# The Mathematics of Encryption: An Elementary Introduction

Midge Cozzens

Steven J. Miller

Wesley Pegden

DIMACS, Rutgers University, New Brunswick, NJ 08901
*E-mail address*: midge6930@comcast.net

Department of Mathematics and Statistics, Williams College, Williamstown, MA 01267
*E-mail address*: sjm1@williams.edu, Steven.Miller.MC.96@aya.yale.edu

Department of Mathematics, Courant Institute, NYU, New York, NY 10012
*E-mail address*: pegden@math.nyu.edu

# Contents

# Chapter 1

# Enigma and Ultra

This chapter's about the Enigma, one of the most famous cryptographic systems ever. The Germans used it during World War II, and believed it provided them with perfect security in their communications. We'll start with a quick review of its history, then move on to some of the mathematics needed to study it. The analysis illustrates a common theme in cryptography: frequently it only takes elementary mathematics to state or describe a problem, but solving it is another story entirely! Essentially all we need in this chapter is elementary combinatorics, at the level of counting how many ways there are to choose a fixed number of people from a larger group (with and without the order of choice mattering).

As the Enigma was in fact broken, it's natural to ask what went wrong, or, from the point of view of the Allies, what went right! The German cryptographers weren't fools. Using these counting arguments, we'll see why they thought there was no chance of the Allies reading their messages, and then see how little mistakes in implementation gave the Allies the needed opening into its secret.

## 1.1. Setting the stage

The movie Patton is about, not surprisingly, the life of General George S. Patton. One of the earliest scenes is General Patton arriving in Africa shortly after Operation Torch, the Allied landings in Africa in November 1942. The U.S. forces have just been routed by the Germans, and Patton takes charge. Shortly thereafter, his units engage and soundly defeat elements of German Field Marshall Rommel's army. In one of the most memorable scenes of the movie, Patton surveys the victory and exclaims: "*Rommel, you magnificent bastard, I READ YOUR BOOK!*"

Rommel *had* written a book on tank strategies, *Infanterie Greift An* (Infantry Attacks), which was published a few years before the war in 1937; however, it is rare to have such insights into your foe. As you would expect, there is a huge advantage if your enemy is kind enough to describe their

methods, preferences, targets and strategy to you. Thus, most of the time, each side goes to great lengths to keep these secret; however, different units must be able to communicate quickly and securely with each other. In the midst of a battle, commanders need to redeploy their forces, either to attack or to reinforce weakened positions. They can't wait hours for a message to be encrypted, transmitted and decoded – these must be done almost instantly, or the information is useless. A vulnerability may exist for only a few minutes, and you have to act fast. At the same time, you don't want the enemy to decode your orders and change their plans.

For centuries, cryptographers had been working on this problem, trying to devise a way to let select people exchange information quickly without others unraveling the message. In the Enigma, the Germans believed they had found a practical and essentially impenetrable solution.

The Enigma was a machine used by the Germans in World War II, and before, to encrypt and decrypt messages. As we'll see when we read about all the different wirings and rotors and parts (see Figure 1 for an illustration), it's a very complicated contraption with an incredibly large number of possibilities that need to be tried to successfully decrypt a message. Because the number of possibilities were so large (over $10^{100}$ as we'll see below!), the Germans were very confident that they had a secure means to communicate, and sent many important messages via the Enigma. Fortunately for the Allies, the Germans made several mistakes in using the Enigma, and these errors were exploitable and led to a significant decrease in security. The Allied efforts to decrypt the German Enigma traffic are known as Ultra.

How valuable were these decryptions? Estimates range from shortening World War II by two years to preventing a German victory. The Allies were able to use the decoded messages for a variety of purposes, ranging from determining Germany's long term strategic goals to dealing with more immediate threats such as the location of German subs hunting Allied convoys.

The Germans placed absolute confidence in the security of the Enigma. As one German cryptographer stated, "From a mathematical standpoint we cannot speak of a theoretically absolute solvability of a cryptogram, but... the solvability is so far removed from practical possibility that the cipher system of the machine, when the distribution of keys is correctly handled, must be regarded as virtually incapable of solution." **(ADD REF)** All branches, all levels of the military in the Third Reich used the Enigma freely. In fact, a large part of Hitler's idea of blitzkrieg was based on the Enigma: as General Erhart Milch stated "the real secret is speed – speed of attack through speed of communication," which the mechanized Enigma provided German commanders, as well as total security, or so they believed.

Breaking the Enigma was a daunting task. The number of possible combinations used to produce a message was enormous. Given that the Germans used a new one every day, trying to guess that combination was prohibitively difficult. Fortunately, due to espionage, mathematics, and some amazing inspiration, the Allies were able to decode German messages daily. As it turns


FIGURE 1. A German Enigma Machine, image from Wikimedia Commons.

out, many of the components of the security of the Enigma could be divorced from each other and dealt with separately, which reduced the difficulty of the task significantly.

It's impossible to do Enigma and Ultra justice in a short chapter. Our goal here is to give a brief overview of how Enigma worked, and how the Allies beat it, and talk just a little about the importance of codes in war. For more on the subject, see **ADD REFS**.

Before we turn to the mathematics of the subject, it's worth looking at the Enigma one last time (Figure 1). One of its great advantages is the ease of use. There are two keyboard on the top, one is like a typewriter with keys to press, the other has blubs underneath the letters which light up. After setting everything up to encrypt a message, all the operator has to do is type; each time he presses a key, what that letter encodes to lights up.

To decrypt a message, again after choosing the correct settings, the person on the other end just types in the encrypted message. Each time she types a letter, the letter that lights up is what that letter decrypts to. In other words, encryption and decryption is done at the speed of the typist! There is no difficult math problem to be solved on either end; the machine takes care of everything. This is a very desirable feature for battlefield situations.

## 1.2. Some Combinatorics

There are several reasons for studying the Enigma early in a cryptography course. It's one of the most important examples ever, and its successful decryption changed the fate of the world. Mathematically, it's extremely accessible. All we need is some elementary combinatorics. We begin by quickly reviewing these through some basic problems, isolating the functions which will appear in later sections when we turn to analyzing the Enigma.

Let's start with a basic example:

***Question #1:*** *How many ways can you arrange three people in a line?*
**Answer:** Suppose we want to seat our friends Alan, Bernhard, and Charles. In this problem, the order of the three people matter. Thus Alan, Charles and Bernhard is different than Charles, Bernhard and Alan. A good way to approach this is to fill the three available slots one at a time. How many choices do we have for the first position? Before anyone is assigned, we have three people: Alan, Bernhard and Charles. We have to choose one of them for the first position. We're now left with two people, and two open spots. We don't need to know *which* two are left, only that there is *one* less person. We thus have two ways to choose which person is second. For the last spot, we only have one choice left, and we must put the remaining person in that final place. The total number of arrangements is 6, coming from 3 ways to choose the first person, 2 ways to choose the second, and one way to choose the last.

A good way to view this is to look at this as a tree (see Figure 2). We start with all positions open. We have three choices for the first person, which creates three branches. At the next stage, we have two remaining people to choose from. Each of our initial branches then branches again, but this time into just two possibilities, one for each remaining person. Note that while the people involved in the split differ from branch to branch, the number of people involved is always two. Finally, each branch is continued once more as we assign the third person. As only one person is left to assign, the choice is forced upon us.

Arguing along the same lines, we see that if we had four people to arrange in a line (with order mattering), the number of possibilities would be $4 \cdot 3 \cdot 2 \cdot 1 = 24$. These descending products occur so frequently in the subject that we have a special name and notation for them. We call these

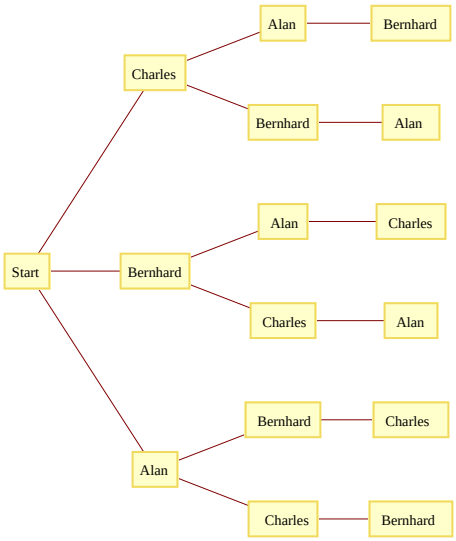


FIGURE 2. Tree diagram indicating the number of ways to put three people in a line (when order matters). Note we end up with $6 = 3 \cdot 2 \cdot 1$ possible arrangements.

**factorials**, and write 3! for $3 \cdot 2 \cdot 1$ and 4! for $4 \cdot 3 \cdot 2 \cdot 1$. Notice that this function grows *very* rapidly; 10! is ten times larger than 9!, which is 8 times larger than 8!, and so on. For each positive integer $n$, we have $n! = n \cdot (n-1)!$.

As the above problem suggests, there is a combinatorial interpretation of the factorial function: $n!$ is the number of ways of arranging $n$ people in a line, when order matters. It turns out to be useful to define 0!. Though it might seem surprising, the right definition is to take 0! to be 1. We may interpret this as follows: there is only one way to do nothing!

We now generalize our first question.

***Question #2:*** *How many ways can you line up three of five people?*
**Answer:** Again, this is a problem where order matters. If the five people are Alan, Bernhard, Charles, Danie and Ephelia, then choosing Charles, Danie and Alan is counted as different than choosing Danie, Alan and Charles. The solution to this problem is similar to the previous. We have already solved how to choose five people from five: the answer is 5!, or $5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$. The difference now is that we're not choosing all five people. We're only choosing three, and thus stop after the third choice. Our answer is therefore $5 \cdot 4 \cdot 3$, or 60. Notice how quickly the number of possibilities grow; drawing

a tree with all the possibilities is no longer practical!

There is another way to write down the answer. Instead of writing $5\cdot4\cdot3$, we could instead write $5!/2!$. The reason is

$$\frac{5!}{2!} \;=\; \frac{5\cdot4\cdot3\cdot2\cdot1}{2\cdot1} \;=\; 5\cdot4\cdot3.$$

What we're doing is multiplying by 1 in a clever way, and multiplying by 1 won't change the answer. Here we wrote $1 = 2!/2!$. The reason this is a good idea is that the product $5\cdot4\cdot3$ looks a lot like a factorial, but it isn't. It's missing the $2\cdot1$. By including that factor, we obtain a factorial. The problem, of course, is that we can't just multiply part of an expression; thus if we multiply by $2\cdot1$ we must also divide by it. If instead we had to choose 4 people from 9 with order mattering, the answer is $9\cdot8\cdot7\cdot6$. To make this into $9!$ we need the factor $5!$. We thus multiply and divide by this, and find the number of ways to choose 4 people from 9 (when order matters) is $9!/5!$.

More generally, if we want to take $k$ people from $n$ people with order mattering, it's just $n\cdot(n-1)\cdots(n-(k-1))$. This bit is a little tricky. Why is this the product? Clearly the first term is $n$, then $n-1$, then $n-2$. What is the last term? We need to have $k$ terms in all. The first term, $n$, is really $n-0$. The second term is $n-1$, and notice that we are subtracting one less than the number term it is. Thus to find the $k^{\text{th}}$ term we must subtract off one less than $k$, or $k-1$. Equivalently, the last term is $n-(k-1)$. We've just shown that the number of ways to choose $k$ people from $n$ people (with order mattering) is $n(n-1)\cdots(n-(k-1))$. This wants to be a factorial, but it doesn't go all the way down to 1. To remedy this, let's argue as before and multiply by $(n-k)!/(n-k)!$. This won't change the answer as we're just multiplying by 1, but now we have all the numbers from $n$ down to 1 being multiplied together, and we see that the answer is just $n!/(n-k)!$.

So far, we've looked at how to arrange people (or more generally anything) when order matters. What happens when you just need to pick some people and the order in which you pick them doesn't matter? We can view this as choosing a subset of a group to serve on a committee, and all committee members are equal; conversely, what we did earlier (arranging people with order mattering) would be assigning people to a group but designating one the president, one the vice-president, one the treasurer, and so on.

For example, imagine we want to pick four people from a group of four people. How many different ways could we do this? Since there are only four people, and we have to pick four, there's no freedom. We *have* to pick all four, and thus there's only one way to pick four people from four people when order does not matter. Equivalently, we can say there is only one group of four people that can be made from four people. Let's consider a

more involved case.

***Question #3:*** *How many ways are there to pick four people from a group of nine people, when it doesn't matter which order the people are selected?*
**Answer:** Now things are more interesting, and it isn't immediately clear what the answer should be. Fortunately, there's a nice way to view this problem. Let's first consider the easier problem of how many ways there are to choose four people from nine. We looked at this earlier, and saw that the answer is $9 \cdot 8 \cdot 7 \cdot 6$ or $9!/5!$. This is the number of ways of choosing four people from 9, but order matters. The clever idea is to realize that this is precisely 4! more than the number of ways to choose four people from nine when order does not matter. Why? For each group of four people chosen, there are 4! ways to order them. Each of these 4! ways has been counted in the $9!/5!$ ways to choose four people from nine with order mattering. Thus, if we take $9!/5!$ and divide by 4!, we remove the extra counting, we remove the ordering we don't care about, and obtain the answer. The reason this works is that *all* groups of four people have the same number of arrangements.

There is nothing special with choosing four people from nine. If instead we wanted to take $k$ people from $n$, not caring about the order, we would argue similarly. We first note that there are $n!/(n-k)!$ ways to choose $k$ people from $n$ when order matters. For each group of $k$ people there are $k!$ ways to assign an ordering; thus each group of $k$ people is counted $k!$ times among the $n!/(n-k)!$ orderings. If we divide by $k!$ we remove this counting, and find that the number of ways to choose $k$ people from $n$ people, not caring about the order in which they are chosen, is simply $\frac{n!}{k!(n-k)!}$.

It was worth defining the factorial function and giving it special notation because of how often it appears in combinatorial problems. Similarly, it is worth introducing some notation for $\frac{n!}{k!(n-k)!}$ so we don't have to write this out every single time. We let $\binom{n}{k} := \frac{n!}{k!(n-k)!}$, and read this as $n$ *choose* $k$. This is called a **binomial coefficient**. The top refers to the number of options we have, and the bottom to how many we are choosing. Order does not matter, and we assume that $0 \le k \le n$.

The binomial coefficients have many useful properties. Doing some simple algebra, we see that $\binom{n}{k} = \binom{n}{n-k}$, because

$$\binom{n}{n-k} = \frac{n!}{(n-k)!(n-(n-k))!} = \frac{n!}{(n-k)!k!} = \frac{n!}{k!(n-k)!} = \binom{n}{k}.$$

While the above argument proves the claim, it's not particularly enlightening. There's a much better argument. Note $\binom{n}{k}$ is the number of ways to choose $k$ people from $n$ when order doesn't matter; however, if we choose $k$ people from $n$, we can view this as *excluding* $n-k$ from $n$. Thus $\binom{n}{k}$ must be the same as $\binom{n}{n-k}$. This proof illustrates a central idea in combinatorics:

*if you can tell an appropriate story, you can give an interpretation to quantities and see the relationships.*

When encountering new ideas, it's usually a good idea to test simpler cases where you have some intuition. Let's imagine the case where we have 3 people, and we want to choose a group of 3 and it doesn't matter what order we choose. Clearly there's only one way to do this (as we have to take everyone), which agrees with our formula since $\binom{3}{3} = \frac{3!}{3!0!} = 1$ (remember $0! = 1$). What if we want to choose 3 people from 4, where order doesn't matter? Our formula tells us there are $\binom{4}{3} = \frac{4!}{3!1!} = 4$ distinct possibilities. To conserve space, let's give the four people the boring names A, B, C and D. There are 24 ways to choose three of four. There are six ways to order A, B and C if they are the three people chosen: {A,B,C}, {A,C,B}, {B,A,C}, {B,C,A}, {C,A,B}, {C,B,A}. These six different orderings have the same people, and thus only contribute one possibility. Similarly there are six ways to order the people A, B and D, another six to order A, C and D, and finally six more for B, C and D. Note $6 = 3!$, and thus the number of unordered possibilities is $4!/3! = 24/6 = 4$, as each unordered possibility of three people can be ordered 3! ways.

To sum up, we have shown **(CANNOT GET BOX TO WORK HERE – BOX THIS LATER)**

Let $n$ and $k$ be integers, with $0 \le k \le n$

- The number of ways to choose $k$ objects from $n$ objects *with order mattering* is $n(n-1)\cdots(n-(k-1))$, which may be written as $n!/(n-k)!$. Here $m!$ denotes $m$ factorial, which is $m(m-1)\cdots 1$, and we have set 0! to equal 1.
- The number of ways to choose $k$ objects from $n$ objects *with order not mattering* is $\frac{n!}{k!(n-k)!}$, which is denoted with the binomial coefficient $\binom{n}{k}$.

Armed with our building block functions of factorials and binomial coefficients, we can now analyze the Enigma's complexity and see why the Germans were unconcerned about Allied attempts to crack their codes.

**Exercise 1.2.1.** *How many ways are there to choose* at most *3 people from 10 people, when order does not matter? What if order does matter?*

**Exercise 1.2.2.** *Five men and eight women go to a dance. How many ways are there to choose three men and three women? How many ways are there to choose three men and three women so that each of the chosen men dances with* exactly *one of the chosen women?*

**Exercise 1.2.3.** This problem is hard, but can be solved with binomial coefficients *if* you look at it the right way. *Imagine we have 10 identical cookies and five (distinct) people: Alice, Bob, Charlie, Danie and Eve. How many different ways can we divide the cookies among the people? Since the cookies are identical, all that matters is how many cookies a person receives, not which cookies.*

## 1.3. Enigma's Security

If the Enigma were used properly, it would've been extremely secure. While today computers are everywhere, during World War II the situation was completely different. People had to examine all the tedious combinations by hand; in fact, the need to quickly exhaust numerous possibilities was a huge impetus in the birth of computers. As we'll soon see from analyzing the Enigma's workings, the Germans were justified in believing that the Enigma would provide secure communications. Fortunately for the Allies, however, the Germans had several practices that greatly weakened the actual security. Before describing their errors, let's take a look at what exactly made the Enigma seem so impregnable. This section relies heavily on *The Cryptographic Mathematics of Enigma* by Dr. A. Ray Miller [**Mi**]. It's a short pamphlet from the NSA and available online, and we highly recommended it to anyone interested in additional reading.

### 1.3.1. The Plugboard

There are five things that contributed to the Enigma's cryptographic strength. The first was a **plugboard** with twenty-six holes for 13 connector cables, each hole associated with a letter (see Figure 3). The operator would plug in these cables as designated in the code book, which in effect would switch the outputs of the connected letters. For instance, if the code book indicated a cable should be plugged into the holes for E and F, the outputs for those two letters would be switched. In that case, the word 'effect' would become 'feefct.' If instead O and Y were connected and P and N were connected, then 'effect' would appear unchanged, but 'pony' would become 'nypo'. The more pairings we have, the more garbled the words become.

The operator could use anywhere between zero and thirteen cables. Not all letter connections are possible. You can't connect two letters to the same letter, or a letter to itself. Let's see how these choices affect the security of a message. Specifically, let's determine how many different letter swaps are available. We first consider the simplest possibility: don't use any cables (and thus no message is changed in typing). There's only one way to do nothing. Another way of looking at this is that there is only one assignment of cables connecting pairs of letters that results in no swaps: use no cables!

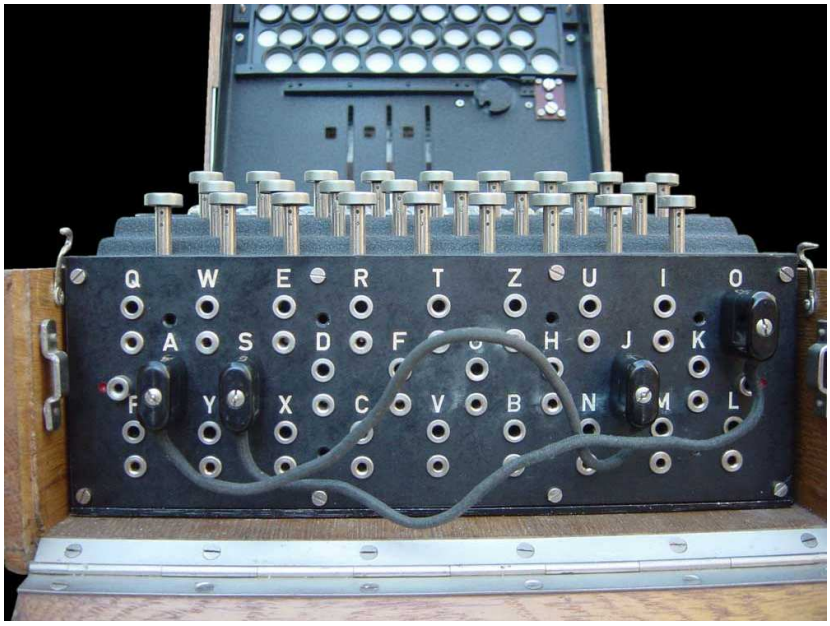


FIGURE 3. The Enigma's plugboard, with O and Y connected and P and N connected. Image from Wikimedia Commons.

Now let's look at a slightly more difficult example. Let's use just one cable. In this case, we're going to switch two letters (like we did for E and F before). How many different ways could we do this? Well, we have 26 letters, and we want to *choose* two of them to switch. This is the definition of the binomial coefficient $\binom{26}{2}$ from last section, as this represents the number of ways to choose two objects from 26 when order doesn't matter. One of the hardest parts of combinatorial problems like these is keeping track of when order matters and when order doesn't matter. At the end of the day, we can't tell if we first plugged in one end of the cable to E and then to F, or the other way around. It doesn't matter; all that matters is that E and F are now connected. Thus, for these cable problems, it is always choosing with order immaterial. As $\binom{n}{k} = \frac{n!}{k!(n-k)!}$, we see $\binom{26}{2} = \frac{26!}{2!(24)!} = \frac{26 \cdot 25}{2 \cdot 1}$, or 325.

We've now figured out how many ways to use zero cables (just 1 way, which is $\binom{26}{0}$) or one cable ($\binom{26}{2}$ or 325 ways). What about two cables? A natural thought is that it should be the number of ways of choosing four letters from 26, with order not mattering, or $\binom{26}{4}$. While this looks like the next natural term in the pattern, it isn't quite right. The reason is that we need to do more work than just choosing four letters; after we choose the four letters, we must pair them into two groups of two. For example, if we choose the four letters A, E, L and Y, then there are three different ways to pair them into two groups of two: we could have the pairings A-E and

L-Y, or A-L and E-Y, or A-Y and E-L. Note that the pairing A-E and L-Y is the same as the pairing E-A and L-Y or L-Y and A-E and so on, as all that matters is which two are paired and not the order. Because of this, we see it is going to be a lot more work to determine the number of ways to use exactly two cables.

There are several ways we can arrange this calculation. One nice way is to first choose the four letters to be paired with the two cables; there are $\binom{26}{4}$ ways to do this. We then have to split the four letters into two pairs of two; we don't care which pair is listed first, or which letter is listed first in each pair. How many ways are there to do this? Similar to our analysis on the number of ways to choose $k$ objects from $n$ objects when order doesn't matter, it's easier to first impose an order and then remove it. What we'll do here is put labels on the pairs (*first pair, second pair*), and then remove them. If we have four letters, there are $\binom{4}{2}$ ways to choose two letters to be the first pair. This leaves us with two remaining letters, which must be chosen to be the second pair; there are $\binom{2}{2}$ ways to choose the two remaining letters for the second pair. Multiplying, we see that there are $\binom{4}{2}\binom{2}{2}$ ways to pair the four letters into two pairs of two, with the pairs being labeled *first pair, second pair*. We now remove the labels. How many ways are there to assign the two labels to the two pairs? There are 2! ways – we have two choices for the label for the 'first' pair, and then one remaining label for the second. Thus, we have overcounted each desired pairing of four letters into two pairs (not caring about which is the first and which is the second pair) by a factor of two. The answer is not $\binom{4}{2}\binom{2}{2}$, but instead $\binom{4}{2}\binom{2}{2}/2!$. Combining this with the number of ways to choose four letters from 26, we see there are $\binom{26}{4}\binom{4}{2}\binom{2}{2}/2!$ ways to use two cables. Multiplying this out, we get 44,850, much larger than the 325 possibilities with just one cable!

To highlight the method, let's consider three cables. There are several different ways to look at this problem. We'll first solve it by modifying our previous argument. We have three cables; as each cable connects two letters, we must choose 6 letters out of 26. There are $\binom{26}{6}$ ways to do this. Now that this has been done, we have to pair the six letters into three groups of two. It doesn't matter which group is called the first pair or the second or the third, or in a pair which letter is chosen first. A good way to count the possibilities is to look at how many ways we can pair the six letters into three pairs with the pairs labeled, and then remove the labels to reduce the over-counting. There are $\binom{6}{2}$ ways to choose two letters from the six letters for the first pair, $\binom{4}{2}$ ways to choose the two letters from the remaining four letters for the second pair, and then $\binom{2}{2}$ ways to choose two letters from the remaining two letters for the third pair. There are 3! ways to assign labels to the three pairs; thus we have over-counted by a factor of 3!, which we must remove. Putting this all together, the number of ways to use three cables is $\binom{26}{6}\binom{6}{2}\binom{4}{2}\binom{2}{2}/3!$.

If we multiply out the last factor, $\binom{6}{2}\binom{4}{2}\binom{2}{2}/3!$, we notice something interesting. It's

$$\begin{aligned}\binom{6}{2}\binom{4}{2}\binom{2}{2}\frac{1}{3!} &= \frac{6!}{2!4!}\frac{4!}{2!2!}\frac{2!}{2!0!}\frac{1}{3!} \\ &= \frac{\frac{6\cdot 5}{2}\frac{4\cdot 3}{2}\frac{2\cdot 1}{2}\frac{1}{3\cdot 2\cdot 1}}{} \\ &= \frac{6\cdot 5\cdot 4\cdot 3\cdot 2\cdot 1}{6\cdot 4\cdot 2} \\ &= 5\cdot 3\cdot 1.\end{aligned}$$

We generalize the factorial notation and write $n!!$ to mean the product of every other term, down to 2 if $n$ is even and 1 if $n$ is odd. We call this the **double factorial**. While we could've set $n!!$ to be the factorial of $n!$, that expression just doesn't turn up that often in problems, whereas the product of every other term frequently arises. We can define anything we want – the question is what's useful to define. Here, it turns out that the every other product is useful, and worth giving a special name. Thus $5!! = 5\cdot 3\cdot 1$, $6!! = 6\cdot 4\cdot 2$, $7!! = 7\cdot 5\cdot 3\cdot 1 = 7\cdot 5!!$ and so on.

We just showed that the number of ways to pair six letters into groups of two, where the pairs are unlabeled and it doesn't matter if a letter is listed first or second in a pair, is $5!!$. One can similarly show that if we had to pair eight letters into groups of four then the answer would be $7!!$, and in general if we had $2p$ letters that had to be paired into groups of two then the answer would be $(2p-1)!!$.

**Exercise 1.3.1.** *Before reading on, where we give the answer, try to prove the above. Specifically, show that there are* $7!!$ *ways to pair eight letters into four groups of 2, and* $9!!$ *ways to pair ten letters into five groups of 2. If you know mathematical induction, extend your argument to show that there are* $(2p-1)!!$ *ways to pair* $2p$ *letters into* $p$ *groups of 2. As this is such an important result, we give its solution below.*

One way to see the last claim is to proceed by induction. We know it's true when we have six letters. Imagine now we have eight letters. The first letter has to be paired with one of the remaining 7 letters; there are 7 ways to do this. We now have 6 letters remaining that must be paired in groups of two. We know there are $5!!$ ways to do this, and thus the number of ways to pair 8 letters in groups of 2 is just $7\cdot 5!!$, which is $7!!$. We now turn to ten letters. Consider the first letter. It must be paired with one of the remaining 9 letters; there are 9 ways to do this. We're now left with 8 letters that must be matched in pairs; however, we've just shown that the number of ways to do this is $7!!$. Thus the number of ways to pair ten letters in groups of 2 is just $9\cdot 7!!$ or $9!!$. Arguing along these lines proves the claim in general.

Based on our calculations above, we see that if we have $p$ cables connecting $2p$ letters, then the number of possibilities is $\binom{26}{2p}(2p-1)!!$; the $\binom{26}{2p}$ comes

from the number of ways to choose $2p$ letters from 26, and the $(2p-1)!!$ comes from the number of ways to match them in pairs.

To get the total number of combinations for any number of cables used, we just add the results for having $p$ cables, where $p$ ranges from 0 to 13. We can just add them because using 1 cable results in a different output than when we don't use any cables ('effect' versus 'feefct'), that is, one setting was independent from all other settings. When we add these together (using $(-1)!! = 1$ to allow us to write things in a common language), we get

$$\begin{aligned}&\binom{26}{0}(-1)!! + \binom{26}{2}1!! + \binom{26}{4}3!! + \binom{26}{6}5!! + \cdots + \binom{26}{26}25!! \\ &\quad = \ 532985208200576 \ \approx \ 5.32 \cdot 10^{14}.\end{aligned}$$

**Exercise 1.3.2.** *Which value of $p$ do you think gives the greatest contribution to the sum? Why? Compute all the values and see if you're right.*

This is an *enormous* number of possibilities. To put it in perspective, imagine the Allies could check a million combinations a second (this is far more than was possible back in the 1940s). How long would it take to go through the approximately $5 \cdot 10^{14}$ possible cable connections? There are about 31,536,000 seconds in a year (60 seconds in a minute, 60 minutes in an hour, 24 hours in a day and 365 days in a year). In one year, checking at a rate of a million wirings per second, the Allies would be able to examine 31,536,000,000,000 or about $3.15 \cdot 10^{13}$, less than a tenth of the possibilities! To make matters worse, the wiring chosen changes – after all this work the Allies would only know the wiring for one particular day, and would have to start all over again for the next day.

To sum up, even assuming the ability to check a million wirings a second (which was well beyond what the Allies could do) only gives us a 10% chance of finding one day's wiring *in an entire year's time!* We can begin to see why the Germans were confident in the security of Enigma, especially as this is just the first of many pieces. We'll now discuss the rotors, whose complication dwarfs the contribution of the cables.

**Exercise 1.3.3.** *Based on the definition of the factorial and the double factorial, what do you think the triple factorial should be?*

**Exercise 1.3.4.** *There is a simple function $f$ such that $(2n)!! = f(n) \cdot n!$. What is $f(n)$?*

**Exercise 1.3.5.** *Show that $\sum_{k=0}^{n} \binom{n}{k} = 2^n$.*

### 1.3.2. The rotors and reflector

The **rotors** were the second component contributing to the Enigma's strength (see Figure 4). These rotors took 26 inputs and wired them to 26 outputs. The number 26 is no accident, as the effect of the rotor was to provide

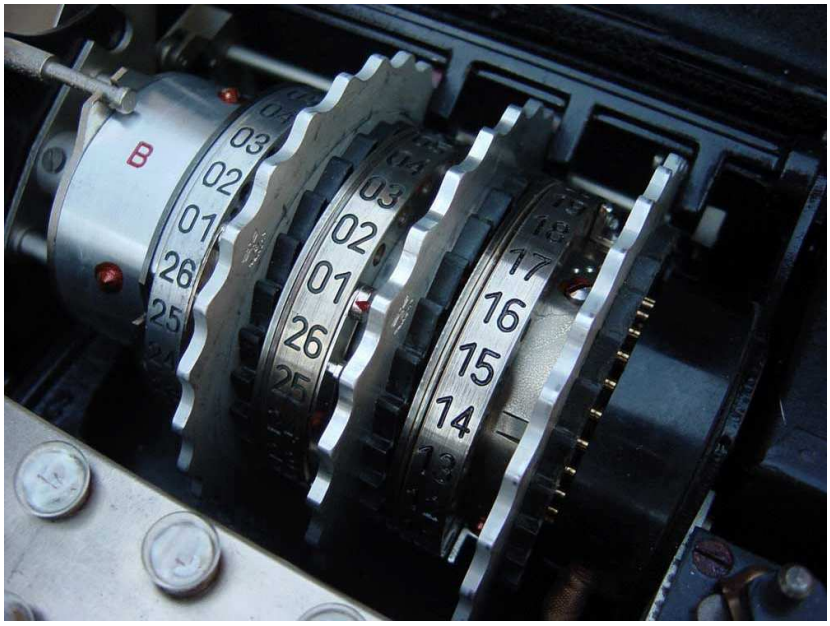


FIGURE 4. The stack of rotors inside an Enigma machine, consisting of three rotors and Umkehrwalze-B (the reflector). Image from Wikimedia Commons from user Matt Crypto.

another exchange between letters. This means the Germans could have constructed 26! different rotors. Why is this? Let's look at how this wiring was done. Each rotor has two sets of the alphabet, and the rule is we connect each letter in the first alphabet with a letter in the second alphabet, never connecting two letters to the same letter (this means we can't send A and B on the first alphabet to C on the second, but we could send A to A).

Let's consider the letter A. We can connect it to any of the 26 letters in the second alphabet. Now that we've connected A somewhere, what about the wire for B? Like before, we can wire it to any one of the remaining outputs. There are only 25 available now because the wire from the letter A is connected to one of them already. The wire from C can connect to any of the 24 remaining outputs, and so on, so we see that there are 26! different choices. Note this problem is the same as our earlier exercise of arranging $n$ people in a line. We may regard the letters in the first alphabet as positions 1, 2, 3, ..., 26, and the second alphabet is then 26 people we need to order on the line, with order counting.

As the standard Enigma machines had three rotors, the number of possibilities for the three rotors is $26!^3$ (26! ways to choose each rotor), which is approximately $6.559 \cdot 10^{79}$. We thought $5.32 \cdot 10^{14}$ was bad; checking rotors at a rate of a *trillion* possibilities per second means we could investigate about $3.2 \cdot 10^{25}$ triples of rotors in a year. As the universe has only existed for a few billion years (a billion is $10^9$), we see that if we check a trillion combinations

per second for the entire life of the universe then we're not even at $10^{40}$, and we need to get up to $10^{79}$. Actually, the Allies had one piece of information on the German rotors: they knew the Germans didn't reuse a rotor in the machine, and thus the three rotors had to be distinct. Therefore, instead of $26^3$ possible triples of rotors, in fact it was $26! \cdot (26! - 1) \cdot (26! - 2)$. The difference between these two is negligible, and the first 26 digits of the two numbers are the same. (The difference between these two numbers is about $4.87 \cdot 10^{53}$. We're not saying that this large number is negligible; what's negligible is the savings. We've saved much less than $10^{-20}$ percent!)

The next variable was how the rotors were initially positioned. Operators could rotate the rotors to any initial position they wanted. Since there were 26 different positions, and three rotors, the number of possible initial positions is $26 \cdot 26 \cdot 26$, or $26^3 = 17,576$.

The forth component to consider is the **reflector**. The reflector was really just like a plugboard with 13 cables. Each letter was connected to another letter via a wire, essentially switching the letters, just like the plugboard. The difference was that the reflector never changed, and switched all 26 letters. From our analysis earlier, we see that the number of possible reflector arrangements is just

$$\binom{26}{26}(26-1)!! \;=\; 7,905,853,580,625 \;\approx\; 7.9 \cdot 10^{12}.$$

The last contribution to the Enigma's complexity were **notches** on the rotors. These notches controlled when the next rotor would move forward by one letter. The rotor positioned on the right would rotate every time a key was pressed, the second rotor would turn every 26 key strokes, and the last rotor would rotate once for every 26 rotations of the second rotor, or once every $26 \cdot 26 = 676$ keystrokes. The notches determined when the second and third rotor first turned. [**PICTURE if we can find one**] If the first notch were in a particular position, the second rotor would turn for the first time after, say, 3 keystrokes. If it were in a different position, the second rotor would not rotate until, say, 19 keys had been pressed. The notch on the first rotor could be in one of 26 positions, and similarly for the second rotor. Since the reflector didn't rotate, the notch on the third rotor did nothing. Therefore, the number of possibilities from this component of the Enigma is just $26^2 = 676$. The notches provided an enormous amount of security. Without the notches, each letter is always encrypted the same way. This means the messages would be vulnerable to a frequency analysis. We'll describe this in greater detail in **ADD REF**, but the gist is easy to describe: if you have a very long message in English, the most common letter is almost surely an 'E'. Using letter frequencies (or, even better, frequencies for pairs and triples of letters), we can make very good guesses at how the letters are being switched. The advantage of the notches is that at first an 'E' is encrypted as a 'Y', and then maybe as an 'N', then perhaps as an 'R', and so on. By constantly changing how each letter is encrypted, it's

impossible to attack the code with a frequency analysis.

To sum up, the number of possibilities introduced by the rotors, notches and reflector is

$$26!(26!-1)(26!-2)\cdot(26)^3\cdot\binom{26}{26}(26-1)!!\cdot 26^2 \;\approx\; 6.16\cdot 10^{99}.$$

In the previous section we looked at the plugboard. Its contribution of about $10^{14}$ looked impressive, but that complexity is dwarfed by these other features.

**Exercise 1.3.6.** *In the rotors for the Enigma, it's possible to connect a letter to a copy of itself; for example, we could send A to B, B to C, C to A, and then D to D, E to E, and so on. What if we added a restriction that no letter could go to a copy of itself? It's now a lot harder to find the number of such rotors. If you know about the Principle of Inclusion / Exclusion, that's a great way to attack this. Instead, let's consider 'simpler' alphabets. Find the number of such rotors if there are only 2 letters in our alphabet. What if there are 3 letters? What about 4 letters?. (As the number of letters in the alphabet grows, the ratio of number of these restricted rotors to the number of rotors converges to* $1-1/e$*.)*

### 1.3.3. Possible Enigma States

Now comes the fun part – we'll see why the Germans had such faith in the Enigma. Since each component could be changed without affecting any other part, to get the total number of Enigma configurations we just need to multiply our previous numbers together. The result is simply astonishing. In its full glory, we get

$$\begin{aligned}&3,283,883,513,796,974,198,700,882,069,882,752,878,379,955,\\ &\quad 261,095,623,685,444,055,315,226,006,433,615,627,409,666,\\ &\quad 933,182,371,154,802,769,920,000,000,000,\end{aligned}$$

or, approximately $3\cdot 10^{114}$.

This number is so much larger than what we've seen before that it takes awhile to view this in a meaningful way. Currently we believe the universe is about 14 billion years old, or approximately $3.2\cdot 10^{22}$ seconds. Using guestimates from the number of stars in the universe, one gets that there should be about $10^{80}$ atoms in the universe (give or take a few orders of magnitude). Imagine each such atom were a supercomputer devoted entirely to checking Enigma possibilities, and capable of looking at a billion (i.e., $10^9$) setups per second. If these supercomputers had been running non-stop since the creation of the universe, by today they would only have checked a little less than $10^{112}$ possibilities. In other words, in such a situation

of unbelievably fantastic computer power at our disposal (and for such an incredibly long period of time), we would have less than a 1% chance of having found the correct wiring for a given day! It's no wonder that the Germans felt the Enigma provided more than enough of a challenge to the Allies, and were not worried about their messages being read.

**Exercise 1.3.7.** *Consider all the pieces in the Enigma, from the plugboard to the reflector. Which steps adds the most number of combinations?*

## 1.4. Cracking the Enigma

In the last section we showed that the Germans had more than $10^{114}$ possible ways to configure their Enigma machines. Against that sort of number, how could the Allies possibly have broken the Enigma? Fortunately for the Allies, the Germans made several mistakes in its use. For a very good read and more detail on some of the ways the Enigma was weak, we recommend *The Code Book* by Simon Singh [**Si**], from which much of the information in this chapter is taken. Another good source is the NSA pamphlet *Solving the Enigma - History of the Cryptanalytic Bombe* by Jennifer Wilcox [**Wi**], available online.

Before looking at some of the German mistakes in implementation, let's look at some things that will bring that terrifying number of $10^{114}$ down a little right off the bat. For one thing, suppose we can solve every challenge the Enigma throws at us except how the plugboard is arranged. Then what we have is essentially a monoalphabetic cipher. In other words, it just switches letters and always switches them the same way. We can use frequency analysis to solve this puzzle. We'll discuss this in detail in Chapter **ADD REF**. As we said earlier, in English the most common letter is E, so if you have lots of encrypted messages, whatever letter appears most often probably decrypts to E. You can get more advanced, and look at the most common pairs of letters, and so on.

Moreover, if not all the letters are switched, it might be possible to just guess what the words are. For instance, take the phrase "sieben und zwanzigste Bataillon", which is German for twenty-seventh battalion. If you were reading a message from the 27th Battalion, you might guess that "aiebenundzwsnzigatebstsillon" really means "siebenundzwanzigstebataillon", and that S and A have been switched. Continuing in this manner, you could, perhaps, find all the plugboard settings. So, assuming we can handle everything else, we've reduced the number of settings we have to check by about $10^{14}$. Unfortunately this still leaves us the problem of solving the rest, and the number of combinations there is of the order $10^{100}$, but we've made some progress.

We now turn to the rotors. We need to choose three different rotors. There are 26! possibilities for each rotor, which led to approximately $6.559 \cdot 10^{79}$ triples of distinct rotors. This is a huge part of the remaining $10^{100}$

possibilities, and if we can make good progress here, we're well on our way to deciphering the Enigma. It's important to remember the distinctions and differences between theory and practice. While in theory a German soldier could have any set of rotors on him, a little though shows that there is no way that he can carry all $26! \approx 4 \cdot 10^{26}$ rotors with him in the field. Reasons for this range from the obvious (it would weigh to much, there's no way German factories could churn out that many rotors) to the more subtle (if he had even $10^{10}$ rotors on him, it would take a long time to find the right ones to use, and the Enigma is supposed to provide not only secure but also rapid communication). So, naturally, in practice there would only be a few available rotors. While this looks promising, unfortunately we still don't know the exact layout of the rotors. How can we solve this problem?

Instead of trying to magically guess the right layout, we use espionage, which in fact is exactly what the Allies did. Even before the war started, the French were able to obtain documents that gave the layout of all of the rotors in use at the time. During the war, the Allies were occasionally able to capture Enigma machines, and so if things changed, they were able to find out how. Thus, instead of having to test $6.559 \cdot 10^{79}$ different rotor wirings, they could test five (or in the German navy's case, ten, as the navy felt (correctly) that more security was needed). This is a huge savings, and without this precious information no further progress could have been made.

Next, let's look at the notches. Remember the notches determine when the rotors advance by one letter, and that only the first two notches affect the code. Therefore, you can imagine that if you could solve every other part of the arrangement, you could just decode until the first notch advanced, which would cause the decoded message to suddenly become gibberish again. Whenever that happened, you would know that another notch had advanced, and in this way, you could easily determine the initial settings of the notches.

All we have left are the rotor settings. If there were some way to separate finding the rotor settings from all the other problems, we could solve the entire cryptogram. As it turns out, there are two ways in which this was done. The first was developed by Marion Rejewski and the Polish cryptanalysts before the war, and the second was developed by Alan Turing and the British once the war had begun.

Once the Germans switched over to using the Enigma in 1926, most European intelligence agencies quickly gave up decoding German messages. There was one nation, however, that could not afford to relax: Poland. They realized that being between Germany and Russia was a dangerous place to be, especially since they now controlled formerly German areas. As a result, they needed all the intelligence they could get from communications intercepts.

When the French got a contact in the German signals corps, the Poles requested all the information that the French had on the Enigma. They also did something that had never been done before: they hired mathematicians

as cryptanalysts. In the past, cryptanalysts were generally linguists, or people who were good with words, but now, with the advent of the mathematical complexity put forth by the Enigma, a different sort of mind was required. These two moves proved to be a winning combination. They got the wiring information from espionage, and the ability to put that information to good use from the mathematicians.

The Germans distributed what was called the 'day key' in a different code book every month. The day key specified the settings for the Enigma for that day: which rotors went where, which letters to switch, and so on. The day key, however, was only ever used to transmit three letters at the beginning of the message, which indicated what the receiver should set his rotors to for the rest of the message. Except, that's not exactly right. Rather than transmit just those three letters, for much of the war the letters were transmitted twice to avoid operator error. This was an enormous mistake, and in fact, was just how Rejewski was able to break the Enigma.

The following example is taken from Simon Singh's *The Code Book* [**Si**]; for more details, see his chapter "Cracking the Enigma." Suppose you intercept a message that starts with PEFNWZ. You know that P and N are the same letter, but are enciphered differently because of the way the Enigma works. Similarly, E and W, and F and Z are the same letters, respectively. After receiving many messages and using this information over and over, you might tabulate the relationships in the following way. You know that the first and fourth letter are the same. If you intercepted four messages that started with LOKRGM, MVTXZE, JKTMPE, and DVYPZX, you would know that L and R, M and X, J and M, and D and P are all respectively the same letter. If you did this enough times, you might come up with a table like this:

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

Let's look at this closely. What is A linked to? F, as we can see by looking in the second row below A. What is F related to? Looking at the second row just below F, we see W. And below W, we see A. Well that's interesting. We've found a chain of letters that loops back on itself: A→F→W→A. Clearly, if the day key were set differently, we would have a different chain, maybe longer, maybe shorter, and probably different letters. However, and this is really important, the length of the chains is determined purely by the rotor settings! The plugboard might change what letter goes where, but the length of the chain would stay the same regardless of the plugboard setting. Using this knowledge allowed Rejewski to 'fingerprint' a specific day key. He then set about using the Cipher Bureau's Enigma replica to catalogue *every single possible rotor setting* and the link numbers associated with it. As you can imagine, this was an enormous and tedious task. However, since the number of rotor settings is only about 105,000, it *was* doable if the need was great enough, and the need was definitely great. It took the Cipher Bureau

more than a year to do it (while this is a long time, it is *magnitudes* less than the life of the universe!), but after that, all they had to do was build the table of relationships, find the size of the chains it generated, and look it up in the catalogue. As we've already established, once this was accomplished, the other parts of the Enigma code (the plugboard, notches and the like) were possible to break.

Later in the war, however, the Germans realized their mistake, and stopped transmitting the three letters for the message key twice. Once this happened, it was no longer possible to build the table of relationships, and the catalog was next to useless. Enter Alan Turing and Blechley Park (where Ultra was headquartered). Turing and his associates had been concerned that the Germans would do just that, and had developed an ingenious plan to get around it. This new method required what is known as a 'crib.' A **crib** is a word that you know will occur in the plaintext of the message. For instance, one of the most common cribs the British made use of was wetter, the German word for weather. When intercepting messages from a weather station, the word wetter is likely to occur. This time, though, instead of using letters encrypted the same way to find chains, Turing wanted to find loops occurring between plaintext letters and cipher text letters. Suppose, for example, that you knew the word 'wetter' was encrypted to 'ETJWPX.' Let's view this as a table of relationships:

$$\begin{array}{c} W\ E\ T\ T\ E\ R \\ E\ T\ J\ W\ P\ X. \end{array}$$

What does W go to? It goes to E, and E goes to T. What does T go to? Well, one T goes to W, forming a chain. If, somehow, you could find rotor settings that enciphered W, E, T, and R in such a way as to create this pattern, you would have a likely candidate for the rotor settings in the message key. Once you had the message key, deducing the day key was a simple matter. The British quickly built machines to do this, and were soon up and running.

## 1.5. Codes in World War II

While the point of this chapter is to describe the Enigma and Ultra, we would be remiss if we didn't mention some of the consequences of deciphering German messages, as well as some of the other cryptographic issues and challenges faced during the war. Numerous books have been written about each of these incidents; our purpose here is to highlight some of these issues, and excite you to read more.

(1) *Battle of the Atlantic:* Throughout the entire war, there was a constant struggle between the Allies, desperate to get food, supplies and troops from North America to assist the Allied cause around

Europe, and the Axis, desperate to sink these loaded ships. The Germans relied heavily on submarine warfare to disrupt and sink Allied shipping. Cracking the Enigma provided a decisive advantage to the Allies, who could not only adjust convoy routes to avoid enemy subs but could also use this intelligence to seek out and sink these ships. There are numerous other examples where decrypted intelligence played a key role in the distribution of Allied forces, ranging from aerial defense in the Battle of Britain (July 10 - October 31, 1940), to Operation Vengeance on April 18, 1943 (the successful attack on Japanese Admiral Yamamoto's plane, leading to his death).

(2) *Coventry: November 14, 1940.* Conventry was an important industrial city for the British, manufacturing many military items. On November 14, 1940, the Germans launched a devastating raid on the city. Some sources say the British knew about the impending attack, but only through an Ultra decrypt, which is why no additional defensive measures or warnings happened. One of the greatest fears of the British was that the Germans would realize their Enigma had been compromised. Thus, Allied forces could never act on Ultra decrypts unless there was a convincing story that would ascribe their actions to something other than having broken Enigma. While the truth about Coventry may be unknown, this was not an academic problem and occurred in various forms throughout the war. For example, Allied forces launched many aerial reconnaissance missions in the Mediterranean to find the Axis ships. The Allies already knew the location of these ships from Ultra decrypts, but it was imperative that the Germans and Italians see the Allied planes, and thus believe that this was how their positions were discovered (see, for example, the Battle of Cape Matapan, March 27 - 29. 1941). Many pilots were shot down and died to preserve the Ultra secret.

(3) *Battle of Midway: June 4 - 7, 1942.* After a horrible defeat at Pearl Harbor on December 7, 1941, American forces were on the defensive in the Pacific, and lost numerous engagements against Japanese forces. It's hard to do justice to this battle in a paragraph; it was one of the turning points in the war, a massively decisive victory for the Americans. One aspect of the battle is worth mentioning. Japanese codes were partially broken, and the Allies knew an attack was coming. Unfortunately, the target was given a code name, so even though the message was decrypted, all battle orders listed the target as 'AF'. Based on the code name and other items, many believed that 'AF' was the Japanese designation for Midway. If this were true, the Americans could deploy their

forces appropriately and surprise the Japanese; however, if 'AF' happened to refer to another target.... There was thus a pressing need to quickly determine whether or not 'AF' was Midway. The solution was ingenious. Commander Joseph J. Rochefort and his team at Station Hypo hit on an elegant solution. They had Midway send a message that their water distillation plant was damaged and request fresh water. Shortly afterwards, decrypted messages stated that 'AF' was in need of water.

(4) *Operation Fortitude: March - June, 1944.* In war, there are secrets, and then there are ***SECRETS***. We've already discussed Ultra, which fell into this category – if the Germans had realized their codes were compromised, they could have easily upgraded their security with disastrous effects. Another super-secret was the Manhattan Project, the development of the atomic bomb which ended the war with Japan in the Pacific. One more worth mentioning is D-Day. This was the Allied invasion of Europe. The fact that the Allies were planning to invade Europe in force wasn't a secret – the Axis knew such discussions were in place from the time America entered the war. What was unclear was *when* the invasion would be, and *where*. Both of these were closely guarded secrets on a need-to-know basis. Though the Allies eventually decided upon landings in Normandy, France, there were other candidate sites. One was Pas de Calais, also in France. The advantage of a landing here is that this was the closest point between British and French soil. Operation Fortitude was part of the Allied effort of disinformation to convince the Germans that the main attack would be in Calais. General Patton was given command of a fictitious army unit with simulated radio traffic, fake equipment, and false information leaked to known German agents. The deception succeeded brilliantly; even after the Normandy landings, Hitler held Panzer tank units in reserve. He was convinced these landings were a feint, and numerous units that could have attacked the Allied beachheads in Normandy sat idle waiting for an attack on Calais that never came. There is a great lesson to be learned here: one can wreak havoc on the enemy through careful misinformation, leading to a disastrous allocation of resources.

There is a common theme above. It's not enough to crack the enemy's code; one has to decide what to do with the information, which ranges from acting on the message to deciding it was a ploy designed to mislead. These decisions are one of the hardest optimization problems ever faced, with trade-offs between what is best in the short term having to be weighed against the long term advantage of decrypting future messages.

## 1.6. Conclusion

The Enigma was a complex machine, among the most advanced of its time. It had lots of great features: it could quickly encode and decode messages, it was extremely portable and easy to use, and most importantly, it appeared perfectly secure. However, due to espionage, mistakes in how it was used, and the brilliant ingenuity of some very talented people, it was broken.

We've made a few simplifications in the few comments we made about breaking the Enigma. Numerous people from many nations labored for years in these attempts; there is no way to do justice to their efforts or give credit to all in just a few pages. Our goal instead is to give a flavor for both the problem and the mathematics behind it. For more details on how it was broken and for more technical readings, we recommend [**Mi, Si, Wi**]. **MORE REFS**

# Bibliography


[CS] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, third edition, Springer-Verlag, New York, 1998.

[Go] M. J. E. Golay, *Notes on digital coding*, Proc. I.R.E. **37** (1949), 657.

[Ka] M. Kanemasu, *Golay codes*, MIT Undergraduate Journal of Mathematics **1** (1999), no. 1, 95–99. Available online at http://www-math.mit.edu/phase2/UJM/vol1/MKANEM~1.PDF

[LS] J. Leech and N. J. A. Sloane, *Sphere packings and error-correcting codes*, Canad. J. Math. **23** (1971), 718–745. Available online at http://cms.math.ca/cjm/v23/cjm1971v23.0718-0745.pdf and http://www2.research.att.com/~njas/doc/leech.html.

[Mi] A. R. Miller, *The Cryptographic Mathematics of Enigma*, NSA Pamphlet, 2001. http://www.nsa.gov/about/_files/cryptologic_heritage/publications/wwii/engima_cryptographic_mathematics.pdf

[MS] S. J. Miller and C. E. Silva, *If a prime divides a product...*, preprint. http://arxiv.org/abs/1012.5866

[MT-B] S. J. Miller and R. Takloo-Bighash, *An Invitation to Modern Number Theory*, Princeton University Press, Princeton, NJ, 2006, 503 pages.

[Si] S. Singh, *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*, Anchor Books (a division of Random House), New York, 1999.

[Th] T. M. Thompson, *From error-correcting codes through sphere packings to simple groups*. The Carus Mathematical Monographs, Number 21, the Mathematical Association of America, 1983.

[Wi] J. Wilcox, *Solving the Enigma - History of the Cryptanalytic Bombe*, NSA Pamphlet, 2001. http://www.nsa.gov/about/_files/cryptologicvheritage/publications/wwii/solving_enigma.pdf