Complexity and encryption

Slides by Dan Fu

Outline

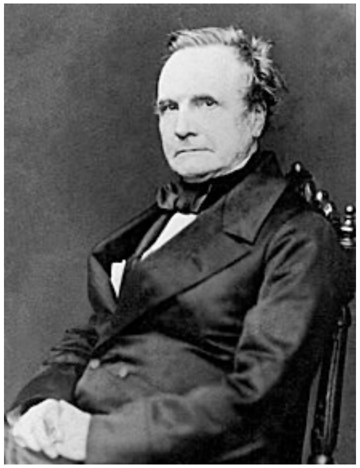
- A history of computation
- The Enigma machine
- RSA cryptography
- The complexity of RSA

Poll: Who’s the father of computing?

Pioneers of Computing

Charles Babbage

The first computer

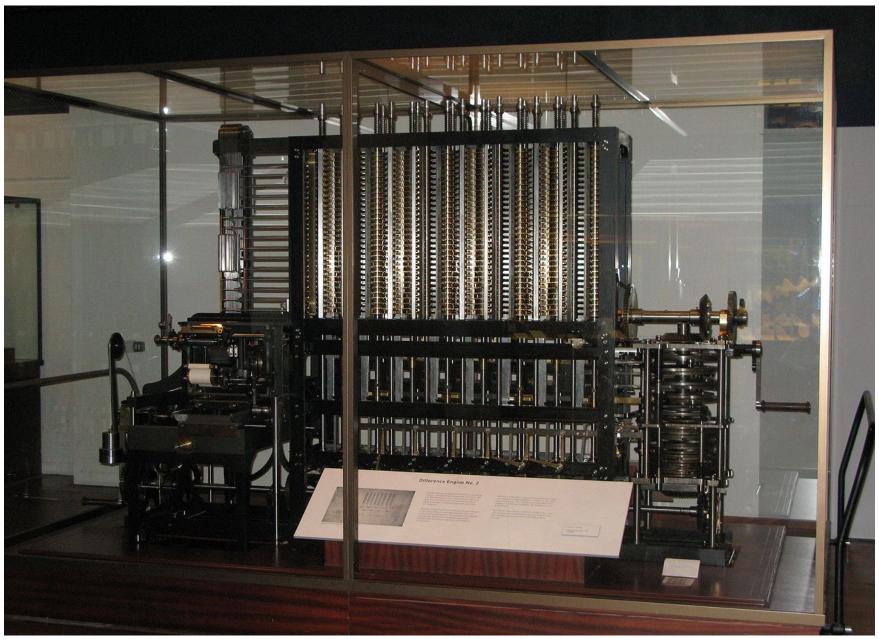


Ada Lovelace

The first algorithm

Pictures from Wikipedia

The Analytical Engine



- Built in 1837
- Next general purpose computer not built until 1941!!!
- First universal Turing machine
- Had memory, ALU, if/for/while

Pictures from Wikipedia

Input

- Took input via punch cards
- Physically move the cards in the machine
- Cards stored instructions and variables
- The code was a lot like assembly
  - Read this card
  - Read the next card
  - Add the value
  - Read the previous card
  - …and so on

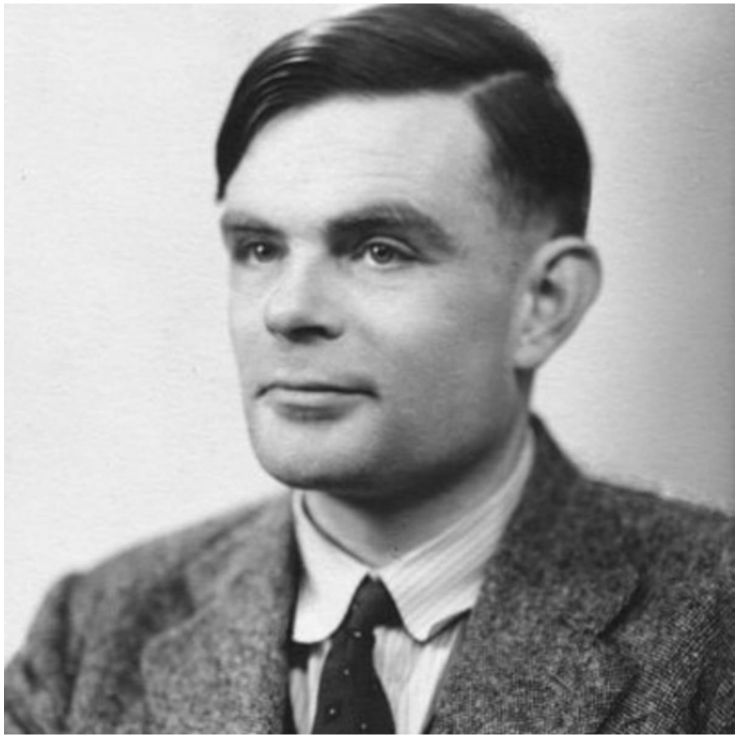
Pictures from Wikipedia

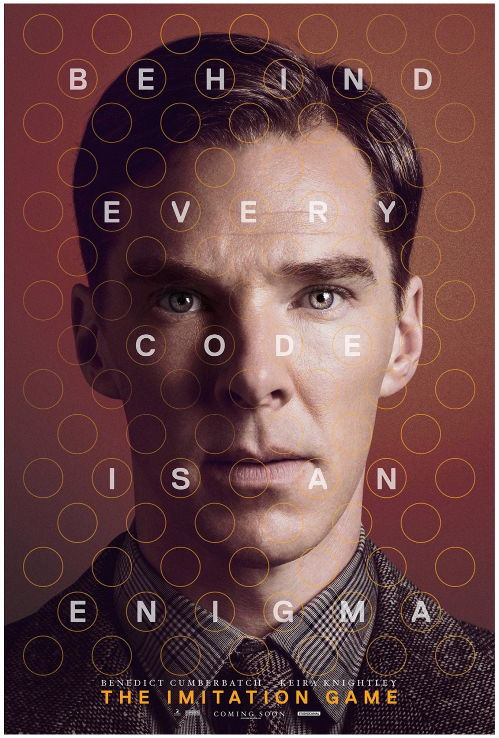
Poll: What could go wrong?

- Charles Babbage already recognized the problem of "computational complexity"
- Bigger programs need more cards, more time, and more memory

Pictures from Wikipedia

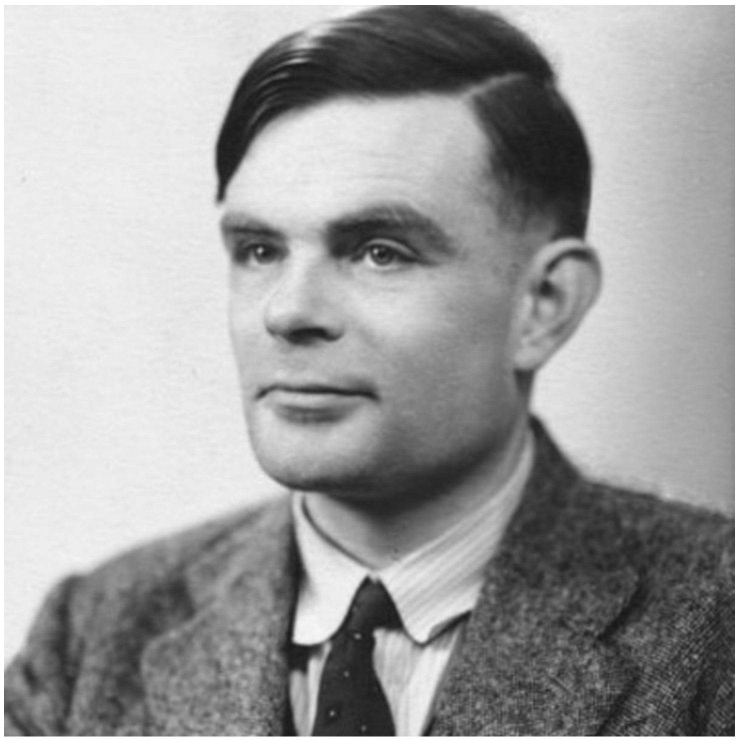
Poll: Who was Alan Turing?





Pictures from Biography.com, imdb

Alan Turing



1912-1954, Died age 41

- Formalization of computation by the Turing machine
- Turing awards given out since 1966
- Developed Turingery, a code-breaking technique implemented in the Bombe to break Nazi encryption by the Enigma machine (not acknowledge until 1970s!)

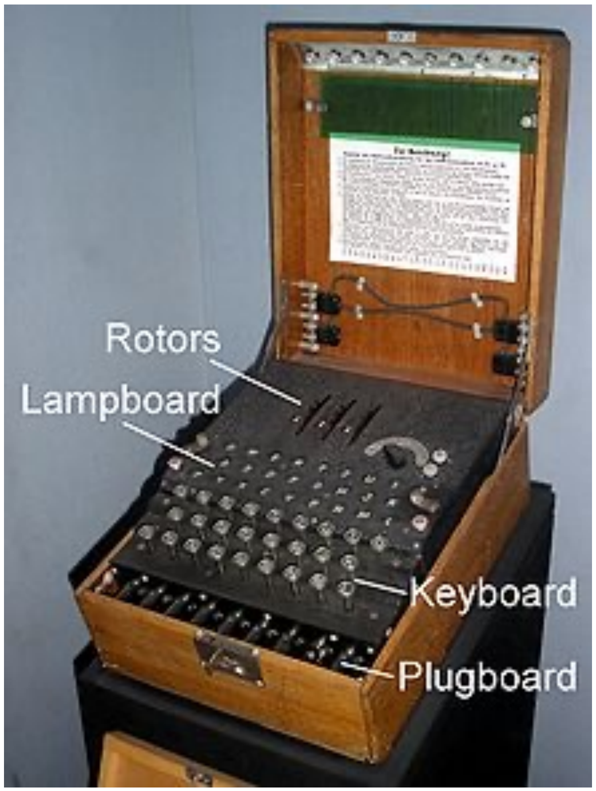
Pictures from Biography.com

German U-boats

- Decimated Allied naval efforts in WW2
- Allies tried radar, underwater mines, but most importantly, broke the Enigma code and could intercept orders sent to Nazi U-boats to know where they would be

Pictures from Wikipedia

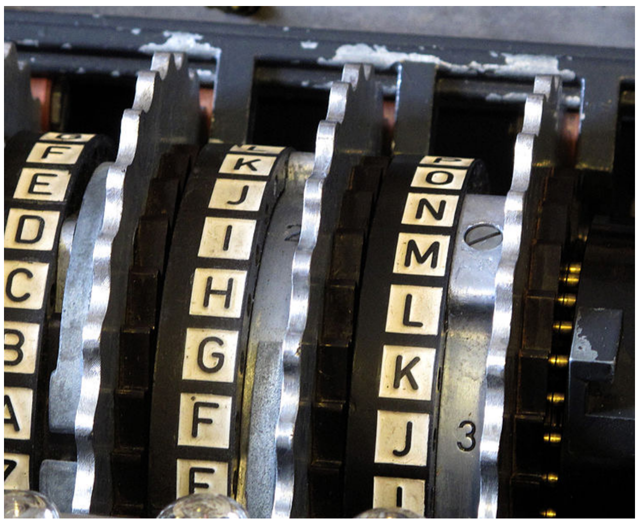
Enigma Machine



- Invented in 1929 by German inventor Arthur Scherbius
- Immediately taken and classified by Nazi government as they wanted to keep secret their actions that would infringe on the Treaty of Versailles (WW1 peace treaty)
- 26 German alphabet letters
- Changes one letter to another
- NOT a computer
- Needed computers to break the encryption
- The encryption was so good that Nazis broadcasted messages in Morse code, publicly, over the radio, without fear of being intercepted

Pictures from Wikipedia

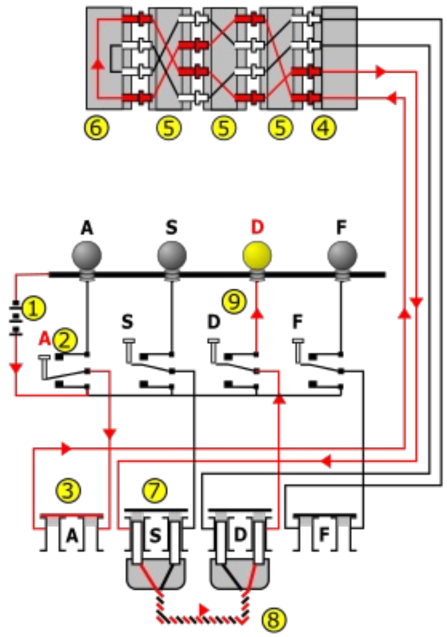
Enigma Machine



- It was just 3 rotors and some wiring
- Press one button, another lights up. One person types the message, the other records the scrambled text.
- Typing the ciphertext reverses the message back to plain language.
- The alignment of rotors changed the path of an electrical signal through the machine.

Poll: How long would it take you to solve a setting?

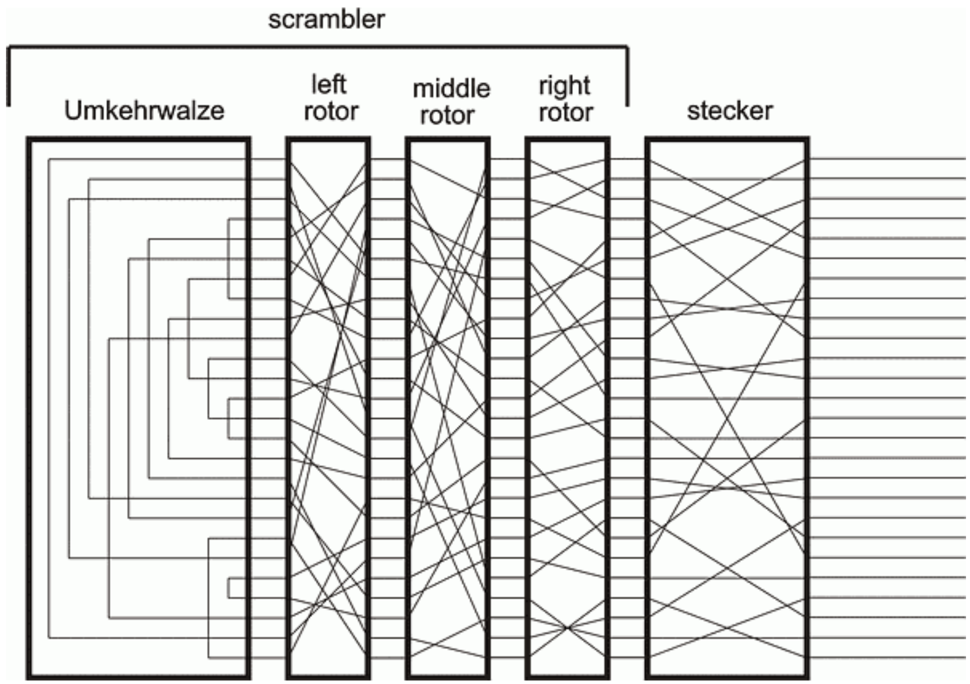
Enigma Machine



- Setting: scramble the wiring
- Nazis began to scramble it every day and different divisions of their military (up to 30) also did so independently. Later added 2 more rotors!
- The encryption itself was "understood" by Polish mathematicians…but…
- 3.3 x 10^{114} possible combinations!
- Would take 100,000 codebreakers more than the lifetime of the universe to try them all…and then it gets changed the next day

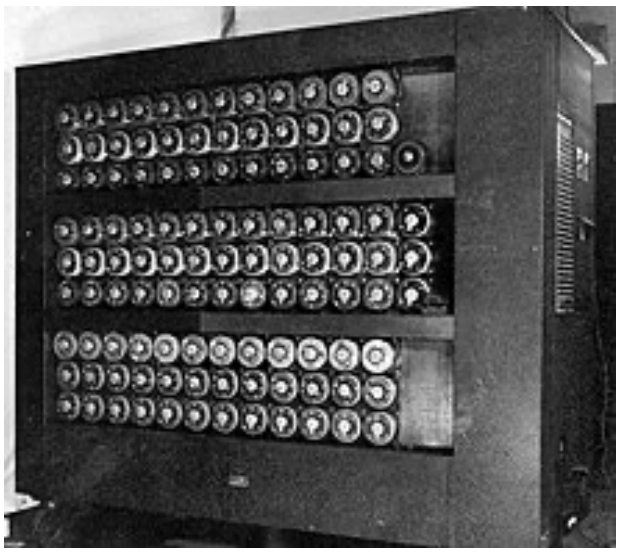
Pictures from Wikipedia

Enigma Machine



Pictures from Graham Ellsbury

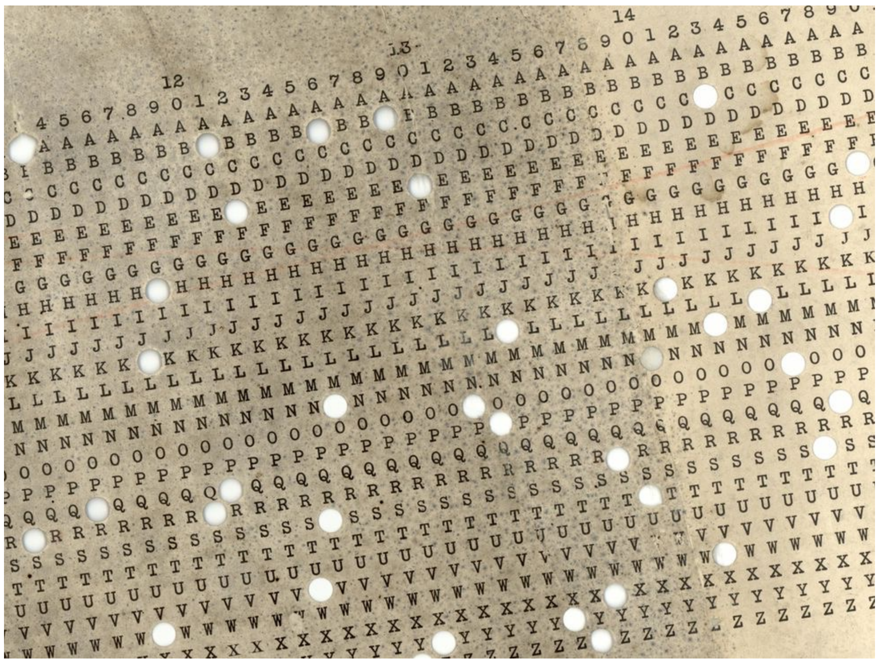
Bombe



- Allied efforts created a "computer" to break the code fast
- Not until 1940s
- More than 100 years later from Babbage
- Not a general purpose machine (not programmable)
- Simulated the Enigma in parallel with a lot of rotors
- That means…
- The encryption was likely **P**, or simply solvable with more computing power

Pictures from Wikipedia

How?



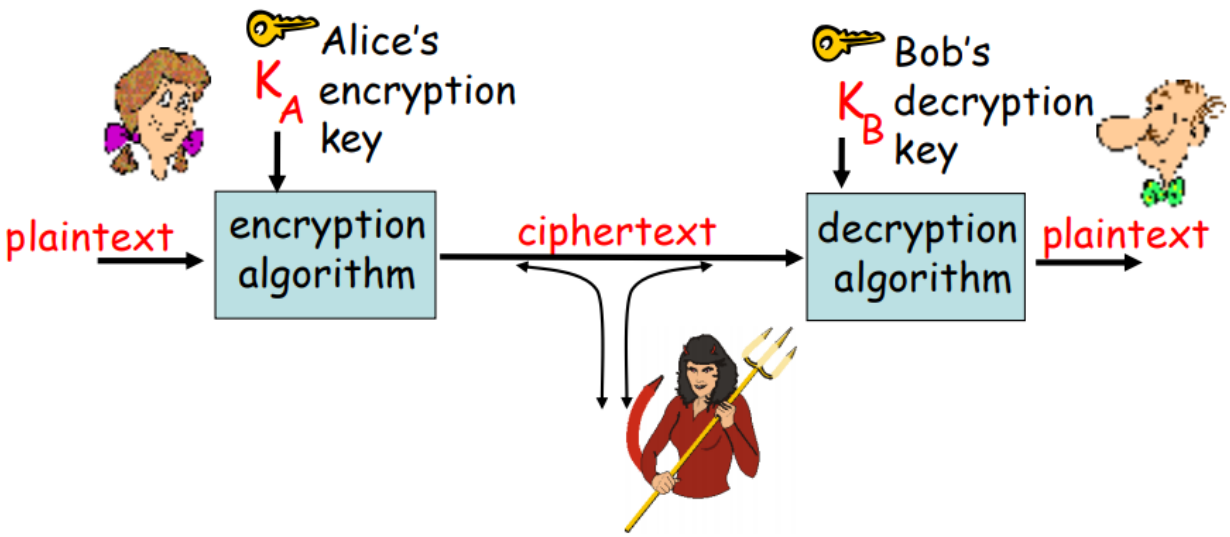
- Polish mathematicians discovered some “rules” of Enigma machines, eliminating possible settings.
- Bombe tried remaining possibilities in parallel.
- But any documents for how the Bombe was really designed were destroyed and their secrets lost forever.

Pictures from Wikipedia

Modern Encryption: RSA

- Encryption is a "one-way" function: a function f that is easy to compute but hard to invert (Arora & Barak)
- You want to send a message, scramble it, and make sure only your intended receiver can understand it
- An interceptor could see the message, but they won't understand it

The language of cryptography



symmetric key crypto: sender, receiver keys *identical*

public-key crypto: encryption key *public*, decryption key *secret* (private)

From Li Yang, UTC

RSA

1. Choose two large prime numbers p, q
2. Compute $n = pq, z = (p-1)(q-1)$
3. Choose e (with $e < n$) that has no common factors with z. (e, z are "relatively prime")
4. Choose d such that $ed \bmod z = 1$.
5. Public key is $P_A(M) = (n, e)$. Private key is $\mathrm{S}_A(C) = (n, d)$
6. A message $S_A\big(P_A(M)\big) = M$ and also $P_A\big(S_A(M)\big) = M$

From Li Yang, UTC; Eli Upfal

Why does it work?

- Easy to calculate in forward direction, but difficult to crack.
- Finding p, q from n is hard.
- Thus breaking RSA is hard.
- But is it NP-Complete “hard”?

What is NP-Complete "hard"?

- NP-Complete is:
  - Decision problem: Does a solution exist? (Yes/No)
  - Solution is polynomial time verifiable
  - *Usually* exponential to find a solution (some exceptions), but this is NOT a characteristic and does not prove that $P \neq NP$
  - Not always apparent by description of a problem. Prove it from reductions
  - Describes difficulty <u>in the **worst** case.</u> ← Important distinction!!!

RSA needs to be difficult <u>in **most** cases.</u>

Complexity of Solving RSA

- Is factoring in NP?

- Is factoring NP-Complete?

Complexity of Solving RSA

- Is factoring in NP?
  - Yes, if you're given a solution, multiplying them and verifying the result is a polynomial time verification

- Is factoring NP-Complete?
  - No
  - ??? Probably
  - Because it is $NP \cap coNP$ (also recall NP ≠ coNP…probably)

Complexity of Solving RSA

- What's the NP decision problem?
  - Is there a prime factor $p < k$ such that p factors n?
  - Verifies that yes, if there is one
- What's the coNP decision problem?
  - Is there a prime factor p that factors n but not $< k$?
  - Verifies that no, there is not a yes solution to the NP problem.

- Why is this difficult:
  - Recall: coNP is a class of problems that are unlikely to have short proofs