Lectures 4,5 - Cryptanalysis of Enigma

Lecture notes taken by Jason Friedman*

The Enigma Cipher

The first patents related to the Enigma cipher date back to 1919 (Koch's reflector). The first commercial version appeared in 1923, and after various improvements it was adopted by the German Navy in 1926, and the Army in 1928. The final "revised standard" for Wehrmacht was issued in June 1930. The Air Force adopted the machine in 1935. It was also used by the Abwehr, the the railways, and other government departments. Each department used its own variant of the machine, following successive refinements and changes in operating instructions until 1945. At least 30,000 machines were used during the war. In 1945, the wirings of the rotors from 1920s was still in use. (from [2])

Brief description of the machine

The Enigma machine (see Figures 1, 2 - these and subsequent photos are from [1]) had a 26-letter keyboard, for the entry of plaintext and ciphertext, and a panel with 26 letters illuminated by glowlamps. The components that performed the ciphering were three rotors, and a fourth reflector (initially).

Each of the three rotors was wired so that it performed a simple substitution, however, no letter was allowed to be connected to itself. The combined substitution would be a product of the individual substitutions.

The reflector also performed a simple substitution, but in pairs (ie if $A \to B$, then $B \to A$). There was also optionally a plug board (see Figure 4), which worked in a similar way to the reflector, but switched only some of the letters.

The machine would operate by an electric current traveling through the switchboard, also called the commutator, plugboard, or stecker (where it may be switched with another letter), through the three rotors (where three substitutions were performed), through the reflector (a substitution of involution type), back through the three rotors in the reverse direction (another three substitutions) and finally through the switchboard again.

*jason@wisdom.weizmann.ac.il

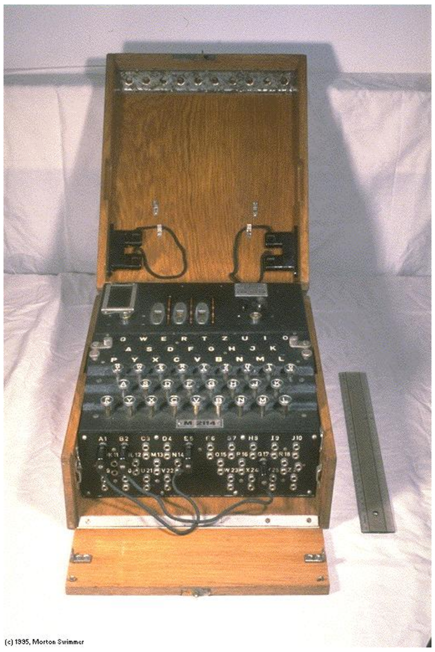


Figure 1: The Enigma machine, with the keyboard, glowlamps and plugboard visible

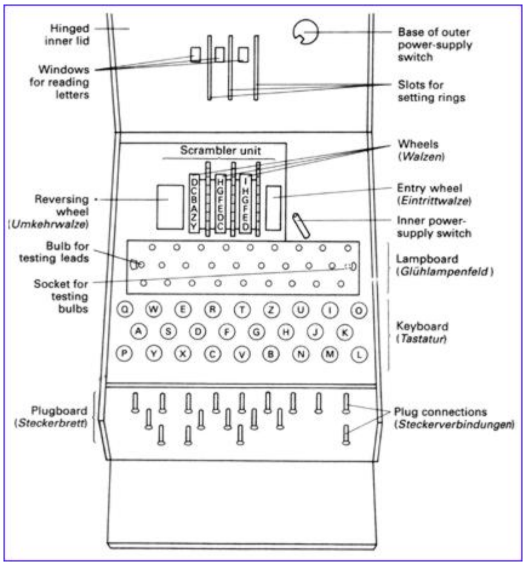


Figure 2: Layout of the Enigma machine

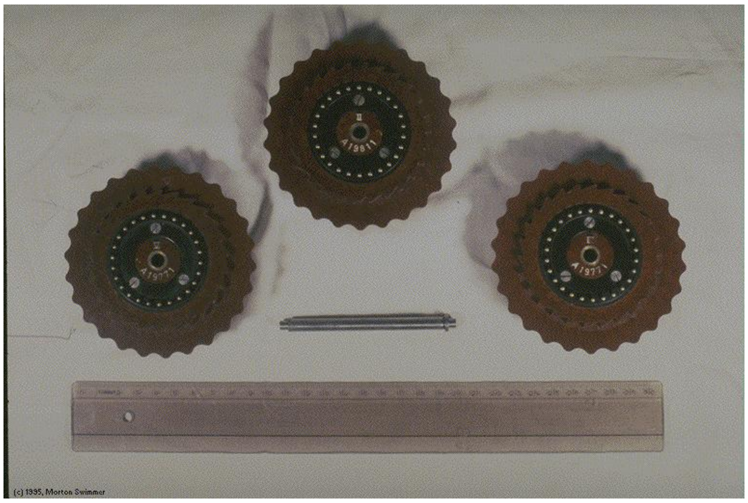


Figure 3: The rotors - each rotor had a number on it - the selection and location of the rotors was part of the key

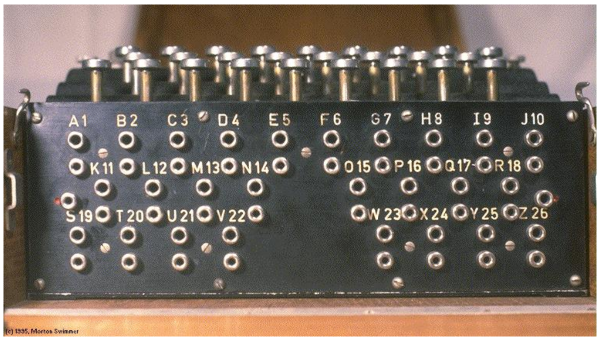


Figure 4: The plugboard - note that each letter had two holes to allow it to be interchanged with another letter

Between the coding of letters, the positions of the rotors would change in a systematic manner. Each rotor had two rings: an alphabetic (or sometimes number) ring and a wired ring, which could be set in 26 relative positions against each other. This relative position was called a ring setting. As seen in Figure 5, the number/letter ring had attached to it a latch which caused the next ring to turn. So although the second ring would rotate every 26 turns of the the first ring, the first time that it rotates would be dictated by the ring setting, and similarly for the third ring. As the position of the rotors changed, so would the product of the substitutions. Before encoding each message, a message key

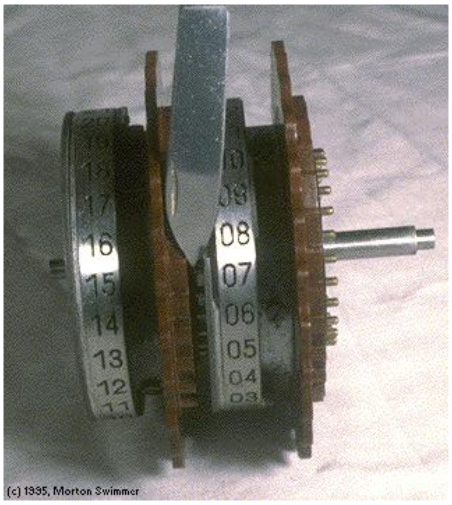


Figure 5: The ring setting - the ring could be set to a number before being attached, and the latch as shown would determine when the next rotor would turn.

would be set, by selecting three numbers (or letters), as shown in Figure 6. This would determine the starting position of the rotors for the rest of the message (after the message key itself was transmitted twice in the base position).

The importance of the Ring setting

Without a ring setting, the Enigma machine would cycle only through about 17,000 different permutations. The ring setting introduced 26^2 variants to the Enigma stepping function. Thus two Enigma machines with the same plug, rotor order and message key setting when used to encrypt identical texts will produce different ciphertexts if their ring settings are different.

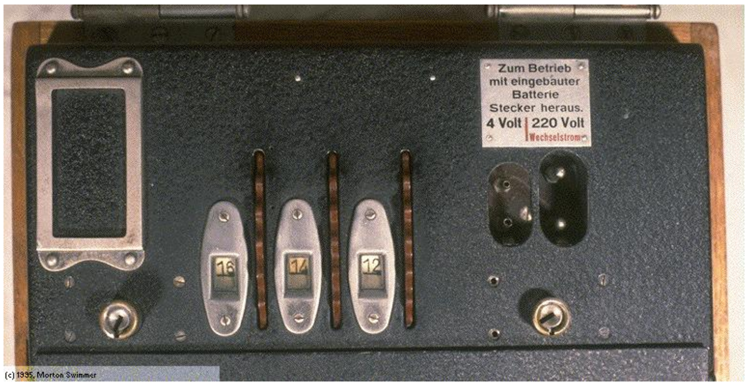


Figure 6: The message key setting - three numbers would be selected corresponding to the message key - these would be transmitted twice in the base settings at the beginning of the message

Modification through the years

It was the introduction of the plug-board (1930) that made German cryptanalysts believe that the Enigma machine was unbreakable. The number of plugs in use varied from four to all 13 depending on the organization using the machine and the year. Abwer used plug-less machines. The Army used six plugs until November 1937, after which the number of plugs varied from five to eight each day. In November 1939 the number of plugs was increased to 10. This last change obviated certain Polish methods of cryptanalysis.

It is easy to get lost in the various modifications and changes of operational procedure of the Enigma cipher. We will thus consider only the Army variant. The standard Wehrmacht practice was to use a day key (Grundstellung - ground setting) known to everyone on the network in order to encrypt message-key pairs. In September 1938 this idea of a common ground setting was abandoned. Each operator now chose his own ground setting at random, which was sent in clear together with the twice-enciphered message key. In May 1940 the double encipherment of the message key was abandoned.

1. Plugboard introduced (June, 1930)
2. Change in a key-setting method (15.1.1938)
3. Rotors IV and V added (15.12.1938)
4. Sharp increase in the number of plug connectors (1.1.1939)
5. End of a key-pair encipherment (1.5.1940)

Method of Batons for plugless Enigma

During the Spanish Civil War in 1938-1939 (which was an international conflict), all sides used plugless Enigma machines for encrypting traffic - the British, the Germans, Italians and Spanish republicans. A simple method was discovered to learn the identity and starting position of the first rotor. (It was for this reason that the Germans added the plugboard). The method was known as the Method of 'Batons' by the French, and 'cliques on the rods' by the British. This method assumes knowing the following initial information:

1. No plugs are in use
2. The wiring of the rotors is known
3. Assume that only the first rotor is moving
4. Probable words are known

Consider the equation of the machine when only the first rotor is moving. Let Z be the combined effects of rotors 2 and 3, the reflector, and rotors 3 and 2 in reverse. As we are assuming that only the first rotor is moving then Z will be a fixed substitution. Let N be the substitution of the first rotor (and subsequently N^{-1} its inverse) - we need to make a guess as to which of the rotors is the first. So for the first letter, the combined permutation will be (applying the permutations from left to right):

$$NZN^{-1}$$

We can write this as an equation, it becomes (where p is the plaintext, c is the ciphertext):

$$N^{-1}ZN(p) = c$$

This can be rewritten by applying N to both sides:

$$ZN(p) = N(c)$$

So now applying the first rotor permutation to both the plaintext guess (using a crib) and the ciphertext, we will have one of the letter substitutions in Z. Notice that Z is an involution[1] and thus we learn two substitutions $(\alpha \to \beta)$ and $(\beta \to \alpha)$ at once. This property allows us to use much shorter cribs in the "Batons" attack. We can repeat this procedure for the entire crib word, however we need to take care to change the N permutation each letter to take into account that the rotor is turning. The correct equation for the i-th step will be $Z \cdot P^{-i}NP^{i}(p) = P^{i}NP^{-i}(c)$, where P denotes a cyclic shift of the alphabet $(a \to b \to c \to d \ldots)$.

[1] Permutation T is called an *involution* if $T^2 = I$

If there are contradictions in the substitutions found in Z, then our choice of the indicator setting for rotor N was incorrect. Thus we need to try the 26 different starting rotor positions — this can be easily performed manually in a short amount of time using strips of paper with the columns of the relevant rotor table written twice on each strip. For example, using the crib 'RECONNAISSANCE', using a known first rotor permutations we get:

starting rotor position (indicator setting)	ZN(P) above N(C)
A	`U F J R Q N X A W B D R M H` `A W C Y R G U Q I N N D S Q`
B	`R B D O N J J P B C I X U E` `H S H N U A B M M O J K X C`
C	`N N Z K J T N L C U C D J A` `Z P I X H W T J P W G L Y G`
...	...
Y	`J G M G F U H R W C N S E W` `U Z C Z B J O T A M Q E S A`

As can be seen, only with setting Y do we get involutions without any contradictions. So this would confirm for us that the first rotor is the correct one being used, and the first indicator setting is indeed Y. If during the encryption of the crib, the second rotor did move, then the isomorphism splits into two pieces, which could also be detected fairly reliably.
The Germans used this method for reading Swiss diplomatic correspondence. They used the above method, but prepared tables to check message for probable words, using about 100 words of 4-10 letters. Once the fast rotor was identified, they used prepared catalogs to give the order and position of the remaining rotors.
This method fails once a plugboard is introduced, because the unknown plugboard permutation destroys the test encipherments of the plain and ciphertext pairs. French, British and US also knew of this attack method and experimented with faster stepping middle rotors (every 3-4 letters), placing the fastest rotor close to the reflector (thus the rest plays the role of a variable plugboard, Friedman) and adding two stators (fixed rings with 26^2 possible settings, British).

Polish Mathematical attacks

The Poles learned in 1928 that the Germans had begun using a new cipher known as *Enigma G.* In 1931, an employee of the German Ministry of Defense, code-named 'Asche' offered to produce documents and key listings to the French. Between 1931 and 1939, these documents were passed through the French to the British and the Poles. In 1932, the Polish Cipher Bureau in Warsaw gave the task of analyzing the Enigma traffic to Marian Rejewski.

He was in possession of the following information:

1. The same key of the day was used by all sites
2. The Poles had noticed that at the beginning of each message, there was a relationship between the first and fourth letters, second and fifth letters and third and sixth letters on a given day. If for example, E was the first letter, then G was always the fourth letter. This led them to deduce that a three letter key was twice repeated at the start of each message. This double encipherment was obviously meant for error correction purposes.
3. The order of the rings is changed every three months

At this point in time, they still did not know:

1. The wirings of the rotors or the reflector
2. The ordering of the rotors (although they did know they only changed once every three months)
3. The wiring of the plugboard - although there were assumed to be no more than 6 plugs used (which was true until 1939)
4. the initial ring settings
5. The key of the day
6. The message key (also called the indicator settings)

From these unknowns, we can calculate the key space:

- Each rotor has 26! possible combinations, so combined they have $(26!)^3$ combinations
- For the reflector, there are 26 possibilities for the first switch, 24 for the second, etc, so there are $\frac{\binom{26}{2}\binom{24}{2}\cdots\binom{2}{2}}{13!} \approx 7.9 \cdot 10^{12}$
- There are 6 possible orderings of the three rotors
- For the plugboard, the first plug has 26 choices, the next 24, etc, so for 6 plugs, there are $\frac{\binom{26}{2}\binom{24}{2}\cdots\binom{16}{2}}{6!} \approx 10^{11}$. The effect of different numbers of plugs is shown in Table 1. Note that the maximum number of permutations is achieved by using 11 rather than all 13 plugs:

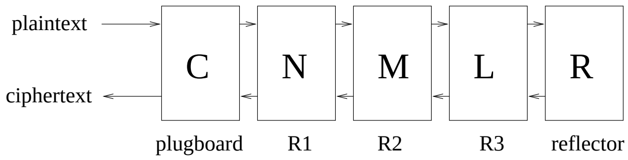
Number of plugs in use	Permutations	% letters influenced
0	0	0
1	325	15
2	44850	28
3	$3.5 \cdot 10^6$	41
4	$1.6 \cdot 10^8$	52
5	$5.0 \cdot 10^9$	62
6	$1.0 \cdot 10^{11}$	71
7	$1.3 \cdot 10^{12}$	79
8	$1.1 \cdot 10^{13}$	85
9	$5.4 \cdot 10^{13}$	91
10	$1.5 \cdot 10^{14}$	95
11	$2.1 \cdot 10^{14}$	98
12	$1.0 \cdot 10^{14}$	99
13	$7.9 \cdot 10^{12}$	100

Table 1: The effect of the plugboard.

- The ring settings of the rotors had 26 possible positions for each of the first two rotors, so 26^2 possibilities in total (the ring setting on the third rotor did not change the stepping function).
- The three letter message key also has 26^3 possibilities.

So in total the key size was: $(26!)^3 \cdot 7.9 \cdot 10^{12} \cdot 6 \cdot 10^{11} \cdot 26^2 \cdot 26^3 \approx 2^{369}$.
This key space is clearly much too big to solve by exhaustive search, even with today's computers, let alone in the 1930s and 40s.[2]

Rejewski's method exploited the double repetition of the encipherment of the message key at the start of the message. First he considered the equation of the machine. The machine can be considered with the following schematic diagram:



So the permutation of the machine will be represented by the product[3]:
$CNMLRL^{-1}M^{-1}N^{-1}C^{-1}$.

[2]For example, the number of atoms in the universe is estimated to be "only" 2^{265}.
[3]We preserve the notations of the original paper, where the product of several transformations is written from left to right

However, each time a key is pressed, the N rotor rotates one position. (We are assuming that in these first 6 letters the other rotor positions are constant, which was a reasonable assumption to make). This can be represented by a cyclic permutation that transforms each letter of the alphabet into the next one:

$$P = (a \rightarrow b \rightarrow c \rightarrow \dots)$$

This now allows the permutations for the first six letters in a message to be expressed (which we denote by letters from A to F):

$$\begin{aligned}
A &= CPNP^{-1}MLRL^{-1}M^{-1}PN^{-1}P^{-1}C^{-1} \\
B &= CP^2NP^{-2}MLRL^{-1}M^{-1}P^2N^{-1}P^{-2}C^{-1} \\
C &= CP^3NP^{-3}MLRL^{-1}M^{-1}P^3N^{-1}P^{-3}C^{-1} \\
D &= CP^4NP^{-4}MLRL^{-1}M^{-1}P^4N^{-1}P^{-4}C^{-1} \\
E &= CP^5NP^{-5}MLRL^{-1}M^{-1}P^5N^{-1}P^{-5}C^{-1} \\
F &= CP^6NP^{-6}MLRL^{-1}M^{-1}P^6N^{-1}P^{-6}C^{-1}
\end{aligned}$$

Note that all A,B,C,D,E,F are involutions due to reflector R. Now these equations can be combined to make a product equations. Note that there is some canceling in the middle:

$$\begin{aligned}
AD &= CPNP^{-1}MLRL^{-1}M^{-1}PN^{-1}P^3NP^{-4}MLRL^{-1}M^{-1}P^4N^{-1}P^{-4}C^{-1} \\
BE &= CP^2NP^{-2}MLRL^{-1}M^{-1}P^2N^{-1}P^3NP^{-5}MLRL^{-1}M^{-1}P^5N^{-1}P^{-5}C^{-1} \\
CF &= CP^3NP^{-3}MLRL^{-1}M^{-1}P^3N^{-1}P^3NP^{-6}MLRL^{-1}M^{-1}P^6N^{-1}P^{-6}C^{-1}
\end{aligned}$$

Note that these double products are not just formal algebraic manipulations, but are available to the cryptanalyst due to the double encipherment feature of the first 6 letters of each message. It can be seen that the repeated product $MLRL^{-1}M^{-1}$ is found in all of them. This is effectively just one substitution, as it does not change. So to simplify the procedure, this was replaced by Q, giving:

$$\begin{aligned}
AD &= CPNP^{-1}QPN^{-1}P^3NP^{-4}QP^4N^{-1}P^{-4}C^{-1} \\
BE &= CP^2NP^{-2}QP^2N^{-1}P^3NP^{-5}QP^5N^{-1}P^{-5}C^{-1} \\
CF &= CP^3NP^{-3}QP^3N^{-1}P^3NP^{-6}QP^6N^{-1}P^{-6}C^{-1}
\end{aligned}$$

We know P already (it is just the permutation function). We are able to find the products AD, BE and CF from the ciphertext. We know from the symmetry of the machine, that if the encoding of x is y, then also the encoding of y is x. So if the letter Q in position A gave a D, and a V in position D, then we know that if we took a D, encoded it at position A, we would get a Q. We also know that that this Q encoded at position D gives a V. In this way, we know the products. Although we do not know the individual transitions (yet), because this would have required knowledge of the plaintext, the products can be useful for us. In these equations, we are left with three unknowns, S, N and Q.

If we have about 80 messages in a given day, then usually all the letters of the alphabet will be found in each of the six positions at the start of the message. This will allow us to reconstruct the product transformations. For example, suppose we received the following 3 message openings:

```
DMQ VBN
VON PUY
PUC FMQ
```

From the first message, we know that `D` encoded by AD will give `V`. Similarly, we can see that `V` goes to `P`, and `P` to `F` by the second and third messages. This way, we can begin to build a cycle — in this case we will have: (... DVPF ...). With sufficient further openings of messages, we can build the entire cycles for AD, BE and CF, for example, we may receive something like this:

```
AD = (DVPFKXGZYO) (EIJMUNQLHT) (BC) (RW) (A) (S)
BE = (BLFQVEOUM) (HJPSWIZRN) (AXT) (CGY) (D) (K)
CF = (ABVIKTJGFCQNY) (DUZREHLXWPSMO)
```

It must be remembered that they are cycles — for example, the `O` in the first cycle of AD has a transition to `D`. From observation, it can be seen that each cycle size is repeated twice. This is not by chance but is a result of the following theorem:

Theorem 1 (Rejewski) *If two permutations of the same degree are involutions, then their product consists of pairs of cycles of equal length.*

In each of the individual permutations, each cycle is of size 2 (because of the nature of the Enigma machine), so this theorem should hold for Enigma.

Proof
Consider the two positions A and D. The trivial case is where both contain an identical permutation (eg `(AB)`) - this would result in two unilateral cycles (`(A)` and `(B)`). So far the theorem holds. Without loss of generality, consider the following permutations for A and D (where $2n$ is their degree, and $k \leq n$:

A	D
$(a_1 a_2)$	$(a_2 a_3)$
$(a_3 a_4)$	$(a_4 a_5)$
$\vdots$	$\vdots$
$(a_{2k-3} a_{2k-2})$	$(a_{2k-2} a_{2k-1})$
$(a_{2k-1} a_{2k})$	$(a_{2k} a_1)$

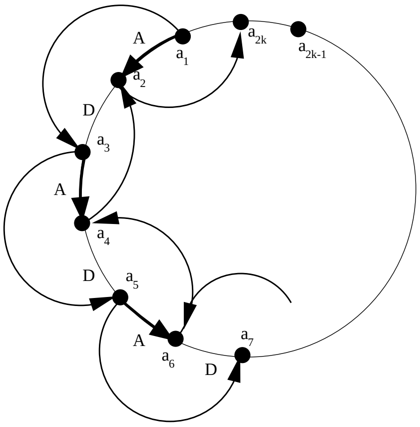


Figure 7: The relationship between letters and their permutations by A and D

It will be in this form, because at some point, a_1 must appear in D. If we now multiply A and D, we will get two cycles of length k:

$$(a_1\ a_3\ \dots\ a_{2k-3}\ a_{2k-1})\ (a_{2k}\ a_{2k-2}\ \dots\ a_4\ a_2)$$

This process can be repeated until all letters in the permutations have been used up.

This completes the proof (see also Figure 7).
■

Now we can see that:

1. Letters that were in the same swap (eg `(AB)`), end up in two different cycles of the same length in AD.
2. If two letters, in two different cycles of the same length of AD belong to the same swap `(AB)`, then the letter to the right of `A` and to the left of `B` are a swap in A. For example, cycles in AD of:

   `(AXNE) (BYTC)`

   correspond to transpositions `(XC) (NT)` and `(EY)` based on original swap `(AB)`. Similarly, `(XB)` will then be a swap in D, and by the same rule, `(NT)(ET)(AY)` will be swaps in D.

It is now possible to 'guess' the factorization of product permutations from these two cycles, as shown above. There are n possibilities for a cycle of length n. So in the the example given before,

- AD - there are 20 possible factorizations $(10 \cdot 2 \cdot 1)$
- BE - there are 27 possible factorizations $(9 \cdot 3 \cdot 1)$
- CF - there are 13 possible factorizations (13)

So we have about 7000 possible factorizations, in general there would be from several thousand to several dozen thousand possible solutions.[4] To extract the true solution at that time was infeasible. Although this theorem does not lead to the goal, it does take us much closer. Note that the number and the length of cycles in AD,BE and CF is the same for any choice of the plugboard setting. This feature is very important, and the method is said to be *plugboard invariant*. The cycles where called by the Poles the characteristic of the day.

At this point, we are stuck. However, we can suppose that German Enigma operators prefer to use as their message keys simple keyboard combinations, for example three identical letters like `JJJ`.[5] This turned out to be a reasonable assumption. A sample of enciphered message keys from 1932-33 from a day's traffic, sorted into alphabetical order can be seen below, (taken from [2]):

```
ABW BFO   ACC BIS   AKJ BTB   ARS BXA   AWB BCL   BAS QJA
BIQ QMV   CCB KIL   CKP KTG   CNZ KYC   CPB KDL   CVC KAC
DDH PKW   DMB POL   DUE PPO   EEH FQW   EHZ FNC   EOH FBW
EUR FPE   GCS CIA   GHG CNK   GJN CVD   GMX COJ   GSG CWK
HDL ZKR   HIU ZMY   HIU ZMY   HIU ZMY   HIU ZMY   HXU ZEY
IAM WJF   IHE WNO   IQR WUE   IVB WAL   IXJ WEB   JKI RTU
JKZ RTC   JZA RSX   KOG EBK   LIH AMW   LOX ABJ   LPK ADI
LQJ AUB   MBZ NFC   MLW NRQ   MLW NRQ   MLW NRQ   MLW NRQ
MSQ NWV   NWJ JOB   NTA JLX   NYM JHF   NYM JHF   DHZ XNC
PGW OCQ   PHA DNX   PXA DEX   QAP HJG   QFR HZE   QGX HCJ
QIE HMW   QMS HOA   QYZ HHC   RAA VJX   RFQ VZX   RRN VXD
RSD VWZ   RWZ VGC   SRO YXN   SUP YPG   TLB GRL   TMV GOM
TQO GUN   UGU ICY   UQT IUP   UQT IUP   UQT IUP   VIY TMH
WLL ORR   WNT OYP   WWP OGG   WZY OSH   XFF LZT   XQK LUI
XQR LUE   YAP SJG   YFV SZM   YQQ SUV   YSG SWK   YSG SWK
YSG SWK   ZMD UOZ   ZPZ UDC
```

[4] In the worst case there can be up to $(2^5 \cdot 3)^3 \approx 9 \cdot 10^5$ factorizations. But it is a very rare case.

[5] Later in 1937 new operating instructions were issued, which prohibited the use of identical letters. However, German operators still used guessable keyboard combinations like QAZ and QWE.

It is not important which letter is repeated, more that the same letter is repeated. Now as well as being able to couple A and D, we can also couple every combination of A to F because they are all the same letter encoded. This reduces the number of possible factorizations, so we are able to find the one correct solution for A to F.
To begin, the cycles are constructed from the data — for example, the data above yields the following cycles:

```
AD: (DP) (SY) (ABQHZUIWOXL) (MNJRVTGCKEF)
BE: (AJV) (HNY) (BFZSWGCIMO) (DKTLRXEQUP)
CF: (AXJBLREONDZCS) (IUYHWQVMFTPGK)
```

Taking one of the repeated message key, `YSG SWK`, we assume that this represents the same starting rotor position on all three rotors. This can then only be `D` or `P` as this is the only other cycle of length 2. However, it can't be `P`, because `S` and `W` are in different cycles in the BE permutation. So we hence have three cycle superpositions:

```
AD:  (YS) BE: (SWGCIMOBFZ) DF: (GKIUYHWQVMFTP)
     (DP)     (DPUQEXRLTK)     (DNOERLBJXASCZ)
     (PD)     (KDPUQEXRLT)     (ZDNOERLBJXASC)
```

Using this to decipher another one of the indicators, `UQT IUP` gives us `?CC ?CC`, which confirms our result, and allows us to construct a further cycle superposition:

```
AD: (UIWOXLABQHZ)
    (CGTVRJNMFEK)
    (KCFTVRJMNFE)
```

From here, the remaining cycle superpositions can be obtained, which will give us the decipherment of all the indicator settings.
It is an open problem if the set of six equations:

$$
\begin{aligned}
A &= CPNP^{-1}MLRL^{-1}M^{-1}PN^{-1}P^{-1}C^{-1} \\
B &= CP^2NP^{-2}MLRL^{-1}M^{-1}P^2N^{-1}P^{-2}C^{-1} \\
&\ldots
\end{aligned}
$$

with three unknowns C,N and Q is efficiently solvable. This set would be solvable if the cryptanalyst could obtain ciphertexts of key pairs for two different days, with different plugboard settings but the same, or almost the same rotor settings. Such a demand may seem unrealistic, since the keyspace in question is $6 \cdot 26^3 = 105,456$, however due to the birthday paradox (see next lectures), one needs to search only within not more than 500 days in order for this condition to be satisfied with high probability.

Date	1.09.1939
Rotor sequence	IV I II
Ring Setting	12 15 21
Indicator setting (day key)	X A E
Plug connections	AB HU YS CZ MI OS TD

Table 2: A sample of the daily key settings — rotor order, day indicator settings, ring settings, plugboard settings.

Such a pair of days can be recognized by having the same cycle characteristic, of AD,BE,CF. So at least in theory, the Enigma wires could be found. In practice, a simpler approach was found.

From here we still do not know the plugboard settings. However, as mentioned earlier, the Poles were given from the French certain information about the Enigma machines keys, see Table 2.

One important piece of information was the plugboard settings (C). This makes it possible to transfer the C permutations to the left hand side of the equation:

$$\begin{aligned}
C^{-1}AC &= PNP^{-1}QPN^{-1}P^{-1} \\
C^{-1}BC &= P^2NP^{-2}QP^2N^{-1}P^{-2} \\
&\dots \\
C^{-1}EC &= P^5NP^{-5}QP^5N^{-1}P^{-5} \\
C^{-1}FC &= P^6NP^{-6}QP^6N^{-1}P^{-6}
\end{aligned}$$

We now have six equations, with only 2 unknowns, N and Q. This set is already solvable, however for error correction purposes (human errors or to filter out cases when the 2nd rotor stepped within six permutations A to F) the Poles have developed an elegant solution method robust to such errors. The main idea is based on permutations isomorphism, namely if we have three permutations G,H,T reflected by equation $G = T^{-1}HT$, then these are solvable when G and H are similar, and there will be as many solutions for T as there are ways or writing G beneath H. If G or H consists entirely of swaps (i.e. G and H are involutions), then there is a very large number of transpositions — for 13 transpositions, there will be $2^{13} \cdot 13!$ ways, clearly impractical to search through. So some auxiliary transformations need to be performed to make this practically solvable.

To do this, we transform both sides of A by P on the right side, and P^{-1} on the right side, by P^2 for B, etc. To make it clearer, we also label the left sides by the letters U to Z:

$$\begin{aligned}
U &= P^{-1}C^{-1}ACP = NP^{-1}QPN^{-1} \\
V &= P^{-2}C^{-1}BCP^{2} = NP^{-2}QP^{2}N^{-1} \\
W &= P^{-3}C^{-1}BCP^{3} = NP^{-3}QP^{3}N^{-1} \\
X &= P^{-4}C^{-1}BCP^{4} = NP^{-4}QP^{4}N^{-1} \\
Y &= P^{-5}C^{-1}ECP^{5} = NP^{-5}QP^{5}N^{-1} \\
Z &= P^{-6}C^{-1}FCP^{6} = NP^{-6}QP^{6}N^{-1}
\end{aligned}$$

Next we form products of two successive expressions:

$$\begin{aligned}
UV &= NP^{-1}\,(QP^{-1}QP)\,PN^{-1} \\
VW &= NP^{-2}\,(QP^{-1}QP)\,P^{2}N^{-1} \\
WX &= NP^{-3}\,(QP^{-1}QP)\,P^{3}N^{-1} \\
XY &= NP^{-4}\,(QP^{-1}QP)\,P^{4}N^{-1} \\
YZ &= NP^{-5}\,(QP^{-1}QP)\,P^{5}N^{-1}
\end{aligned}$$

By substituting, we can eliminate the common middle term $QP^{-1}QP$:

$$\begin{aligned}
VW &= NP^{-1}N^{-1}\,(UV)\,NPN^{-1} \\
WX &= NP^{-1}N^{-1}\,(VW)\,NPN^{-1} \\
XY &= NP^{-1}N^{-1}\,(WX)\,NPN^{-1} \\
YZ &= NP^{-1}N^{-1}\,(XY)\,NPN^{-1}
\end{aligned}$$

We now have four equations, and only one unknown, N. Due to the isomorphism property, all the products VW, WX, XY and YZ must have the same structure unless an error in calculations or stepping of the rotor M has occurred. Now from the first equation, we can obtain several dozen possibilities for NPN^{-1}. The second equation will give us the same number of solutions - only one of these will be identical with high probability, and this will be the correct solution. We no longer need the last two equations. From here we can obtain N^{-1} by comparing with P (applying the isomorphism method again), after which we will obtain 26 possible solutions for N^{-1} which are essentially the same, which gives us N - the wiring of the right most rotor. As the rotor positions were changed every three months, this enabled the Poles to find the wirings of two rotors. From here it was not hard to derive the third rotor and the reflector.

Example

We will now consider an example, taking the cycles for AD, BE and CF used before:

```
AD = (DVPFKXGZYO) (EIJMUNQLHT) (BC) (RW) (A) (S)
BE = (BLFQVEOUM) (HJPSWIZRN) (AXT) (CGY) (D) (K)
CF = (ABVIKTJGFCQNY) (DUZREHLXWPSMO)
```

By using the theorem above, and together with knowledge of enciphering habits, we can obtain the permutations A through to F:

```
A = (AS)(BR)(CW)(DI)(EV)(FH)(GN)(JO)(KL)(MY)(PT)(QX)(UZ)
B = (AY)(BJ)(CT)(DK)(EI)(FN)(GX)(HL)(MP)(OW)(QR)(SU)(VZ)
C = (AX)(BL)(CM)(DG)(EI)(FO)(HV)(JU)(KR)(NP)(QS)(TZ)(WY)
D = (AS)(BW)(CR)(DJ)(EP)(FT)(GQ)(HK)(IV)(LX)(MO)(NZ)(UY)
E = (AC)(BP)(DK)(EZ)(FH)(GT)(IO)(JL)(MS)(NQ)(RV)(UW)(XY)
F = (AW)(BX)(CO)(DF)(EK)(GU)(HI)(JZ)(LV)(MQ)(NS)(PY)(RT)
```

Intelligence provided the plug settings (C) for the given day:

$$C = \mathtt{(AP)(BL)(CZ)(FH)(JK)(QU)}$$

The permutations P we know, and can easily computer its powers:

$$\begin{aligned}
P &= \mathtt{(A\ B\ C\ D\ E\ F\ G\ H\ I\ J\ K\ L\ M\ N\ O\ P\ Q\ R\ S\ T\ U\ V\ W\ X\ Y\ Z)} \\
P^2 &= \mathtt{(A\ C\ E\ G\ I\ K\ M\ O\ Q\ S\ U\ W\ Y)\ (B\ D\ F\ H\ J\ L\ N\ P\ R\ T\ V\ X\ Z)} \\
P^3 &= \mathtt{(A\ D\ G\ J\ M\ P\ S\ V\ Y\ B\ E\ H\ K\ N\ Q\ T\ W\ Z\ C\ F\ I\ L\ O\ R\ U\ X)} \\
P^4 &= \mathtt{(A\ E\ I\ M\ Q\ U\ Y\ C\ G\ K\ O\ S\ W)\ (B\ F\ J\ N\ R\ V\ Z\ D\ H\ L\ P\ T\ X)} \\
& \ldots
\end{aligned}$$

Now using the methods described before, find U, V, W and X (we have no need for Y and Z):

```
U = (AX)(BU)(CK)(DR)(EJ)(FW)(GI)(LP)(MS)(NZ)(OH)(QT)(VY)
V = (AR)(BV)(CO)(DH)(FL)(GK)(IZ)(JP)(MN)(QY)(SU)(TW)(XE)
W = (AS)(BZ)(CP)(DQ)(EO)(FW)(GJ)(HL)(IY)(KR)(MU)(NT)(VX)
X = (AP)(BF)(CU)(DV)(EI)(GR)(HO)(JN)(JY)(LX)(MZ)(QS)(TW)
```

and then their products:

```
UV = (A E P F T Y B S N I K O D) (R H C G Z M U V Q W L J X)
VW = (A K J C E V Z Y D L W N U) (S M T F H Q I B X O P Q R)
WX = (A Q V L O I K G N W B M C) (P U Z F T J R Y E H X D S)
```

As expected, the products have the same configuration of cycles. Now we write product VW below UV in every possible, and similarly WX below UV. Only one way will give the desired result:

```
UV = (A E P F T Y B S N I K O D) (R H C G Z M U V Q W L J X)
VW = (Y D L W N U A K J C E V Z) (I B X O P Q R S M T F H Q)
VW = (Y D L W N U A K J C E V Z) (I B X O P Q R S M T F H Q)
WX = (U Z F T J R Y E H X D S P) (C A Q V L O I K G N W B M)
```

In both of these cases, we get the same expression for NPN^{-1}:

$$NPN^{-1} = (\texttt{A Y U R I C X Q M G O V S K E D Z P L F W T N J H B})$$

To extract N from here is simple - we write the permutation P underneath, then rearrange the top row to put it into alphabetical order. There are of course 26 ways of writing P underneath, we cannot know which of these gives the correct N without more information. Which one on the solutions is the correct one depends now only on the ring settings.

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

Concluding remarks

There were several additional complications which made actual attack much more involved than the description above: the keyboard —the way the keyboard is wired to the rotors was different between commercial and military versions of Enigma (it was an identity $A \to A$, $B \to B$, ...). This complication caused large time loss. Another complication was the changes in the order of the rotors, ring settings, and the stepping of the rotors M and L.

The cyclometer

At this point, they knew the wirings of the rotors, but the rotor order, the indicator settings, representing the daily key, and the daily plug settings were still unknown. So it was needed to find these daily keys so that messages could be rapidly decrypted. For each day, there would be 6 possible arrangements of three rotors on the axis, and $26^3 = 17,576$ ways of placing the starting location of the rings, giving a total of 105,456 possibilities. To solve this problem, the Poles utilized the fact that the number and lengths of the cycles in the three permutations (AD, BE and CF) are invariant with respect to the plug board combinations. For each rotor order, we need to generate the cycle structures at each starting position (this is based on the assumption that only the fast rotor is moving). With the help of the cyclometer, they constructed a catalog involving the necessary 105,456 possibilities.[6] Using the catalog, it took 20 minutes to produce correct rotor order and plugboard settings C. The indicator settings were solved from a list of daily interceptions and using the guesses about frequently used triplets of letters for several indicators in the list. The other, more carefully chosen indicators could be solved based on this entry information.

Perforated sheets

On September 15, 1938, the Germans changed the rules for encoding messages with the Enigma system. Now the Enigma operator would select the basic position, selecting a new one for each message. He would transmit this unenciphered at the head of the message. The message key, however, would still be encoded twice as before. This meant that the products AD, BE and DF could no longer be calculated for each day because each message began from a different basic position. However, there was still a relation between the first and fourth, second and fifth, and third and sixth letters. Sometimes, a message would begin with a phrase such as `PST PWA`, where the first and fourth letter were the same (or similarly for the other letters). This meant that in AD (or BE or DF depending on the case) there is a cycle of length 1. Since the length of the cycles in these products is invariable with respect to the plugboard permutation C, if there were these cycles of length 1, also called constant points, then these were also invariable with respect to the transformations.

A catalog was now constructed of constant points for all 17,576 possible rotor arrangements. Henryk Zygalski found a quick way to check these positions. A piece of paper was prepared for each of the 26 positions of L (the third rotor) - the sheet was divided into 51 x 51 small squares. Along the sides were written the alphabet to correspond to these squares, one in full from A to Z, the second time from A to Y. This became a set of points, whereby the axes represented the possible positions of rotors M and N. Any position that referred to a constant point was perforated. When these sheets were superimposed and moved in the proper sequence and manner, the number of visible apertures decreased. With sufficient data, there would remain only one aperture, which referred to the daily key, which would then give enough information to be able to decipher messages.

Turing-Welchman's bombe (probable word attack)

At a historical meeting in Pyry on 25.7.1939, the Polish taught the French and the British cryptanalysts their methods of solving the Enigma. They also presented the allies with the copies of the German cipher machine, that has been reconstructed in Poland on the basis of the theoretical work described above.

[6] A work which required one year, and in November 1937, just when the catalogs were ready, the Germans changed the writes on the reflecting rotor and the process had to be restarted.

From this point, the British took the lead in breaking the Enigma. During the war, they employed about 30,000 people of various occupations (mathematicians, engineers, paleontologists, crossword puzzle hobbyists, etc) to solve the Enigma encrypted traffic. This massive effort took place at Bletchley Park. One of the leading figures in this work was mathematician Alan Turing.

The initial information available to the British was:

1. The rotor wirings for all the rotors

2. Probable plaintexts

However, they did not know the plugboard settings, for which there are up to $\frac{\binom{26}{2}\binom{24}{2}\cdots\binom{2}{2}}{13!} \approx 7.9 \cdot 10^{12}$ possibilities (far too many to search exhaustively), the order of the rotors, the ring and indicator settings.

The British mathematician Alan Turing based his method for breaking the Enigma code on the use of probable plain text. His idea was to design a high-speed machine that would step through all the possible rotor positions [7] while testing for a probable plain text. Due to the prohibitive number of plugboard settings, there needed to be a way of testing without exhaustively searching every possibility. One had to find a characteristic of the crib, that was plugboard invariant. Such a characteristic was found: crib loops. Consider the following ciphertext lined up against a crib (this was a common message beginning):

```
           1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4
PLAIN TEXT:O B E R K O M M A N D O D E R W E H R M A C H T
CIPHERTEXT:Z M G E R F E W M L K M T A W X T S W V U I N Z
```

In positions 7,9 and 14, we can form a loop from the substitutions:

```
      9       7      14
  A  -->  M  -->  E  -->  A
```

Recall that in decryption the Enigma machine is the same as encryption because it is an involution. Defining C as the effect of the plugboard, R_i as the combined effect of the rest of the machine in position i, we can write the transition from M to E at position 7 as (note that the permutations are being performed right to left):

$$\mathtt{E} = CR_7C(\mathtt{M})$$

[7] In the fall of 1938, Polish mathematicians invented a device which they called 'Bomba' or cryptological bomb (probably due to a ticking sound that it made). This device consisted of six electrically powered Enigmas and was used to automate and accelerate the process of reconstruction of the daily keys. This expensive device (six were built in Warsaw until September 1939) took the place of about 100 workers and shortened the time for obtaining the key to about 2 hours.

and the other two permutations:

$$\mathtt{M} = CR_9C(\mathtt{A})$$

$$\mathtt{A} = CR_{14}C(\mathtt{E})$$

The first two equations can then be combined:

$$\mathtt{E} = CR_7CCR_9C(\mathtt{A})$$

Note that two adjacent plugboard permutations cancel each other. Substituting A by the third expression we obtain:

$$\mathtt{E} = CR_7R_9CCR_{14}C(\mathtt{E})$$

$$C(\mathtt{E}) = R_7R_9R_{14}C(\mathtt{E})$$

This equation now involves only the plug-board settings for E and the rotor permutations. At this point, we could guess what $C(\mathtt{E})$ is (there are only 26 possibilities), and for all possible message keys (there are 17,576) test out our three permutations (i.e. Enigmas at positions 7,9 and 14) and see if we get back the same result. However, with this system, we would expect to get back the same letter about one time in 26, even if it is the wrong settings. So we need to use another loop, which indeed is present in the crib and luckily for us it has a common letter (E) with the previous loop:

```
     4        15        8        7
  E  -->  R   -->  W  -->  M  -->  E
```

This gives us the equation:

$$C(\mathtt{E}) = R_4R_{15}R_8R_7C(\mathtt{E})$$

Now we can expect a false alarm once every 676 tests. We have 26 possibilities for the plug and 26^3 possibilities for the indicator setting. A set of two loops poses a condition which can be satisfied only by a fraction of $(\frac{1}{26})^2$ possible key guesses. With 4 loops, we will have only few possibilities for the key. In Figure 8, we present a graph of the connections produced by the 24-letter crib shown above. This graph exhibits four loops necessary for Turing's attack and a short loop[8]: ($\mathtt{R} \xrightarrow{15} \mathtt{W} \xrightarrow{19} \mathtt{R}$). The British bombe contained batteries of simulated

[8]Such short cycles could alone be very useful. When the number of plugs is not large (6 or 7) there is a 25% chance that both occurrences are not plugged. Thus for such a loop we need to try only the 26^3 possible indicator settings and the loop leaves only $\left(\frac{1}{26}\right)^2 \cdot 26^3 = 26$ of them. These few possible indicator settings could be found by means of so called 'Turing sheets' or 'Jeffrey sheets' in a way similar to the 'perforated sheets' method described above.

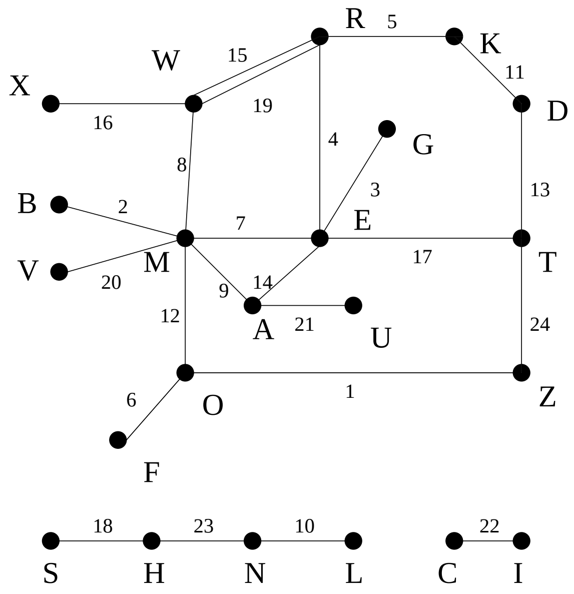


Figure 8: A graph showing the connections (i.e. the machine at which time position) with the 24-letter crib and ciphertext

Enigma machines, each of them starting at a different displacement (10 machines are needed for the crib in our example). The bombe was constructed to utilize feedback so as to reduce the time complexity. This was done by using a 26-terminal test-register to keep track of the results and a 26-line bus connecting all the machines. At each rotor position, current was fed into the test-register letter corresponding to some point on the loop (for example, A with the first loop). If the rotor positions were correct, then:

1. If this was the correct point, then the current will return to terminal A because it was a loop, forming a closed circuit, and thus a single test lamp will be lit.

2. If not, it will return to some other point, and be fed back into some other point by the test register. This will continue until all test register terminals (except for the correct one which forms a closed circuit) will sense current, or a loop is formed before this happens.

If the rotor positions are incorrect, case 2 usually holds, but now all 26 terminals can be reached by the current. Thus we watch for two cases: single test lamp is lit, or all lamps except one are lit (this assumes that we have enough loops, and current reaches all terminals in incorrect cases).

Below is an example of the feedback process for the correct guess of the indicator setting and an incorrect plug guess in the loop involving Enigmas at positions 7, 9 and 14. Enigma alphabets for these positions are given below:

```
       ABCDEFGHIJKLMNOPQRSTUVWXYZ
R(7)   RNVQMXLYSPWGEBZJDAIUTCKFHO
R(9)   MCBPJURIHEOWATKDSGQNFYLZVX
R(14)  EODCALMQJIVFGXBRHPYWZKTNSU.
```

Note that the currents that pass through the test register (the first column) would be almost the entire alphabet:

	7		9		14	
	$\rightarrow$		$\rightarrow$		$\rightarrow$	
A		R		G		M
M		E		J		I
I		S		Q		H
			$\vdots$			
L		G		R		P
P		J		E		A

This incorrect loop shows that initial guess $C(\texttt{E}) = \texttt{A}$ was incorrect, and moreover, $C(\texttt{E})$ can not be any of the letters M, I, H, ..., P (if the rotor position is correct). Thus there remain only 3 possible correct plugs: (B,E,W), $C(\texttt{E}) = \texttt{E}$ is a correct one.

Turing's method had two major drawbacks: it required long (more than 20 letter) cribs in order to encounter four loops; it assumes that the middle rotor was not stepping in the crib. Due to the unknown ring setting the latest assumption was wrong for most of the long cribs, and thus one had to guess the location of the "turnover" (usually dividing the crib into 5 sections: turnover between 1 and 5, 5 and 10, 10 and 15, etc. and omitting the 5-letter region). There was also a technical problem of wiring 26 cables to the Enigma, since due to the reflecting property it will return only 13 of the input channels. A simple design solution was to use a set of 7 rotating drums representing the rotors: III, II, I, R, I, II, III. Gordon Welchmann suggested a clever way to overcome these difficulties. The idea was to follow all the implications of a single plug guess $C(\texttt{B}) = \texttt{C}$ feeding current in both B and C terminals. This was implemented using a "diagonal board" - a 26x26 matrix of reciprocally connected terminals. A feedback line had to be connected from the proper rotor set to the row of the board corresponding to the appropriate letter. This allowed the disconnected letters to also provide feedback, and greatly reduced the number of false alarms. With Welchman's idea, cribs of half the length could be used (see Table 10). The avalanche of implications caused by a short 13-letter crib (even without loops) was sufficient to provide a single "stop" (possible correct guess) per rotor order. See Figure 11 for an example. In this example, initially the plug B$\rightarrow$C is guessed. By using our knowledge of the rotor and reflector wirings, and the initial indicator setting guess, we know that the C (at position 2) will give us a G. However, the ciphertext at this position is an M, implying the plug G$\rightarrow$M.

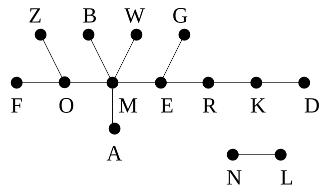


Figure 9: A diagram showing the relationship between the letters for a 12-letter portion of the crib: OBERKOMMANDO .... Notice that this crib has no loops.

So in the next step, we can assume this plug, and try to follow the chain of implications. In this process, we have a possible contradiction between the third and the seventh positions which both imply the plug value for the letter E. At this point let us assume that there is no contradiction and that in both cases $C(\mathtt{F}) = \mathtt{E}$. Proceeding with the line of implications we receive: L→W, A→P and K→O. Now both the fourth and fifth positions imply a plug for R. If the fourth position implies R→H and the fifth R→S we receive a contradiction, also in the sixth position both plugs are known thus if $R_6(\mathtt{K}) \neq \mathtt{E}$ it is a contradiction. This means that plug guess B→C (at this indicator setting) is wrong. Such a process needs to be performed at all 17,576 indicator settings. We will find a "stop" if there are no contradictions.

Number of loops	Number of non-loop letters
4	0
3	9
2	11
1	12
0	13

Figure 10: Bombe set-ups necessary for one "stop" per rotor order.

The plugboard and loop-testing occurs at close to the speed of light, so the bottle neck is the mechanical stepping of the rotors. The bombes operated at 1800 rpm, and so took about 11 minutes to run through all the rotor positions for a certain rotor ordering - to change the ordering, the order would have to be changed by hand in the whole bank, and the process repeated. The Bombes allowed for daily reading of the German traffic. By the end of the war, about 200 bombes were operational.

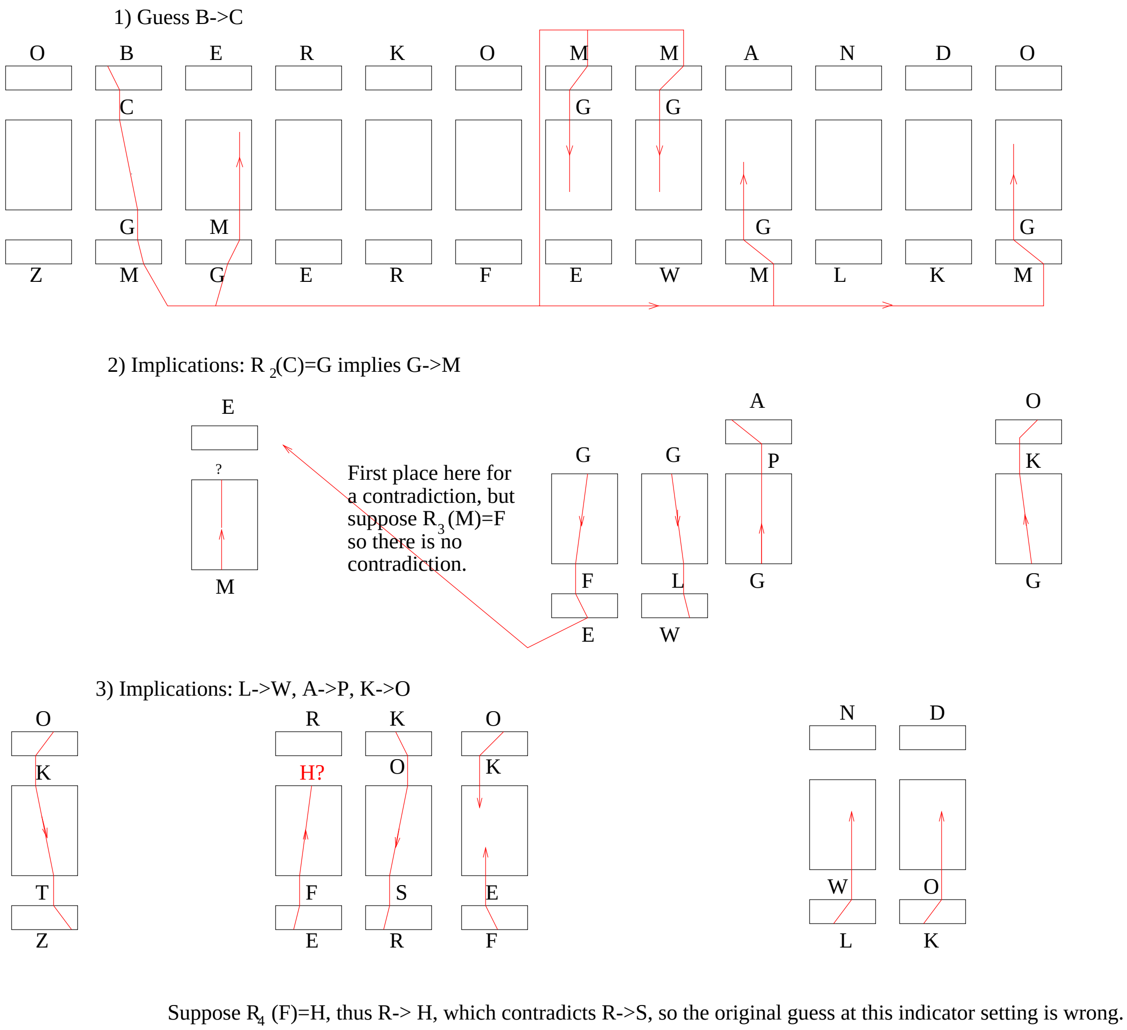


Figure 11: An example of the guess B $\rightarrow$ C and its implications at a certain indicator setting.

Ciphertext only analysis by Gillogly

The attacks we have studied so far were known plaintext (probable-word) attacks. In 1995, Gillogly[4] described a ciphertext-only attack on Enigma based on the following assumptions:

1. The wirings of the rotors in use, and the reflector were known.
2. Long messages were available (300-600 letters). Messages of this length were prohibited by German instructions, however Gillogly claims to have found examples in the Bletchley Park archives, although the historical accuracy is questioned[5]. In his paper, he says that on 6 Aug 1943, there was an average of 340 letters (from 107 messages), and on 13 Nov 1943, there was an average of 550 letters. On this day, there were 10 messages with lengths over 1000 letters.
3. Only six plugs were used, although up to 10 plugs were used later in the war.

The Germans themselves knew how to break the code of an Enigma without a plugboard - the Germans would read Swiss diplomatic correspondence that was encoded using a modified version of the commercial Enigma[2], with a probable word attack, described earlier in these lecture notes. They believed the strength of the Enigma was based on the plugboard. Gillogly exploits the fact that a significant portion of the stream is not effected by the plugboard, if few plugs (less than 6) are used. If all the other key settings are correct (the correct rotor order, ring settings, and rotor message key) but with no plugs in the plugboard, some of the letters in the decrypted plaintext will have their correct plaintext value - those letters that did not pass through a plug on the way from the keyboard, or out to the lights. Although it is not possible to know which of these letters were changed, if the message is sufficiently long, it is possible to look at the message statistics. If 6 plugs were used, then 12 letters are interchanged, so 14 are unchanged.

Some considerations on the Gillogly's attack

1. Note that at point B(in Figure 12) we have "plugged German", some letters are real German, some are swapped. Plugged German has the same IC statistic as real German, since frequency profile (f_i) is not changed.
2. At point D the cipher stream can be considered uniformly random and thus which letters are plugged (E$\leftrightarrow$A—frequent letters, or Q$\leftrightarrow$X) is not important. We know that if there are p plugs, then $\frac{2p}{26}$ fraction of the stream is effected and $\frac{13-p}{13}$ is not.
3. In Gillogly's attack we decrypt the text with the plugless Enigma, thus with a correct indicator setting, for $\frac{p}{13}$ of the stream, we receive random letters, and for $\frac{13-p}{13}$ we receive text in "plugged German". Given a

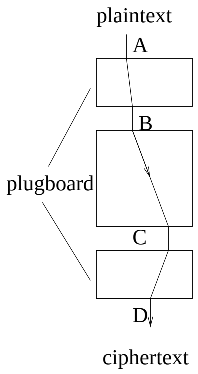


Figure 12: Encryption stages in the Enigma cipher.

message of length n, we have $\frac{n \cdot p}{13}$ random letters and $n \cdot \left(1 - \frac{p}{13}\right)$ plugged-German. The probability for the ith letter in a random sample is $\frac{1}{26}$ and for German it is g_i. Then the total frequency of the ith letter in a text of length n is:

$$f_i = \frac{1}{26} \cdot \frac{n \cdot p}{13} + g_i \left(1 - \frac{p}{13}\right) n$$

4. We can calculate the IC of such a combined text:

$$\begin{aligned}
IC &\triangleq \sum_1^{26} \frac{f_i(f_i - 1)}{n(n-1)} \\
&= \sum_1^{26} \left(\frac{1}{26}\frac{np}{13} + ng_i\left(1 - \frac{p}{13}\right)\right) \cdot \frac{\left(\frac{1}{26}\frac{np}{13} + ng_i\left(1 - \frac{p}{13}\right) - 1\right)}{n(n-1)} \\
&\approx \frac{p^2}{26 \cdot 13^2} + \left(1 - \frac{p}{13}\right)\frac{p}{13^2} + \left(1 - \frac{p}{13}\right)^2 \mathrm{IC(German)} - \frac{p}{13} \cdot \frac{1}{n} - \frac{1}{n} + \frac{p}{13} \cdot \frac{1}{n} \\
&= \frac{p}{13}\left(2 - \frac{p}{13}\right)\mathrm{IC(Random)} + \left(1 - \frac{p}{13}\right)^2 \mathrm{IC(German)} - \frac{1}{n}.
\end{aligned}$$

Gillogly's method assumes that is sufficient for the message statistics to be affected enough that the correct settings could be identified. His method worked as follows:

1. Set the rings to their initial settings (the impact of this will be discussed later), then run through all possible message keys (ie AAA to ZZZ) - there are $26^3 = 17,576$ possibilities. Also try all possible rotor orders (there are 5 rotors, so $5 \cdot 4 \cdot 3 = 60$ possibilities. This is not a prohibitive number of trials for a computer program.

2. For each key setting, calculate the Index of Coincidence (IC). Random 26-letter text gives an IC or approximately 0.038, while standard German is about 0.07. We can use the formula for the IC, breaking it up into the random and German parts, as shown before. We can expect an IC of 0.04218 for our random and German mix of letters, with a variance of 0.0011. This means that we should be able to see a difference from random text (IC=0.038).

3. Now look at those trials which gave the best IC. For example:

Order	IC	Message Key
II I III	0.0416	2 11 6
III I IV	0.0414	20 8 17
V II I	0.0413	20 16 3
II III IV	0.0412	20 20 26
III I V	0.0411	19 16 20
IV I II	0.0411	4 20 7

The key space is now drastically reduced, however, now the ring settings need to be taken into account.

4. The ring setting was part of the daily key. Setting the ring would determine when the next rotor would step. Every letter would cause the first rotor to step, however, the next rotor would step only once every 26 letters. The first time it steps is determined by the ring setting - after that it would step 26 letters later. The ring setting for the second rotor worked in a similar way. It is not necessary to consider the ring on the third rotor, because it has no effect on the stepping function. The method used was to first try the first rotor in all 26 positions, and see which gave the best IC score, then do the same with the second rotor. For example, changing the ring setting on the first rotor (note that we simultaneously need to change the message key so the wirings remain the same):

```
        Rotor index 2, ring  1, message key  6:  0.0416
        Rotor index 2, ring  2, message key  7:  0.0418
        Rotor index 2, ring  3, message key  8:  0.0416
--->    Rotor index 2, ring  4, message key  9:  0.0422
        Rotor index 2, ring  5, message key 10:  0.0420
        Rotor index 2, ring  6, message key 11:  0.0416
        Rotor index 2, ring  7, message key 12:  0.0416
```

Ring setting 4 was chosen as it had the highest IC. The same process was repeated with the second ring:

```
       Rotor index 1, ring 20, message key  4:  0.0432
       Rotor index 1, ring 21, message key  5:  0.0428
       Rotor index 1, ring 22, message key  6:  0.0437
--->   Rotor index 1, ring 23, message key  7:  0.0440
       Rotor index 1, ring 24, message key  8:  0.0437
       Rotor index 1, ring 25, message key  9:  0.0429
       Rotor index 1, ring  0, message key 10:  0.0423
```

Gillogly initially put the ring settings to `1 1 1`. If the ring settings were not equal to these, then there would be a period of time when the wrong transition function was being used (see Figure 13). If the selection of ring

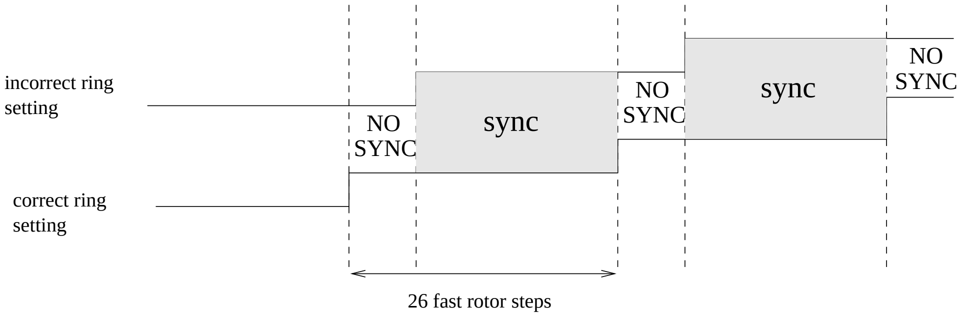


Figure 13: Temporary synchronization loss due to incorrect ring setting

settings was close (as it was in the example), then this period of wrong transitions can be 'fixed' by rotating the ring (note that the message key simultaneously also needs to be changed) to find which setting gives the highest IC. If the guess of ring settings was bad, then half or more of the text would be random still, making the analysis very difficult. This problem could potentially be overcome by trying each message key setting with a number of ring settings (say 3 for each of the 2 rotors that have any effect, would increase the number of tests necessary by a factor of 9). Note that if messages are considerably shorter than 676 letters the chance for 3rd rotor stepping decreases.

5. From here, the only thing left is to try and recover the plugboard settings. There are 625 possible plugs ($25 \cdot 25$), so it is suggested to try each pair individually and decrypt the message using this setting. At this stage, Gillogly recommends the use of trigraphs scores, and a hill-climbing method with a table of base-2 logarithms of raw trigraph counts. The score is simply calculated by summing the values found in the table (summing the logarithms has an effect similar to multiplying trigraph probabilities). For example, he began with the text:

```
577 AIZLJWZHODZTFUWGSRZYLZRNQLJHAVUCPCPZIBAFGCSTSEGEFC
```

The number indicates the trigraph score. The best plug setting was found to be using the $E - Z$, plug which gave:

```
E-Z 1029 AIELJWEHBDETJUWGSREYLEFNQLJHAVUCPCPEIUAFGCSTSZGZFC
```

From here this step was repeated with other plugs:

```
R-W 1258 AIFLEREHBDETJURGSWENLEFEQLJHAVECPCUEIUAFGBSESZGZFP
M-V 1457 AIFLEREHBDETJUEGSWENLEFEHLJHAMERSSUEIUAFABSESZPZTP
I-U 1751 AUFLEFEHBDETJIEGSWENLEFEHLJHALERSSIEUIAFABBESZPZTP
B-L 2121 AUFBEFEHLDETJIEGSWENBEFEHLJHABERSSINDIMFALLEPZPZTP
P-X 2454 AUFBEFEHLDESJIEGSWENBEFEHLSHABERSSINDIMFALLEXZXZTX
J-O 2814 AUFBEFEHLDESOBERSTENBEFEHLSHABERSSINDIMFALLEXZXZTX
```

From here the plaintext was readable.

Effectiveness

Gillogly performed an experiment by altering the number of plugs from 0 to 13 and message lengths from 161 to 1463 letters with 100 randomly-selected settings. Figure 14 shows the results of his experiment. He claims for the level of traffic seen at Bletchley Park, it would have been possible to recover at least one message a day.

Improvements by Heidi Williams

Gillogly's algorithm was improved by Heidi Williams[6], by choosing a more effective statistical test and improving the algorithm. Her method allowed deciphering of a 500-letter message encrypted with 10 plugs. She made two significant changes — the first was rather than keeping only the best result after running through all the indicator settings (with the ring settings set at AAA), she kept the best 3000. The other change was to use a more effective statistical test in place of the Index of Coincidence. After comparing several methods, she found the 'Sinkov statistic' to be more effective, which is defined as:

$$S = \sum_{1 \le i \le m} n_i \ln p_i$$

where n_i is the frequency of letter i in the test string and p_i is the probability of letter i in the language. The best result will be the one with the highest value for the Sinkov statistic. Her method allowed solving 10 plug Enigma messages of around 500 letters in between 2.5 to 10 hours on a 266MHz PC.

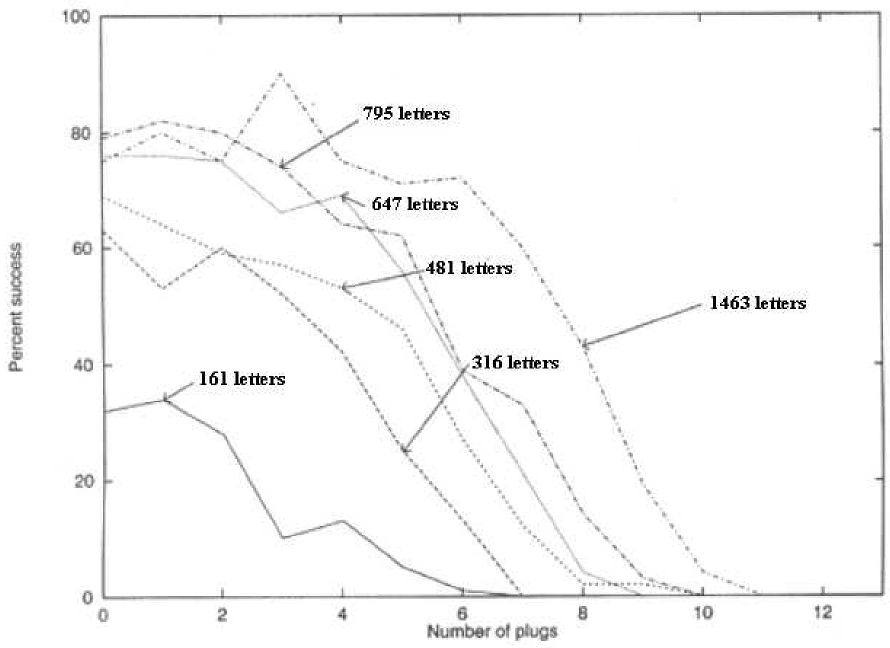


Figure 14: Key Recovery as a function of plugs and message length

Bibliography

[1] http://www.und.nodak.edu/org/crypto/crypto/graphics/enigma/, part of the ACA (American Cryptogram Association web site).

[2] Cipher A. Deavours, Louis Kruh, *Machine Cryptography and Modern Cryptanalysis*, Artech House, 1985.

[3] M. Rejewski, *Mathematical Solution of the Enigma Cipher*, Cryptologia, Volume VI, Number I (January 1982).

[4] J. Gillogly, *Cryptologia October 1995;Volume XIX, Number 4*, also found at
http://www.fortunecity.com/skyscraper/coding/379/gillog1.htm

[5] R. Erskine, *Cryptologia October 1996;Volume XX, Number 4M*, letter to the editor, also found at
http://www.fortunecity.com/skyscraper/coding/379/erskin.htm

[6] H. Williams, *Cryptologia, January 2000;Volume XXIV, Number 1*, Applying Statistical Language Recognition Techniques in the Ciphertext-only Cryptanalysis of Enigma