The History of the Twentieth Century

Episode 339

“The Enigma Machine”

Transcript

[music: Fanfare]

During the 1920s, the German military came to learn exactly how successful the British had been at decrypting their coded messages during the First World War, and this led them to search for a more secure way to send coded messages. By the Second World War, they had what they believed to be a system for sending coded messages that could not be broken.

It was called Enigma.

Welcome to *The History of the Twentieth Century.*

[music: Opening War Theme]

Episode 339. The Enigma Machine.

Today we are going to talk about cryptography and cryptanalysis. Or to put it in plain English, writing messages in code and breaking the code to read the message, respectively.

One simple way of encoding a message is by substituting letters. For example, you might decide to change every A in your message into an H, and every B in your message into a Y, and so on, until you have a table in which each and every letter of the alphabet is assigned to its own unique code letter. If you write your message using the code letters, it will come out as gibberish. Other people will not be able to decipher your message unless they have a copy of the table that tells which code letter represents which letter in the message.

I already know what you’re thinking. It is, in fact, possible to decipher such a message without a copy of the table. In fact, in the United States there is a daily syndicated newspaper feature called “Cryptoquote,” which consists of a saying from a well-known person coded in exactly this way, which is offered to the reader as a puzzle to solve. You can buy puzzle books that include cryptograms for you to decipher; you can find web pages and apps that offer you cryptograms to solve for your entertainment.

If people are doing cryptanalysis of this kind of code for recreation, that's a pretty clear sign that's it isn't all that difficult. How is it done? Well, in entertainment cryptograms, the spaces between the words are unchanged, which gives the person attempting to break the code some useful information right off the bat. For example, there are only three one-letter words in the English language: "a," "I" and maybe "o." If you see a one letter word in the cryptogram, you already know that one letter must represent either *a* or *i* or possibly *o*.

You could make the coded message much harder to decode by simply eliminating the spaces between the words and presenting it as one long string of letters, but it would still be possible to break the code. For example, *e* is the most common letter in the English language, so if you are presented with this long string of letters, you could try counting how many times each code letter appears, and it's a safe bet that the one that appears most often represents *e*; if it doesn't represent *e*, it probably represents *t* or *a* or *o*, which are the next three most common letters in English.

Okay, so a simple one-to-one letter substitution code is pretty easy to break. But let's put that issue to the side for a moment and think about something else. I want you to consider how you might use a substitution code, or technically, a substitution cipher, in the day-to-day communication of an organization. Let's say, for example, that you are the head of a business that has multiple offices. These offices have to communicate with each other, but your competitors are monitoring these messages in the hope of stealing some of your trade secrets, so you want to use a substitution cipher to keep your trade secrets safe. As a practical matter, how would you instruct your employees to send and receive messages in the cipher? You could just hand out tables and tell them to work out the coded message letter by letter, by hand, but that would take a lot of time and introduce a lot of mistakes into the messages.

Here's a simpler method. Let's imagine a device that looks like a typewriter. Does anyone but me remember what a typewriter looks like? Never mind. Imagine an ordinary keyboard in the QWERTY arrangement. Now imagine that just behind the keyboard is a set of lights. Each light has a letter of the alphabet printed on it, and they are laid out in the same QWERTY arrangement as the keyboard. When you press one of the keys on the keyboard, one of the lights comes on.

You could wire them directly, as in A lights up A, B lights up B, and so on, but what good is that? Imagine though, that you wire this machine so that A lights up H and B lights up Y and so on. Now you have a simple machine that anyone can use to encode a message. They just type in the message letter by letter, and then write down which lights come on—or maybe an assistant does that for them—and there's your coded message.

How would someone on the other end decode the message, though? Here's a simple solution to that problem: the letter assignments go both ways, by which I mean if A is encoded as H, then H would be encoded as A, and so on. If you use that system, then the same machine that encoded the message could also decode that message. So if you built a lot of these machines and distributed them among your offices, your employees could send and receive coded messages to each other much more quickly and accurately.

The weakness of this system is that this simple substitution cipher is easy to break. And even worse, since the substitutions are hard-wired into your encryption machines, once one of your competitors breaks your code, they will be able to decode every message your company ever sent, or will send in the future, at least until you build and distribute a whole new set of machines.

But what if the substitutions weren't hard wired into the machine? Imagine this same machine, but let's add a new feature. Below the keyboard, on the front side of the machine, there are 26 holes, each hole marked with a letter and laid out in the same QWERTY arrangement as the keyboard and the lights. These holes are jacks, and you also have a set of 13 cables with a plug on each end that fit the jacks, and if you plug one end of the cable into the B hole and the other end into the Y hole, this will cause the machine to light up Y when you push B and light up B when you push Y.

The machine is now programmable for different substitutions. Now you could distribute a list of cable connections to all your offices, let's say a different set of cable connections for each day of next month, and instruct your employees to change the cable settings on the machine every day, per the instructions on your list.

Now, even if your competitor manages to break your code, they will only be able to read one day's worth of messages. Tomorrow your company will be using a different code and today's code will be useless.

All right, that's an improvement, but many people can solve a simple substitution cipher like this in a matter of minutes, so changing the code every day is not that big an impediment. But let's take the machine we already have and think about how we can move beyond simple substitution and make the code much harder to break.

Let's add a rotor to our machine. This rotor sits inside the machine and it has 26 possible settings, one for each letter of the alphabet, and it is inserted into the wiring of the machine. Suppose that the rotor has its own internal wiring that swaps the letters around in the same way as the cables on the front do. What goes in one side of the rotor as an A comes out the other side as an H and vice versa.

In other words, it does the same job as the cables. But here's the brilliant bit. Every time you press a key on the keyboard, one of the letters lights up. Every time you release the key, the rotor turns one notch. This means the substitution code changes with every letter. So for example, if you were sending the word SEE, S-E-E, it would come out the other end as three different letters, Q-H-F maybe. The double *e* would not produce a double code letter, because the rotor moved between the first *e* and the second *e* and changed the code. This would not only eliminate double letters, it would also mess up the letter distribution. Someone trying to break the message couldn’t simply look for the most common letter in the message and assume that was *e*, because the *e*s in the message were encoded into different letters.

But the recipient of the message will have the code machine, and they will know how to set the cables and how to set the rotor for today by looking at the list of settings you previously distributed, and they will be able to decode the message easily.

The machine I’m describing here is fairly easy to use but the messages it produces are nearly impossible to break.

But what if “nearly impossible” isn’t good enough for you? I have a few more suggestions that will make this machine positively diabolical. Suppose instead of one rotor, it has three, and the letter to be coded gets changed three times, once by each rotor. Suppose every time you release a key on the keyboard, the rotor on the right turns one notch, and also when the rotor on the right has made a complete revolution, it turns the middle rotor one notch. When the middle rotor makes one revolution, it turns the left rotor one notch.

And let’s add a fourth rotor, which we’ll call a reflector. When the three rotors encode a letter, the coded letter goes into the reflector and right back out again into the three rotors. The three rotors will then each scramble the letter a second time, and the signal that comes out the other end will determine which light comes on. Let’s also suppose that the three rotors can be removed and changed or swapped around, so that not only the rotor setting but the rotor order will change from day to day.

During the Second World War, the German government and the Wehrmacht had exactly the code machine I have just described to you. They called it Enigma. The total number of possible settings on an Enigma machine is on the order of 10^{20}. That means that if you wanted to decode a message encoded by Enigma, even if you had your own Enigma machine to work on, it would do you no good if you did not also knew what today’s settings were. If you tried to crack the code by trial and error on your Enigma machine, and you could test one setting every second, which is faster than anyone could possibly work, the Universe would end before you were finished.

In other words, it is absolutely impossible for a human being to decrypt a message sent by Enigma. Full stop. End of story. The Germans had a code they believed was impossible to break, and they had every reason in the world to believe that.

[music: Mozart, Rondo alla turca]

The Enigma machine was invented by a German electrical engineer named Arthur Scherbius during the First World War. He filed for a patent in February 1918. Scherbius was only one of four inventors who came up with the same basic idea, using a rotor machine for encryption and decryption, at about the same time. The other three were in Sweden, the Netherlands, and the

United States. It must have been the war that inspired all this thought about how to send and receive coded messages.

Scherbius would prove the most successful inventorof this group. He founded a company with another engineer called Chiffriermaschinen, Aktiengesellschaft, or Cipher Machines, Inc., in Berlin. The new company began marketing their cipher machine under the trade name Enigma in 1923. The first Enigma machines were the size of a cash register and weighed 50 kilograms, about 110 pounds. In later models, the company was able to pare that down to a much smaller portable device, built inside a wooden case that could be closed up and carried by its handle, like a small briefcase. These early models had the three rotors and the reflector, but not yet the plugboard.

Scherbius tried to market his machine to the German military of the early post-Versailles period, but initially found no interest, so his company offered the device to businesses. There is a commercial market for an encryption machine, as I suggested earlier.

You may recall from our Great War episodes that the British had been highly successful at decrypting German coded messages during the last war. These decryption efforts were centered in Room 40 of the Admiralty building on Whitehall in London, and their most important success was the decryption of the Zimmerman Telegram, the event that gave the United States the final push to enter the war on the Allied side. I talked about this in episode 134.

As the 1920s progressed, the German military came to learn exactly how successful the British had been at decrypting their coded messages during the war and this increased their interest in improved encryption, and the Enigma device was the state of the art in encryption. The German Navy adopted Enigma in 1926; the German Army in 1928. That same year, 1928, the plugboard was introduced. If you think about it, the plugboard is effectively another rotor, but unlike the other rotors, this one can be reprogrammed.

It is the plugboard that truly makes Enigma effective. Without the plugboard, you have only the three rotors, each of which can be set in one of 26 positions, for a total number of settings of 26x26x26, which equals 17,576. Oh, and did I mention that the three rotors in the Enigma machine are interchangeable? This means that arranging the rotors in the correct sequence is also part of the settings you need to know to decrypt an Enigma message. Three rotors can be arranged in six possible ways, so that increases the total number of possible Enigma settings to 105,456.

That sounds like a lot, but if we’re talking military applications here, where cracking the code is a high-stakes business and people are willing to invest a lot of resources, it is not insuperable. You could crack the code by brute force, meaning someone sits at an Enigma machine and tries to decrypt a coded message using every possible setting, one at a time. If it takes an Enigma operator 30 seconds to test a setting, decide whether it is the right setting or not, and if not, move on to the next setting, then the total process would take less than 900 hours. If you had multiple

Enigma machines, say ten, and teams of operators to run them around the clock, you could decode the message within four days. Most likely it would take only one or two. That's doable.

It is the plugboard that makes Enigma diabolical. Even though Enigma machines never used more than ten cables out of a potential 13, since each cable connects two letters, those ten cables are what increase the possible settings from about 10^5, which is a one with five zeroes after it, to 10^{20}, which is a one with twenty zeroes after it. It is the plugboard that makes it impossible to defeat Enigma using brute force, even if you know how the machine works and have access to a number of them.

The particular value of Enigma at this time was linked to the increasing importance of radio in military communications. Radio was valuable in communicating with ships at sea and with Army units in the field. Radio allowed for rapid response in combat, such as calling in a dive bomber attack on a fortified enemy position. This was especially important for the German Army, whose doctrine emphasized rapid movement.

If the German Army simply sent their reports over the radio in spoken German, the enemy would be able to eavesdrop and the element of surprise would disappear. The technology to scramble voice communications was in its infancy. The equipment was bulky, making it unsuitable for use in the field, and it was fairly simple for an enemy to decrypt anyway.

On the other hand, radio signals in Morse code sending messages encrypted by Enigma offered a communication system that was simple, easy to use, and impossible for an enemy to decrypt. Enigma machines were simple, portable, and they could be manufactured in quantity. I've had the opportunity to push a few keys on an Enigma machine myself, and I can vouch for the fact that they are easy to use.

Enigma machines became the pride of the German military and famous in military intelligence circles all over the world. Alas for poor Arthur Scherbius; he died in a horse-drawn carriage accident on a Berlin street in 1929, at the age of 50, just as his invention was really taking off.

The three foreign intelligence agencies most interested in mastering Enigma were in France, Britain, and Poland. The British were monitoring German naval communications. The French and the Polish, just by reason of geography, were in an excellent position to monitor German Army communications, especially Poland, which had a long and convoluted border with Germany. The Polish Corridor sits between East Prussia and the rest of Germany, for goodness' sake. And then there were German naval exercises in the Baltic. But no one in any of these countries could make sense out of communications encoded by Enigma.

Enter Hans-Thilo Schmidt. He had been an officer in the German Army during the First World War. After the war, he was discharged from the Army, and found himself unemployed, but his big brother, Rudolf Schmidt, who retained a position as a staff officer in the Army, found Hans-Thilo a civilian job in the military, which happened to be in the Cipher Office.

I guess his job didn't pay very well, because when the new military version of Enigma, the one with the plugboard, was introduced, Schmidt contacted French intelligence and in 1931 offered to sell them copies of instruction manuals and other information related to Enigma.

The recipient of this offer was Captain Gustave Bertrand of French military intelligence. He was the head of Section D, D for *decryptment,* or decryption. These documents gave the French a clear picture of what Enigma looked like, including their first photograph of an Enigma machine with a plugboard, and instructions on how to use it, but not the internal wiring of the machine, which might give a code-breaker a leg up on how to decrypt messages encrypted with Enigma. Without that information, his colleagues in French intelligence told him, the information he had acquired was interesting, but practically useless. Bertrand shared this intelligence windfall with France's British and Polish allies, but the British couldn't see any way to make use of it either.

But in the Polish Cipher Bureau, headquartered in Warsaw, they had an idea.

[music: Mozart, *Rondo alla turca*]

Even before they had received the information from the French regarding Enigma, somebody in the Polish Cipher Bureau had the bright idea that mathematicians would make good cryptologists. In 1929, the Bureau created a course on cryptology at the University of Poznań. Students who did well in the course were invited to take part-time jobs working for the Cipher Bureau.

Okay, mathematicians make good cryptanalysts; that tracks. But why go all the way to Poznań? Why not the larger and more prestigious University of Warsaw, just down the street from the Cipher Bureau? Remember that before the last war, the city of Poznań was called Posen and was part of the German Empire. A graduate student at the University of Poznań in 1929 would have been a grammar school student back then, which means they would have been studying in a German school where German was the only language taught or spoken and children were punished for speaking Polish. In other words, virtually every grad student at the University of Poznań was fluent in German, which is also a skill you would want in someone working to crack German codes.

For whatever cultural or historical reasons, there have been a lot of Polish mathematicians. Wikipedia has a list of 130 notable Polish mathematicians, which is interesting to me for reasons that you will understand if you remember episode 180, in which I spoke about being Polish-American and being frequently told while growing up that Polish people were stupid and that no Polish person in all of history had ever accomplished anything notable, so thank you Wikipedia for proving all those people wrong.

Three of the 130 Polish mathematicians on Wikipedia's list are Marian Rejewski, Henryk Zygalski, and Jerzy Różycki. All three of them were students at the University of Poznań when they were recruited by the Cipher Bureau. In 1932, when the Cipher Bureau got those materials from the French, these three young men were offered full-time positions working for the Cipher Bureau in Warsaw. Among their assignments was breaking the Enigma machine.

It was Rejewski who first identified the crack in Engima's armor, but to explain what it was, first I have to tell you a little about German procedures for using Enigma. Enigma operators were given a specified setting to use every day, but it would have been dangerous for every Enigma operator to use the same set of settings, as that would give an enemy too many samples and make it too easy to work out which settings were in use. So Enigma operators were told first to choose their own settings for each individual message. This would be a string of three letters representing the settings for the three rotors. Since I'm an American, let's say for example that the operator chooses USA as the settings for this message.

Because Enigma-encoded messages were sent out over the radio in Morse code, and sometimes Morse code messages get garbled by the radio or misunderstood by the receiver, Enigma procedure called for the operator to transmit the three-letter code twice, to make sure the receiver got it. Obviously, you wouldn't send it in the clear, as in USAUSA, because anyone listening in who had an Enigma machine would now know which settings you are using. So they encoded the settings using the daily setting, which the receiver would already have. So they would encode USAUSA through Enigma using the daily setting and it might come out something like RFEAVO. The recipient of the message would decode those first six letters back to USAUSA and would then know that they needed to change their rotor settings to USA to read the rest of the message.

But whoever decided to encode the three letters twice in succession made a critical error. They should have had the operator encode the settings once and transmit it twice. So if USA encodes to RFE, they should have transmitted RFERFE. But instead they encoded the settings twice and broadcast both versions; RFEAVO.

Marian Rejewski realized that this procedure tells you something about the rotors, even if you have no idea how to decode the message. The rotor on the right, the one that changes with every key press, produced an R for the first letter, and then three changes later, encoded that same letter as A. It produced an F for the second latter, then three changes later, encoded that same letter as a V.

Once you've collected a sufficient number of encoded messages and put all these clues together, you can work out the wiring in the rotor on the right. That's just one of three rotors, but wait, every three months the Germans change the order of the rotors, so now you can work on another one. Using these techniques, Rejewski was able to work out the internal wiring of the three Enigma rotors without ever seeing one.

German radio operators helped Rejewski by making poor choices in their rotor settings. Sometimes two operators at different stations chose the same three-letter key for their message. Rejewski realized that it would be highly improbable that two different people would independently come up with the same random choice of letters; therefore the choice was not random. These operators were choosing easy letter combinations, like AAA, for their rotor settings, which provided valuable clues. This would be the 1932 version of choosing the word *password* as your password. All that elegant security, brought down by laziness.

But Rejewski got stuck on the problem of how to take the information about the rotor wiring and use it to build an actual Enigma simulator. His formulas were getting inconsistent results, and Rejewski thought he knew why. It had to do with the order in which the letters on the Enigma keyboard are wired to the rotors. In the commercial Enigma machine, which anyone could buy and examine, the keyboard letters were wired in keyboard order, that is: QWERTY and so on. Clearly, the military version of Enigma was wired differently, and on the face of it, the only way to find out would be to test every possible wiring sequence, and there are 26! possibilities, which works out to roughly $4x10^{26}$. It would be impossible to test every one of them.

Would it be possible to guess the wiring order? Remember Rejewski spoke German. He went to a German school as a child. He'd had German teachers. He was acquainted with how Germans think. And the answer came to him: how about alphabetical order? That proved to be the answer. With this information, the Polish Cipher Bureau was able to build replica Enigma machines.

By the end of 1937, the Poles were able to decrypt 75% of the Enigma communications they intercepted. But as the Poles were making progress, the Germans were also making their work harder. In November 1937, the Germans changed the reflector disk on the Enigma machine, which meant the Poles had to do substantial recalculations, which took time.

Henryk Zygalski worked out a method of using a set of perforated sheets with a 51x51 grid. Each sheet represented a set of Enigma rotor settings. By stacking certain of these sheets on a light table and moving them around in the right sequence, the holes in one sheet would be covered by another sheet until ultimately you found just one place where holes on every sheet lined up, which would tell you the rotor setting.

In 1938, Rejewski developed an automated device for testing Engima settings. Enigma machines had three rotors, which could be arranged in six different ways. Rejewski's machine had six sets of rotors, simulating all six possible rotor permutations. The machine would automatically cycle its six sets of rotors and compare the output to an intercepted message. If it detected inconsistencies, it would keep going. When it got to a place where one of the rotor sets was generating code that seemed to match the intercepted message, the machine would stop and signal a human cryptographer to evaluate the setting. If it didn't seem right, the machine could be started up again and continue its work.

The Poles called this device *bomba kryptologiczna*, the cryptologic bombe. Why it was given this name is unclear. One story claims that Jerzy Różycki, the most junior of the three mathematicians, gave it that name, and he was not thinking of bombs in the military sense but in the culinary sense, that is, the ice cream dessert known as a bombe. Another story claims the name is derived from the rhythmic clicking sound the machine made as it ran, reminiscent of a ticking time bomb. I like the first story better, but the second one is more plausible.

The Polish Cipher Bureau's Enigma project was producing amazing results, and no one else knew it. This was perhaps the most closely guarded secret in the Polish government. But the Germans continued to improve Enigma; at the end of 1938, they added two new rotors and increased the number of plugboard cables. This was the end of the line for the Poles. Now Enigma operators would choose three out of a possible five rotors for their machines, meaning instead of six possible rotor combinations, there were now 60. Zygalski's perforated sheets and Rejewski's bombe were now ineffective. The Cipher Bureau would have needed to cut hundreds more Zygalski sheets or build a bombe with 60 sets of rotors, rather than six. Neither of these options was feasible for the Cipher Bureau and its limited resources.

These improvements to Enigma couldn't have come at a worse time for the Poles, as 1939 was the year the German government began its propaganda campaign against Poland, and by summer, war was a distinct possibility. Germany repudiated its non-aggression pact with Poland, while France and Britain pledged to go to war to defend Poland's borders.

In the last week of July 1939, the Cipher Bureau invited representatives of French and British intelligence to a series of meetings, which was held in a suburb outside Warsaw, as a precaution against a surprise German bombing of the capital. At these meetings, the Poles shared with their astonished French and British colleagues everything they had learned about Enigma, including giving each ally one of the Polish-built Enigma replicas. The French and British pledged to help the Poles construct the Zygalski sheets and the bombes that would be necessary to crack the new and tougher Enigma.

Five weeks later, the Second World War began. The Cipher Bureau and a truckload of their equipment were shipped across the Polish border into Romania. The three mathematicians were afraid that if they were identified, the Romanians might hand them over to the Germans, so they posed as civilians, escaped internment, and traveled to Bucharest, where they went straight to the British Embassy. The staff at the British Embassy told them it would be much more convenient if they came back in a few days, if you'd be so kind. They didn't have a few days, so they went to the French Embassy. Fortunately, the French understood the urgency of the situation and arranged for the three Polish mathematicians to be evacuated to France.

There is of course more to this story, but we'll have to stop here for today. I thank you for listening, and I'd especially like to thank Julie and Christopher for their kind donations, and thank you to Gerald for becoming a patron of the podcast. Donors and patrons like Julie and Christopher and Gerald help cover the costs of making this show, which in turn keeps the podcast available free for everyone, so my thanks to them and to all of you who have pitched in and helped out. If you'd like to become a patron or make a donation, you are most welcome; just visit the website, historyofthetwentiethcentury.com and click on the PayPal or Patreon buttons.

The podcast website also contains notes about the music used on the podcast. Sometimes it's my own work, sometimes it's licensed, but many times, the music you hear here is free and downloadable. If you hear a piece of music on the podcast and you would like to know more about it, including the composer, the performers, and a link to where you can download it, that would be the place to go. While you're there, you can leave a comment and let me know what you thought about today's show.

I'm pleased to be able to tell you that a short story of mine appears in the recently released fantasy anthology, *Artifice and Craft*. It's a collection of stories about magical artifacts. It is available as an ebook or a paperback at Amazon, Barnes and Noble and Kobo.

And I hope you'll join me next week, here on *The History of the Twentieth Century,* as we continue the story of Enigma. After the war began, the job of breaking Enigma fell to the British, who deployed their secret weapon: the mathematician Alan Turing. The Ultra Secret, next week, here, on *The History of the Twentieth Century*.

Oh, and one more thing. I just said that the job of breaking Enigma fell to the British after the fall of Poland, but what became of Marian Rejewski, Henryk Zygalski, and Jerzy Różycki? The French brought them to Paris, where they continued to work on Enigma for French intelligence.

After France fell, French intelligence moved them to Vichy, where they continued to work for the French government until Vichy was occupied by the Germans. Jerzy Różycki died in January 1942. He had made a visit to Algiers as part of his work for French intelligence, and was on his way back to France when the passenger ship he was on sank in a storm, killing 222 passengers including Różycki and two other Poles who had formerly worked for the Polish Cipher Bureau. At the time of his death, Różycki was 32 years old.

When Vichy France fell to the Germans, the other two mathematicians, Marian Rejewski and Henryk Zygalski, were wanted by the Gestapo. They narrowly avoided arrest and escaped to Spain, and then to England, where they worked for the Polish government in exile. The British Ultra project had by this time advanced far beyond anything the two of them were capable of alone, so they spent the rest of the war working on decrypting Soviet codes.

After the war, Henryk Zygalski opted to remain in England. He lectured on mathematics at the University of Surrey, but the Official Secrets Act forbade him to speak of his role in breaking Enigma. He died in 1978, at the age of 70.

Marian Rejewski had a wife and two children in Poland, so after the war he chose to return to his native country, where he kept a low profile, as he was investigated repeatedly by the Polish security ministry, who were suspicious of his wartime connections with the Polish government in exile and the British. In the 1970s, when the story of Ultra became public, Rejewski began to publish articles and speak publicly about his role and the role of his fellow Poles in breaking

Enigma, a story that was being minimized in the West. In 1978, the Polish government awarded him a medal for his work. He died in 1980, at the age of 74.

[music: Closing War Theme]