How to Crack Engima?

Arjun Agarwal, CMI

Based on work with Prof. Jyothi Krishnan and Aaditya Shah

Contents

Introduction

- Cryptography originated somewhere around 2000 years ago with Julius Caesar. Caesar was one of the first rulers to have battles far away from the capital.
- So now if the king wanted to change the battle plan or ask the General to do something, they had to send a message.
- However,
  1. If the opposing army intercepts the message...
  2. If a general defects and tells our scheme...
  3. Someone could send fake messages...
- This created a need for secure and authenticated messages.

Caesar's idea was simple: shift the letter's some places. For example:

$$A \to C, B \to D, ..., X \to Z, Y \to A, Z \to B$$

Say I want to transmit the policy of utmost importance: ***"Once we win the Gaelic War, get croissants from France."***

And we can't let the enemies know that we want croissants! So instead we transmit ***"Xwln fn frw cqn Pjnurl Fja, yunjbn pnc laxrbbjwc oaxv Oajwln."***.

This is probably utter gibberish, unless this summons a devil from the nine hells or something. But if you knew that we have shifted by 9 places, you can undo it rather quickly!

Some Terms

Plain Text: ***“Once we win the Gaelic War, get croissants from France.”***

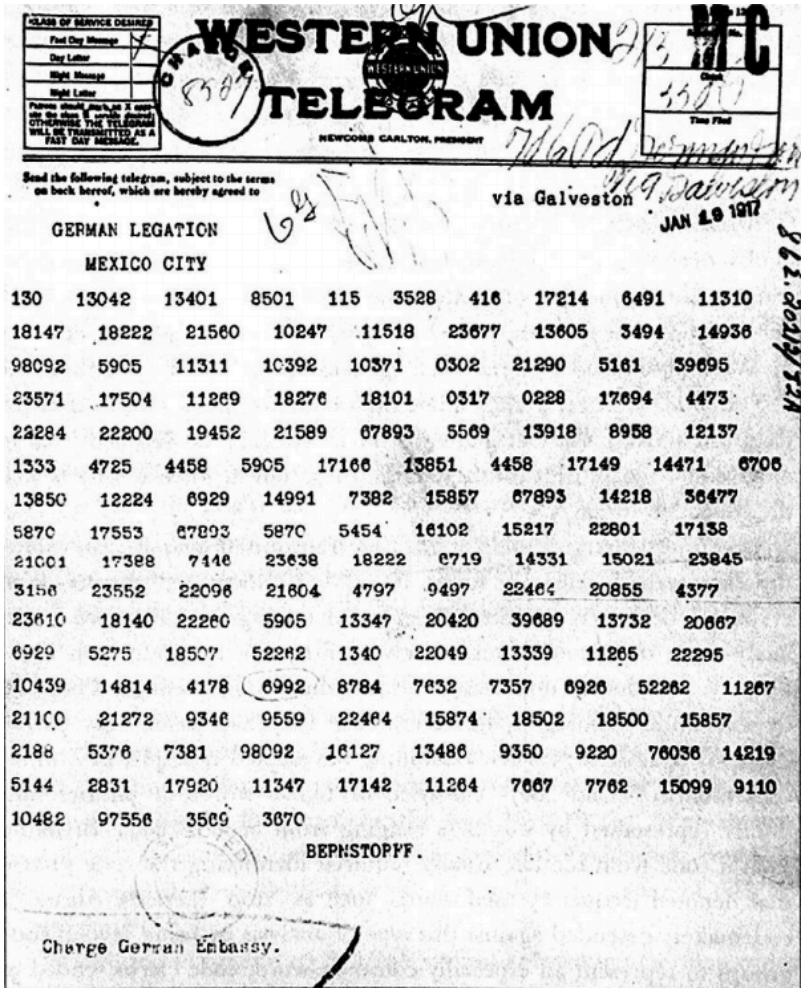
Algorithm: Shift the alphabet

Key: 9

Cipher Text: ***“Xwln fn frw cqn Pjnurl Fja, yunjbn pnc laxrbbjwc oaxv Oajwln.”***

During the first World War, the following telegram was intercepted. And in what followed; a group of linguists, language professors and crossword enthusiasts basically changed the trajectory of history.

- This Telegram was from the German foreign secretary to the German ambassador in Mexico City seeking alliance against the United States.
- This message was intercepted and decrypted by Room 40, Britain's cryptoanalysis department. Notably, mathematicians and (a job that doesn't yet exist) computer scientists are yet to enter the picture.
- About 3 months later, on 6 April 1917, US declares war on Germany. The war ended on 11 November 1918.



WESTERN UNION TELEGRAM

Send the following telegram, subject to the terms on back hereof, which are hereby agreed to

via Galveston JAN 19 1917

GERMAN LEGATION
MEXICO CITY

130 13042 13401 8501 115 3528 416 17214 6491 11310
18147 18222 21560 10247 11518 23677 13605 3494 14936
98092 5905 11311 10392 10371 0302 21290 5161 39695
23571 17504 11269 18276 18101 0317 0228 17694 4473
22284 22200 19452 21589 67893 5569 13918 8958 12137
1333 4725 4458 5905 17166 13851 4458 17149 14471 6706
13850 12224 6929 14991 7382 15857 67893 14218 36477
5870 17553 67893 5870 5454 16102 15217 22801 17138
21001 17388 7446 23638 18222 6719 14331 15021 23845
3156 23552 22096 21604 4797 9497 22464 20855 4377
23610 18140 22260 5905 13347 20420 39689 13732 20667
6929 5275 18507 52262 1340 22049 13339 11265 22295
10439 14814 4178 6992 8784 7632 7357 6926 52262 11267
21100 21272 9346 9559 22464 15874 18502 18500 15857
2188 5376 7381 98092 16127 13486 9350 9220 76036 14219
5144 2831 17920 11347 17142 11264 7667 7762 15099 9110
10482 97556 3569 3670

BEPNSTOPFF.

Charge German Embassy.

What Was The Enigma?

But all of this changed a few decades later, with the creation of the Enigma.

Enigma refers to a German encryption machine used during World War II.

This was used by them to share battle plans and strategies. If somehow the encryption could be broken, the tides of the war would shift. But none of our older methods were working!

Basically, Hitler's best-kept secrets lay behind, what we shall see, was a math problem.

To solve this this: A different kind of an army, one of mathematicians, engineers, linguists and chess players; was amassed in the small radio factory of Bletchley Park.

What follows is the story of some of history's greatest minds and their work to solve that problem. It is a story of spies and deception; perspectives and puzzles; logic and language and machines built to think faster than any human.

Winston Churchill saw this math problem for what it was: the key to saving millions of lives. Understanding the project's urgency, he issued a memo demanding that the utmost attention and resources be given to the Bletchley team. Churchill tagged this memo with a red stamp reserved for matters of the highest priority. It read: **Action This Day**

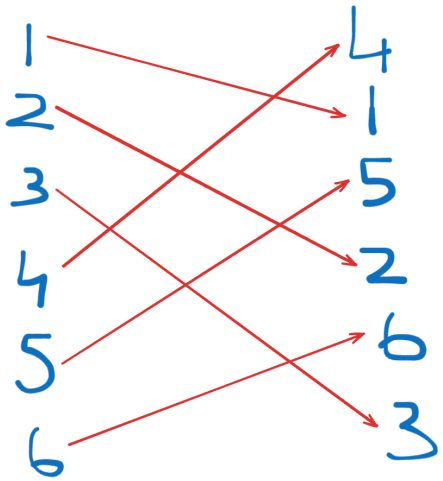
Interlude: Permutation Theory

A Seemingly Unrelated Problem

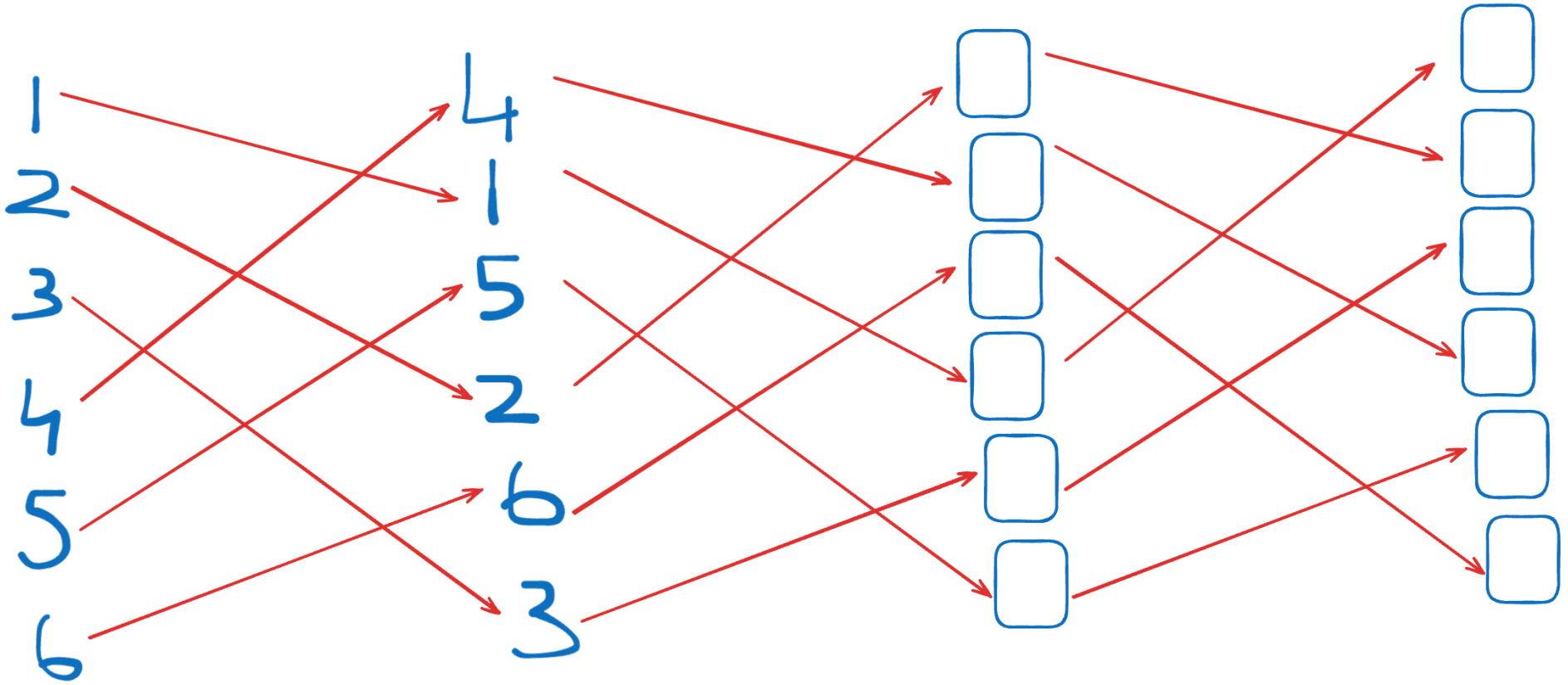
Given a standard deck of cards, an expert magician cuts it into 2 equal piles and then interleaves them (changing the top and bottom card).

How many times must he repeat this move before the deck is in back to the same order as start? Can that even happen?

Let's try to see what is happening here with a smaller example.



The red arrows in some way represent a shuffle. If we wish to shuffle again, we can copy the arrows and move things according to them...


1
2
3
4
5
6
4
1
5
2
6
3

Notice, in 3 shuffles, we are back to where we stated.

A set of these arrows is called a **permutation**. It rearranges the items.

Towards a better notation!

One could argue the arrow notation is actually cumbersome. Maybe instead we should note permutations as pairs (i, j) of where an element starting at index i is mapped to index j.

In the case of shuffle on 6, this would look like $\{(1, 2), (2, 4), (3, 6), (4, 1), (5, 3), (6, 5)\}$.

We can (ab)use the fact that both sides have same number of elements and instead write this as: $(1, 2, 4)(3, 5, 6)$ meaning index 1 goes to index 2, index 2 goes to index 4 and index 4 goes to index 1. Same with $(3, 5, 6)$.

This notation also makes it clear why applying this permutation three times returned us to the same state.

Composing Permutations

Also, we could apply one set of arrows after another. Consider $p_1 = (1, 2, 3)$ and $p_2 = (1, 2)(3)$. If we applied p_1 after p_2, we would get

$$1 \underset{p_2}{\rightarrow} 2 \underset{p_1}{\rightarrow} 3$$

$$2 \underset{p_2}{\rightarrow} 1 \underset{p_1}{\rightarrow} 2$$

$$3 \underset{p_2}{\rightarrow} 3 \underset{p_1}{\rightarrow} 1$$

or p_1 after $p_2 = (1, 3)(2)$.

Similarly, if we applied p_2 after p_1, we would get

$$1 \underset{p_1}{\rightarrow} 2 \underset{p_2}{\rightarrow} 1$$
$$2 \underset{p_1}{\rightarrow} 3 \underset{p_2}{\rightarrow} 3$$
$$3 \underset{p_1}{\rightarrow} 1 \underset{p_2}{\rightarrow} 2$$

or p_2 after $p_1 = (1)(2,3)$. We could represent these using function composition notation that is: f after $g = f \circ g$.

Notice, from the above example, we can see that $p_1 \circ p_2 \neq p_2 \circ p_1$.

Bye-Bye Fixed Points

As a final notational convenience, we can ignore all the fixed points. That is we can write $p_2 = (1,3)(2) = (1,3)$.

This removes the need to specify what n we are working on as anything not mentioned is going to remain where it is. We will denote $(1)(2)(3)...(n) = \mathrm{id}$ as it changes nothing.

As permutation is just a bunch of arrows, we can also reverse them. If $X = \mathrm{rev}(Y)$ then $X \circ Y = Y \circ X = \mathrm{id}$. This is called **inverse** of a permutation and is denoted as $X = Y^{-1}$.

Another side effect of our notation is that we have decomposed our permutations into cycles. One could argue that $(1,2,3)(4,5) = (1,2,3) \circ (4,5) = (4,5) \circ (1,2,3)$.

Substitution Cipher and Frequency Analysis

- Notice, the Caeser Cipher was just us applying the permutation $(abc...z)$ some number of times to the whole message. But this just gave us **25** options.
- That's bad as if someone figures out our algorithm, well, they can just run through the possibilities and crack the code. This is called a **brute force attack**.
- But why limit ourself? What if we could apply any permutation? That is exactly what a **substitution cipher** is.

Algorithm: Apply a permutation to all the letters of an message

Key: The permutation applied

- Notice, now there are $26 \times 25 \times ...2 \times 1 \approx 4 \times 10^{26}$ keys so brute forcing is no longer feasible, neither by hand and nor on modern computers.

Breaking Substitution Clues

- However, context clues might still give it away, eve in short messages. For example:

Zit vgskra ajqkktaz cogkoh stqkkn httra qh qxrothet, ag oy O rg hgz yohr agjtwgrn aggh, O'kk wkgv xf ohzg ajozitsttha

In general, to combat this, we can replace all spaces with ‘x’ and write as 5 letter words (and pad with ‘x’s) and also, ignore grammatical symbols and uppercase. The text from the problem can be written as

zitbv gskra bajqk ktazb cogko hbstq kknbh ttrab qhbqx rothe tbagb oybob rgbhg zbyoh rbagj twgrn baggh bokkb wkgvb xfboh zgbaj ozits tthab

However, one could argue that if one just looks hard enough, the weird placements of one of the letters (in this case *b*) would be apparent and then this is susceptible to the same context decoding as above.

If you thought that way, congratulations, you have hit the nail on the head. All substitution ciphers are susceptible to something called the **frequency attack**.

The First Idea in Statistics and Cryptanalysis

- In 860, an Arab polymath Al-Kindi wrote perhaps the first book on statistics and code breaking. This was a major idea there.
- **What is the most common word in an English text?**

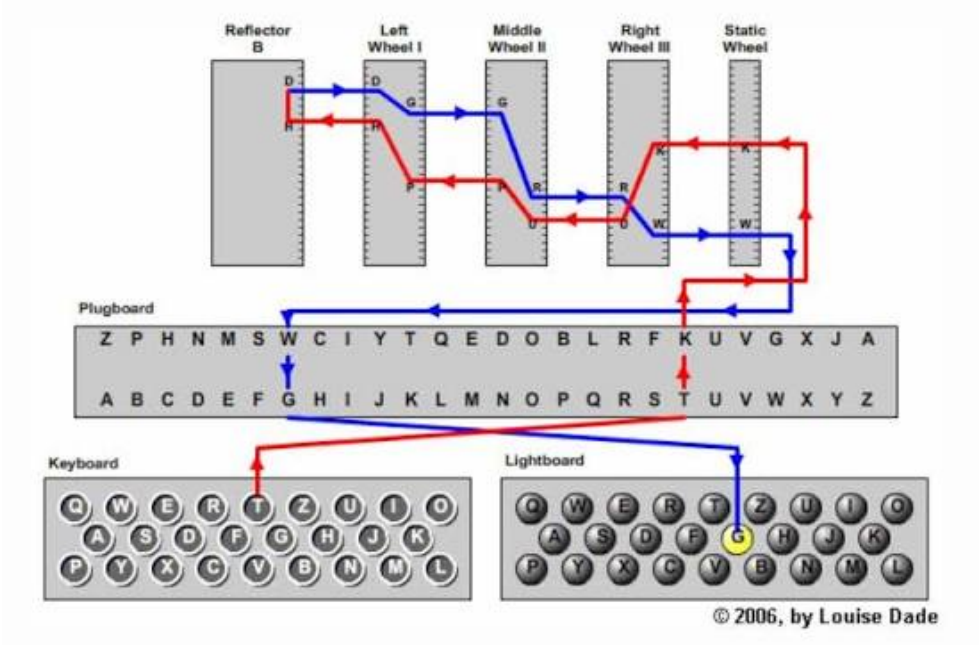
The First Idea in Statistics and Cryptanalysis

- The answer is obviously ‘e’, the next is maybe ‘t’ or ‘a’ and so on.
- Well, the substitution cipher didn’t modify the frequency, it just relabeled them.
- This means, given a long enough message, we can just substitute back the letters by frequency and then maybe fix errors (if any) using context clues and voila, we have broken the message.
- This attack can be made much more lethal using digrams (two letter pairs), trigrams and quadgrams because the odds of a coincidence are rarer there.

The Engima Machine!

The Enigma machine had basically 3 components: Rotors, Reflector, and Plugboard. Let's look at each.


Nur Glühlampen mit 12 mm Durchmesser verwenden


Reflector B
Left Wheel I
Middle Wheel II
Right Wheel III
Static Wheel
Plugboard
Z P H N M S W C I Y T Q E D O B L R F K U V G X J A
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Keyboard
Lightboard
© 2006, by Louise Dade

The Plugboard

The plugboard is simply a panel of 26 sockets. Pairs of letters are connected by cables. If **A** is plugged to **Q**, then every **A** entering becomes a **Q** before going further, and vice versa.

The plugboard typically had 10 cables, swapping 10 pairs of letters. The remaining 6 letters passed through unchanged.

This is just a substitution and not a particularly good one at that as it had the additional structure if $A \to Q$ then $Q \to A$.

By itself, nothing special. But it's our first layer.

The Rotors

This is where Enigma gets interesting. A rotor is a disk with 26 electrical contacts on each side. Inside, each contact on the right is wired to a **different** contact on the left. So a signal entering at position **A** might exit at position **G**.

A rotor is like a substitution cipher encoded in hardware. The signal enters on one side and exits somewhere else on the other.

But why call it a rotor? Cause it turns!

After every single keystroke, the first rotor advances by one position. Like an odometer. When it completes a full revolution, it nudges the second rotor. When the second completes a revolution, it nudges the third.

This means every single letter you type is encrypted with a different substitution. The first **A** you type maps to, say, **G**. The second **A** you type maps to something entirely different, perhaps **T**.

Mathematically, if we think of the rotor as a permutation σ on 26 letters, and the rotor position as r, the effective permutation is:

$$\pi_r = \rho^{-r} \circ \sigma \circ \rho^r$$

where ρ is shift by one position or the permutation $(abcd...yz)$.

The rotor stepping changes r with every keypress.

Three rotors in series means the plaintext letter passes through three such permutations, one after another.

The Reflector

After the signal passes through the three rotors left-to-right, it hits the **reflector**. This is a fixed permutation that pairs up all 26 letters and bounces the signal **back**. The signal then passes through all three rotors again, right-to-left, through entirely different internal paths this time (since it's entering from the other side).

This might seem like a very weird inclusion. Remember, WW2 was in a age before computers so this machine had to work mechanically. This means making completely different encode and decode machines for such a complex system would be logistically difficult and also, impractical to carry.

The reflector had this beautiful effect of making the machine symmetric. If encrypting **A** gives **Z**, then encrypting **Z** gives **A**. This means the **same** machine setup can both encrypt and decrypt. The receiver just needs to set their Enigma identically and type the codetext. Out comes the plaintext. No separate decryption machine or mode needed.

This beauty will be the crack in armour that will allow us to breakthrough the encryption.

How Many Keys?

- **Rotor selection and order** (choosing 3 from 5 rotors): $5 \times 4 \times 3 = 60$ choices
- **Starting positions** (3 rotors, 26 positions each): $26^3 = 17576$ choices
- **Plugboard** (10 pairs from 26 letters): approximately 1.5×10^{14} choices

Multiplied together, this gives roughly: $\approx 10^{23}$

Even if you could check a billion keys per second, it would take longer than the age of the universe to try them all. This is why Germany was so confident.

Furthermore, Germany used more rotors for Navy so this space could be even larger.

If you think about it, Enigma is just a combination of Caesar Cipher and Substitution Cipher in the strictest sense.

The rotors alone don't have a large enough search space to defend against a simple brute force attack but by rotating after every keypress, make the machine less stagnant.

On the other hand, the keyspace is massively increased by the plugboard, but it alone is not very complicated and can be broken easily.

It seems to have the best of both but the weaknesses of neither!

Interlude: Rejewski's Theorem

Math-y terms

- A permutation of order 2 is called an **involution**.
- A **proper involution** on $\{1, 2, \ldots, 2n\}$ has no fixed points, ie no cycles of types $(1), (2), \ldots, (2n)$.

A permutation can be expressed as the product of two involutions without fixed points if and only if every cycle length occurs an even number of times.

This is not very easy to prove. So let's do it step by step.

Lemma 1

Given involutions X and Y such that they don't contain identical transpositions, we can decompose

$$X = X_1 X_2 ... X_k$$
$$Y = Y_1 Y_2 ... Y_k$$

such that

$$X_i = (a_{i,1} a_{i,2})(a_{i,3} a_{i,4}) ... (a_{i,2m_i-1} a_{i,2m_i})$$
$$Y_i = (a_{i,2} a_{i,3})(a_{i,4} a_{i,5}) ... (a_{i,2m_i} a_{i,1}).$$

Proof. We provide the following algorithm to find this decomposition.

- Choose an arbitrary transposition in X and name it (a_1, a_2).
- Find the transposition in Y with a_2 and name it (a_2, a_3). Notice, $a_3 \neq a_1$ as the involutions don't contain identical transpositions.
- Find the transposition in X with a_3 and name it (a_3, a_4). And so on.
- After m_1 iterations, if we find $\left(a_{2m_1}, a_1\right)$ in Y, we name

$$X_1 = (a_1 a_2)(a_3 a_4)...\left(a_{2m_1-1} a_{2m_1}\right) \quad Y_1 = (a_2 a_3)(a_4 a_5)...\left(a_{2m_1} a_1\right).$$

- Repeat the process with $X \setminus X_1$ and $Y \setminus Y_1$.

Notice, that $X \setminus X_1$ and $Y \setminus Y_1$ are still involutions on the same set of elements as we only got rid of the same $a_1, ..., a_{2m_1}$ elements from both of them.

Lemma 1

The algorithm terminates as every iteration of the loop reduces the number of elements in the involutions.

Correctness is obvious and follows immediately.

Lemma 2

For two given involutions X and Y defined on a set S, such that $|S| = 2n$,

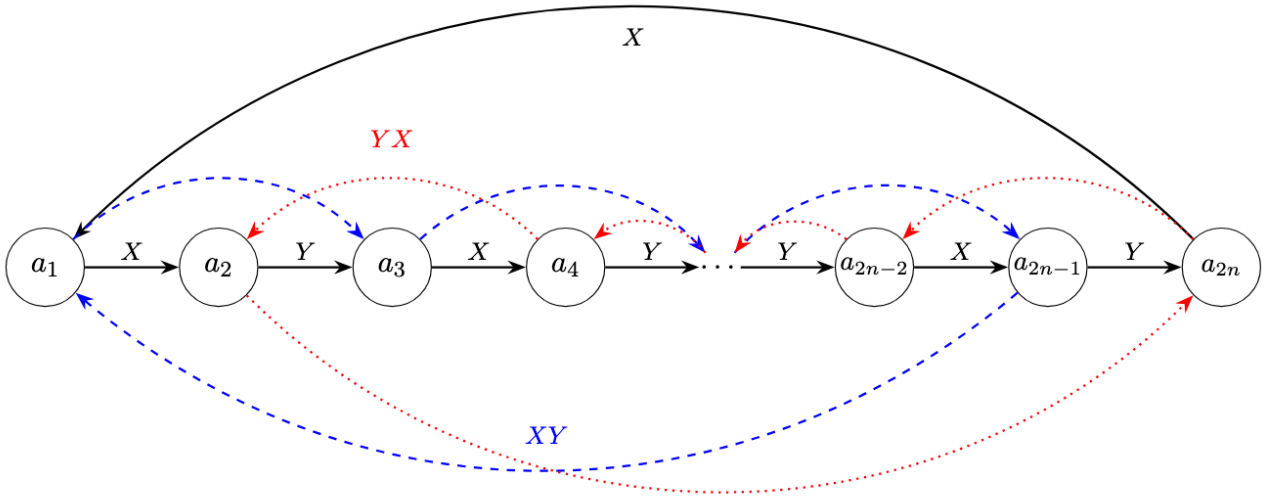
$$X = (a_1 a_2)(a_3 a_4)...(a_{2n-1} a_{2n}) \quad Y = (a_2 a_3)(a_4 a_5)...(a_{2n} a_1)$$

The product of X and Y will consist of two disjoint cycles of equal length n, that is given by

$$XY = (a_1 a_3 ... a_{2n-1})(a_{2n} ... a_4 a_2).$$

Proof.

Lemma 2



It can be seen from the above figure that product of X and Y will form two disjoint cycles of length n, such that

$$XY = (a_1 a_3 ... a_{2n-1})(a_{2n} ... a_4 a_2).$$

Proof of Rejewski's Theorem

We can now prove our theorem!

($\Longrightarrow$) Let X and Y be two proper involutions defined on the set S, that is, $X : S \to S$ is a permutation such that $X(X(a)) = a$ for all $a \in S$, and $Y : S \to S$ is a permutation such that $Y(Y(b)) = b$ for all $b \in S$, both not having any fixed points and $|S| = 2n$.

Let us ignore the identical transpositions between X and Y to form X' and Y'.

Using lemma 1, we can decompose X' and Y' into X_i''s and Y_i''s such that

$$X_i' = (a_1 a_2)(a_3 a_4)...(a_{2k-1} a_{2k}) \quad Y_i' = (a_2 a_3)(a_4 a_5)...(a_{2k} a_1),$$

where $a_i \in S$ for all $i \in \{1, 2, ..., 2k\}$ and $k \leq n$.

Using lemma 2, $X_i' Y_i'$ can be written as product of two cycles of length k, say $C_i C_i'$ where $|C_i| = |C_i'|$ Thus,

Proof of Rejewski's Theorem

$$\begin{aligned} X'Y' &= (X'_1X'_2...X'_m)(Y'_1Y'_2...Y'_m) \\ &= (X'_1Y'_1)(X'_2Y'_2)...(X'_mY'_m) \\ &= C_1C'_1C_2C'_2...C_mC'_m. \end{aligned}$$

Finally, notice that $(xy)(yx) = (x)(y)$.

Thus, product of identical transposition produces 2 singleton cycles of the elements of the transposition.

Thus, product of two proper involutions on the same elements can be expressed as a product of pairs of cycles of the same length.

($\Longleftarrow$) Given $P = C_1C'_1C_2C'_2...C_mC'_m$ such that $|C_i| = |C'_i|$.

We can find involutions X, Y that product up to P as follows:

Proof of Rejewski's Theorem

- If $C_1 = (a_1 a_2 ... a_k)$ and $C_1' = (b_1 b_2 ... b_k)$ add to X the transpositions $(a_i b_i)$ and to Y the transpositions $(b_i a_{i+1})$ along with $(b_k a_1)$.
- Repeat.

This terminates as m is finite.

The correctness is implied by lemma 2.

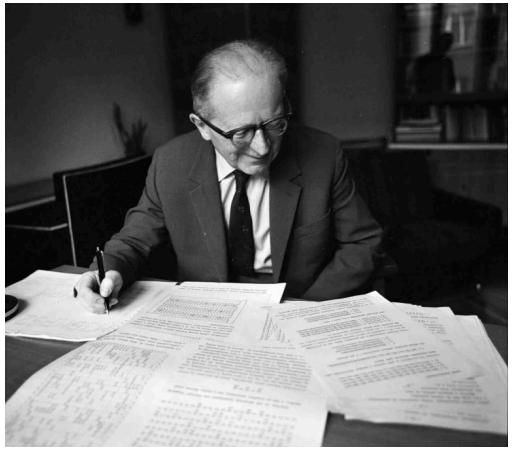
What was all this?

And what does this have to do with Enigma?

The Rejewski Attack

- In cryptography, we usually like to assume the complete encrypting algorithm is given to us.
- Bad news: the Germans were not kind enough. The wirings of the rotors was kept top secret. What the Allies had was how the machine worked but not the internal wirings of the rotors or reflectors.
- Good news: this was the early in the war. This meant spies were quite effective and we had the key sheets (literally a calender of what key was in use) and loads of intercepted messages. Finally, the operators were yet to get comfortable with this machine and would make some errors.

- Marian Rejewski was the Polish mathematician who figured out how to use the information they had and exploit the operator errors to figure out the wiring of the machine.
- He also figured out a very effective attack on the Enigma once the key sheets got secured.



Hexagram

- This attack depended on the fact that every enigma message started with a six letter code, a **hexagram**.
- Let's say the day key is **LKG**. The sender chooses a personal key, say **ABC**. They set their enigma to **LKG** and encodes **ABCABC** and puts it at the top of the message.
- They then set their enigma to **ABC** and encode the rest of the message.
- These allowed a sort of a check of correctness as well as authentication for the receiver. If the 6 letters didn't decode to something of this form, there had to be an error.

The issue is moving the rotors was hard and painful (remember this was a mechanical machine). So a lot of senders just used the same keys over and over. And sometimes they were dumb things like `ABC` or `QWE` or `CCC`.

We now just needed our spies on the inside to tell us about some men who let the things slide.

This meant that once we had intel that some sender used a particular key, these 6 letters told us everything we needed to know.

We can say that the letter at the first position is encoded by π_0 as:

$$\pi_0 = P^{-1}\sigma_1^{-1}\sigma_2^{-1}\sigma_3^{-1}R\sigma_3\sigma_2\sigma_1 P$$

This expression, despite it's rather toothy appearance, is just the working of enigma written using our notations.

Assuming that we don't face a turnover in the first 6 letters (ie the second rotor turning), we can say

$$\begin{aligned}\pi_i &= P^{-1}\left(C^{-i}\sigma_1^{-1}C^i\right)^{-1}\sigma_2^{-1}\sigma_3^{-1}R\sigma_3\sigma_2\left(C^i\sigma_1 C^i\right)P \\ &= P^{-1}C^i\sigma_1^{-1}C^{-i}\sigma_2^{-1}\sigma_3^{-1}R\sigma_3\sigma_2 C^{-i}\sigma_1 C^i P \\ &= P^{-1}C^i\sigma_1^{-1}C^{-i}QC^{-i}\sigma_1 C^i P\end{aligned}$$

where $C = (abc...z)$ and $Q = \sigma_2^{-1}\sigma_3^{-1}R\sigma_3\sigma_2$.

Notice, if the hexagram at the start of the message is **HQVRWS** and suppose the operator's key was **XYZ**.

$$\pi_0(X) = H$$
$$\pi_1(Y) = Q$$
$$\pi_2(Z) = V$$
$$\pi_3(X) = R$$
$$\pi_4(Y) = W$$
$$\pi_5(Z) = S$$

This implies $\pi_3(R) = X$ (involution) and thus, $\pi_0\pi_3(R) = H$ and same for $\pi_1\pi_4$ and $\pi_2\pi_5$.

This means given enough hexagrams (around 85), we can get the cycle structure of $\pi_0\pi_3, \pi_1\pi_4$ and $\pi_2\pi_5$.

Note, we can sort of detect if a turnover occurred as these have to have every cycle length occur an even number of times due to Rejewski's Theorem. So the error has to have been in the violating cycles.

This is sort of enough to break the Enigma, only if we had the wiring. Let's suppose we do.

Let’s hope future Arjun doesn’t mess this up!

So what did we do?

- Using some magician fiat, I had the deck arranged as:

A♠ K♠ Q♠ J♠ 10♠ ... A♥ K♥ Q♥ J♥ 10♥

- After your cut, in guise of counting, we are left with

A♠	10♥
K♠	J♥
Q♠	Q♥
J♠	K♥
10♠	A♥
⋮	⋮

Notice, no matter how you shuffle, the top 5 cards will be an ace high straight!

This is called the **Gilbreath's Pricipal**.

Properties Preserved Upon Shuffling

- A lot of magic tricks (particularly the ‘self working’) ones rely on certain properties of permutations being preserved by composing with certain other family of permutations. As it turns out shuffling is not really ‘random’!
- If we can find some property that is preserved by the enigma family of permutation, maybe an attack can be mounted?

Properties Preserved Upon Shuffling

Given a permutation P, $(c_1, c_2, ..., c_n)$ is characteristic of P where c_i is the count of i-cycles in P. We shall denote the characteristic of P as $\text{char}(P)$

For permutations P, Q, $\text{char}(P) = \text{char}(Q)$ if and only if exists a permutation X such that $X^{-1}PX = Q$

Proof of Conjugacy-Characteristic Theorem

Proof. ($\Longleftarrow$) If $\text{Char}(P) = \text{Char}(Q)$, then we can construct an X such that $X^{-1}PX = Q$ as follows:

- For each cycle of length k in P, pick a cycle of length k in Q.
- Define X to map elements of P cycle to the Q cycle, in order.

This terminates as P has finitely many cycles and the correctness is obvious.

($\Longrightarrow$) We have $Q = X^{-1}PX \iff Q(s) = X^{-1}PX(s)$.

Take any cycle in P, say $a_1 \to a_2 \to \ldots \to a_k \to a_1$.

Let $X^{-1}(a_i) = b_i$.

Note, as X^{-1} is also a permutation, we can't have $X^{-1}(a_i) = X^{-1}\left(a_j\right)$ if $i \neq j$.

Notice,

Proof of Conjugacy-Characteristic Theorem

$$
\begin{aligned}
& Q(b_i) \\
&= X^{-1}PX(b_i) \\
&= X^{-1}P(a_i) \\
&= X^{-1}a_{i+1} \\
&= b_{i+1}
\end{aligned}
$$

This implies we have a cycle in Q of the form $b_1 \to b_2 \to \ldots \to b_k \to b_1$.

By induction, this can be done to all the cycles of P and thus, $\text{char}(P) = \text{char}(Q)$.

This has an immediate consequence:

For permutation P and involution V, $\text{char}(VPV^{-1}) = \text{char}(P)$.

This basically tells us that the plugboard settings don't affect the characteristics of $\pi_0\pi_3, \pi_1\pi_4$ and $\pi_2\pi_5$. That means only the rotor settings affect this and that is a small enough space for us to catalog.

Rejewski made a machine named Cyclometer that (using the rotor wirings) just went through all the rotor settings and cataloged what the 3 characteristics would be.

We still have a plugboard to decode, but as we discussed, the plug board is easy to break using our ideas of frequency analysis.

Um, but we don't have the wirings!

Remember, we had mentioned that some operators used to use bad keys like **CCC** or **VVV** or **QWE**. This means, we could basically figure out $\pi_0, \pi_1, ..., \pi_5$.

So we have these 6 knowns to figure out the 3 unknows P, σ_1 and Q. Problem is, we don't really know a way to solve these kind of equations on permutations...

So what do we do? Well, remember we had the day keys for a huge number of days. This means we have the plugboard settings. So we really are looking at only 2 unknowns, σ_1 and Q.

Um, but we don't have the wirings!

Let $P\pi_i P^{-1} = P\pi_i P = \varphi_i$.

We have:

$$\begin{aligned}
\varphi_0 &= C^0\sigma_1^{-1}C^{-0}QC^{-0}\sigma_1 C^0 \\
\varphi_1 &= C^1\sigma_1^{-1}C^{-1}QC^{-1}\sigma_1 C^1 \\
\varphi_2 &= C^2\sigma_1^{-1}C^{-2}QC^{-2}\sigma_1 C^2 \\
\varphi_3 &= C^3\sigma_1^{-1}C^{-3}QC^{-3}\sigma_1 C^3 \\
\varphi_4 &= C^4\sigma_1^{-1}C^{-4}QC^{-4}\sigma_1 C^4 \\
\varphi_5 &= C^5\sigma_1^{-1}C^{-5}QC^{-5}\sigma_1 C^5
\end{aligned}$$

Notice,

Um, but we don't have the wirings!

$$\varphi_0^{-1}\varphi_1 = \sigma_1^{-1}Q\sigma_1 C\sigma_1^{-1}C^{-1}QC^{-1}\sigma_1 C$$
$$\varphi_1^{-1}\varphi_2 = C^{-1}\sigma_1^{-1}CQC\sigma_1 C\sigma_1^{-1}C^{-2}QC^{-2}\sigma_1 C^2$$
$$\varphi_2^{-1}\varphi_3 = C^{-2}\sigma_1^{-1}C^2QC^2\sigma_1 C\sigma_1^{-1}C^{-3}QC^{-3}\sigma_1 C^3$$
$$\vdots$$

Doing this again (painfully!) and some math later, we can eliminate Q and recover σ_0.

I will not be doing that in full. But I suppose it makes sense how this can over a period of days expose the wiring of all the rotors.

Turing's Attack

Unfortunately, starting 1940, the German's made one small change.

The message key would only be enciphered once. This meant we could no longer figure out the cycle structure of $\pi_i \pi_{i+3}$ and use our Cyclometer catalogue. So what does one do?

Remember, we had observed that the Enigma never enciphered a letter to itself? This means if we could find a word that had to be a part of the message, we can try to align it and see if none of the letters match. If that was indeed the case, then take that as our hypothesis and start solving from there.

Originally, the plan was to intercept weather reports or do light firing at some front and intercept communications after. But this was inefficient and well, how many times can there be random shelling at the Maginot Line before they get suspicious?

This problem was alleviated when an internal memo was leaked. Hitler's hubris and self-importance had led to a critical mistake: every communication was required to end with "_____"

However, as keys were changing every 8 hours now and messages were transported in much more security, using the older method of waiting till we have the cycle was simply infeasible.

A fact Turing abused was that the rotor configurations were pretty limited and the thing stopping a simple brute force was the plug board.

Let's say using a letter, we have the following pieces of data:

$$H \underset{32}{\mapsto} T$$
$$E \underset{33}{\mapsto} Q$$
$$\vdots$$
$$H \underset{36}{\mapsto} U$$
$$\vdots$$
$$R \underset{42}{\mapsto} W$$

Please note, the letters are chosen randomly and any resemblance to some extremely politically, morally and simply incorrect phrase is coincidental.

We make the assumption that rotors **I, II, III** are being used with the key **ABC**.

By this time, it was usual to use a plug board of 13 wires that is all letters were connected to something. So let's make the assumption that $H \leftrightarrow A$.

Passing 'A' through the rotors and plugboard with said key would say outpur 'R'. This means $R \leftrightarrow T$.

Let's now we use the second H and pass A through the rotors moved some 4 times and get say T. This would imply $T \leftrightarrow U$

But that is absurd as $R \leftrightarrow T$ and $T \leftrightarrow U$ are both deduced. Thus, our initial connection must be wrong. We try again with $H \leftrightarrow B$.

If none of these work, our key must be wrong. We try again with **ABD**.

If still nothing works, our rotors must be wrong. We try again with **I, III, II**.

Notice, this could take $\binom{5}{3} \times 3! \times 26^3 \times 25 = 26364000$ tries.

And that is not counting how long each deduction goes on for.

This would take loads of time, so what do we do?

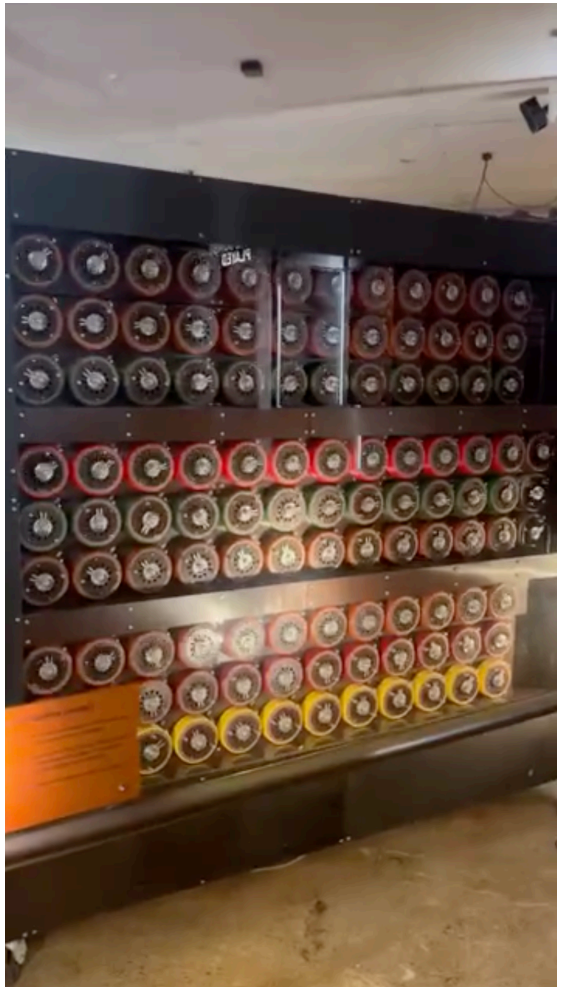


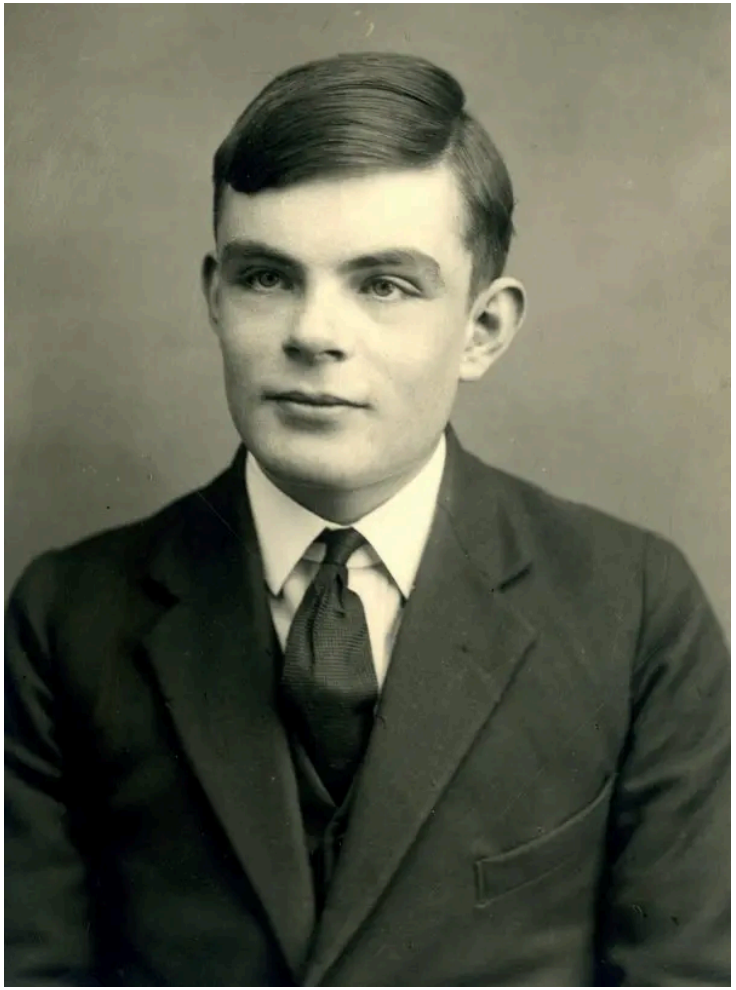
Figure 7: From Bletchley Park Museum, clicked by Ketaki Sahasrabudhe (CMI)

Bombe

Alan Turing, unlike a lot of other logicians, was interested in circuits and using electrical current to make logical deductions.

During his time at Princeton, he had designed come circuits trying to emulate logical equations (though he was clumsy) to various degrees of success.

So at Bletchley Park, they made this huge machine called the Bombe that could emulate 36 Enigma machines at one.



While a 36 times smaller is good, we would still require ≈ 700000 tries.

A further reduction was found by reducing the number of initial plugboard settings we need to try as well as increase the speed of refutation. Notice, if we have

$$H \leftrightarrow A \Rightarrow R \leftrightarrow T \Rightarrow T \leftrightarrow U : \textbf{Refuted!}$$

then $H \leftrightarrow B \Rightarrow T \leftrightarrow U$ can also be refuted without wasting time looking for a contradiction as $T \leftrightarrow U$ lead to a contradiction elsewhere (with the same rotor setting and key).

Gillgoly's Attack

However, after all this, there were still some messages that neither Rejewski or Turing were able to decipher. Rejewski's methods were made moot by the removal of hexagrams and Turing's method were either too time consuming (just out of coincidence) or no words they guessed would match.

In 1995, James J. Gillogly of RAND decided to take a crack at it (pun intended).

Rejewski ignored the plugboard and tried to figure out the rotors first and use frequency analysis on the plugboard. While Turing tried to figure out the plugboard by brute forcing over the rotors.

This indicates that neither one of these alone was strong enough encryption.

Another Magic Trick!

Let's hope future Arjun doesn't mess this up...

How did that work?

- To begin with, the deck was stacked in an order such that given a card, I can tell you both the neighbors.
- When I asked you give the deck a shuffle behind your back, I was sort of taking a chance you don't really know a one handed riffle shuffle.
- This means somewhere along maybe 5-10 cards are separated from their neighbors.
- And after that, it was just keeping fingers crossed! (I had a separate out...)

- Something I want to hint at is that a single riffle shuffle or cuts or over-hand shuffle is not that good cause some amount of initial structure can be retained. So what is a good shuffle? It's random.
- Well, how do we measure randomness?
- Persi Diaconis has a lot of work on this and it is a really interesting topic in mixing markov chains and cut-off functions.

- However, we can give a much simpler answer for text ciphers.
- A good encryption is almost random, otherwise frequency analysis would probably be able to break it. Which means, a bad encryption is sort of less random.
- One way to measure randomness here is called the **index of coincidence** which is the probability that two randomly chosen letters from the message will be the same. We can calculate this as

$$IC = \frac{\sum_i f_i(f_i - 1)}{N(N-1)}$$

where f_i is the frequency of some letter i.

- For a random string of characters, $IC \to \frac{1}{26} \approx 0.03846154$ as $n \to \infty$, while for standard English, it is 0.0667.

Gillgoly's Attack

As the plug board only shuffles around the letters, it doesn't affect the index of coincidence. This means we can brute force the rotor combinations (keeping the same day key) and choose the ones with the highest index of coincidence.

Then brute force the day key and choose the ones with the highest index of coincidence.

Now with only the plugboard remaining, we can perform standard frequency analysis.

Notice, this attack didn't use any knowledge about the message other than the fact it was written in English!

This is a cipher-text only attack. No hexagrams or cribs needed!

Conclusion

With the age of digital communications and transactions, Cryptography is no longer a military specific idea. It is everywhere now.

Since then there has also been a lot of development; from stronger ciphers to better methods of breaking them.

One major change was the choice of transmitting numbers in place of words. There are far less patterns to observe with long streams of numbers.

This has also made this field much more mathematical, with a lot of work being championed by abstract algebraists and number theorists.

We are still cracking Enigma!

There is this bimonthly journal named Cryptologia that deals with a lot of historic ciphers, coding schemes and ways to crack them. In 2023, the editor's note read:

"In the second issue of Cryptologia, way back in 1977, a contributor noted, "The golden age of decipherment may have been the first half of the nineteenth century, …

But there's a new and different golden age of decipherment that we are presently in the midst of, namely the recovery of messages intended to be kept secret and therefore hidden behind the best ciphers of the time."

That issue had yet another mordern attack at Enigma which combining frequency analysis along with methods like Hill Climbing and Gentic algorithms, solved some of shorter codes that had resisted other attacks.

And there are still more messages left! (needless to say even other ciphers stronger than Enigma!)

Thanks You!
Any Questions?

References

- Karandikar, R. *The Mathematics of Ciphers*, CMI Lecture, Lodha Genius (2025).
- throwittomebro. *How was the Zimmermann Telegram encrypted?*, r/AskHistorians (2018).
- Ostwald, O. & Weierud, F. *Modern breaking of Enigma ciphertexts*, Cryptologia 41(5), 395–421 (2017).
- Bauer, C. P. *The new golden age of decipherment*, Cryptologia 47(2), 97–100 (2023).
- Guballa, J. *SubstitutionBreaker*, GitLab (2021).
- Guballa, J. *Substitution Solver*, guballa.de (n.d.).
- Pound, M. *Cracking Enigma in 2021*, Computerphile (2021).
- Grime, J. & GingerGM. *How Chessplayers helped Crack Enigma and Win WW2*, Outray Chess (2020).
- Brailsford, D. *Tackling Enigma (Turing's Enigma Problem Part 2)*, Computerphile (2014).
- *Flaw in the Enigma Code*, Numberphile (2013).
- Hoffstein, J., Pipher, J. C. & Silverman, J. H. *An Introduction to Mathematical Cryptography*, Springer (2008).

References

- Poskitt, K. & Baker, I. *Codes — How to Make Them and Break Them*, Scholastic (2007).
- *Cyclometer*, Wikipedia (2025).
- Weinbaum, J. *Action This Day: The Mathematics and Machinations that Bested the German Enigma*, Dartmouth College Master's Theses (2025).