Codebreaking has moved on since Turing's day, with dangerous implications

November 24 2014, by Bill Buchanan

After decades of cracking, the cracks are appearing in cryptography. Credit: infobunny, CC BY

We have always been been intrigued by keeping secrets and uncovering the secrets of others, whether that's childhood secret messages, or secrets and codebreaking of national importance.

With a film, The Imitation Game, reprising the life of Alan Turing and

his role in breaking the Nazi's Enigma cipher of World War II, how does one codebreak, then and now?

It's all in the cipher

Imagine that Bob and Alice wish to secretly communicate, and Eve, who wishes to listen in. Here "plain text" refers to the original message, and "cipher text" as the coded message.

There are two ways of creating the cipher text:

- Having an algorithm (a cipher) that only Bob and Alice know, so that one applies the cipher to encode the text and the other applies the cipher in reverse to decode.
- Using a well-defined algorithm, but adding something that changes the way it operates which is easy for Bob and Alice to convert, but difficult for Eve to find.

In the first case, to read their messages Eve will have to crack the cipher – to work out what method it uses to change plain text to cipher text. For example, the famous Julius Caesar cipher is a form of substitution cipher that shifts letters in the alphabet a number of places, for example a shift of 13 means A becomes N, B becomes O, C becomes P and so on. This is an easy code to crack, as there are only 25 unique shifts.

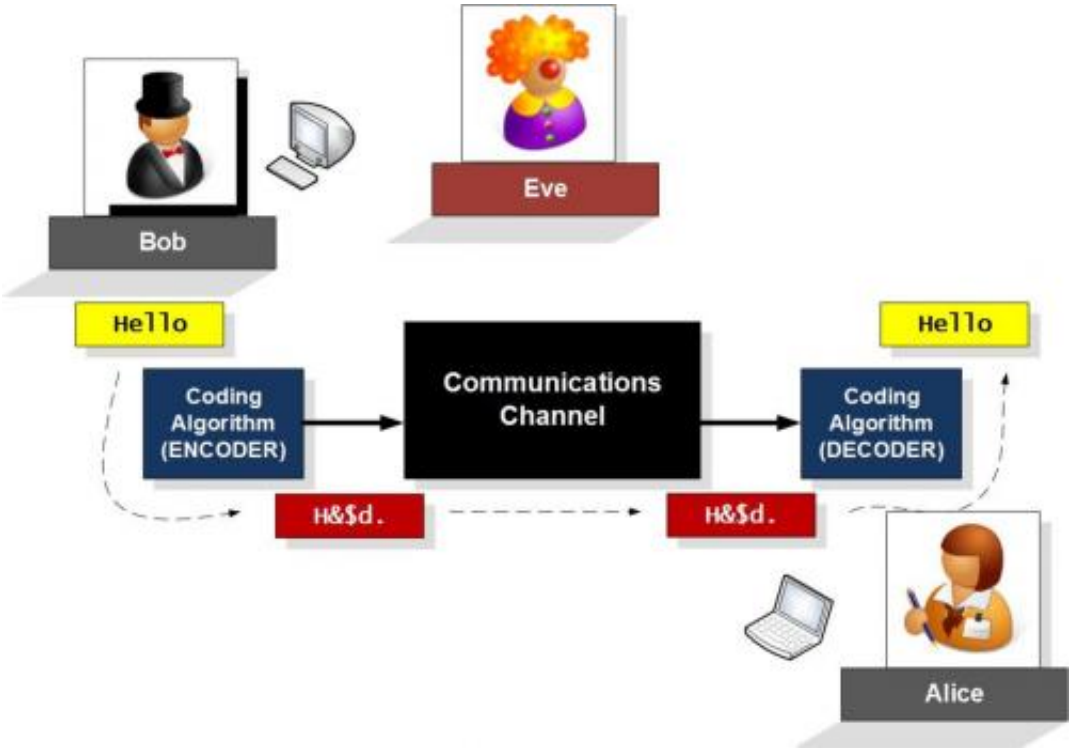
Far harder would be a scrambled alphabet, in which any letter can be mapped to any other letter without the same substitution shift applying to all, as with the Caesar cypher. This leads to a colossal number of permutations – 403 million billion billion – that, even with a supercomputer with which to try a billion mappings per second, would still take an average 6.3 billion years to crack.

However, these ciphers' fundamental weakness is the occurrence of the

letters: in English, the letter with the most occurrences is likely to represent an E, the most common in the language. Performing a frequency analysis with this in mind makes much shorter work of it – about five minutes.

In the early days it was the ciphers' text scrambling method that was kept secret. But if Eve manages to crack the cipher, neither Bob nor Alice will know – just as the Nazis didn't know the Allies had cracked Enigma. So modern cryptography uses a different approach: a public method to create the cipher, but a private key to use the cipher that Eve will find difficult to find. This is public key encryption.

Codebreaking Enigma



Secret messaging, ciphers, and those listening in. Credit: Bill Buchanan, Author provided

In the days before computers, ciphers were mechanically generated – the Enigma cipher rotor machine is a good example. It used a polyalphabetic substitution cipher – with three rotors to generate three alphabetic substitution shifts – and a secret key. The challenge was to determine both the algorithm used and the key.

Enigma's weakness was that the machine prevented a plain text letter from being ciphered as itself (that is, from A ending up after three substitutions as A). This made the challenge easier as the codebreakers could dismiss any code that mapped to the same letter, bu this still left many alternatives – too many for a human to crack.

The mathematical prowess of the Polish Cypher Bureau had secretly first broken Enigma codes in 1932, with the aid of French intelligence. At the eve of the war they handed their work to the Allies, who were amazed. But by now the German military were using more advanced versions of the Enigma machine, with extra rotors and other features adding complexity to the cipher. Dilly Knox, the British chief codebreaker, had some success but better equipment was required.

Developing the work of the Polish Cypher Bureau, Turing and Gordon Welchman designed the electro-mechanical Bombe, a device designed to imitate Enigma machines wired back-to-back, which given certain information could narrow down the possible permutations of the Enigma machines' settings from 150 million million to a more manageable number.

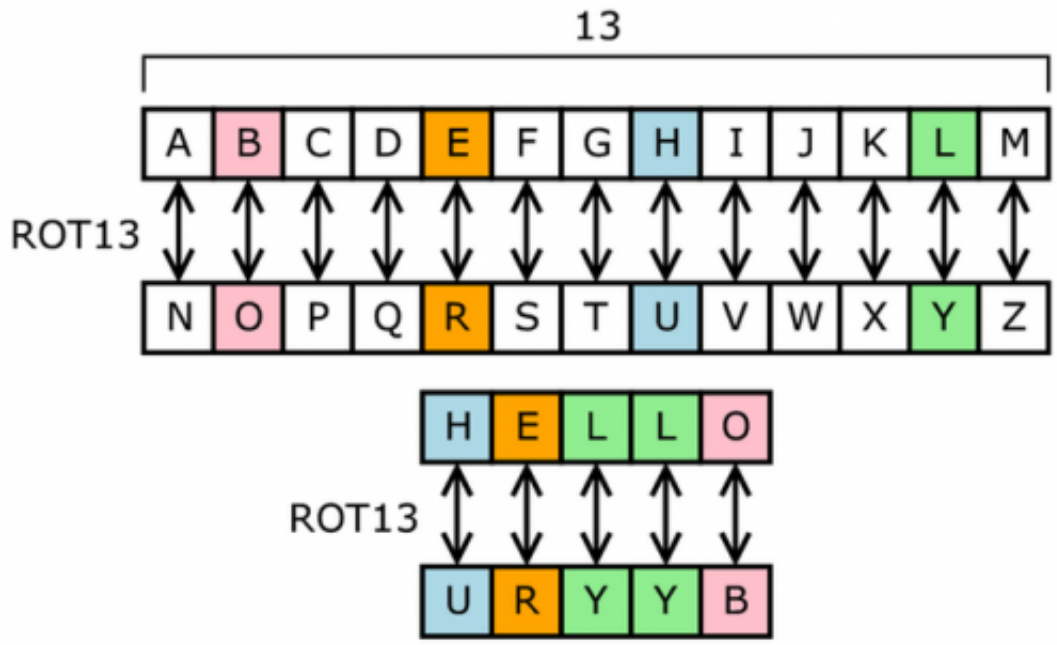
But the true father of code cracking is Colossus, the world's first programmable electronic digital computer, which was created by engineer Tommy Flowers in order to crack the Lorenz cipher, the more sophisticated successor to Enigma.

Cracking crypto today

Computers are so much more powerful now than in Turing's day that their raw power can crack passwords or uncover encryption methods simply by crunching through all the different possible permutations.

Most passwords are stored in a hash – a fixed-length string of characters generated by a mathematical function from text of any length. This is a one-way process, so the hash cannot be reversed to gain the original text.

However today it's possible to compile look-up tables of pre-hashed values, essentially a dictionary of hashes to compare to the plain text passwords they represent.



A substitution cypher with the alphabet shifted 13 times (ROT13). Matt Crypto

As an illustration: a seven character password in lower case letters could be one of over 8 billion possible permutations. The graphics processor

on a typical computer graphics card (which are excellent for this task) can check over 150m words per second, meaning the maximum time to crack any password is a mere 53 seconds.

Adding upper case letters gives one thousand billion combinations, and a maximum time to crack of 114 minutes. Even when adding 20 more numeric characters and common punctuation marks to create more than ten thousand billion permutations, the maximum time is only 18 hours. Eight character passwords are a bit more difficult with 722 thousand billion permutations, but even so these can be cracked at an acceptably quick time of 1,337 hours or 55 days.

The worry is that the distributed computing power of the cloud is making this process significantly easier. One of the most common hashing functions, MD5, has been shown to be too weak to cope with the computing power that can now be thrown at it.

Each MD5 hash is 128 bits long, so to store every possible seven character password hash would require 160 terabytes of storage space. While that seems like a lot, you can buy a 4TB drive for about £100, Microsoft advertises no limits to its storage, and Dropbox offers several terabytes for little cost. Similarly, using cloud processing power such as the Amazon Cloud it would be possible to rent processing time on 1,000 of these graphics cards for just pennies, spread the task between them, and crack an eight-character password in an hour and a half.

Add salt to taste

The true way to increase the strength of a password is to add a salt, a random string of text used to add complexity to a hashed password.

For example, the seven-character salt "eFUqsfi" is added to the password "password" to create the string "eFUqsfipassword". This produces a hash

that is sufficiently random that it won't appear in a hash look-up table and, being 15 characters instead of eight, massively increases the complexity of trying to crack the password using brute force attacks, from 722 billion billion permutations to over 742 million billion billion permutations.

Thus we have gone from the millions of dollars of investment to build and maintain Colossus, to a time where crackers have managed tocreate the same hash signature for a different image, and where it tookjust 10 hours and cost only 65 cents plus tax on a GPU instance on the Amazon Cloud.

So some of the pure intellect of Turing's day has gone, and now it's down to who has the fastest computer. The cloud itself is a supercomputer that is expanding by the day, and with websites dedicated to analysing and storing as many hashed passwords as possible, the whole foundations of our password cryptography are starting to crumble, with profound implications for the security of the internet.

This story is published courtesy of The Conversation *(under Creative Commons-Attribution/No derivatives).*

Source: The Conversation

Citation: Codebreaking has moved on since Turing's day, with dangerous implications (2014, November 24) retrieved 20 March 2024 from https://phys.org/news/2014-11-codebreaking-turing-day-dangerous-implications.html