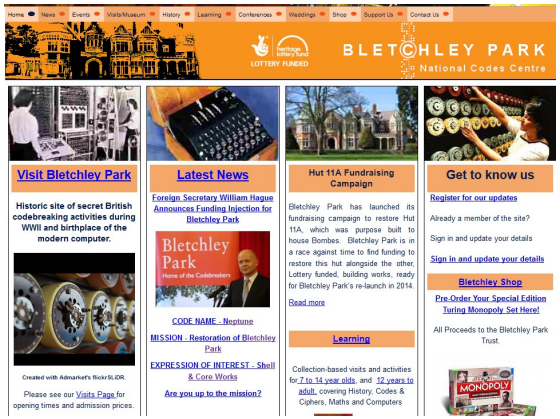


Bletchley Park

Bletchley Park



Bletchley Park

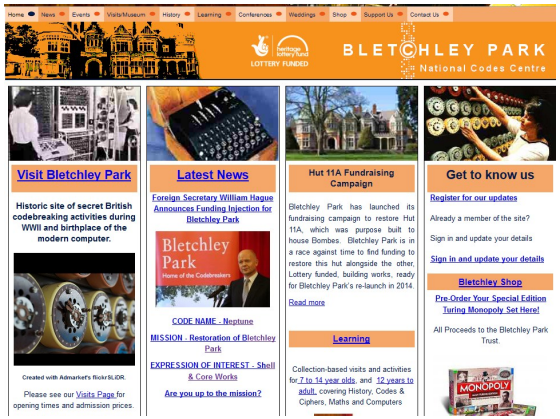


"The Government Code and Cypher School" originariamente di stanza a Londra aveva bisogno di un posto più sicuro dove lavorare e nel 1938 decise di installarsi a Bletchley Park. Nell'agosto 1939 divenne il centro operativo del controspionaggio inglese.

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.

Bletchley Park

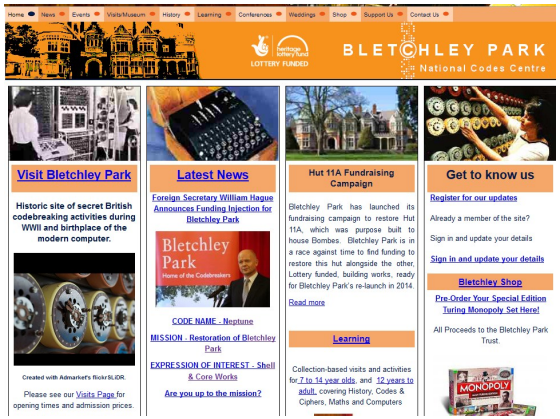


"The Government Code and Cypher School" originariamente di stanza a Londra aveva bisogno di un posto più sicuro dove lavorare e nel 1938 decise di installarsi a Bletchley Park. Nell'agosto 1939 divenne il centro operativo del controspionaggio inglese.

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.

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)

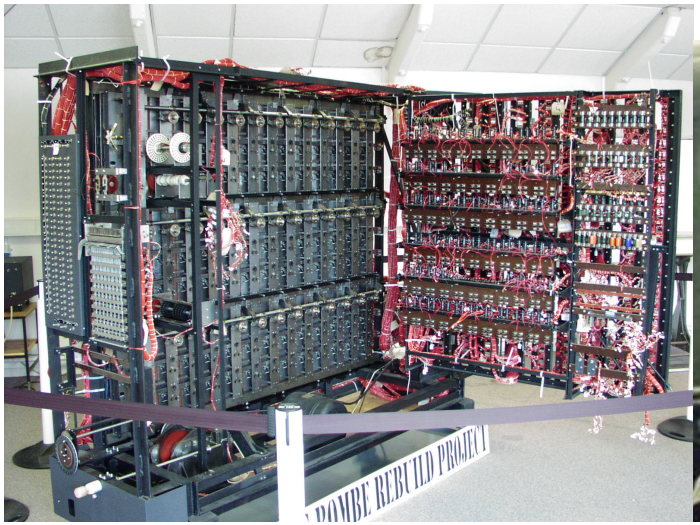
Alan Turing ebbe un ruolo cruciale nel team per la decrittazione dell'Enigma, la "macchina da cifra" impiegata dall'esercito e dalla marina tedesca.

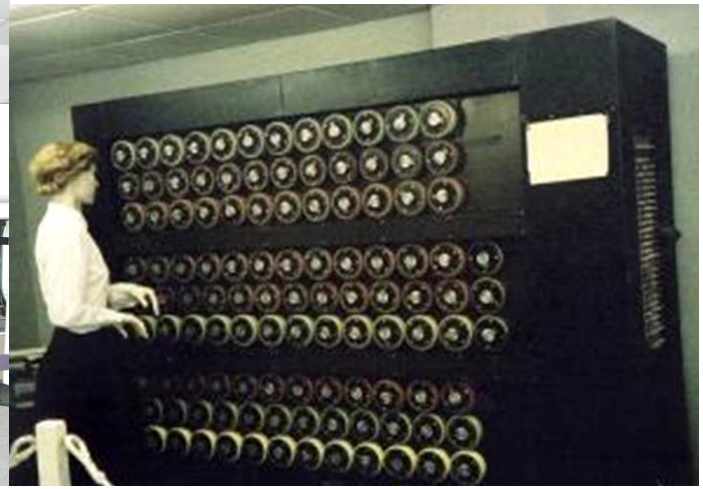
Enigma: attacco

- 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:

A	*B*	*C*	*D*	*E*	*F*	...
Q	*W*	*E*	*R*	*T*	*Y*	...
E	*N*	*I*	*G*	*M*	*A*	...
M	*A*	*L*	*I*	*G*	*N*	...

- 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.
- 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: (POCHE) DEBOLEZZE

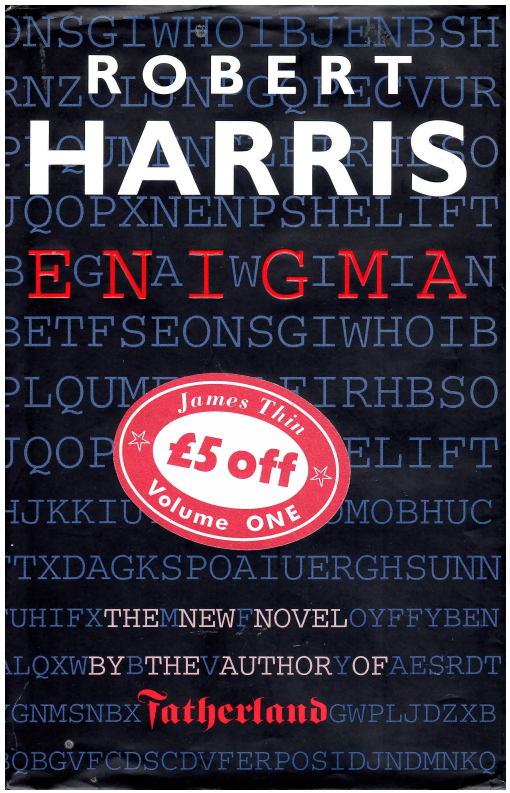
- Rejewski et al. realizzarono (a mano) un catalogo di 105456 settings iniziali e i corrispondenti parametri quali lunghezza ciclo di permutazioni.
- Alla fine degli anni '30 le comunicarono agli inglesi.
- Turing mise insieme queste informazioni più altre che derivavano dalla cattura di macchine ENIGMA 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: (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.

Robert Harris

Enigma (1995)



Robert Harris

Enigma (1995)

The staircase was just as he remembered it, except that now this wing of the college was closed and the wind had blown dead leaves into the well of the steps. An old newspaper curled itself around his legs like a hungry cat. He tried the light switch. It clicked uselessly. There was no bulb. But he could still make out the name, one of three painted on a wooden board in elegant white capitals, now cracked and faded.

TURING, A.M.

How nervously he had climbed these stairs for the first time – when? in the summer of 1938? a world ago – to find a man barely five years older than himself, as shy as a freshman, with a hank of dark hair falling across his eyes: the great Alan Turing, the author of *On Computable Numbers*, the progenitor of the Universal Computing Machine …

Turing had asked him what he proposed to take as his subject for his first year's research.

'Riemann's theory of prime numbers.'

'But I am researching Riemann myself.'

10

Robert Harris

Enigma (1995)

The staircase was just as he remembered it, except that now this wing of the college was closed and the wind had blown dead leaves into the well of the steps. An old newspaper curled itself around his legs like a hungry cat. He tried the light switch. It clicked uselessly. There was no bulb. But he could still make out the name, one of three painted on a wooden board in elegant white capitals, now cracked and faded.

TURING, A.M.

How nervously he had climbed these stairs for the first time – when? in the summer of 1938? a world ago – to find a man barely five years older than himself, as shy as a freshman, with a hank of dark hair falling across his eyes: the great Alan Turing, the author of *On Computable Numbers*, the progenitor of the Universal Computing Machine …

Turing had asked him what he proposed to take as his subject for his first year's research.

'Riemann's theory of prime numbers.'

'But I am researching Riemann myself.'

10

Robert Harris

Enigma (1995)

ENIGMA

'I know,' Jericho had blurted out, 'that's why I chose it.'

And Turing had laughed at this outrageous display of hero worship, and had agreed to supervise Jericho's research, even though he hated teaching.

Now Jericho stood on the landing and tried Turing's door. Locked, of course. The dust smeared his hand. He tried to remember how the room had looked. Squalor had been the overwhelming impression. Books, notes, letters, dirty clothes, empty bottles and tins of food had been strewn across the floor. There had been a teddy bear called Porgy on the mantelpiece above the gas fire, and a battered violin leaning in the corner, which Turing had picked up in a junk shop.

Robert Harris

Enigma (1995)

Turing had been too shy a man to get to know well. In any case, from the Christmas of 1938 he was hardly ever to be seen. He would cancel supervisions at the last minute saying he had to be in London. Or Jericho would climb these stairs and knock and there would be no reply, even though Jericho could sense he was behind the door. When, at last, around Easter 1939, not long after the Nazis had marched into Prague, the two men had finally met, Jericho had nerved himself to say: 'Look, sir, if you don't want to supervise me …'

'It's not that.'

'Or if you're making progress on the Riemann Hypothesis and you don't want to share it …'

Turing had smiled. 'Tom, I can assure you I am making no progress on Riemann whatsoever.'

'Then what …?'

'It's not Riemann.' And then he had added, very quietly: 'There are other things now happening in the world, you know, apart from mathematics …'

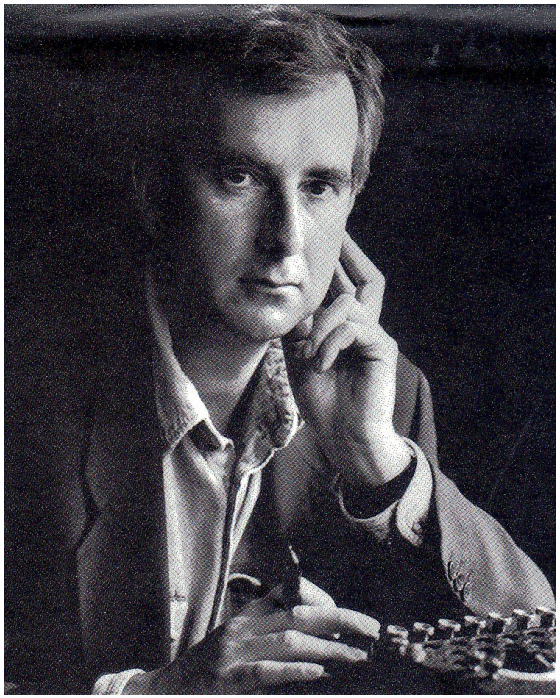
Two days later Jericho had found a note in his pigeonhole.

11

Robert Harris

L'autore

- Nato a Nottingham nel 1957
- Studia a Cambridge
- Giornalista BBC, Observer, Sunday Times
- Fatherland (1992), Enigma (1995), Archangel (1999), Pompeii (2003)



- Imperium (2006), The Ghost (2007), Lustrum (Cospirata—2009)
- Saggi (tra cui Selling Hitler: Story of the Hitler Diaries (1986))
- Films: Fatherland, Enigma, The Ghost Writer, Archangel (BBC mini serie), ...

Michael Apted

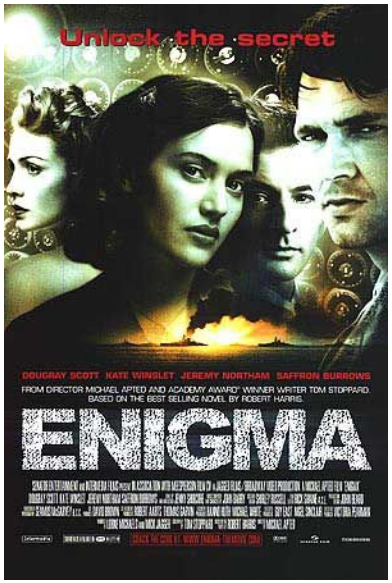
Il regista

- Nato a Aylesbury il 10/02/1941
- Studia a Cambridge
- Dirige numerosi films, tra cui: Stardust (1974), Il segreto di Agatha Christie (1979), Gorky Park (1983), Gorilla nella nebbia (1988), Agente 007 - Il mondo non basta (1999)

- Enigma (2001), Le cronache di Narnia: il viaggio del veliero, ...
- E documentari, ad esempio Bring on the night (Sting) La grande finale (documentario ufficiale dei mondiali FIFA di Germania).

Michael Apted

Enigma (2001)



Il film fu prodotto (in parte) da Mick Jagger che mise a disposizione il proprio ENIGMA.
Critiche principali: non menzionare mai (se non in una scena marginale sulla scelta del nome “squalo”) Alan Turing nè dare credito ai crittanalisti polacchi del Cipher Bureau.

Crittografia informatica

Crittografia informatica

- Si deve assumere il principio di Kerchoffs 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

Breve storia

- 1973 L'NBS (National Bureau of Standards) ora NIST (National Institute of Standards and Technology) richiede un algoritmo standard di cifratura
- 1975 IBM (Horst Feistel et al) pubblica il DES nel Federal Register
- 1976/77 Il DES è approvato/pubblicato come standard
- Il funzionamento del DES è pubblico (soddisfa il principio di Kerckhoffs). Non sono necessariamente rese note le *ragioni* di alcune scelte.

Data Encryption Standard

Chiave

- Del DES tutto è noto tranne la chiave.
- La chiave del DES è costituita da 8 bytes.
- L'ultimo bit di ogni byte è un bit di disparità.
- Dunque la vera lunghezza della chiave è 56 bits (spazio di $2^{56} \approx 7.2 \times 10^{16}$).
- La codifica di un messaggio avviene dividendolo in blocchi di 64 bits e processandone uno alla volta.

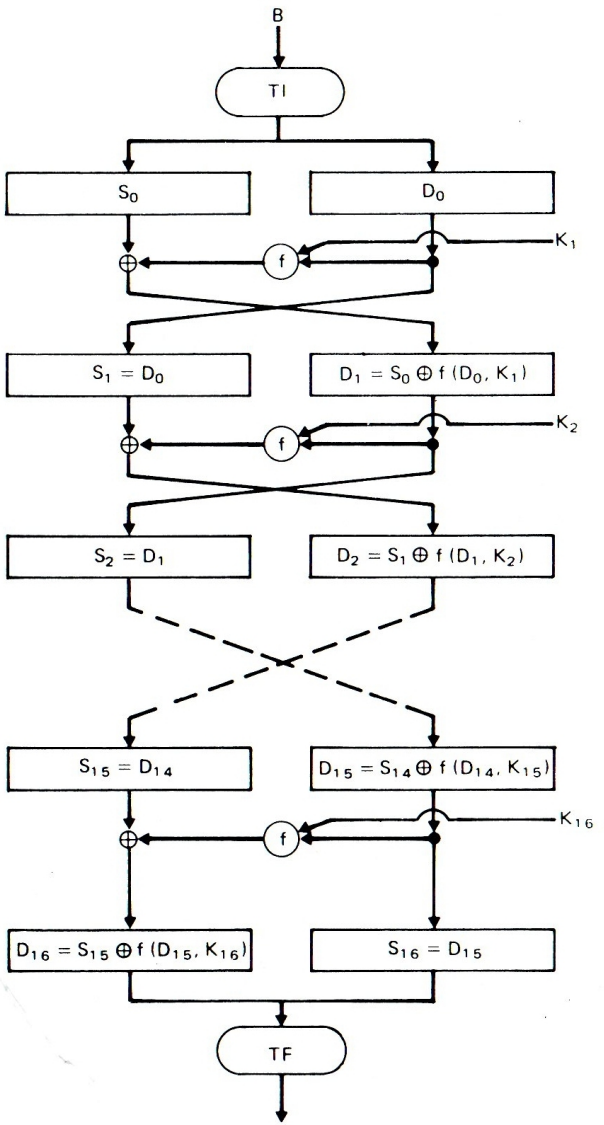
Data Encryption Standard

Chiave

- Del DES tutto è noto tranne la chiave.
- La chiave del DES è costituita da 8 bytes.
- L'ultimo bit di ogni byte è un bit di disparità.
- Dunque la vera lunghezza della chiave è 56 bits (spazio di $2^{56} \approx 7.2 \times 10^{16}$).
- La codifica di un messaggio avviene dividendolo in blocchi di 64 bits e processandone uno alla volta.

Data Encryption Standard

Fase 1–16: Algoritmo principale



- $T_0 = \mathrm{TI}(I)$
- Per $i > 0$, sia T_i il risultato (64 bit) della i-esima iterazione.
- Per $i \geq 0$, siano S_i e D_i t.c. $T_i = S_i D_i$, $|S_i| = |D_i| = 32$.
- Per $i = 1, \ldots, 16$, sia:

$$\begin{cases} S_i &= D_{i-1} \\ D_i &= S_{i-1} \oplus f(D_{i-1}, K_i) \end{cases}$$

- Non entriamo nei dettagli ma sono tutte operazioni (tante) di sostituzione e trasposizione!

Data Encryption Standard

Qualche numero

- 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

- 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

- 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

- 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

- 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

- 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

Decrittazione: forza bruta (e HW dedicato)

- Già nel 1975 Diffie e Hellman stimarono che con circa 20 milioni di dollari si potesse costruire un calcolatore in grado di forzarlo in circa un giorno.
- Nel 1977 fu approvato come “standard”.
- Ogni 5 anni nasceva discussione circa il rinnovo di tale licenza
- Nel 1987, in barba a Kerckoffs, l’NSA propose un nuovo sistema (SW) di cui solo loro sapevano l’architettura (e garantivano forte protezione verso reverse engineering). Non andò in porto.
- Gli attacchi differenziali si dimostrarono inefficaci (e basati su un’ipotesi comunque molto forte).
- Il DES fu ricertificato anche nel 1992.

Data Encryption Standard

Decrittazione: forza bruta (e HW dedicato)

- Nel 1996 maturarono due linee di attacco al DES:
  - Parallelismo massivo ditribuito (forza bruta a basso costo)
  - Architettura ad hoc (Michael Wiener, Bell, 1993)
- Nel 1997 i “rivali” della RSA (di cui parleremo poi) misero in palio 10000 $ per chi sarebbe stato in grado di trovare la chiave di un messaggio creato e cifrato da loro.
- Rocke Verser preparò un programma in grado di distribuire il lavoro su PC vari collegati in Internet. La partecipazione era invogliata economicamente (si sarebbe ricevuto il 40% della vincita nel caso il proprio PC fosse quello che trovava la chiave).

Data Encryption Standard

Decrittazione: forza bruta (e HW dedicato)

- Il messaggio era: Strong cryptography makes the world a safer place.
- Furono necessari 5 mesi (ovviamente con carico piuttosto irregolare). Fu esplorato circa il 25% dello spazio.
- L'anno successivo (1998) l'RSA ripetè il concorso, con il messaggio Many hands make light work.
- In questo caso la chiave fu trovata dopo 39 giorni, visitando l'85% dello spazio di ricerca.

Data Encryption Standard

Decrittazione: forza bruta (e HW dedicato)

- Nel 1998 la EFF (Electronic Frontier Foundation) mise a disposizione un budget di 200.000$ ad un team per lo sviluppo del DES cracker.
- Ad un PC tradizionale furono associati circa 1500 chips dedicati (a 40MHz) che emulavano il DES.
- Ogni chip aveva 24 unità di ricerca ed era delegato ad una porzione delle chiavi possibili.
- Un testo di 16 Bytes era diviso in due blocchi da 8 Bytes da analizzarsi in cascata.

Data Encryption Standard

- La chiave era promettente se sui primi 8 Bytes generava un testo in cui i simboli erano numero, lettera, o simbolo di punteggiatura.
- Se la chiave era promettente, si ripeteva sulla seconda metà.
- Numeri: $\frac{69}{256} \simeq \frac{1}{4}$ sono gli ASCII buoni. Dunque la chance che sia promettente è $\sim \left(\frac{1}{4}\right)^8 = 2^{-16}$. Combinando con la seconda metà si arriva a 2^{-32}. Rimangono per la CPU principale $2^{(56-32)} = 2^{24} \approx 16$ milioni di tentativi.
- A 100 tentativi al secondo sono 40 ore nel caso peggiore.
- La storia del DES finisce qui.

Data Encryption Standard

- La chiave era promettente se sui primi 8 Bytes generava un testo in cui i simboli erano numero, lettera, o simbolo di punteggiatura.
- Se la chiave era promettente, si ripeteva sulla seconda metà.
- Numeri: $\frac{69}{256} \simeq \frac{1}{4}$ sono gli ASCII buoni. Dunque la chance che sia promettente è $\sim \left(\frac{1}{4}\right)^8 = 2^{-16}$. Combinando con la seconda metà si arriva a 2^{-32}. Rimangono per la CPU principale $2^{(56-32)} = 2^{24} \approx 16$ milioni di tentativi.
- A 100 tentativi al secondo sono 40 ore nel caso peggiore.
- La storia del DES finisce qui.

Advanced Encryption Standard

Introduzione

- AES fu annunciato dalla NIST come standard nel 2002, dopo diversi di discussioni e una competizione, vinta dall'algoritmo Rijndael
- Gli autori di Rijndael sono i belgi Joan Daemen and Vincent Rijmen.

Advanced Encryption Standard

Introduzione

- 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.
- La cifrazione avviene comunque sempre con blocchi di 128 bits.
- Usa una “substitution permutation network” invece di una “Feistel network”.
- Può essere esteso con blocchi fino a 256 bits, ma con chiavi di lunghezza illimitata.
- C’è largo uso dell’algebra in $GF(2^8)$.

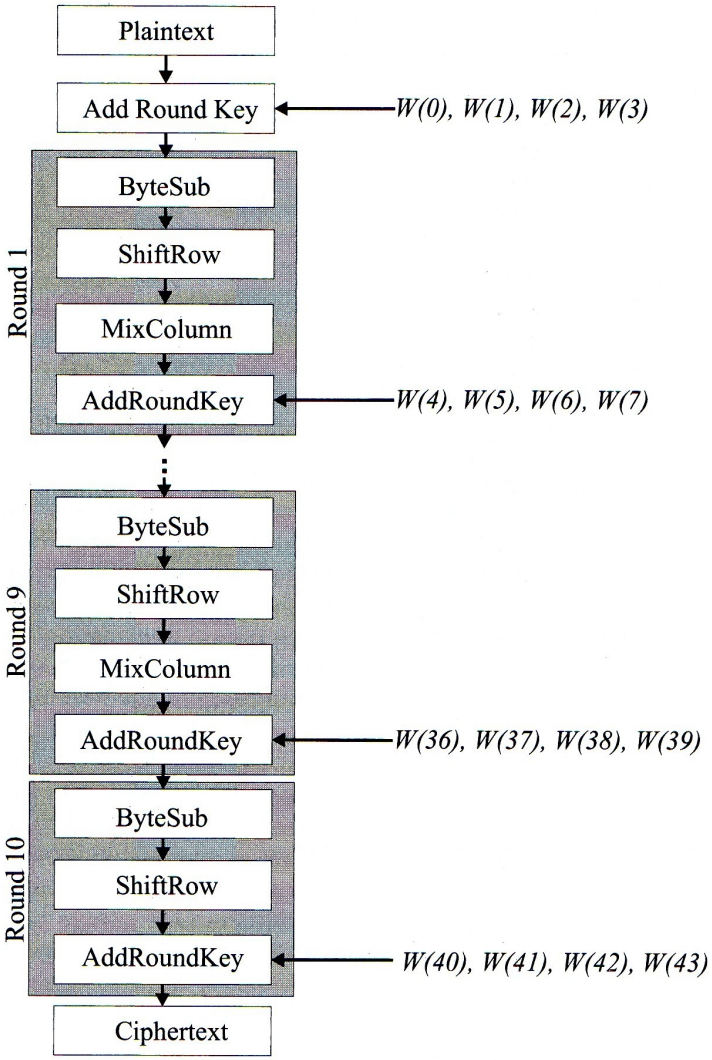
Advanced Encryption Standard

Introduzione

- 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.
- La cifrazione avviene comunque sempre con blocchi di 128 bits.
- Usa una “substitution permutation network” invece di una “Feistel network”.
- Può essere esteso con blocchi fino a 256 bits, ma con chiavi di lunghezza illimitata.
- C’è largo uso dell’algebra in $GF(2^8)$.

Advanced Encryption Standard

Funzionamento



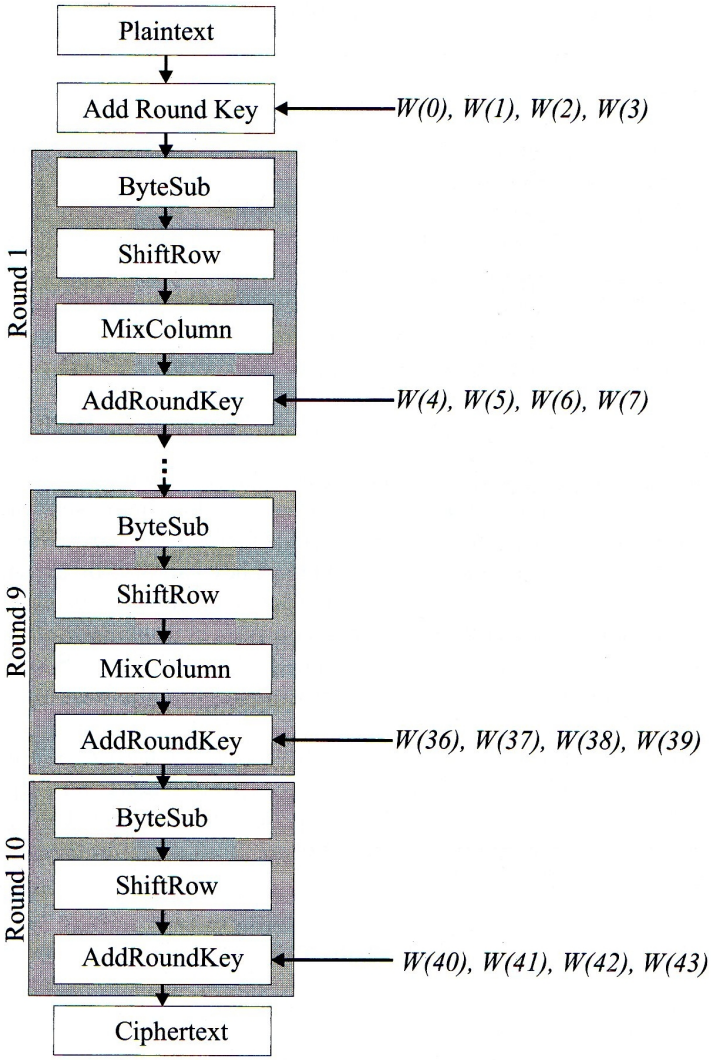
Con la chiave a 128 bits (192, 256), l'algoritmo a sinistra consta di 10 (12, 14) iterazioni.

In ogni iterazione ci sono 4 passi:

1. ByteSub Transformation.
2. ShiftRow Transformation.
3. MixColumn Transformation.
4. AddRoundKey.

Advanced Encryption Standard

Funzionamento



Con la chiave a 128 bits (192, 256), l'algoritmo a sinistra consta di 10 (12, 14) iterazioni.
In ogni iterazione ci sono 4 passi:

1. ByteSub Transformation.
2. ShiftRow Transformation.
3. MixColumn Transformation.
4. AddRoundKey.

Advanced Encryption Standard

Considerazioni finali

- I vari passi sono studiati molto bene per resistere agli attacchi
- Finora nessuno ha trovato un varco
- La forza bruta non è applicabile su quei numeri (attualmente)
- Rimane lo storico problema di scambiarsi la chiave.

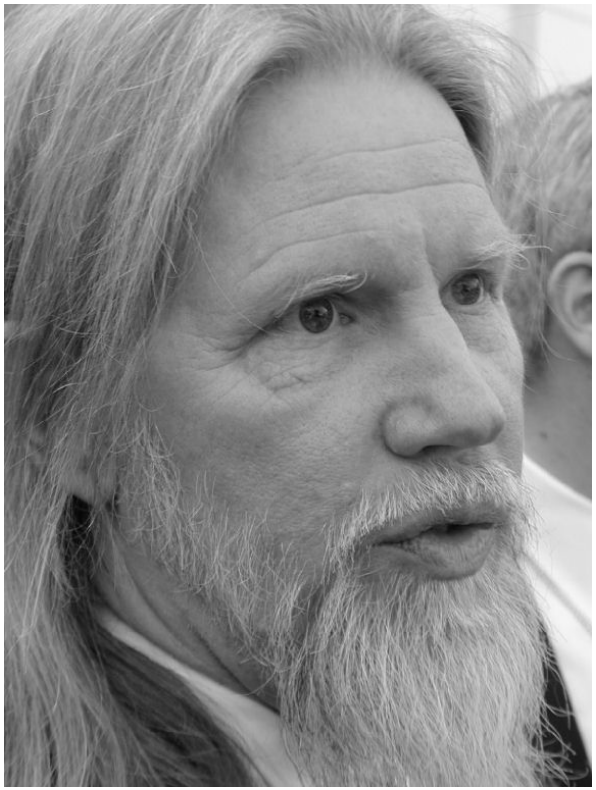
Advanced Encryption Standard

Considerazioni finali

- I vari passi sono studiati molto bene per resistere agli attacchi
- Finora nessuno ha trovato un varco
- La forza bruta non è applicabile su quei numeri (attualmente)
- Rimane lo storico problema di scambiarsi la chiave.

Crittografia a chiave pubblica

Diffie e Hellman



Crittografia a chiave pubblica

Diffie e Hellman

- Nel 1976 Whitfield Diffie e Martin Hellman pubblicano una metodologia rivoluzionaria per la comunicazione cifrata.
- Alla stessa idea erano giunti prima Malcolm J. Williamson, James H. Ellis, Clifford C. Cocks nei primi anni 70. Lavorando al *Government Communications Head Quarter (GB)*, il loro lavoro fu secretato (e all'epoca giudicato tecnologicamente impraticabile)
- In cosa consiste?

Crittografia a chiave pubblica

Diffie e Hellman

- Nel 1976 Whitfield Diffie e Martin Hellman pubblicano una metodologia rivoluzionaria per la comunicazione cifrata.
- Alla stessa idea erano giunti prima Malcolm J. Williamson, James H. Ellis, Clifford C. Cocks nei primi anni 70. Lavorando al *Government Communications Head Quarter (GB)*, il loro lavoro fu secretato (e all'epoca giudicato tecnologicamente impraticabile)
- In cosa consiste?

Crittografia a chiave pubblica

Diffie e Hellman

- Nel 1976 Whitfield Diffie e Martin Hellman pubblicano una metodologia rivoluzionaria per la comunicazione cifrata.
- Alla stessa idea erano giunti prima Malcolm J. Williamson, James H. Ellis, Clifford C. Cocks nei primi anni 70. Lavorando al *Government Communications Head Quarter (GB)*, il loro lavoro fu secretato (e all'epoca giudicato tecnologicamente impraticabile)
- In cosa consiste?

Crittografia a chiave pubblica

Idee generali

La crittografia a chiave pubblica poggia le sue fondamenta sulla comunicazione privata tra due personaggi chiave.

Crittografia a chiave pubblica

Idee generali

Bob ed Alice

Crittografia a chiave pubblica

Idee generali

Bob ed Alice

Inoltre c'è bisogno della spia.

Crittografia a chiave pubblica

Idee generali

Charlie

Crittografia a chiave pubblica

Idee generali

Charlie's

Crittografia a chiave pubblica

Idee generali

- Alice e Bob (e tutti) pubblicano (in un elenco telefonico, negli anni '70, in rete oggi) una volta per sempre la loro chiave pubblica (K_A quella di Alice, K_B quella di Bob, etc.), nota a tutti.
- Alice vuole inviare il messaggio m a Bob.
- Alice codifica il messaggio m per Bob con la chiave pubblica di Bob (K_B) e invia il messaggio cifrato $COD(m, K_B)$.
- Bob riceve il messaggio $COD(m, K_B)$ e usa la sua chiave privata H_B per decodificare il messaggio.

Crittografia a chiave pubblica

Alice spedisce un messaggio a Bob

```
Questo è un
messaggio
riservato che
pertanto non ti
invio in chiaro
ma usando
crittografia a
cgiave
pubblica,
usando
ovviamentela
tua chiave
pubblica.
```

Crittografia a chiave pubblica

Alice spedisce un messaggio a Bob

Questo è un messaggio riservato che pertanto non ti invio in chiaro ma usando crittografia a cgiave pubblica, usando ovviamentela tua chiave pubblica.

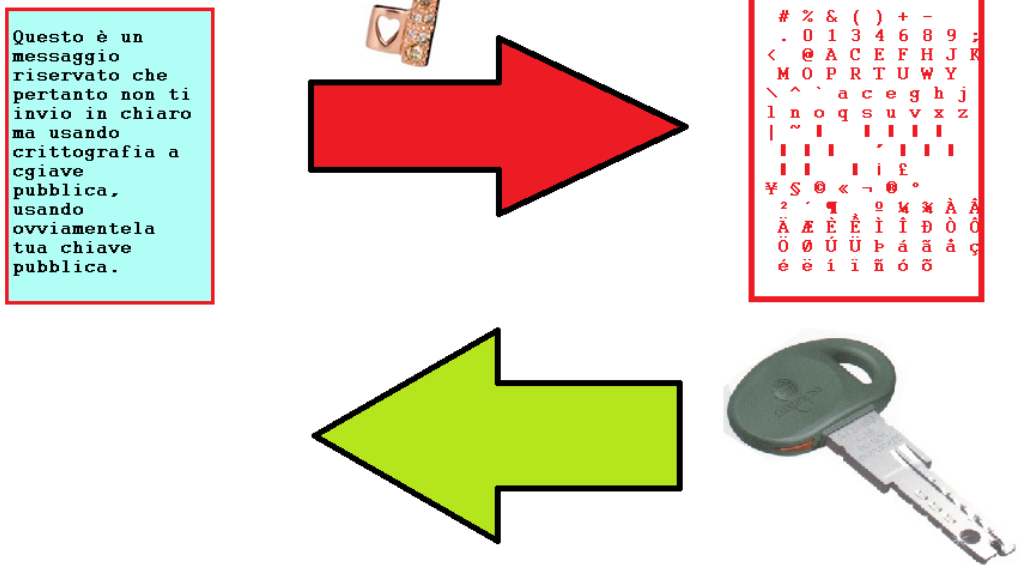
Crittografia a chiave pubblica

Alice spedisce un messaggio a Bob

Questo è un messaggio riservato che pertanto non ti invio in chiaro ma usando crittografia a cgiave pubblica, usando ovviamentela tua chiave pubblica.

Crittografia a chiave pubblica

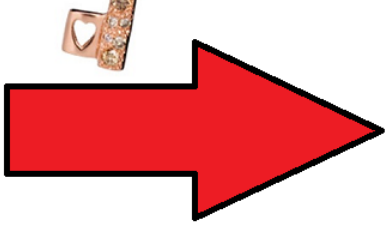
Alice spedisce un messaggio a Bob



Crittografia a chiave pubblica

Alice spedisce un messaggio a Bob

Questo è un messaggio riservato che pertanto non ti invio in chiaro ma usando crittografia a cgiave pubblica, usando ovviamentela tua chiave pubblica.



????

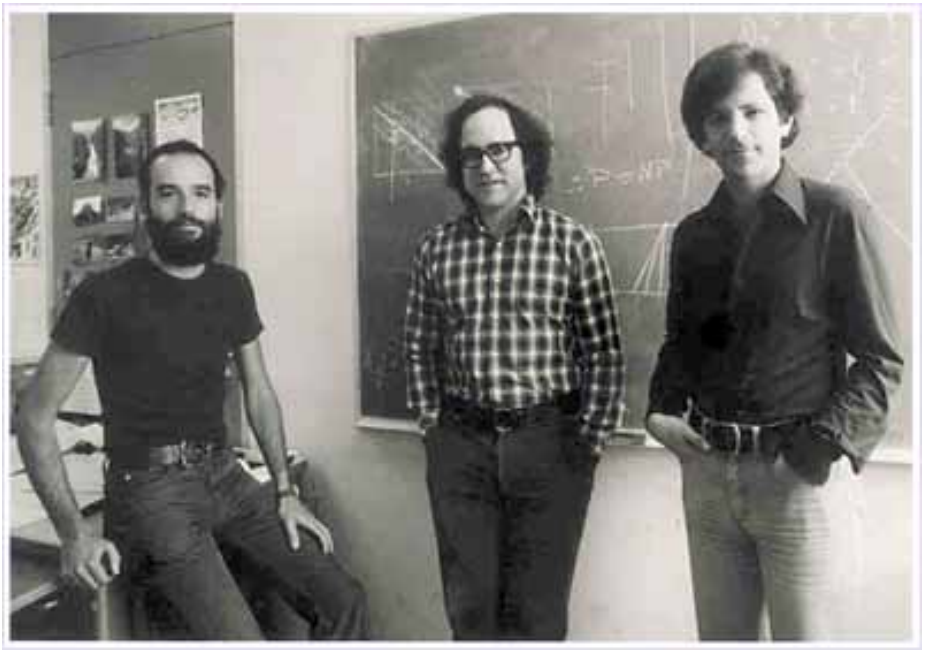
Crittografia a chiave pubblica

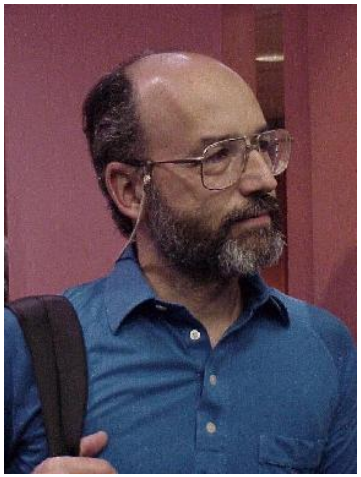
Idee generali

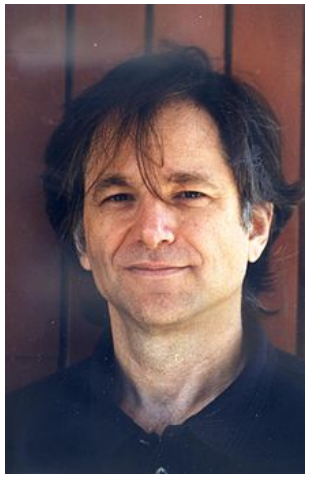
- Le chiavi pubblica e privata sono progettate in modo tale che
$$DEC(COD(m, K_B), H_B) = m$$
- L'operazione di decifrazione, sapendo la chiave privata dev'essere algoritmicamente facile
- L'operazione di decrittazione (per Charlie) deve essere algoritmicamente impraticabile.
- Anche se l'impresa è possibile: conoscendo K_B e $c = COD(m, K_B)$ si possono generare uno ad uno i messaggi di lunghezza opportuna, $m_1, m_2, \ldots, m_\ell$.
- Dunque si codificano uno ad uno con la chiave K_B e si vede se $COD(m_i, K_B) = c$.

Crittografia a chiave pubblica

Rivest, Shamir, e Adleman — Turing award 2002







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

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

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

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

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.

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.

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.
   - 4.1 $MCD(m, n) = 1$ (essendo n prodotto di due primi, questo fatto è molto probabile).
   - 4.2 $MCD(m, n) \neq 1$ (meno probabile, ma va considerato).
5. In entrambi i casi $m^{ed} \mod n = m$

Sia $MCD(m, n) = 1$

- Per il Teorema di Eulero

$$m^{\Phi(n)} = 1 \mod n \tag{1}$$

- Per costruzione, da EE so trovare h tale che:

$$de + h\Phi(n) = 1 \tag{2}$$

Sia $k = -h$.

- Dunque

$$\begin{aligned} c^d &= m^{ed} \mod n \\ &= m^{k\Phi(n)+1} \mod n && (2) \\ &= m\left(m^{\Phi(n)}\right)^k \mod n \\ &= m\,1^k \mod n && (1) \\ &= m && \text{Poiché } m < n \end{aligned}$$

Sia ora $MCD(m, n) \neq 1$

- n è prodotto di due primi p e q
- $m < n$ per costruzione.
- Lemma. Se $p \neq q$ sono due primi, $a = b \mod p$ e $a = b \mod q$, allora $a = b \mod pq$.
- $MCD(m, n) \neq 1$ significa che $(p|m)$ oppure $(q|m)$ (ma non entrambi, anche se quanto segue varrebbe comunque).
- Sappiamo per costruzione che $de = 1 \mod \Phi(n)$ dunque $\Phi(n)|de - 1$.
- Ma $\Phi(n) = (p - 1)(q - 1) = \Phi(p)\Phi(q)$
- Dunque $\Phi(p)\Phi(q)|de - 1$, pertanto vale sia $\Phi(p)|de - 1$ che $\Phi(q)|de - 1$, ovvero $de = h\Phi(p) + 1$ e $de = k\Phi(q) + 1$ per h e k opportuni.

- Se $MCD(m,p) = 1$, per Teorema di Eulero $m^{\Phi(p)} = 1 \mod p$, dunque $m^{de} = m(m^{\Phi(p)})^h = m \mod p$
- Se $MCD(m,p) \neq 1$, siamo nel caso $(p|m)$. Dunque $m = pr$ per r opportuno. Pertanto

$$\begin{aligned} m^{de} &= (pr)^{de} \\ &= p^{de} r^{de} \\ &= 0 \mod p \\ &= pr \mod p \\ &= m \mod p \end{aligned}$$

- Similmente, sia con $MCD(m,q) = 1$ che $MCD(m,q) = 1$, vale che $m^{de} = m \mod q$
- Dunque, per il Lemma sopra, in ogni caso $m^{de} = m \mod pq$

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.

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.

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.

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.

Crittografia a chiave pubblica

Fattorizzazione

```
fattorizza(n)
    for i = 2 to sqrt(n)
        if n mod i = 0 {
            print(i," e' un divisore di ",n);
            exit;
        }
```

Nel caso peggiore compie un numero di passi pari a $\sqrt{n}$.

Possiamo togliere i pari, i multipli di 3, i multipli di 5, ma la sostanza non cambia.

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

Fattorizzazione

```
fattorizza(n)
    for i = 2 to sqrt(n)
        if n mod i = 0 {
            print(i," e' un divisore di ",n);
            exit;
        }
```

Nel caso peggiore compie un numero di passi pari a $\sqrt{n}$.
Possiamo togliere i pari, i multipli di 3, i multipli di 5, ma la sostanza non cambia.

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

Fattorizzazione

```
fattorizza(n)
    for i = 2 to sqrt(n)
        if n mod i = 0 {
            print(i," e' un divisore di ",n);
            exit;
        }
```

Nel caso peggiore compie un numero di passi pari a $\sqrt{n}$.
Possiamo togliere i pari, i multipli di 3, i multipli di 5, ma la sostanza non cambia.
Se n è un numero di 100 cifre, ogni divisione mi costa $10^{-10}s$, e dispongo di 10^9 processori, mi serve tempo

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

```
fattorizza(n)
    for i = 2 to sqrt(n)
        if n mod i = 0 {
            print(i," e' un divisore di ",n);
            exit;
        }
```

Nel caso peggiore compie un numero di passi pari a $\sqrt{n}$.
Possiamo togliere i pari, i multipli di 3, i multipli di 5, ma la sostanza non cambia.
Se n è un numero di 100 cifre, ogni divisione mi costa $10^{-10}s$, e dispongo di 10^9 processori, mi serve tempo

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

Conclusioni

- AES e RSA vengono spesso usati in combinazione (p.es. PGP).
- Usando RSA ci si passa la chiave (corta) per l'AES e si usa AES (più veloce) per cifrare il messaggio (lungo).
- Astraiamo pure dai dettagli delle varie operazioni sui campi finiti di AES e RSA (ci sono i programmi già pronti che li fanno per noi).
- 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
- Ma nessuno ha mai dimostrato che non si può fare!

Conclusioni

- AES e RSA vengono spesso usati in combinazione (p.es. PGP).
- Usando RSA ci si passa la chiave (corta) per l'AES e si usa AES (più veloce) per cifrare il messaggio (lungo).
- Astraiamo pure dai dettagli delle varie operazioni sui campi finiti di AES e RSA (ci sono i programmi già pronti che li fanno per noi).
- 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
- Ma nessuno ha mai dimostrato che non si può fare!

Conclusioni

- AES e RSA vengono spesso usati in combinazione (p.es. PGP).
- Usando RSA ci si passa la chiave (corta) per l'AES e si usa AES (più veloce) per cifrare il messaggio (lungo).
- Astraiamo pure dai dettagli delle varie operazioni sui campi finiti di AES e RSA (ci sono i programmi già pronti che li fanno per noi).
- 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
- Ma nessuno ha mai dimostrato che non si può fare!

Conclusioni

E ora tocca a voi!!!

Ci vediamo ad aprile!