Breaking Enigma
Codebreaking Techniques from WWII

Kaleb Davis
Courtney McGorrill
May 2017

1 Introduction

(TODO: Add introduction content)

2 History

(TODO: Add history content (Courtney)) Take a look at chapters/oldfiles/10-tutorial.tex for formatting tips

3 Background

The following will provide an introduction into some of the core concepts necessary to understand how Polish cryptologists were able to break Enigma in 1932. We will introduce the hardware of the Enigma machine, as well as provide a brief overview of permutation theory. We will also prove a theorem which is integral in creating the permutations used in the set of equations that model the electrical circuit inside Enigma.

3.1 Enigma Hardware

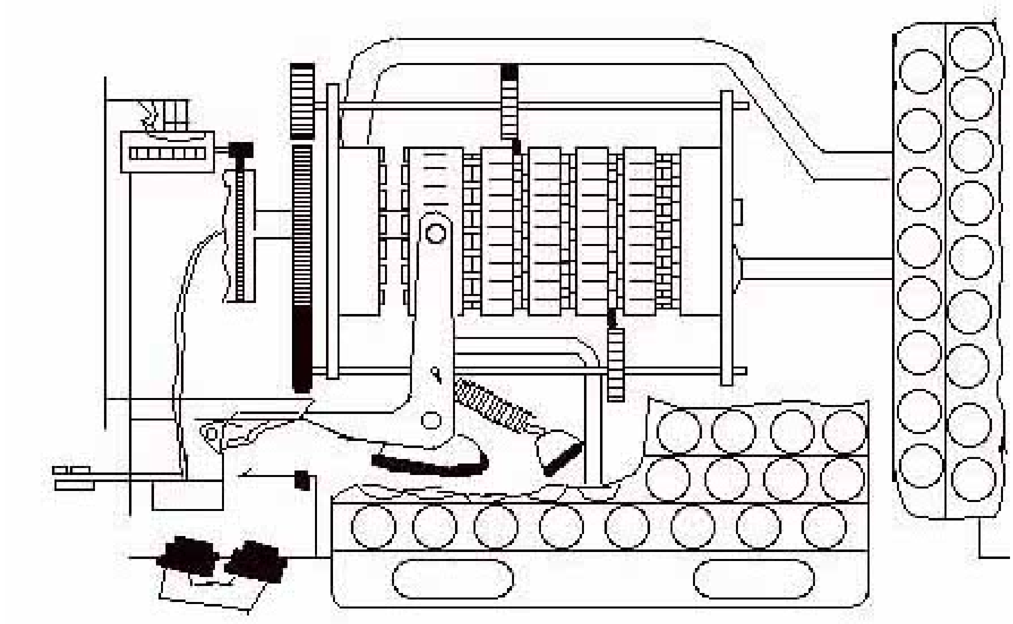

Figure 1: Hardware of the Enigma Machine

In order to understand how the ciphertexts created by the Enigma machine were broken, it is important to understand the inner workings of the machine itself. Figure 1 shows a schematic of the hardware.

On the outside of each machine, there is a keyboard and a row of glowlamps. Each key on the keyboard is connected to a glowlamp through a changing electric circuit, so when a key is pressed it lights up a corresponding glowlamp. Below the keyboard, there is a plugboard with between six (6) and twelve (12) switches. These switches allow for two letters of the alphabet to be transposed prior to being sent into the machine's hardware. It introduced a "reciprocal monoalphabetic substitution between the keyboard and the first rotor" [1]. This adds a layer of security beyond the rotors on the inside of the machine.

Inside each machine, there are anywhere from three (3) to as many as eight (8) rotors and a reflector (or reversing drum). These rotors are the main ciphering components. Each rotor has the alphabet inscribed on the rim, twenty-six (26) fixed contacts on one face, and twenty-six (26) spring loaded contacts on the other face [2]. Each rotor has a unique circumference, as well as a unique set of connected contacts. These contacts are randomly connected, and are different on each rotor [1]. The reversing drum is responsible for creating the reciprocal nature of the machine, meaning that if an 'A' is pressed on the keyboard and an 'F' lights up on the glow lamps, it also means that if an 'F' is pressed on the keyboard, the 'A' will light up on the glow lamps.

Each rotor inside the machine is set up in such a way that it will rotate corresponding to different key presses. The rotor closest to the keyboard rotates every time a key is pressed, meaning that the substitution changes every time a key is pressed. The other two (2) to seven (7) rotors rotate at variable rates, depending on how the hardware is configured. The second rotor's rotation is dependent on the first rotor's rotation, the third rotor is dependent on the second, and so on and so forth. This rotation of the rotors adds another level of complexity on top of the already complex substitution cipher that Enigma creates.

3.2 Important Mathematical Concepts

To understand some of the mathematical theory later on in this paper, one must first understand some basics of permutation groups. A permutation is a rearrangement of elements in a set. An example of this kind of permutation is rearranging the numbers $1, 2, 3, 4, 5$ into $3, 5, 4, 2, 1$. This could be expressed in the standard matrix form $P = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 4 & 2 & 1 \end{pmatrix}$, or in cyclic notation as $P_c = (13425)$. Notice in cyclic notation that there is an implied transposition from 5 to 1 from the last element in P_c.

Permutations can be multiplied as well. In multiplying permutations, order is important. If we have

$$P = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 4 & 2 & 1 \end{pmatrix}$$

$$Q = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 1 & 4 & 3 & 5 \end{pmatrix}$$

One can multiply

$$QP = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 1 & 4 & 3 & 5 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 4 & 2 & 1 \end{pmatrix}$$

To multiply, rearrange the columns of the Q permutation (leftmost) so that the first row matches the P permutation (or the rightmost). Then, take the non-matching rows of P and Q as the product. In this example, we get

$$QP = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 1 & 4 & 3 & 5 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 4 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 5 & 4 & 2 & 1 \\ 4 & 5 & 3 & 1 & 2 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 4 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 5 & 3 & 1 & 2 \end{pmatrix}$$

In cyclic notation, $QP = (14)(25)(3)$.

Another important concept necessary in order to fully understand the theory behind cracking the Enigma machine cipher is the *Theorem on the Product of Transpositions*. First, a transposition is a 2-cycle permutation, for example $P = (13)$. A group of disjunctive transpositions, then, is a group of non-overlapping 2-cycles.

The Theorem on the Product of Transpositions states that:

If two permutations of the same degree consist solely of disjunctive transpositions, then their product will include disjunctive cycles of the same lengths in even numbers.

The proof is as follows:

We assign two permutations to be multiplied to be X and Y, with total degree $2n$. If both X and Y have identical transpositions within them, such as (ab), then the product will have two distinct cycles (a) and (b). This makes up our base case, as any two permutations with identical transpositions will have an even number of disjunctive transpositions.

Our next step occurs as such. If permutation X includes a transposition (a_1a_2), there must be a permutation in Y that begins with a_2, such as (a_2a_3). We have already ruled out the possibility of Y including (a_2a_1) in our base case. We can continue this logic and say the following:

$$(a_1a_2), (a_3a_4), ..., (a_{2k-3}a_{2k-2}), (a_{2k-1}a_{2k}) \in X$$

$$(a_2a_3), (a_4a_5), ..., (a_{2k-2}a_{2k-1}), (a_{2k}a_1) \in Y$$

From these two sets X and Y, when we multiply them together we will always obtain two cycles of the same length $k \leq n$:

$$(a_1a_3...a_{2k-3}a_{2k-1})(a_{2k}a_{2k-2}...a_4a_2)$$

We continue this step until there are no more elements in X and Y, with the result that the product will only include disjunctive cycles of the same lengths in even numbers, which concludes the proof. This proof was adapted from Appendix E from Kozaczuk [2].

In the breaking of Enigma, Polish mathematicians also used the converse of the above proof, which states that *if a permutation of even-numbered degree includes disjunctive cycles of the same lengths in even numbers, then this permutation may be regarded as a product of two permutations, each consisting solely of disjunctive transpositions.*

4 Examples

4.1 Encryption Process

In accordance with Kerchoff's principle, the method of encrypting a message was known by the Polish and British cryptologists attempting to break the cipher. In this paper, we will touch on two (2) separate methods of encrypting a message that were used during World War II. The first method was common practice until September 1938, when Germany decided to increase security by changing to the second method, which was used from then on.

4.2 Encryption Process Pre-1938

Prior to the security increase in 1938

4.3 Mathematical Theory

4.4 Encryption Process Post-1938

4.5 Cryptological Machines

5 Conclusion

(TODO: Add conclusion content)

6 References

(TODO: Add references)

7 Introduction

This is a template for an undergraduate or master's thesis. The first sections are concerned with the template itself. If this is your first thesis, consider reading Section 7.3. Of course, the structure of this thesis is only an example. Discuss with your adviser what structure fits best for your thesis.

7.1 Template Structure

- To compile the document either run the makefile or run your compiler on the file 'thesis_main.tex'. The included makefile requires latexmk which automatically runs bibtex and recompiles your thesis as often as needed. Also it automatically places all output files (aux, bbl, ...) in the folder 'out'. As the pdf also goes in there, the makefile copies the pdf file to the parent folder. There is also a makefile in the chapters folder, to ensure you can also compile from this directory.

- The file 'setup.tex' includes the packages and defines commands. For more details see Section 7.2.

- Each chapter goes into a separate document, the files can be found in the folder chapters.

- The bib folder contains the .bib files, I'd suggest to create multiple bib files for different topics. If you add some or rename the existing ones, don't forget to also change this in thesis_main.tex. You can then cite as usual [1, 2, 3].

- The template is written in a way that eases the switch from scrbook to book class. So if you're not a fan of KOMA you can just replace the documentclass in the main file. The only thing that needs to be changed in setup.tex is the caption styling, see the comments there.

7.2 setup.tex

Edit setup.tex according to your needs. The file contains two sections, one for package includes, and one for defining commands. At the end of the includes and commands there is a section that can safely be removed if you don't need algorithms or tikz. Also don't forget to adapt the pdf hypersetup!!
setup.tex defines:

- some new commands for remembering to do stuff:
  - `\todo{Do this!}`: **(TODO: Do this!)**
  - `\extend{Write more when new results are out!}`: **(EXTEND: Write more when new results are out!)**
  - `\draft{Hacky text!}`: **(DRAFT: Hacky text!)**
- some commands for referencing, 'in `\chapref{chap:introduction}`' produces 'in Chapter 7'
  - `\chapref{}`
  - `\secref{sec:XY}`
  - `\eqref{}`
  - `\figref{}`
  - `\tabref{}`
- the colors of the Uni's corporate design, accessible with `{\color{UniX} Colored Text}`
  - UniBlue
  - UniRed
  - UniGrey
- a command for naming matrices `\mat{G}`, $\mathbf{G}$, and naming vectors `\vec{a}`, $\mathbf{a}$. This overwrites the default behavior of having an arrow over vectors, sticking to the naming conventions normal font for scalars, bold-lowercase for vectors, and bold-uppercase for matrices.
- named equations:

  ```
  \begin{align}
  ```

```
    d(a,b) &= d(b,a)\\ \eqname{symmetry}
\end{align}
```

$$d(a,b) = d(b,a) \tag{1}$$

symmetry

7.3 Advice

This section gives some advice how to write a thesis ranging from writing style to formatting. To be sure, ask your advisor about his/her preferences.
For a more complete list we recommend to read Donald Knuth's paper on mathematical writing. (At least the first paragraph). `http://jmlr.csail.mit.edu/reviewing-papers/knuth_mathematical_writing.pdf`

- Don't use passive voice. It's harder to read, more likely to produce errors, and most of the times less precise. Of course there are situations where the passive voice fits but in scientific papers they are rare. Compare the sentence: 'We created the wheel to solve this.' to 'The wheel was created to solve this', you don't know who did it, making it harder to understand what is your contribution and what is not.
- If you use formulas pay close attention to be consistent throughout the thesis!
- Usually in a thesis you don't write 'In [24] the data is..'. You have more space than a paper has, so write 'AuthorXY et al. prepare the data... [24]'. Also pay attention to the placement: The citation is at the end of the sentence before the full stop with a no-break space. `... last word~\cite{XY}.`
- Pay attention to comma usage, there is a big difference between English and German. '...the fact that bla...' etc.
- Do not write 'don't ', 'can't' etc. Write 'do not', 'can not'.
- If an equation is at the end of a sentence, add a full stop. If it's not the end, add a comma: $a = b + c \quad (1)$,
- Avoid footnotes if possible.
- Use ``` ``'' ``` for citing, not `""`.

Bibliography

[1] B. J. Winkel, C. A. Deavours, D. Kahn, and L. Kruh, *The German Enigma Cipher Machine: beginnings, success, and ultimate failure*. Boston, MA: Artech House, 2005.

[2] W. Kozaczuk, *Enigma: how the German machine cipher was broken, and how it was read by Allies in World War Two*. S.l.: Univ. Publ. of America, 1985.

[3] W. Trappe and L. C. Washington, *Introduction to Cryptography with Coding Theory*. Upper Saddle River, NJ: Pearson Prentice Hall, 2006.