Enigma in the 21st Century

***Abstract*—The idea of cryptography is as ancient as civilizations, communicating sensitive information which can be read by anyone but only understood by the ones who have the right knowledge of the message was crucial. Cryptography is not bound by languages but by the concepts of mathematics and the definition of information itself. There is a mathematical difference between everyday information we communicate and messages that are encrypted. We start by understanding how certain versions of the Enigma worked under the hood that existed as a necessary means of communicating confidential information of the militia in the 20th Century, some of which was cracked by different military intelligence organizations of the Allies. Cracking this eventually led to the outcome of the war and how it shaped modern society. We will then do cryptanalysis of some influential derivatives of the Enigma which at the time was thought to be unbreakable and crack it with the power of the modern silicon chips and by the techniques that were used to break them during the war. We will also learn how encryption has changed throughout the century, what's different in modern encryption standards and how well it can stand against future attacks.**

***Index Terms*—Enigma, Cryptography, Bombe, Substitution ciphers**

I. Introduction

The most basic method of encryption you can do is a substitution cipher. Most of you have probably done it when you were passing notes or when writing down something secret. There are Mono-alphabetic substitution ciphers like Caesar Cipher and ROT13, which remained secure and widely used for centuries until frequency analysis was discovered to crack this. Poly-alphabetic substitution ciphers are a bit more complex than Mono-alphabetic ones and were first introduced in 14th century by Al-Qalqashandi [1]. One of the most widely used Poly-alphabetic substitution cipher was the Vigenère cipher, it too was cracked, by Friedrich Kasiski [2] which first determines the key length then essentially using each of the letters of the key as individual Caesar ciphers you are able to find out the key by using the frequency analysis yet again. There have been countless number of enumerations of ciphers created and cracked, this feedback mechanism is what allowed modern cryptographic algorithms to take shape and learn how to make better ciphers.

We will be breaking apart each individual component of Enigma to better understand how it works under the hood and at the end, crack it using modern computation power by writing our own emulator and crack with a modern programming language like JavaScript. Even though JavaScript is not one of the most performant language, it should be more than enough when we compare it to the computational speed of Bombe.

II. Background

Enigma is a sophisticated Poly-alphabetic substitution cipher used widely by the Nazi Germany during WWII to coordinate and communicate their large scale military operations. Even though it was a Poly-alphabetic substitution cipher like Vigenère or Alberti cipher, it was a more complex variant. This is because the cipher changes every time we input a character and the sheer number of explosive possible states the machine can have. The Fig. 1 shown below is a fully functioning Enigma that has been recovered.

Fig. 1. Enigma model I [3]

A. Definitions

A simple substitution cipher will replace a character in the alphabet with another character, also called Mono-alphabetic substitution cipher. Caesar Cipher is the easiest of substitution ciphers, unlike characters being replaced by arbitrary characters; Caesar Cipher will replace every character to another character with a fixed length. As an example the characters A,B,C,D,... goes to E,F,G,H... here the fixed length is 4 thus 4 is the key, You can also use a key that encrypts characters to different lengths; as an example we can use the key ENCRYPTIOABDFGHJKLMQSUVWXYZ to encode "PLAINTEXT" to "JDEOGQYWQ" Here E is substituted to A, B to N and so on.

To create an even secure cipher, the method of Poly-alphabetic substitution was adopted. Instead of using a single

substitution for letters, it chains multiple substitution ciphers Vigenère cipher is a simple Poly-alphabetic substitution cipher. To encrypt the plaintext ”WEDNESDAY” with the key ”CENTURY” you first repeat the key to fit the plaintext so in this case it becomes ”CENTURYCE”, The encryption process is simple you create a table shown in Fig. 2 Now to encrypt

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

Fig. 2. Vigenère Table

you go letter by letter of the key and plaintext, then pick the character at column of the key and row of the plaintext. For our first character of the ciphertext it's Y because it's in column C and row W, our second character is at column E, row E which is I, and so on. The ciphertext in this case is YIQGYJBCC.

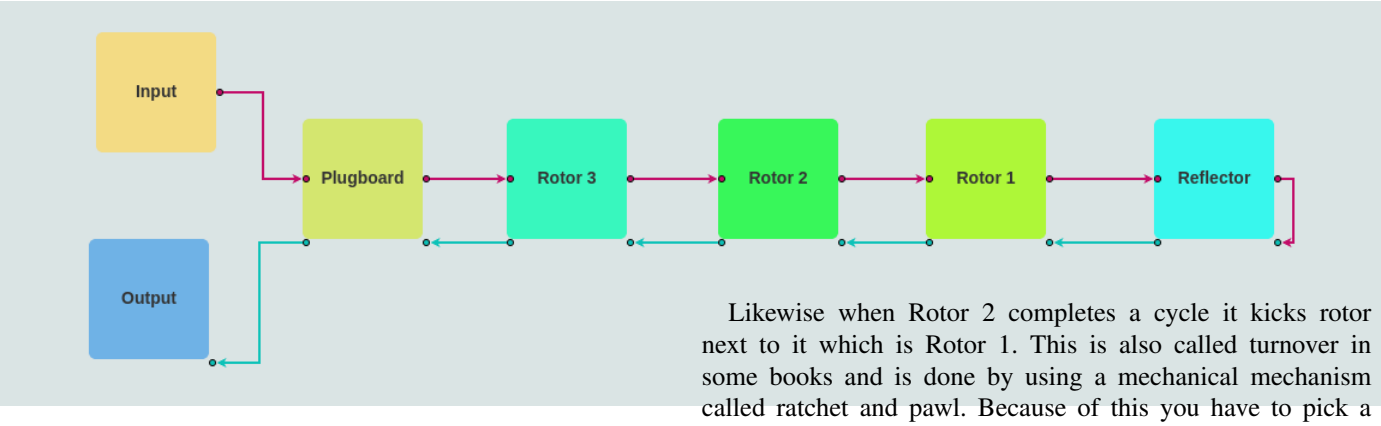


Fig. 3. Abstract layout of enigma

To decrypt we go backwards that is we find the ciphertext character in the column of our key to get back the row. Because it's more complex than a simple substitution cipher, trivial frequency analysis wont work, however there are methods to attack this cipher which we will discuss in the following sections.

III. Investigations

A. Overview of Enigma

Enigma was a mechanical machine that operated with electricity. Although it may seem primitive it was astonishingly complicated. It consists of 3 different parts, the rotors, plugboard and the reflector, the plugboard was added in later by the German army to be used for military communication. Like other substitution ciphers enigma too worked entirely in English alphabet which means you cannot encrypt other numbers or symbols into it explicitly. When a key is pressed, the electrical current first goes through the plugboard, then to Rotor 3, which passes to Rotor 2, through Rotor 1 and the reflector. The reflector runs the current backwards through a different circuit through the rotors in reverse order and out through the plugboard to the output, in which the encrypted letter is indicated by the glowing bulb. Different parts of Enigma can be broken down as shown in Fig. 3

One property of enigma is that it is it's own inverse, for example: if you input the letter 'C' and the bulb lights up on 'F', to decrypt it, you can input 'F' in the same configuration to get back 'C', we will prove this in the next section entirely mathematically, so that when we later introduce methods to crack Enigma, the notations should be easier to understand. The rotors work in a relatively simple principle, it maps a given character to another character just like a simple substitution cipher, but what makes this complicated is that the left most rotor, Rotor 3 changes its position every time a letter is clicked, when Rotor 3 completes a cycle, it kicks the next rotor which is Rotor 2 to advance one position.

Likewise when Rotor 2 completes a cycle it kicks rotor next to it which is Rotor 1. This is also called turnover in some books and is done by using a mechanical mechanism called ratchet and pawl. Because of this you have to pick a letter to indicate when one cycle is finished and often these letters are different for each of the rotors that's being used. In some later versions of Enigma, there were two letters in which the turnover took place. To fully understand how the rotor advances look at Fig. 4 which shows a simplified mapping without every connection before its advanced, the next figure Fig. 5 shows exactly how those simplified connection got moved around after the turnover.

The reflector stays fixed, it does not change throughout the whole process. It redirects current from one circuit to another in reverse. However the rotors will not advance when current is being sent back. It encodes every character to its pair, a simplified version of a typical reflector is shown in Fig. 6. Plugboard is just another way to redirect current and scramble the letters. But just like the reflector each letter is being mapped to it's pair. So when 'A' and 'G' are paired with the plugboard, it will output 'A' when you input 'G', and 'G' when you input 'A', which then goes to the rotors and so on. There were a total of 8 rotors and 3 reflectors that you can select to place in the position of rotors 3, 2 and 1 and the reflector. The plugboard can be hooked up to a maximum of 10 pairs of characters. If we try to measure the total number of possibly states for a given Enigma machine so that each number represents a single Mono-alphabetic substitution cipher using these factors we get

$$\frac{8!}{5!} \times \frac{26!}{6! \times 10! \times 2^{10}} \times 26^3 \approx 8.9 \times 10^{20} \tag{1}$$

In which the first number is what we get if we choose 3 rotors from a total of 8, the equation where we choose r number of things from a total of n can be given by $\frac{n!}{(n-r)!}$ without repetition and where order matters. [4] The next factor is the number of plugboard positions given that we choose total pairs of 10, 20 letters from a total of 26 and since the order doesn't matter we divide it again my 2 to the power of pairs, and since the order of pairs does not matter we divide it by 10!. The next number shows the total number of positions each of the 3 rotors can have. As you can see this is an astronomically large number and clearly is not susceptible to simple brute-force attacks. The number of rotors and the plugboard pairs used in this calculation was after the introduction of the improved enigma by 1939. [5] And we ignored the choice of reflectors because in most cases they used the default one.

B. Enigma in practice

In this section we are going to encrypt a given letter and explain thoroughly how it actually works. Note that we are using the actual rotors and reflectors used in the German Navy in 1930 [6]. Where each R1 uses I, R2 uses II, and R3 uses III rotors respectively, all in their initial position 'A', the reflector used is the UKW-B. To simplify explanations we will use the same notation we used before to indicate the substitution that's taking place. We use ***M('OPKLNMUIHJVBXCZASDFGTYERQW')*** to indicate that mapping M substitutes 'A' to 'O', 'B' to 'P', 'C' to 'K' and so on. The initial mappings of each of the components are as follows.

P('NBHDEFXCUJKLMAOPQRSTIVWGYZ')
R3('BDFHJLCPRTXVZNYEIWGAKMUSQO')
R2('AJDKSIRUXBLHWTMCQGZNPYFVOE')
R1('EKMFLGDQVZNTOWYHXUSPAIBRCJ')
RE('YRUHQSLDPXNGOKMIEBFZCWVJAT')

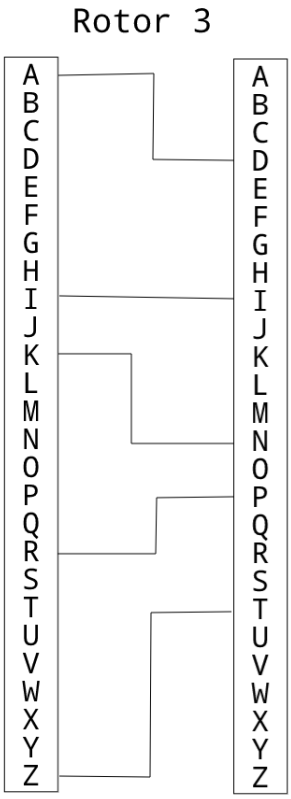


Fig. 4. Rotors before advance

Here, P indicates the mapping for the plugboard, R3 indicates the mapping for Rotor 3, R2 for Rotor 2, R1 for Rotor 3, and RE for the reflector mapping. When we press the character 'N', Rotor 3 advances one character, in the notation R3 now becomes ***R3('CEGIKBOQSWUYMXDHVFZJLTRPNA')*** which has the effect of advancing the rotor by 1 tooth forward and having the same layout of internal circuitry which links to the same position as before. After the key press, the plugboard turns 'N' into 'A', which passes through Rotor 3 to become 'C', which passes through Rotor 2 to become 'D', which becomes 'F', which then enters the reflector to become 'S', which pass through Rotor 1 to be 'S' again, which then becomes 'E', and then Rotor 3 turns 'E' into 'B', because the plugboard did not have a pair for 'B', it'll output back 'B'. At the end 'N' becomes 'B', if you input 'B' back into the machine with the same state it encrypted, you'll get back 'N'. The following diagram illustrates the complete picture

$$\text{N} \xrightarrow{\text{Plugboard}} \text{A} \xrightarrow{\text{Rotor 3}} \text{C} \xrightarrow{\text{Rotor 2}} \text{D} \xrightarrow{\text{Rotor 1}} \text{F} \xrightarrow{\text{Reflector}} \text{S}$$
$$\xrightarrow{\text{Rotor 1}} \text{S} \xrightarrow{\text{Rotor 2}} \text{E} \xrightarrow{\text{Rotor 3}} \text{B} \xrightarrow{\text{Plugboard}} \text{B}$$

The following snippet shows code taken from [7], this is an abstraction over many of the parts of Enigma we discussed earlier, key functions include advance, map and connect. The code is written in JavaScript and uses method chaining to simulate the internal wiring's in code.

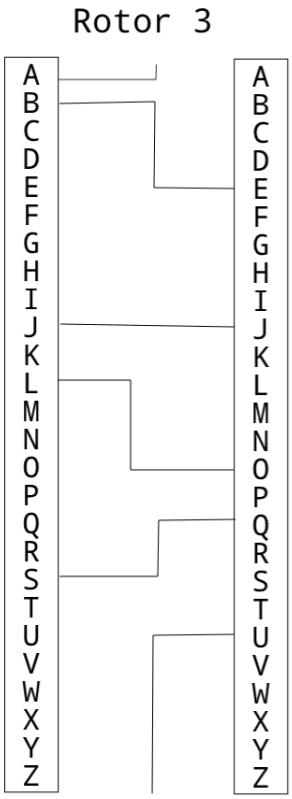


Fig. 5. Rotors after advance

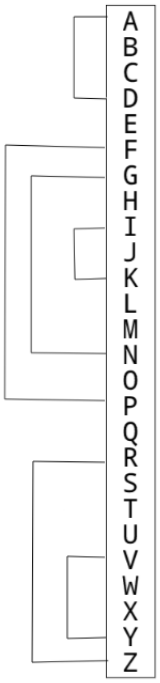


Fig. 6. Reflector

```
class CharMap {
    constructor(charmap) {
        if (charmap === undefined)
            this.charmap = "
    ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        else
            this.charmap = charmap;
        this.ptr = null; // holds mapped character
    }

    advance() { // used to advance the CharMap
    associated with a rotor
        let temp = '';
        for (let i = 0; i < this.charmap.length; i
    ++) {
            let newC = ((toCharI(this.charmap[i]) +
    25) % 26);
            temp += String.fromCharCode(65 + newC);
        }
        temp = temp.slice(1) + temp[0];
        this.charmap = temp;
    }

    switcheroo(a, b) { // switches the characters a,
     b of the map.
        let bI = this.charmap.indexOf(b);
        this.charmap = this.charmap.replaceAt(this.
    charmap.indexOf(a), b);
        this.charmap = this.charmap.replaceAt(bI, a)
    ;
    }

    map(char) { // chainable function, 'char' can be
     a simple character or another instance of
    charmap
        if (typeof char === "object" && char.
    constructor.name === "CharMap")
            this.ptr = this.charmap[toCharI(char.ptr
    )];
        else
            this.ptr = char;
        return this;
    }

    connect(char) {
        // Instead of mapping this "connects" the
    character to another, used after reflection
        this.ptr = String.fromCharCode(65 + this.
    charmap.indexOf(char.ptr));
        return this;
    }
}
```

Listing 1. Enigma example

Proof of Enigma's Inverse relationship: This sections shows the mathematical proof of the fact that enigma is its own inverse. Let the functions P, R1, R2, R3, R, $R3^{-1}$, $R2^{-1}$, $R1^{-1}$, P^{-1} denote the plugboard, Rotor 3, Rotor 2, Rotor 1, Reflector, Rotor 3 after reflection, Rotor 2 after reflection, Rotor 1 after reflection and plugboard after reflection of the Enigma respectively.

Because the reflector transposes A to B in pair the inverse is the function itself, i.e. we are given

$$R(A) = B \; and \; R(B) = A,$$

$$\implies R(R(B)) = B$$

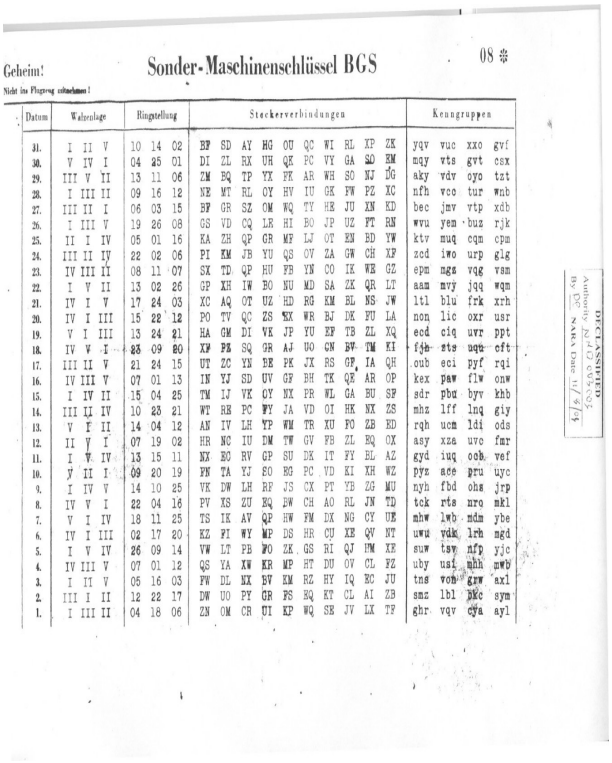
$$\implies R = R^{-1}(2)$$

Each transposition RN is a one-to-one function. Also note that the entire encryption process can be shown as a chain of functions like so,
$P^{-1} \circ R1^{-1} \circ R2^{-1} \circ R3^{-1} \circ R \circ R3 \circ R2 \circ R1 \circ P(\alpha) = \beta$
$P^{-1} \circ R1^{-1} \circ R2^{-1} \circ R3^{-1} \circ R \circ R3 \circ R2 \circ R1 \circ P(\beta) = x$
$\implies P^{-1} \circ R1^{-1} \circ R2^{-1} \circ R3^{-1} \circ R \circ R3 \circ R2 \circ R1 \circ P \circ P^{-1} \circ R1^{-1} \circ R2^{-1} \circ R3^{-1} \circ R \circ R3 \circ R2 \circ R1 \circ P(\alpha) = x$
Note that for any function F with an inverse, $F^{-1} \circ F(a) = a$, thus $P^{-1} \circ P$ cancels out in the middle
$\implies P^{-1} \circ R1^{-1} \circ R2^{-1} \circ R3^{-1} \circ R \circ R3 \circ R2 \circ R1 \circ R1^{-1} \circ R2^{-1} \circ R3^{-1} \circ R \circ R3 \circ R2 \circ R1 \circ P(\alpha) = x$
You can repeat this until you get to R, i.e. $P^{-1} \circ R1^{-1} \circ R2^{-1} \circ R3^{-1} \circ R \circ R \circ R3 \circ R2 \circ R1 \circ P(\alpha) = x$ Remember that in equation (2) we proved $R = R^{-1}$ thus we can cancel out the two R's in the middle and the following inverse functions to get $\alpha = x$, this proves that the inverse to decrypt enigma is itself.

C. Military use of Enigma

German Military used Enigma all throughout the war in all of army, naval and air force branches. There were slight differences in the model, but in principle they were all just Enigma machines. They also used different procedures to encrypt their messages. These procedures changed throughout the course of the war, we'll be focusing on procedure called Spruchschlüssel, German for message key, which was used before May 1940. This is an important procedure because they also used variations on this procedure later on. To encrypt a message in this procedure you first need the key sheet which contained the information on the order of wheels, the initial positions of the rotor encoded as numbers on the alphabet, and the plugboard settings. These key sheets contained the base settings for an entire month, each row represents the setting for the day of the month. Fig. 7 shows a genuine key sheet that was used. Because encrypting every message with the same setting for a session could leak information, the operator picked a random 3 letter key say IYU then using the ground setting encrypt it twice so IYUIYU could become KJOQGI. He then moves the rotor position to IYU and encrypts the message, to send the message you prepend KJOQGI to the encrypted message and transmit over radio signal. Decryption procedure is simple you first put the machine in the ground setting then decrypt the first 6 letters and using the output of the letters to set the machine's rotor position, now you can decrypt the rest of the message with the setting. They encoded the rotor position twice to account for any errors that can happen during the transmission. The Polish Cipher Bureau used a flaw in this procedure to break enigma, however, German cryptologists were aware of the flaw in this procedure and adopted a better one thereafter [8].

The new procedure avoided sending the random rotor position twice. In Fig. 7 you can see a column called Kennugruppen, this is used as part of the improved procedure. At the start of every message the operator had to insert a five



Geheim! Sonder-Maschinenschlüssel BGS 08 ✻

Nicht ins Flugzeug mitnehmen!

Datum	Walzenlage	Ringstellung	Steckerverbindungen	Kenngruppen
31.	I II V	10 14 02	BF SD AY HG OU QC WI RL XP ZK	yqv vuc xxo gvf
30.	V IV I	04 25 01	DI ZL RX UH QK PC VY GA SO EM	mqy vts gvt csx
29.	III V II	13 11 06	ZM BQ TP YX PK AR WH SO NJ DG	aky vdv oyo txt
28.	I III II	09 16 12	NE MT RL OY HV IU GK FW PZ XC	nfh vco tur wnb
27.	III II I	06 03 15	BF GR SZ OM WQ TY HE JU XM KD	bec jmv vtp xdb
26.	I III V	19 26 08	GS VD CQ LR HI BO JP UZ FT RN	wvu yem buz rjk
25.	II I IV	05 01 16	KA ZH QP GR MF LJ OT EN BD YW	ktv muq cqn cpm
24.	III II IV	22 02 06	PI EM JB YU QS OV ZA GW CH XF	zcd iwo urp glg
23.	IV III II	08 11 07	SX TD QP HU FB YN CO IK WB GZ	epm nga vqg vam
22.	I V II	13 02 26	GP XH IW BO NU MD SA ZK QR LT	aam mvy jqq wqm
21.	IV I V	17 24 03	XC AQ OT UZ HD RG KM EL NS JW	ltl blu frk xrh
20.	IV I III	15 22 12	PO TV QC ZS SX WR BJ DK FU LA	non lic oxr usr
19.	V I III	13 24 21	HA GM DI VK JP YU EF TB ZL XQ	ecd ciq uvr ppt
18.	IV V I	23 09 20	XF FS SQ GR AJ UO GN BV TM KI	fjs sts uqu oft
17.	III II V	21 24 15	UT ZC YN BE PK JX RS GF IA QH	oub eci pyf rqi
16.	IV III V	07 01 13	IN YJ SD UV GF BH TK QE AR OP	kex paw flw onw
15.	I IV II	15 04 25	TM LJ VK OY NX FR WL GA BU SF	sdr pbu byv khb
14.	III II IV	10 23 21	WT RH PC WY JA VD OI HK NX ZS	mhz lff lnq giy
13.	V I II	14 04 12	AN IV LH YP WM TR XU FO ZB ED	rqh ucm ldi ods
12.	II V I	07 19 02	HR NC IU DM TW GV FB ZL BQ OX	asy xza uvo fmr
11.	I V IV	13 15 11	MX EO RV GP SU DK IT FY HL AZ	gyd iuq ook vef
10.	V II I	09 20 19	FN TA YJ SO HG PC VD KI XH WZ	pyz ace pru uyc
9.	I IV V	14 10 25	VK DW LH RF JS CX PT YB ZG MU	nyh fbd oha jrp
8.	IV V I	22 04 16	PV XS ZU EQ DW CH AO RL JN TD	tck rfs nro mkl
7.	V I IV	18 11 25	TS IK AV QP HW FM DX NG CY UE	mhw lwb mdm ybe
6.	IV I III	02 17 20	KZ FI WY MP DS HR CU XB QV NT	uwu vdk lrh mgd
5.	I V IV	26 09 14	VW LT PB EO ZK GS RI QJ HM XF	suw tsy nfp yjc
4.	IV III V	07 01 12	QS YA XW KR MP HT DU OV CL FZ	uby usf nhh mwb
3.	I II V	05 16 03	FW DL NX BV EM RZ HY IQ BC JU	tns vob grw axl
2.	III I II	12 22 17	DW UO PY QR FS EQ KT CL AI ZB	smz lbl pkc sym
1.	I III II	04 18 06	ZN OM CR UI KP WQ SE JV LX TF	ghr vqv cya ayl

Fig. 7. Genuine Special Machine Key BGS used in WWII [8]

letter identifier called Buchstabenkenngruppe, which consists of two randomly picked letters of the operators choosing followed by one of the four three letters in the column. The Buchstabenkenngruppe identifier is sent without encrypting. As an example the operator can send AVNON at the start of the message which points to the setting for day 20th. Before sending the message the operator chose a random ground setting from the sheet say IOP and a random message key say UTQ. He set the rotors to the position of IOP and encrypted the message key, lets say it outputs FDM. Now he can actually encrypt the message using the position of UTQ. He transmits the Buchstabenkenngruppe corresponding to the ground setting along with the encoded FDM and the message. When another operator receives the transmission he first finds the corresponding ground setting from Buchstabenkenngruppe then using it to decrypt the random message key, which you can use to decrypt the message entirely. Whenever the message needed to be divided, the operator chose a new ground position and message key.

IV. Cracking of Enigma

We will attempt to crack the first Spruchschlüssel procedure, the double encoding of the random setting in the message was a security flaw that was first exploited by Marian Adam Rejewski working for the Polish Cipher Bureau. [5] He noticed that because the same letter is encrypted twice you can make some deductions about the internal wiring.

Suppose that the operator picked AJP for the random key message and after the double encoding of AJPAJP this became

QFKGTR which is sent over the radio. This means that at some initial ground setting each of these letters double encoded to QFKGTR, we can use the notation $E_1, E_2, E_3, E_4, E_5, E_6$ to represent each of the mono alphabetic substitution, then $E_1(A) = Q,\ E_2(J) = F,\ E_3(P) = K,\ E_4(A) = G,\ E_5(J) = T,\ E_6(P) = R$ but using the inverse relationship of Enigma we proved above, Rejewski made the breakthrough observation, since $E_1(Q) = A\ and\ E_4(A) = G$, we can deduce $E_4(E_1(Q)) = G$, using the same reasoning you can show $E_2(F) = J\ and\ E_5(J) = T \implies E_5(E_2(F)) = J$ and $E_3(K) = P\ and\ E_6(P) = R \implies E_6(E_3(K)) = R$. Throughout the day there will be a number of messages that will be encrypted using different message keys, we can construct a table for each of the permutation pairs shown above. Suppose that we gathered the information shown in Fig. 8 from the first six characters of each of the intercepted transmission

OYAGSI	KRCDLY	UXSODU	BJLVAG
XSONGQ	LDDQXF	ZBYYMW	AGHHFX
SOUIQO	RIRCUS	PUIJIR	CXBFDA
WKWBNM	JGFUFZ	THIKTR	VWZPCC
YXGZDH	DXKRDE	ILXSRD	VATPKT
MMDMPF	FYNLSN	VQKPEE	VUQPIP
QBJTML	VZEPBB	ETNEON	GMNWPN
NBPXMV	DPHRHX	VBNPMN	NCMXWJ
INVSYK	VFEPVB	VVSPJU	HBDAMF
DETRZT			

Fig. 8. Sample Interception of Enigma

Using the method specified earlier we can construct a table for each of the pairs of permutation like so shown in figures from Fig. 9 - 11. You can observe some interesting properties here in Fig. 9 and Fig. 11, $E_4(E_1(E)) = E$ and $E_4(E_1(M)) = M$ encoded to itself, this will always occur in pairs, likewise in 11 $E_6(E_3(N)) = N$ and $E_6(E_3(T)) = T$. If by any case you don't have enough unique message keys, you'll have to make use of such properties to construct the table. Within each of these tables, Rejewski observed cycles where characters are encoded. In the table of $E_4 \circ E_1$, if we start with A, we get H, which when inputted into the state gives back A, now we can pick another letter say B, which becomes V; continue this process until you exhaust all the characters in the alphabet, the total sum of all the cycle lengths should always be 26. We can create the following cycle for $E_4 \circ E_1$
A→H→A
I→S→I
N→X→N
Y→Z→Y
E→E
M→M
B→V→P→J→U→O→G→W→B

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

Fig. 9. Permutation for $E_4 \circ E_1$

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

Fig. 10. Permutation for $E_5 \circ E_2$

C→F→L→Q→T→K→D→R→C
If we take the length of each of the individual cycles for this whole permutation we can denote it with (2-2-2-2-1-1-8-8) Likewise do this for each of the pairs of permutation to get (2-2-2-2-1-1-8-8, 2-2-2-2-9-9, 12-12-1-1). This cycle is analogous to a signature for each of the starting positions of the rotor, although they are not unique. To show this we need to prove the plugboard settings are independent to this signature. We can denote the entire permutation of the internal components without the plugboard settings using the notation ρ_n thus, we can denote the first pair with $\rho_4 \circ \rho_1$ because the Enigma is one to one we can say $\rho_4 \circ \rho_1(\alpha) = \beta$. Now if we include the plugboard back into the equation this becomes $P^{-1} \circ \rho_4 \circ P$ and $P^{-1} \circ \rho_1 \circ P$, by composing the permutation we get $P^{-1} \circ \rho_4 \circ P \circ P^{-1} \circ \rho_1 \circ P$, using the inverse function relationship we know this is equal to $P^{-1} \circ \rho_4 \circ \rho_1 \circ P$. This is the key result, because the plugboard permutation is immutable throughout the process, we know it will always substitute a unique β for every α, thus preserving the cycle even though the characters could be different for every possible plugboard setting.

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

Fig. 11. Permutation for $E_6 \circ E_3$

You can make another catalog for all the possible rotor positions and initial starting positions and store each of their signatures. Luckily we can ignore all the plugboard settings because of the relationship above. Assuming there are 3 rotors in every possible permutation and starting positions we know there is a total of $3! \times 26^3 = 105456$ settings to index the signature. This number becomes 1054560 if we choose 3 rotors from a total of 5, which is still manageable. All this worked under the assumption that the second and third rotors did not advance, which is far less likely to happen using the method shown below. To construct the cyclic signature for all three permutation pairs you first set the Enigma machine to the appropriate setting of initial rotor position and rotor order. Let's say we choose the rotor order of III-I-II and initial settings B-M-C, starting from the initial position encrypt a letter 6 times to create a row for each of the characters in the alphabet as shown in Table I.

Like before you can identify cycles from this table and the pairs of permutation. In this example you'll get the following cycles

$R_4 \circ R_1 =$ (NOHXYJIBGLWCR) (APFKVMSUZTEQD)

$R_5 \circ R_2 =$ (HINOXLFEPC) (AGQVTYRJZB) (DWU) (KMS)

$R_6 \circ R_3 =$ (KTZQJGYXHU) (AWPEBFDRVL) (CS) (MO) (I) (N)

Here we use parenthesis to group each of the cycles. Thus we can identify the signature associated with the setting III-I-II B-M-C as (13-13, 10-10-2-2, 10-10-2-2-1-1). We sorted the cycle length in descending order for consistency however, this could very well work in ascending order. Because the signatures are different we know this is not the setting the operator used. Suppose that we found the match for the signature in our catalog to be I-II-III G-J-C. And the cycle we got without any plugboard settings is

$R_4 \circ R_1 =$ (BVPJUOGW) (CFLQTKDR) (AH) (IS) (NX) (YZ) (E) (M)

$R_5 \circ R_2 =$ (AKNYSGFVJ) (BMPHTOQEZ) (CW) (DX) (IU) (LR)

$R_6 \circ R_3 =$ (AIRSUOQPVKEB) (CYWMJLGHXDFZ) (N) (T)

TABLE I
PERMUTATION TABLE FOR III-I-II B-M-C

Letter	R_1	R_2	R_3	R_4	R_5	R_6
A	N	H	K	O	I	T
B	S	I	Y	U	N	X
C	F	G	M	K	Q	O
D	O	S	J	H	K	G
E	X	V	X	Y	T	H
F	C	T	G	R	Y	Y
G	M	C	F	S	H	D
H	Q	A	P	D	G	E
I	U	B	N	Z	A	N
J	Z	O	D	T	X	R
K	W	U	A	C	D	W
L	V	Y	T	M	R	Z
M	G	W	C	L	U	S
N	A	Z	I	P	B	I
O	D	J	S	A	Z	C
P	R	Q	H	N	V	U
Q	H	P	R	X	C	V
R	P	X	Q	F	L	J
S	B	D	O	G	W	M
T	Y	F	L	J	E	A
U	I	K	W	B	M	P
V	L	E	Z	W	P	Q
W	K	M	U	V	S	K
X	E	R	E	Q	J	B
Y	T	L	B	E	F	F
Z	J	N	V	I	O	L

From figures Fig. 9 - 11 we know the interception of the cycle is the following

$R_4 \circ R_1 =$ (LGWBVZJU) (AHOQTKMR) (CF) (IS) (NX) (PY) (D) (E)

$R_5 \circ R_2 =$ (BDZFTLQEP) (CKNYSGHVJ) (RO) (AW) (IU) (MX)

$R_6 \circ R_3 =$ (CIRSULQZVKEB) (AYWDJOGFXMHP) (N) (T)

We just need to identify the plugboard settings to fully decrypt the intercepted transmission. We can do so by going over each of the pairs of permutation and matching the characters inside a cycle. For example, in the permutation $R_4 \circ R_1$ we can first try to match the cycle (BVPJUOGW) with (LGWBVZJU), because (BVPJUOGW) identifies a cycle we can rotate around characters until we see a strong similarity, by rotating three times to the right we see that (OGWBVPJU) has similar characters in the corresponding places to (LGWBVZJU). We can thus deduce that the plugboard connects O to L and P to Z, Note that at this time there were only a maximum of 6 pairs used in the plugboard, so we can ignore any other rotor order or initial positions if the least difference between the cycle of interception and our guess permutation is greater than 6. Sometimes when you have multiple cycles with the same length they can be switched to make the deductions If we choose the permutation $R_5 \circ R_2 =$, We will see that (AKNYSGFVJ) does not align with (BDZFTLQEP), but (BMPHTOQEZ) does, now we know M goes to D and F goes to H. Repeating this process until we reach a maximum of 6 pairs. In this case the plugboard connects $P \rightarrow Z, O \rightarrow L, M \rightarrow D, F \rightarrow H, C \rightarrow A$. We can conclude the operator used the rotors I-II-III with the initial settings G-J-C and the

plugboard settings given above. With this you can intercept and decrypt the messages for a given day to know what exactly your enemies are planning.

V. Conclusion

There are numerous other ways to crack enigma that we have not discussed in this paper. One major method used was developed by the famous Mathematician Alan Turing, it exploited certain properties of Enigma rather than using flaws in procedures to crack. It used cribbing which is an educated guess of some part of the message to narrow down the possibilities. And with the help of Bombe machines which he developed, it showed how we can use mechanical and electrical machines to check huge numbers of possibilities with an electrical circuit. By cracking Enigma, the Allies knew every move of their enemy, without it it's possible that they would've never won the war. Vigenère cipher too can be cracked using the method of cribbing. The emulator written in JavaScript is available at [7]. This also includes various other tools to crack Enigma in the web browser. In the modern world information becomes the weapon in which we fight each other for, encryption becomes crucial when we want confidentiality of information. WWII was a turning point in our view towards using cryptography which has its Mathematical roots in Number Theory.

References

[1] B. Lennon, *Passwords: Philology, Security, Authentication*. Harvard University Press, 2018.

[2] F. W. Kasiski, *Die Geheimschriften und die Dechiffrir-Kunst [Cryptograms and the art of deciphering] (in German)*. E.S. Mittler und Sohn, Berlin Germany, 1863.

[3] [Online]. Available: https://en.wikipedia.org/wiki/Enigma_machine

[4] (2017) Combinations and permutations. [Online]. Available: https://www.mathsisfun.com/combinatorics/combinations-permutations.html

[5] J. Grime. Maths from the talk alan turing and the enigma machine. [Online]. Available: https://www.singingbanana.com/enigmaproject/maths.pdf

[6] (2019) Enigma wiring. [Online]. Available: https://www.cryptomuseum.com/crypto/enigma/wiring.htm

[7] A. Johnson. (2020) Enigmaproject. [Online]. Available: https://github.com/Anex007/EnigmaProject

[8] D. Rijmenants. Enigma message procedures. [Online]. Available: http://users.telenet.be/d.rijmenants/en/enigmaproc.htm