Chapter 2

Background

This is not a history book. Yet to understand the basics of where modern cryptographic engineering has gone, it's helpful to understand where cryptography came from and how it reached its present state. If you already know a bit about cryptography, most of this material is already familiar to you. However, some of it may not be.

2.1 Confidentiality, Adversaries and Symmetric Ciphers

In Greek, cryptography literally means "hidden writing." Since antiquity, cryptographers have primarily been concerned with communicating privately, usually in the face of a person — called an *adversary* — that wishes to read their messages. This means that much of cryptography historically dealt with a single property: *confidentiality.*

Traditionally, confidentiality was dealt with using a *cipher.* Named from the Arabic word for "zero", a cipher is an algorithm that translates a *plaintext* message into another form called a *ciphertext* that cannot easily be read by an adversary. A legitimate recipient can *decipher* and recover the message. What separates the authorized recipient from the adversary is a piece of special knowledge called a *key.* The nature of a key varies between ciphers. Historically, sometimes the design of the cipher was kept secret, and itself formed the "key". However this approach makes improvement slow and difficult, and the result has been many broken ciphers. In more recent ciphers, the design of the cipher is assumed to be well-known. Thus the key is instead a separate piece of information, such as a long number, a table of substitutions, or a collection of machine settings.

2.2 Historical ciphers

Shift ciphers

There are many classical ciphers, and nearly all of them are broken in one way or another. The most famous, and also the least secure, is the *shift cipher* (or "Caesar") cipher.[1] En-

[1] Julius Caesar did not invent this cipher, but he allegedly used it frequently.

cipherment involves substituting each letter in the original plaintext message with a letter that has been shifted by some number of positions further into the alphabet. For example, with a shift of two, each instance of the letter `A` would become `C`, and the letter `T` would become `V`. For pesky letters like `Y`, the shift simply wraps around back to the beginning of the alphabet, ensuring that `Y` becomes `A` and so on.

The shift cipher is a bit easier to think about if we represent each letter as a number between 0 and 25 (i.e., `A` becomes 0, `B` becomes 1, and so on). In this case, applying the shift cipher key to a single letter can be represented using the following simple equation:

$$C = P + K \ mod \ 26$$

Here P and K are variables that represent the plaintext and key (shift value) respectively. The *mod* operator represents the process of dividing the sum $(P + K)$ by 26 and then outputting only the remainder from that division.[2] Decryption uses a similar equation, though with subtraction substituted for addition.

The problem with the shift cipher becomes obvious when you think about how you'd break, or *cryptanalyze*, someone else's message. Since there are only 25 unique shift values in the English alphabet (technically there are 26 possible shifts, but a zero shift will produce an enciphered message that is identical to the plaintext) an adversary can simply try each key until they produce a recognizable decipherment. Even using pencil and paper, this takes only a modest amount of time. This style of attack is known as a *brute force* attack.

Substitution ciphers

The obvious vulnerability in the shift cipher is the small number of possible keys, or — in different language — the small size of the *key space*. An ideal direction, therefore, is to design a cipher with a much greater set of possible keys. The *substitution cipher* is one example. It replaces the simple shift with a table: each letter is substituted with another letter of the alphabet, mathematically forming a permutation of the alphabet. This approach produces many more possible keys: there are 26 possible substitutions for the first letter, then 25 for the second, and so on, giving a total number of keys as 26!, or about 403 *trillion trillion* (in powers of two, this can be expressed as about 2^{88}.[3]) This is an extremely respectable key size, and unlikely to be vulnerable to simple brute force guessing.

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

Figure 2.1: Example key for a simple substitution cipher.

Unfortunately, simply having a larger key does not make a cipher secure. The substitution cipher falls prey to a different attack. Since every plaintext letter always enciphers to a

[2] We discuss modular arithmetic in more depth in Chapter 4.

[3] Throughout this book we will use powers of 2 to represent the number of possible keys, for reasons that will be more apparent later on.

specific ciphertext letter — for example, each `A` in the plaintext enciphers to `O`, patterns from the underlying plaintext still poke through in the resulting ciphertext. For example, the word `LOOK` might encipher to `WMMP`, revealing the distinctive double-letter pattern. This problem is even more serious than it looks, because in many languages certain letters appear more frequently. If, for example, the letter `E` occurs most commonly in the plaintext, and the cipher maps each one to the letter `R`, then an adversary may be able to determine that `R` is a good candidate to be the letter `E`, just by noticing how frequently it occurs in the text. This technique, known as *frequency analysis*, does not work in every single example — but it works well enough to be very effective against substitution ciphers.

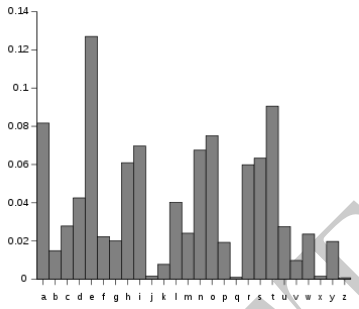


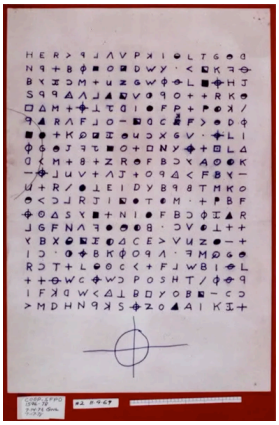
Figure 2.2: An example chart showing letter frequencies for a corpus of English text.

From one perspective, a large part of the history of classical cryptography can be viewed as an effort to overcome the very serious limitations of the simple substitution cipher. Indeed, many later ciphers can be thought of as "hacks" built on top of the substitution cipher, in order to fend off simple frequency analysis. These include homophonic substitution ciphers, which map each plaintext letter to several alternative ciphertext symbols (including letters, numbers and cute hand-drawn symbols); polyalphabetic ciphers that use a distinct substitution alphabet (or shift key) to encipher letters at different positions in the plaintext; digraph ciphers that encipher pairs of letters at a time, and transposition ciphers that reorder letters. The common feature of each approach is that it attempts to foil frequency and pattern analysis by reducing the incidence of those patterns.

Vigenère (Bellasso) cipher

To address the known weaknesses of substitution ciphers, the 16^{th} century cryptographer Giovan Battista Bellaso invented a new twist on the shift cipher. Although Belasso's invention is clearly first, his invention was incorrectly credited to the French cryptographer Blaise de Vigenère and thus bears the latter's name today. Unfortunately, sometimes this happens, and we will mostly use the latter's name as well.

The Vigenère/Bellasso cipher begins with the same idea as the Shift cipher. However, it has an important difference. Instead of a single shift applied uniformly to all letters in the plaintext, the "key" is actually a sequence of different shifts. These shift values are applied



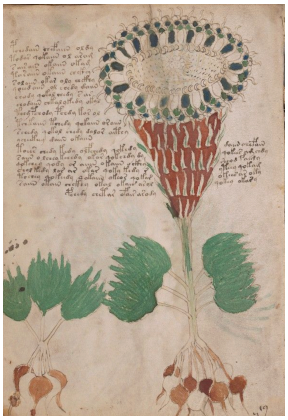


Figure 2.3: **Left:** 340-character enciphered message from the "Zodiac" killer, which appears to be a combination of a transposition cipher (in which letters are re-ordered) and a homophonic substitution cipher. This was decoded in 2020 by David Oranchak, Sam Blake and Jarl Van Eycke. **Center:** Page from the Voynich manuscript, a potentially ciphered work from the early 15^{th} century. The meaning of the text has not been uncovered. **Right:** Portion of a deciphered homophonic substitution cipher used by Mary, Queen of Scots. Matt: Permissions, Smithsonian

to the letters of the plaintext one by one, in sequence. For example, the first letter in the plaintext might be enciphered using a shift of 2, while the second letter of the plaintext might be enciphered with a shift of 10, and so on. To make remembering these shifts easier for human beings, Bellasso proposed to represent the key as a sequence of letters: here `A` represents a shift of 0, and `B` represents 1 and so on. Thus an encryption key can be chosen as a simple word, like `PASSWORD`. The enciphering party simply "adds" the first letter of the plaintext to the first letter of the key using modulo 26 arithmetic, and continues until they have run out of key material. At this point, they can simply repeat the key a second time. Encipherment looks like this:

```
  A T T A C K A T D A W N
+ P A S S W O R D P A S S
  P T L S Y Y R W S A O F
```

The use of multiple shifts dramatically increases the number of possible keys. For example, a 10-letter Vigenère/Bellasso key can have $26^{10} \approx 2^{47}$, or approximately 141 trillion different values. Of course, this assumes that the key is any combination of random letters. If the key is chosen to be a meaningful English word or phrase, the number of keys is much smaller.

Baudot Code and the Vernam Cipher

In the late 1800s, a French engineer named Émile Baudot invented an efficient means of mechanically encoding telegraph signals. His machine used a special binary code in which

every letter was converted into a sequence of five binary digits, which could then be transmitted as electrical pulses over a wire. English text could be converted into symbols by a first machine, multiplexed with many other transmissions, and then decoded by a second machine that would print the readable text. Baudot code is thus an early use of binary encoding for the English alphabet, and it can be viewed as a precursor of the ASCII code or Unicode lettersets used in modern computers.

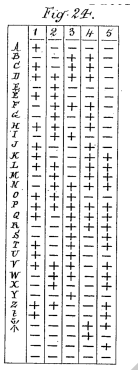


Figure 2.4: Baudot code table [24]. Plus and minus symbols can be thought of as binary digits.

Although Baudot's invention was not a cipher, the widespread use of teleprinter devices inspired the need for better security. To address this, in 1917 an AT&T engineer named Gilbert Vernam invented a machine that could encipher Baudot-encoded messages using a "key" stored on a tape. Vernam's device was essentially a mechanical version of the Vigenère/Bellasso cipher. However, rather than enciphering an alphabet of 26 characters, Vernam's machine worked directly on the binary alphabet produced by the Baudot teletype systems.

Vernam's realization was simple: implementing the classical Vigenère cipher would require a complicated circuit that could compute addition modulo 26. However, the binary Baudot code itself has only a *two-symbol* "alphabet": 0 and 1. If P and K are both single binary digits (or bits), then encipherment of a single bit can be realized using the equation $(P + K) \ mod \ 2$, which was much easier to realize using the simple technology of the day. As an added bonus, when computed modulo 2, addition and subtraction turn out to be exactly the same operation! This meant that encipherment and decipherment could be performed using exactly the same circuit. Indeed, modern engineers will recognize that addition (and subtraction) modulo 2 is equivalent to the bitwise *exclusive OR* (XOR) boolean logic gate (which we denote by $\oplus$, see Figure 2.6), something that is particularly easy to implement using electrical circuitry.

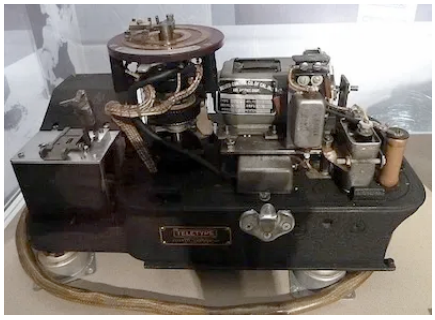


Figure 2.5: A U.S. military SIGTOT cipher machine, which is directly based on Vernam's 1919 patent. This one was used in President Roosevelt's airplane. Courtesy: National Cryptologic Foundation.Matt: Permission

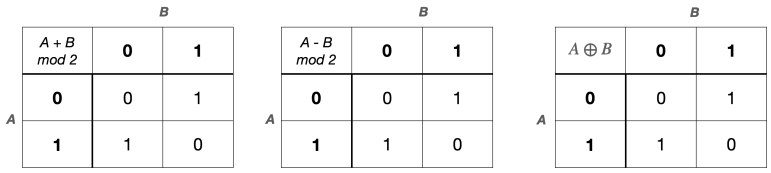
Mauborgne and One-Time Ciphers

Shortly after the release of Vernam's invention, a military cryptographer named Joseph Mauborgne made a surprising discovery: if the key used in a Vernam cipher was *completely random* and it was only used to encrypt one message — meaning, it was at least as long as the message to be encrypted, and never repeated — then the resulting ciphertext would be theoretically unbreakable. Although Mauborgne was not the first (or last) to realize this,[4] he was the first inventor with an efficient means to put it into practice.

The rough intuition behind Mauborgne's *one time cipher* is as follows. Imagine that you, an adversary, intercept a Vernam-machine ciphertext: call it C (this may be a single bit, or many bits long) but you do not know the key. Now consider what you can learn about the underlying plaintext, assuming that the key was chosen perfectly at random and used only once. The intuition Mauborgne and others developed was as follows: consider trying every possible key K one after another, and then attempting to decipher C with each one. In this setting, you will end up deciphering C to *every single possible plaintext* of the same length. Put differently, for every possible plaintext message of this length, there exists exactly one key K that will decipher C to that plaintext message! Moreover, every possible key is equally likely to have been the real one, in a system where the real key was chosen completely at random.

Put more concretely: if C is 14 letters long, then there is one key that causes C to decipher to `ATTACK AT DAWN` and a second key that causes C to decipher to `RETREAT TONGHT`. Given only knowledge of the ciphertext, both plaintexts are equally likely to be the correct decryption. (You might, of course, have a guess about which message is more likely based on extrinsic factors, such as your observation of which way the army seems to be moving. But crucially, *the ciphertext itself* does not tell you anything new.) This remarkable observation would later be mathematically formalized by the "father" of information theory, Claude

[4]The first discovery of one-time ciphers is typically credited to Frank Miller in 1882, and the theory was formalized by Claude Shannon in 1945. It has been independently discovered and re-discovered many other times.



A	B	0	1
	A + B mod 2		
	0	0	1
	1	1	0

A	B	0	1
	A - B mod 2		
	0	0	1
	1	1	0

A	B	0	1
	$A \oplus B$		
	0	0	1
	1	1	0

Figure 2.6: The *truth tables* for the functions $A + B \ mod\ 2$, $A - B \ mod\ 2$, and $A \oplus B$.

Shannon [156]: ciphers that have this property are now referred to as possessing *Shannon secrecy*.

Downsides of one-time ciphers. Of course, the existence of an *unbreakable* cipher raises an obvious question: why don't we use it everywhere? The answer lies in the stringent requirements that apply to one-time cipher keys: each key must be distributed to the recipient, and can only be used for encrypting one message. This is so onerous in practice that the one-time cipher (also called a *one-time pad*) is only used for a handful of extremely sensitive applications: these include espionage, as well as radio broadcasts intended for sensitive assets such as submarines.

If you're tempted to bend these rules, beware: the Vernam-Mauborgne one-time cipher can fail catastrophically if a key is ever used to encrypt two different messages. This is because of a simple property of exclusive-OR: given ciphertexts $C_1 = M_1 \oplus K$ and $C_2 = M_2 \oplus K$, an adversary can simply compute $C_1 \oplus C_2$ to cancel out the key K from both ciphertexts, revealing the result $M_1 \oplus M_2$. In many cases it is then possible to figure out parts of each plaintext from that mixture. Something like this occurred in the 1940s, when rushed Soviet printers accidentally distributed multiple sets of identical one-time cipher keys to Soviet intelligence stations throughout the world. When U.S. cryptanalysts learned of the issue, they were able to painstakingly decipher thousands of Soviet communiqués, revealing precious information about Soviet espionage.[5]

Mechanical ciphers

Given the difficulty of distributing keys for one-time ciphers, military cryptographers kept searching for ways to securely encrypt messages with less key material. This involved a common approach: using machinery to make the encipherment process more *complex*, while still allowing human operators to use the system easily. With the assistance of mechanical devices, an operator merely needed to configure the machine: set it up with a key, type a message into a keyboard, and then read out the resulting ciphertext. The machine could then handle the remaining details, applying a degree of mathematical sophistication that would be too difficult for a human operator to perform with pencil and paper alone.

[5]The Soviet one-time cipher was slightly different from the Vernam-Mauborgne binary cipher, but used similar additive encipherment. The resulting decipherment effort was codenamed the VENONA project, and its results have been declassified by the National Security Agency [7].

State ciphers and the Enigma machine. While there are many mechanical ciphers, the most famous by far is the German Enigma machine. Invented at the end of World War I, the Enigma is a simple and compact device that resembles a typewriter. Internally, it uses a combination of electrical wiring and mechanical rotors to efficiently encipher long messages.

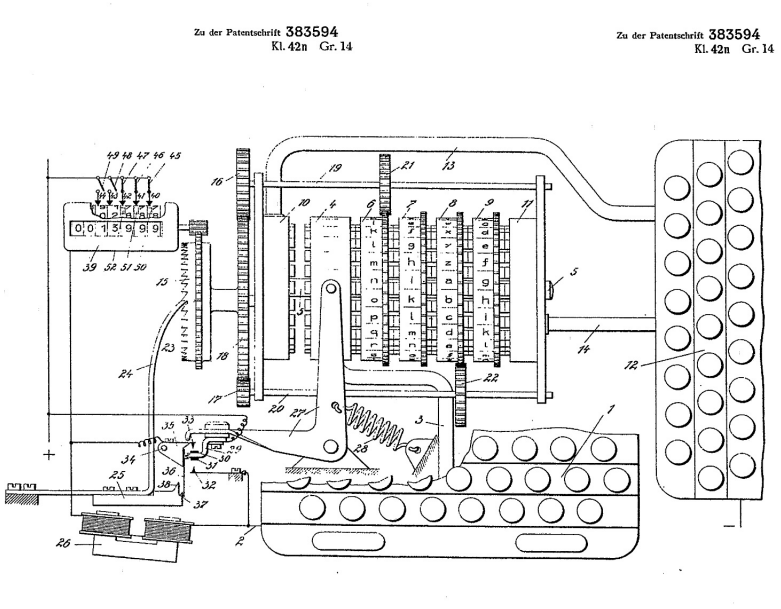


Figure 2.7: Overview diagram of an early Enigma machine showing keyboard, lights and rotors [1].

At one level, the principle behind the Enigma is simple: it is a form of substitution cipher. A keyboard with 26 letters is electrically connected, via a series of plugs and rotors, to the same number of light bulbs. Depressing the `A` key completes a circuit through the machinery, lighting up a labeled bulb – for example, representing the ciphertext letter `X`. Due to the special wiring of the Enigma, the inverse relationship is also true: with the machine at the same settings, depressing `X` will light up the bulb labeled `A`. This means that an Enigma in the right setting can be used to both encipher and decipher messages.

If the internal wiring of the Enigma remained static, it would be *exactly* a substitution cipher, with all of the vulnerabilities that would entail. However, the wiring does not remain static: from an initial configuration, each keypress causes the rotors to turn slightly, in much the same manner as the odometer of a classic car. This changes the substitution alphabet realized by the machine. The crucial ingredient here is the electrical wiring of the rotors themselves. Each contains 26 electrical contacts on each side: electricity flows into one side, then out a different point based on a tangle of wiring inside the rotor. By combining three or more rotors and changing their relative position, the Enigma can produce trillions of substitution alphabets that will differ at each position of the message.

Many books have been written about the design and cryptanalysis of the Enigma, and

this book will not add much to them. What is important for our purposes is to note that each keypress changes the configuration, or *state*, of the machine. Ciphers with this nature are called *state ciphers*. The use of mechanical and electrical components vastly increases the number of states that can be achieved by the machine, making cryptanalysis surprisingly difficult.

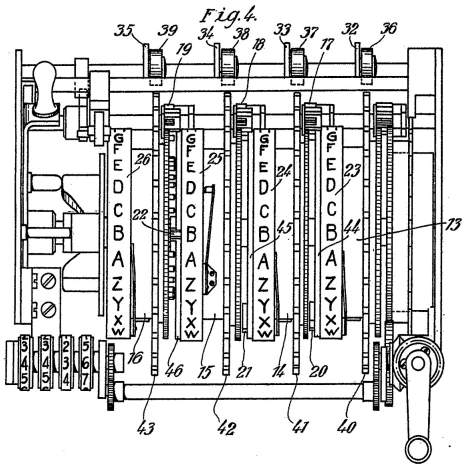


Figure 2.8: Detail of the Enigma rotors [2].

Key distribution and session keys. Although the Enigma design is no longer considered a secure cipher, its usage highlights some important lessons about key distribution that are still relevant today.

To facilitate communication with many remote sources, German high command produced codebooks that contained lists of "daily keys" to be used for communication (see Figure 2.9.) A key comprised the exact set (and order) of the rotor wheels to be inserted, the position of the rotor wiring relative to the outer lettering, the configuration of plugs on the plugboard, as well as the initial position of the wheels. The latter was identified by the letters that would show through a slot in the Enigma casing.

Because daily keys were widely shared by many transmitters, they were typically not used to encipher message contents. Instead, operators were instructed to select a random initial position for the Enigma rotors. This formed a *session key* that would, in some usages, then be enciphered using the daily key, and this short ciphertext would be transmitted. The message content itself would then be enciphered using the session key. In principle this avoided a situation where the same daily key would be used to encipher many long messages.[6] This practice — enciphering messages using a short-lived session key, then enciphering the

[6]For reasons specific to the nature of the Enigma cipher, this early practice was ultimately deemed to be flawed. Later usage also required operators to select a random initial starting position, but transmitted this position in the clear — relying on the remaining unknown settings from the daily key to ensure security.

session key under a long-lived *key encrypting key* (KEK) — is still widely used in modern cryptography.

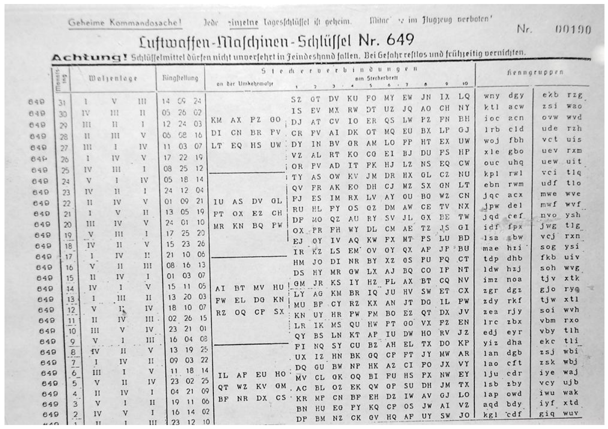


Figure 2.9: List of daily Enigma keys for the German Luftwaffe circa 1944. *Public domain image.*

The Enigma machine was famously cryptanalyzed by a British effort that included the famous computer scientist Alan Turing; more than one film has been made about those events. The British team built mechanical (and later digital) computers that were explicitly designed to test many Enigma keys quickly, in order to successfully decrypt Enigma messages. Because the number of possible keys is so large, many of the successes of that team relied on exploiting operational mistakes in the way that Enigma operators enciphered their messages: this allowed the British to massively reduce the number of keys they had to test. Turing's ideas were later used to devise one of the earliest digital codebreaking computers, but this work was classified and only revealed decades later. Modern digital computers can now make short work of Enigma ciphertexts.

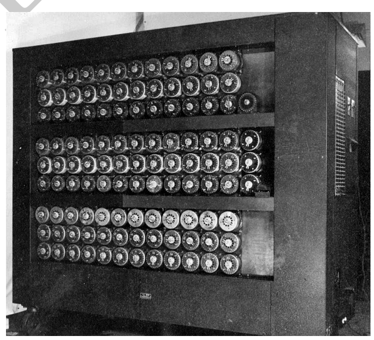


Figure 2.10: One of the Enigma "bombes" designed by Turing and the cryptanalysts at Bletchley Park. The bombe was a mechanical computer designed to test many Enigma keys at one time.

Other mechanical ciphers The Enigma is only one of a large class of rotor-based ciphers. Many other cipher machines use a very design similar to the Enigma, though often with a larger numbers of rotors and more complex wiring. One particularly sophisticated later example is the Soviet Fialka machine (Figure 2.11) that features ten rotors, each of which turns in the opposite direction from its neighbors. The Fialka was used for nuclear command and control as late as the 1990s.

Figure 2.11: Soviet Fialka machine (*source: Nick Gessler*)Matt: NEEDS PERMISSION