The Enigma Cipher Machine

Anna Hurlbut

2018-11-27

Abstract

The Enigma cipher machine was invented in the early 1920s to help businesses protect commercial secrets, but the German military soon saw its potential application to military communications. By the mid-1930s, every branch of the German military was using Enigma for nearly all their encrypted communications. The electromechanical cipher machine works using a system of wires, rotors, a reflector and a plugboard to send an electrical signal from the keyboard to the lampboard, encrypting each message letter by letter. Enigma masterfully balances ease of use and security, the opposing forces of any cryptosystem. The key space is approximately 15 million million million, and it could be compared to a Vigenére cipher with a key length of 16900. Though it seemed infallible, the infamous machine eventually failed. While the cryptosystem did have inherent technical flaws, its defeat was mainly a result of errors in human operation. British intelligence at Bletchley Park was able to find patterns in German messages and used these patterns as cracks to break into the entire system. Enigma represents a vital moment in the field of cryptography. It initiated the transition from by-hand methods to automated and mechanized methods and acted as a catalyst for the development of computers. Additionally, Enigma's ultimate failure illustrated the inadequacy of security through obscurity and pointed the future of cryptography towards more advanced methods.

Contents

1 Historical Context

The Second World War provided a catalyst for technological and mathematical progress. Out of the war effort was born innumerable improvements in various fields, and cryptography was no exception. The field of cryptography is one that stretches far back in human history, nearly to the beginning of language itself. For as long as mankind has sought to communicate with one another, the communicators have sometimes attempted to do so in secret. The word *cryptography* comes from the Greek words *kryptos*, meaning hidden, and *graphia*, meaning writing. It is an ancient field with a long history of innovations, but the primitive methods of pre-war cryptography were ill fitted to the fast pace of the Second World War. During World War I, military communications were accomplished primarily using codebooks and clunky, time-consuming, and confusing methods of encrypting and decrypting messages by hand.[3] Consequently, out of necessity arose the advent of machine ciphers. After their defeat in the Great War, Germany realized that its intelligence operations were greatly lacking—a large contributor to their failures—and recognized need for improved military communications and intelligence.[3] German Marine Captain Heinz Bonatz stated, "Neither the German High Seas Fleet nor the Naval War Staff hit upon the idea that it was German naval radio traffic which supplied the British their knowledge."[3, p. 13] It was clear that the current methods would no longer be adequate, especially since the German military forces needed rapid and frequent communications to execute their new Blitzkrieg war tactics.[3] They needed a way to encrypt and decrypt messages that was quick and easy to use, but also secure and nearly impossible to break.

Hugo Alexander Koch, a Dutch inventor, designed and constructed the Enigma machine just after World War I for protecting commercial secrets in the business world. The patent was soon purchased by Arthur Scheribus, a German manufacturer, who also marketed the machine to the business world.[3] In 1926, the German Marines took up Enigma, and the army did so shortly after. They made numerous modifications and improvements to increase security, each military branch eventually ending up with a slightly different version of the machine.[3] By the mid-1930s, all branches of the German military were using Enigma for nearly all of their encrypted communications.[3]

Polish intelligence made great strides in cracking Enigma throughout the latter half of the decade by rebuilding Enigma's internal structure and analyzing the signals they received, but as the German invasion of Poland loomed, the Poles passed off their acquired intelligence to the British and French.[3] The British then set up facilities at Bletchley Park, about fifty miles north of London, and stationed men and women of the Government Code and Cypher School there to devote undivided attention to German radio communications.[3]

Though it initially seemed unbreakable, the British and their allies took advantage of Germany's human errors and found much success deciphering messages by looking for repeated or predictable segments and using these to uncover the rest of the message.[3] The decrypted messages and intelligence gleaned from them were given the code name "Ultra" and were used with much discretion.

Ultra intelligence could only be acted on if there was a plausible cover story regarding how the Allies got the information. Still, decrypted Enigma messages made it possible for the Allied forces to "avoid waiting U-boats, anticipate surprise attaches, and send their own troops to the German's most vulnerable points."[3, pg. 4] However, the Germans were so sure of Enigma's security that they failed to question the Allies' mounting intelligence, crediting Allied power knowledge and success to other possible information leaks, exactly as the British intended.[3] Even after war ceased, as Enigma rumors spread, German military leaders obstinately asserted its infallibility. They reasoned that the Allies would be publicly boasting of a broken Enigma had they succeeded in decrypting German codes. After all, who could stand to keep such an accomplishment to themselves? The Germans also claimed they would have seen evidence of Enigma-cracking in intercepted and decoded Allied communications. Therefore, they presumed Enigma to be in tact and German cryptography to be entirely secured. This insistence persisted even into the 1970s when the Allies finally announced publicly that they had indeed cracked Enigma during the war.[3]

So how did Enigma work exactly? Why were the Germans so sure of its invulnerability? Why were they wrong? How did the analysts and mathematicians at Bletchley Park manage to break the unbreakable code? This, and more, we will explore, but before we can confront the infamous Enigma machine, we must begin with some cryptography basics.

2 Cryptography Basics and Simple Substitution Ciphers

One of the simplest examples of a cryptosystem is a simple substitution cipher, of which there are many forms. The basic premise of a substitution cipher is to encrypt each letter in the alphabet to a predetermined letter. No two letters can be encrypted to the same letter. In other words, it must be an injective function.[2] For this type of encryption to work, it is essential that both parties, the sender and the receiver, know the substitution alphabet being used. Otherwise, the receiver would be unable to decrypt the message. Consider the following example.

Let Table 1 indicate the encryption alphabet. Assume that Alice would like to send a message that says, "I LOVE CRYPTOGRAPHY," to her friend Bob. This original message in readable form is known as the *plaintext.*[2] Before she sends the message, Alice must encrypt it, so she uses the encryption alphabet to encrypt each letter in the plaintext message to its corresponding letter in the second column of the table. Starting from the beginning, I is encrypted to X, L is encrypted to P, O is encrypted to O, and so on. She continues this process until she has encoded the entire message so it reads, "X POGJ HFMRDOYFSRAM." This is known as the *ciphertext* and is what is sent to Bob. Once Bob receives the message, he does the same process in reverse to recover the plaintext, "I LOVE

Table 1: Substitution Cipher Encryption Alphabet

Letter	**Encrypted As**
A	S
B	C
C	H
D	V
E	J
F	K
G	Y
H	A
I	X
J	U
K	L
L	P
M	Z
N	E
O	O
P	R
Q	B
R	F
S	T
T	D
U	W
V	G
W	I
X	N
Y	M
Z	Q

CRYPTOGRAPHY." Bob can then send a response using the same method.

In this type of cipher, the encryption alphabet forms the *key*, the information needed to transform plaintext into ciphertext and vice versa.[2] A large factor in determining the security of a cryptosystem is its *key-space*, which is the number of possible keys. How big is the key-space for a simple substitution cipher using the English alphabet? The answer can be calculated using a simple combination. First of all, the letter A can be encrypted to any one of the 26 letters in the alphabet. Next, B can be encrypted to any of the remaining letters (25 possibilities). There are 24 choices for C, and we continue on until we get to Z, at which point there is only one remaining letter that we can use. Therefore the size of the key-space is $26! = 403,291,461,126,605,635,584,000,000$. A random guess of the encryption alphabet is pretty unlikely. Even a *brute force attack*, systematically trying every possible key, is nearly impossible. If one could use a computer to test 1 billion keys per second, it would still take 12,753,347,700 years to try each possible key.

2.1 Caesar Cipher

One well-known version of a simple substitution cipher is the Caesar Cipher, so named because it was used by Julius Caesar in his military communications.[2] In a Caesar cipher:

1. Alice and Bob agree on a key k, which is an integer between 0 and 25.
2. Alice writes the plaintext message and shifts each letter k places in the alphabet to form cipher text.
3. She sends ciphertext to Bob, who decrypts by shifting each letter k places backwards. [2]

Going back to our previous example, let's use a key 5 to encrypt Alice's message, "I LOVE CRYPTOGRAPHY." We can also represent this key in a tabular form, which is shown in Table 2.

Following the same process as the previous example, the message is encoded to "N QTAJ HWDUYTLWFPMD" and is sent to Bob, who decrypts by shifting each letter 5 places backwards. Though this system is convenient for Alice and Bob—each only has to know the English alphabet and remember an agreed upon number—it is also very easily broken by anyone who intercepts the message and understands the nature of the encryption. The small key-space of 26 means that a single person could easily discover the key relatively quickly with nothing more than a pen and paper.[2]

As it turns out, though, the Caesar cipher and every other simple substitution cipher are relatively easily broken using a technique known as *frequency analysis*. This method is based on the fact that certain letters, such as E and T, are used more frequently in the English language than others, like Q and X. Table 3 lists the relative frequency of letters in the English language.

The frequency analysis technique is relatively easy to execute. Assume Alice sends an encrypted message to Bob, and it is intercepted by Eve. Looking at the ciphertext, Eve can assume that the letters used most frequently correlate to commonly used letters in the English Language. For example, if G is the most common letter in the ciphertext, Eve could try assuming that E is encrypted to G. Using the process in a trial and error manner, Eve will likely be able to piece together Alice's original message. As we can see, frequency analysis is a straightforward way to decipher any simple substitution cipher. Therefore, it becomes necessary to introduce a more complicated cryptosystem, which is not susceptible to basic frequency analysis.

3 The Vigenére Cipher

One method to protect against frequency analysis, to add another layer of security, is to establish a cryptosystem that is not an injective function. What if we

Table 2: Caesar Cipher Encryption Alphabet

Plaintext	**Ciphertext**
A	F
B	G
C	H
D	I
E	J
F	K
G	L
H	M
I	N
J	O
K	P
L	Q
M	R
N	S
O	T
P	U
Q	V
R	W
S	X
T	Y
U	Z
V	A
W	B
X	C
Y	D
Z	E

could devise a system so that every L in the plaintext is not encrypted into the same letter? One L becomes P, another A, and another H. Similarly, what if every G in the ciphertext does not come from the same plaintext letter? One G is an encrypted F, while another is an encrypted N. This may seem clunky and nearly impossible. How will Bob deccrypt the message if one ciphertext letter can refer to many plaintext letters? But with the Vigenére Cipher, the process is actually quite simple.

The Vigenére Cipher is so named because it was found in Blaise de Vigenére's 16th century book, which describes numerous ciphers. Vigenére is a *polyalphabetic cipher*, meaning that each letter is encrypted using a different ciphertext alphabet than the preceding letter.[2] It works like this:

1. Alice and Bob agree on a key word or phrase.

2. Each letter of the key determines how far to shift each plaintext letter. (A

Table 3: Frequency of Letters in English Text[?]

Letter	**Frequency**
E	13.11%
T	10.47%
A	8.15%
O	8.00%
N	7.10%
R	6.83%
I	6.35%
S	6.10%
H	5.26%
D	3.79%
L	3.39%
F	2.92%
C	2.76%
M	2.54%
U	2.46%
G	1.99%
Y	1.98%
P	1.98%
W	1.54%
B	1.44%
V	0.92%
K	0.42%
X	0.17%
J	0.13%
Q	0.12%
Z	0.08%

indicates a shift of 0, B indicates a shift of 1, and so on.)

3. The key is repeated until all of the plaintext has been encrypted.

To better understand how this works, let's look at an example. Assume Alice wants to encrypt the message "CIPHERS ARE FUN" using the key word "APPLE". Table 4 depicts the encryption process.

The result is that the message is encrypted to "CXESIRH PCI FJC". The decryption process is nearly identical. Notice that the two R's in the plaintext were encoded to different letters. This is one of the major advantages of Vigenére. Because of this, it cannot be deciphered by simple frequency analysis. However, it is still far from unbreakable.

Table 4: Vigenére Encryption

Key	**Shift By**	**Plaintext**	**Ciphertext**
A	0	C	C
P	15	I	X
P	15	P	E
L	11	H	S
E	4	E	I
A	0	R	R
P	15	S	H
P	15	A	P
L	11	R	C
E	4	E	I
A	0	F	F
P	15	U	J
P	15	N	C

3.1 Vigenére Cryptanalysis

The first step in cryptanalysis of a Vigenére cipher is to determine the length of the keyword. One approach is to use the *Kasiski method*, developed by Friedrich Kasiski in 1863.[2] This method involves searching for repeated fragments in the ciphertext and recording the distances between these fragments. It is likely that the key length divides these distances. Another method can be used to check if the key length found using the Kasiski method is correct or to guess and check various key lengths when the ciphertext contains no repeated fragments, rendering the Kasiski method unhelpful. This method begins by assuming the key length is k. Extract every kth letter from the ciphertext, beginning with the 1st letter, and concatenate them into a single string of letters. If k is indeed the length of the key, each letter in this string will have all been encrypted using the same letter of the key, making it a Caesar cipher. In order to check if this is truly the key length, we can compare the relative frequencies of letters to what is expected based on letter frequencies in the English language, represented in Table 3. One method of accomplishing this task is using a metric called the *index of coincidence*, which is the probability that two randomly chosen characters in a string are identical.[2]

Let $s = c_1c_2c_3 \cdots c_n$ be a string of n characters, and let letters $a, b, \cdots, z$ be represented by the numbers $0, 1, \cdots, 25$. For a value $i = 0, 1, \cdots, 25$, let F_i be the number of times the letter represented by i appears in the string s. The number of possible ways to choose two of the ith letter from s is found by the combination $\binom{F_i}{2} = \frac{F_i(F_i-1)}{2}$. To get the total number of ways to choose any 2 identical letters, we must sum this value for every i. Divide this sum by the number of ways to choose any 2 letters from s, $\binom{n}{2} = \frac{n(n-1)}{2}$, and we have a formula for index of coincidence.[2]

$$\text{IndCo}(s) = \frac{1}{n(n-1)} \sum_{i=0}^{25} F_i(F_i - 1)$$

If the string consists of random letters, the probability that one randomly chosen character is the same as another is $\frac{1}{26} \approx 0.0385$ since there are 26 letters in the alphabet. In English text however, some letters are much more likely than others. In a string consisting of 10,000 characters, the following index of coincidence would be expected.[2]

$$\frac{815 \cdot 814 + 144 \cdot 143 + 276 \cdot 275 + \cdots + 8 \cdot 7}{10000 \cdot 999} \approx 0.0685$$

Though the index of coincidence will fluctuate from one string of text to the next, we can generally assume that strings of random letters will have an index of coincidence closer to 0.0385 while strings of English text will have an index of coincidence closer to 0.0685.

To determine the key length of a message encrypted with Vigenére, we must extract characters from the text at various intervals of length k—every kth letter for $k = 5$, $k = 6$, and so on—and form strings from these characters. The k for which the corresponding strings of characters have index of coincidences nearest to 0.0685 is likely the key length.

Knowing the key length k, we can again group the letters of the ciphertext so that we have multiple Caesar ciphers. For the example when $k = 5$, we should have 5 strings. If c_i is the ith character in the ciphertext, then our 5 strings will be

$$s_1 = c_1 c_6 c_{11} \cdots$$
$$s_2 = c_2 c_7 c_{12} \cdots$$
$$s_3 = c_3 c_8 c_{13} \cdots$$
$$s_4 = c_4 c_9 c_{14} \cdots$$
$$s_5 = c_5 c_{10} c_{15} \cdots$$

To determine the key exactly, we must compare each of these strings to one another using the *mutual index of coincidence.* The mutual index of coincidence for two strings is the probability that 2 randomly chosen letters, one from each string, will be the same.[2]

Let $|s|$ be the number of characters in a string s. Let $F_i(s)$ be the number of times the ith letter appears in string s. Then the number of ways to choose the ith letter from both strings, s and t, is $F_i(s)F_i(t)$. The total number of ways to choose the same letter from both strings is the sum of this product for all possible values of i: $\sum_{i=0}^{25} F_i(s)F_i(t)$. Divide this by the number of ways to choose one letter from each string, and we get the definition for mutual index of coincidence.[2]

$$\text{MutIndCo}(s,t) = \frac{1}{|s||t|} \sum_{i=0}^{25} F_i(s)F_i(t).$$

The mutual index of coincidence can be interpreted in this way: Strings encrypted by the same substitution cipher will have a large mutual index of coincidence since the frequencies of various letters are expected to be the same between the two strings. Conversely, two strings encrypted by different substitution ciphers should have a small mutual index of coincidence.

Now assume we have a ciphertext with a known key length of k and have divided that ciphertext into k strings. Let's say that string s_i is shifted by some amount β_i. Then if we have 2 strings, s_i and s_j, we can shift s_i by $\sigma = \beta_j - \beta_i$. After this additional shift, s_i will have the same shift as s_j.

So in order to determine the key, we must compare s_i with s_j shifted by various amounts σ in a process of ascertaining when $\sigma = \beta_j - \beta_i$. Let $s_j + \sigma$ notate the string that results when the characters in s_j are shifted by σ. To determine when $\sigma = \beta_j - \beta_i$, we are looking for a value of σ that makes makes the mutual index of coincidence of s_i and $s_j + \sigma$ large. We compute mutual indices of coincidence for values of σ from 0 to 25, and select those which are larger than 0.065. For these values of σ, it is likely that $\beta_i - \beta_j \equiv \sigma \mod 26$. This results in a system of equations.

$$\beta_2 = \beta_1 + \sigma_2$$

$$\beta_3 = \beta_1 + \sigma_3$$

$$\beta_4 = \beta_1 + \sigma_4$$

$$\vdots$$

$$\beta_k = \beta_1 + \sigma_k$$

Now the shift value for each string s_2, s_3, $\cdots$, s_k is written in terms of the shift value for s_1. Therefore, we have effectively decreased our keyspace to 26. All there is left to do is try each key, shift value for s_1, until one results in the plaintext.[2]

4 The Enigma Cipher Machine

Enigma is an electromechanical machine, accomplishing ciphering and deciphering using wires, rotors, lights, and electrical currents.[3] Its structure can be broken down into six major components. In order to understand the machine as a whole, it is helpful to first examine each component on its own.

4.1 Keyboard

The first element of the Enigma machine is the *keyboard.* The keyboard is nothing unfamiliar. It is simply a normal typewriter keyboard with wires connecting each letter to the plugboard.

4.2 Plugboard

The *plugboard* is the first step of encryption and basically functions as a symmetrical substitution cipher. Enigma's plugboard had an electrical port for each letter, and operators were given 10 chords to connect various letters. Essentially, 10 letters would be swapped with 10 other letters. For example, imagine the operator was told to connect the 10 wires in the following way: AZ, BP, CH, DN, EM, FS, GW, JY, KT, LQ. Then the plugboard would essentially be a simple substitution cipher with the encryption alphabet depicted in Table 5.

Table 5: Plugboard Encryption

Plaintext	**Ciphertext**
A	Z
B	P
C	H
D	N
E	M
F	S
G	W
H	C
I	I
J	Y
K	T
L	Q
M	E
N	D
O	O
P	B
Q	L
R	R
S	F
T	K
U	U
V	V
W	G
X	X
Y	J
Z	A

Notice that some letters—in this case I, O, R, U, V, and X—are encrypted to themselves. The other 20 letters in the alphabet are swapped with another letter.

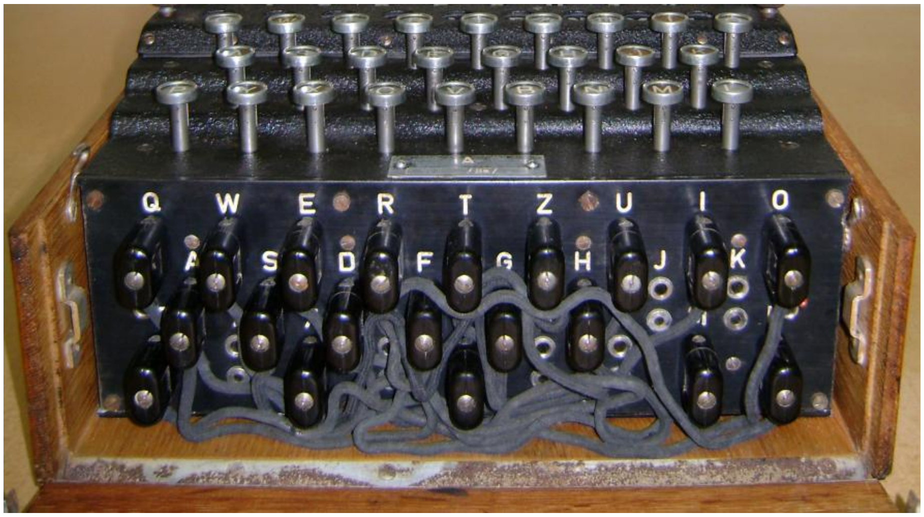

Figure 1: Enigma Plugboard

4.3 Static Wheel

The next element of Enigma is the static wheel. This piece does nothing for encryption purposes. It serves as a connecting piece between the plugboard and the rotors.

4.4 Rotors and Scrambler

The three rotors form the *scrambler*, which is arguably the most vital component of Enigma. This element is where the bulk of the encryption takes place. Each rotor, on its own, is essentially another simple substitution cipher. Conceptually, it consists of two alphabets, one on each face of the rotor, and a wiring maze in the middle connecting each letter to one letter in the other alphabet. For example, one rotor could use the encryption alphabet shown in Table 6. In this case, if the electrical current enters the rotor through the C port, it will leave the rotor through the M port.

In the Enigma machine, one side of a rotor has 26 copper pins where the electrical current enters, one for each letter, and the other side had 26 copper contacts where the current continues into the next rotor. Three rotors are placed side-by-side, copper pins of one touching the contact points of the next (pictured in Figure 2). As the electrical current travels through the second rotor, it is transformed by a different substitution cipher and then transformed yet again by the third rotor.

While having three back-to-back substitution ciphers sounds complex, it is in reality no more secure than having one substitution cipher. To illustrate this, let Table 7 represent the substitution alphabets of three rotors and consider the following example.

Table 6: Rotor Encryption

Plaintext	**Ciphertext**
A	E
B	K
C	M
D	F
E	L
F	G
G	D
H	Q
I	V
J	Z
K	N
L	T
M	O
N	W
O	Y
P	H
Q	X
R	U
S	S
T	P
U	A
V	I
W	B
X	R
Y	C
Z	J

When the electrical current enters through port C, it is sent through port M. Then in the second rotor, M is sent to W. Finally, in the third rotor, W is sent to U. This process is effectively the same as simply sending C directly to U in the first place. Therefore, any encrypted message using simple substitution ciphers, even three of them in a row, is still susceptible to frequency analysis.

However the true genius of the Enigma scrambler lies in what rotors are designed to do: rotate. After a character is typed, rotor I rotates one notch, altering the positions of the copper pins and contact points in relation to the static wheel, the other two rotors, and the reflector (to be discussed). The internal wiring maze remains the same, but the entry point of the electrical current is shifted.

After the operator has typed 26 characters—meaning rotor I has made a complete rotation—rotor II also rotates one notch. Rotor I then makes another complete rotation with the rotor II in this new position. When rotor I completes

Table 7: Encryption by Three Rotors

Plaintext	**Rotor I Ciphertext**	**Rotor II Ciphertext**	**Rotor III Ciphertext**
A	E	A	B
B	K	J	D
C	M	D	F
D	F	K	H
E	L	S	J
F	G	I	L
G	D	R	C
H	Q	U	P
I	V	X	R
J	Z	B	T
K	N	L	X
L	T	H	V
M	O	W	Z
N	W	T	N
O	Y	M	Y
P	H	C	E
Q	X	Q	I
R	U	G	W
S	S	Z	G
T	P	N	A
U	A	P	K
V	I	Y	M
W	B	F	U
X	R	V	S
Y	C	O	Q
Z	J	E	O

another rotation, rotor II clicks one more notch forward. Rotor II eventually makes a complete rotation—$26^2 = 676$ characters into the message—at which time rotor III rotates one notch. Note that the design of the rotors causes a *double-step* at this point. In other words, rotor II and rotor III both shift one notch at the same time. Then the whole cycle repeats again. Another 675 characters later, rotor III shifts again one more notch.

Finally, after $26 \times 25 \times 26 = 16900$ characters have been typed, the three rotors will all be back in their original positions, and the cycle starts over from the beginning. This system could be compared to a Vigenére cipher with a key length of 16900. As an additional measure of security, the German military had eight rotors with different internal wiring mazes. Any permutation of three could be used at once. That is $\frac{8!}{(8-3)!} = 336$ possibilities.

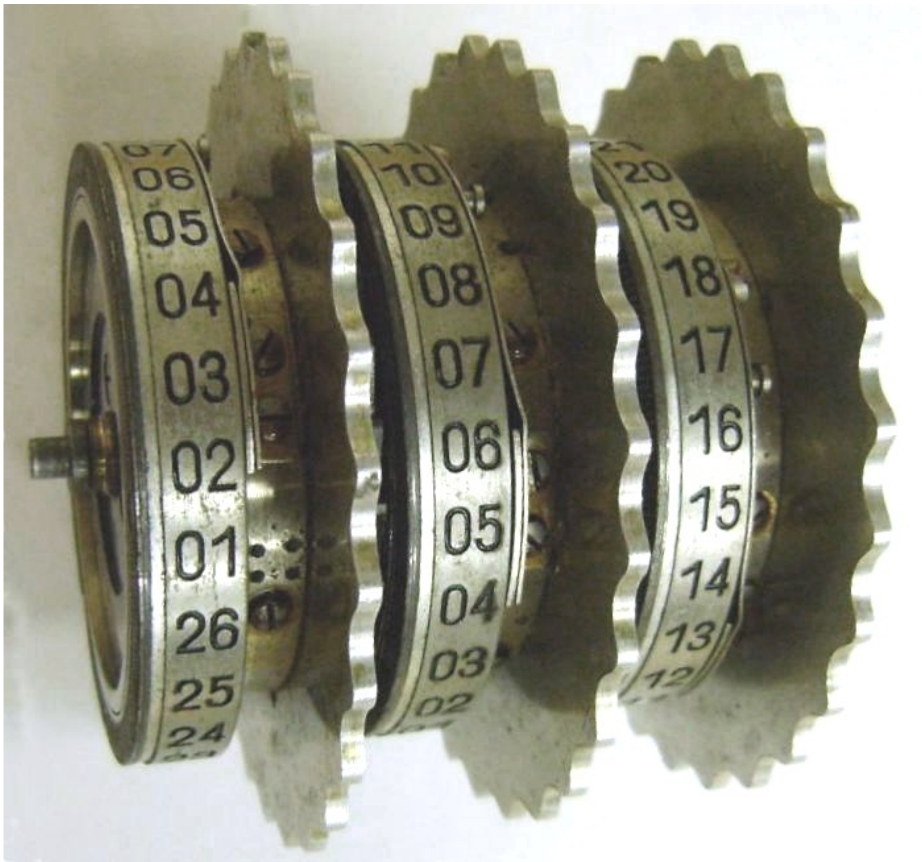

Figure 2: Enigma Rotors

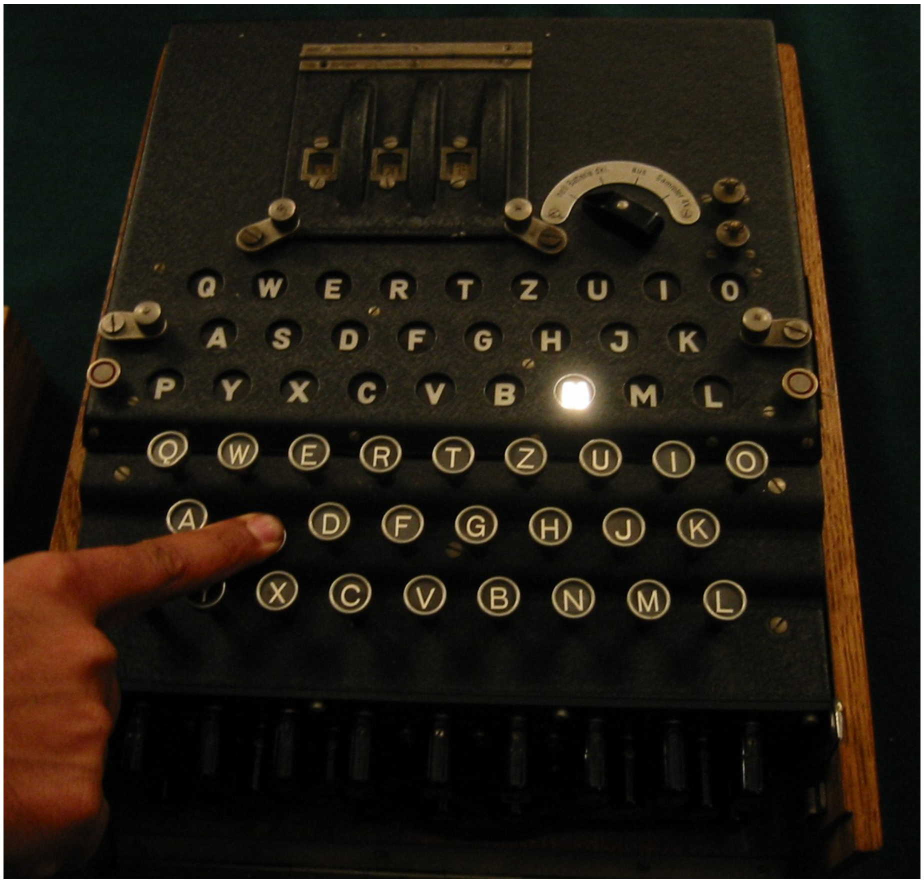

Figure 3: Enigma Keyboard and Lampboard

4.5 Reflector

The *reflector* follows the three rotors. It performs a symmetrical encryption and then sends the electrical signal back through the machine. This symmetrical encryption ensures that decryption and encryption settings are the same.

4.6 Lampboard

After the current is reflected, it travels back through the three rotors, the static wheel, and the plugboard. Finally, the signal makes it back to the *lampboard*, the final component. The lampboard is arranged like a keyboard, and each letter has a small light bulb beneath it. When the electrical current reaches a letter, the corresponding bulb flashes on, indicating to the operator that this is the encrypted letter.

5 Enigma Strengths

The German Enigma cipher machine was the most advanced cryptosystem of its time. It presented a monumental challenge to the Allied forces' cryptanalysts because of its incredible strengths. Enigma masterfully balanced the two opposing elements of any cryptosystem: security and ease of use.

The tangible machine itself was "flexible, portable, reasonably rapid, and easy to use." [3, pg. 30]. It was small, relatively lightweight, and could be easily carried by one person from location to location. It was reasonably easy for any operator to use, not requiring any mathematics or troublesome encryption tables and charts. Once daily settings were in place, encrypting and decrypting messages was no more complicated than using a standard typewriter.

Despite its operational simplicity, Enigma was uncommonly complex and secure. First and foremost, it was not only safe against straightforward frequency analysis, but was also safe against the Kasiski and Index of Coincedence methods, used for Vigenére cryptanalysis, because of the extraordinarily large key length. Recall that the spinning action of the rotors results in a the key length of 16900. It was also secure against brute force attacks because there are approximately 15 million million million (that's 15,000,000,000,000,000,000) possible settings. Even if cryptanalysts were able to check 1000 settings every second, it would take over 475,321,317 years to check every possible setting. Therefore, without knowing the daily plugboard and rotor settings, even possessing an Enigma machine would be of little use to the enemy because of the statistically overwhelming number of possible settings.[3]

Another strength of Enigma is that it "allowed extensive exploitation of radio."[3, pg. 30]. It allowed German military communications to capitalize on radio technology, which was prevalent and easy to use but also easily intercepted, without compromising security.

In addition to these strengths in design, the Germans also instituted operational practices to ensure the security of Enigma. In the early days, the Germans changed the settings only every three months. They soon recognized that more frequent changes were necessary. Settings were then changed monthly, then daily, and eventually multiple times throughout a single day.[3] Additionally, the *ring settings*—the starting positions of the three rotors—were different for each message. In fact, operators growing lax on this security measure was actually a key element in Enigma's demise, which will be discussed in the following section.

The Germans also put numerous security measures into operation to ensure that physical compromise would not dismantle all communications. For example, the German navy printed all codebooks in water-soluble ink and instructed operators to throw them into the ocean if capture was anticipated.[3]

6 Enigma Vulnerabilities and German Blunders

Some of Enigma's vulnerabilities lie in the fundamental design of the machine. First of all, a great convenience of Enigma is its symmetry. The settings to encipher a message and decipher that same message are the same.[3] This is made possible by the reflector element, but the reflector also guaranteed that no letter could ever be encrypted to itself. Of course, this is very useful information for the cryptanalysts trying to crack the code. Additionally, due to the natures of the plugboard and reflector, Enigma only substituted letters reciprocally. For example, if A was encrypted to C, then C was encrypted to A under identical settings.[3]

Though these design weaknesses are not insignificant, most of the security vulnerabilities in Enigma were actually due to military practices and human errors. German intelligence had multiple rules that drastically decreased the key space of Enigma. In regards to the plugboard, the settings always used exactly 10 wires, and connecting sequential letters on the plugboard was not allowed.[3] With these limitations and more, the number of possible settings was reduced from 3×10^{114} to 1×10^{23}.

To their credit, the Germans did introduce updates throughout the war to increase the security of Enigma, but these modifications were usually implemented one at a time.[3] Thus the cryptanalysts had time to understand and adjust to one change before another came down the pipe. Initiating multiple improvements at one time likely would have presented more of a challenge for Allied code-breakers at Bletchley Park.

Individual operators also contributed to Enigma's vulnerabilities. The Germans recognized that if they used the same settings for every message throughout an entire day, the enemy would be able to use a form of cryptanalysis similar to the method previously discussed for the Vigenére cipher. The first letter of every message would have been encrypted using the same encryption alphabet. The same is true for the second letter of every message, and so on. In order to prevent this, it was standard procedure to start encryption with the three rotors in different starting positions for each message. In order to indicate these ring settings, the operator would set up the machine in accordance with the daily settings and type three random letters, then type the same three letters again to account for radio communication errors. The operator would then turn the three rotors so that the previously typed three letters appeared in the windows above the rotors and proceed with the rest of the message. The receiving operator would type the first six letters of the message with the machine set according to the daily settings. Then, reading the ring settings, he would reset the rotors to the indicated positions and decode the remainder of the message. This practice theoretically made frequency analysis impossible. However, operators often got into the habit of using the same three letters for every message. This gave cryptanalysts a huge crack because it allowed them to immediately rule out many possible settings.[3] In fact, code-breakers were able to use repetition to their advantage in other ways as well. For example, the content of some messages were predictable, and the Allies were able to use these predictable components to eliminate Enigma settings.[3]

7 Breaking Enigma

Though every cryptosystem has technical vulnerabilities, and Enigma is no exception, its defeat in this case "arose less from a technological flaw than from the systematic failure of an entire intelligence system" and from the adept cryptanalysis efforts of the Allied forces.[3]

German cryptographic successes in the early days of war lead to complacency in cryptographic communications. Because of pride and perceived impenetrability, the German military used Enigma almost exclusively for wartime communications of every class, from routine weather reports to highly sensitive material.[3] And routine was Enigma's biggest downfall. Certain types of messages were sent habitually and predictably. They were sent regularly from a particular sender to a certain receiver at a typical time of day, and often, the content of the message was similar. Allied intelligence was able to use key words and phrases to find a crack in the system. By knowing a portion or portions of the message, cryptanalysts could eliminate certain possible settings from the onset and drastically reduce the number of Enigma settings to try.

7.1 Polish Cryptanalysis

Before the war even began, Polish intelligence had already made substantial progress on breaking Enigma by exploiting the German operating procedure of typing the ring settings twice at the beginning of each message. They invented an electromechanical machine, called Bomba, which could eliminate impossible settings from the onset and run through potential settings.[4] Just before Germany invaded Poland in 1939, the Poles passed on all the intelligence they had gathered to the British and French.

7.2 British Cryptanalysis

The British, including its highest government officials, well understood the crucial role of signals intelligence and devoted substantial resources towards its success, including financial and human capital. Their efforts were not haphazard, but well organized and strategic throughout the duration of the war. The British consolidated all cryptanalysis efforts in a centralized signals intelligence organization, the Government Code and Cypher School, rather than having separate operations for German, Japanese, naval, military, diplomacy, and other code breaking operations.[3] Previously, several separate departments attacked foreign codes and ciphers with little or no coordination between them. By "concentrating its complete cryptanalytic effort in a single centralized organization," British intelligence was able to collaborate and extrapolate information and techniques.[3] The hub of cryptographic efforts was at Bletchley Park, and this physical centralization also helped intelligence staff to understand the role their piece played in the grand scheme and understand its importance.[3] People working at Bletchley Park included military personnel and civilians alike. It is estimated that the total number of people working at Bletchley Park during the war was as high as 10,000, a large proportion of which was civilian staff.[3] The British put a lot of focus on recruiting highly talented, intelligent university students and professors in the fields of linguistics, mathematics, history, physical sciences, and others.[3] Two mathematicians who ultimately became very crucial to the work at Bletchley Park were Alan Turing and Gordon Welchman, who were both recruited because of their notable chess skills.

It was Turing and Welchman who contributed most to Enigma's downfall. By 1940, the Germans had realized the vulnerability in keying the ring settings twice at the beginning of each message and stopped this practice, making the Polish Bomba ineffective.[4] Fortunately, Turing had already developed another machine, called the Bombe, which used a different method. Instead of relying on ring setting indicators at the beginning of each message, Turing's machine used assumptions about known plaintext within the message, called a *crib*. The process of finding cribs is greatly aided by the fact that a letter is never encrypted to itself. Using a crib as a starting point, the cryptanalysts would create a *menu*. For example, assume that a section of ciphertext "BNXILLLRAJZIQJQQF" is thought to read "THEWEATHERTODAYIS" in plaintext.

Table 8: Bombe menu created using a crib

Ciphertext	**Plaintext (Crib)**
B	T
N	H
X	E
I	W
L	E
L	A
L	T
R	H
A	E
J	R
Z	T
I	O
Q	D
J	A
Q	Y
Q	I
F	S

Using this we can create a diagram, which we will use to determine the Bombe's settings. In the diagram shown in Figure 4, a line connecting two letters indicates that Enigma encrypts one letter to the other at the indicated

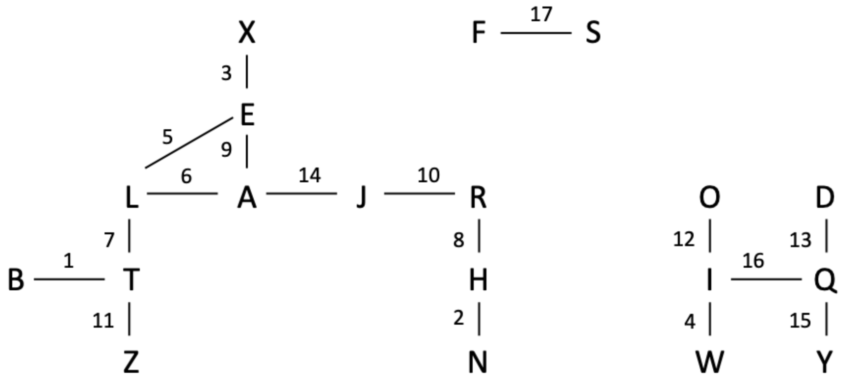

Figure 4: Bombe Menu Diagram

position. For example, T is encrypted to B at position 1. Because of symmetry in design, B is also encrypted to T in this same position.

An operator would set up the Bombe using a similar diagram. The machine would then run through and eliminate candidate settings that were incompatible with the given menu.[1] Welchman further improved the Bombe in 1940 by adding what is known as the *diagonal board*.[4] This addition reduced the steps necessary to determine the proper settings.

The result is that by the end of the war, the Allied powers' code-breakers were reading German messages with regularity. Still, they chose which pieces of intelligence to act on very carefully so as not to arouse German suspicion. This they did with undeniable success, as even after the war German military leaders remained confident in Enigma's impenetrability.

8 Implications in the Field of Cryptography

The development and subsequent defeat of the German Enigma cipher machine has had a substantial impact in the fields of cryptography and computing. First of all, it initiated the shift of cryptography from by-hand methods to automated and mechanized, which in turn created the need for automated and mechanized cryptanalysis.[3] Therefore, Enigma acted as a catalyst for the development of computers. Today, computers constantly utilize cryptographic communications, using complex mathematical algorithms, for even the simplest everyday tasks. One lesson learned from Enigma though is that in the modern world, relying on complexity for security is inadequate. "If we of the twenty-first century rely on the sheer mathematical capabilities of computers for protection, we will be repeating the Germans' blindness."[3] It is imperative to never underestimate the human factor in a system's vulnerability. Though cryptosystems used by modern computers are incredibly complex, it is the human factor that causes trouble. For example, people using predictable passwords or the same password for multiple purposes. Therefore, cryptosystems in the modern world must be designed not for absolute security, which is impossible, but *reasonable security*. This is security such that a single failure does not compromise the entire system.[3] Above all, the most valuable lesson learned is that securing a cryptosystem is a continual process. Once a method is in place, it must constantly be testing for vulnerabilities and improved.

References

[1] F. Carter. From bombe ‘stops’ to enigma keys.

[2] J. Hoffstein, J. Pipher, and J. H. Silverman. *An Introduction to Mathematical Cryptography*. Undergraduate Texts in Mathematics. Springer, New York, second edition, 2014.

[3] R. A. Ratcliff. *Delusions of Intelligence : Enigma, Ultra and the End of Secure Ciphers*. Cambridge University Press, 2006.

[4] P. Reuvers and M. Simons. Crypto museum. https://www.cryptomuseum.com/crypto/bombe/, Nov 2012. Accessed on 2018-11-20.