‡3 Code-breaking in the second world war

by Hugh Thurston[1]

A The initial break into Enigma

A1 Some time before the outbreak of the second world war the Poles began intercepting German code messages with a very distinctive feature. Here are the beginnings of some messages received on the same day. (Not the actual messages, of course; I have made up some examples to illustrate the Polish cryptographers' methods.)

1	**d**	**h**	**z**	**e**	**c**	**n**	**w**	**k**	...
2	**a**	**l**	**w**	**r**	**w**	**p**	**f**	**u**	...
3	**s**	**v**	**g**	**w**	**s**	**r**	**j**	**q**	...
4	**d**	**e**	**b**	**e**	**k**	**h**	**p**	**q**	...
5	**h**	**q**	**x**	**h**	**a**	**z**	**m**	**c**	...
6	**h**	**e**	**c**	**h**	**k**	**l**	**w**	**o**	...
7	**u**	**q**	**b**	**g**	**a**	**h**	**e**	**l**	...
8	**a**	**q**	**c**	**r**	**a**	**l**	**w**	**g**	...
9	**z**	**w**	**j**	**v**	**i**	**c**	**k**	**d**	...
10	**m**	**u**	**m**	**x**	**r**	**o**	**w**	**j**	...
11	**r**	**s**	**e**	**o**	**p**	**u**	**m**	**n**	...
12	**s**	**p**	**q**	**w**	**q**	**q**	**x**	**o**	...
13	**e**	**f**	**d**	**p**	**h**	**f**	**r**	**j**	...
14	**e**	**m**	**a**	**p**	**e**	**d**	**l**	**v**	...
15	**f**	**n**	**a**	**m**	**f**	**d**	**p**	**q**	...

A2 The feature is that whenever two messages have the same first letter they have the same fourth letter. For example, messages 2 and 8 both have first letter **a** and fourth letter **r**; messages 1 and 4 both have first letter **d** and fourth letter **e**, and so on. Similarly, whenever two messages have the same fourth letter, they have the same first letter. The same relation holds between the second and fifth letters and between the third and sixth letters. This is something which any cryptographer would spot, but the Poles continued with something a good deal more subtle. They looked at the correspondences between the first and fourth letters:

First	**d**	**a**	**s**	**h**	**u**	**z**	**m**	**r**	**e**	**f**
Fourth	**e**	**r**	**w**	**h**	**g**	**v**	**x**	**o**	**p**	**m**

and began to build them up into chains. From **de** and **ep** they formed **dep** and so on. They had far more than the fifteen messages I have displayed (at least eighty each day) and managed to incorporate all the letters of the alphabet, the final result being

$$(\mathbf{depzvlyq})(\mathbf{aronjfmx})(\mathbf{gktu})(\mathbf{icsw})(\mathbf{b})(\mathbf{h}) \qquad (1)$$

[1] Professor Emeritus of Mathematics, University of British Columbia. It should be emphasized here that Thurston took part in the very code-breaking that is the subject of this paper — and which helped the Allies win World War 2, one of the few just war-efforts human history can boast of.
[Note added 2008/11/5. At Midway (1942/6/4), the Allies' code-breaking edge was instrumental in the Japanese invading fleet's sudden simultaneous loss of 3 of its 4 aircraft-carriers and thus most of its c.300 airplanes & veteran ace pilots . (Its commanders had learned of the US' nearby-lurking 3 carriers only 2^{h} earlier and could not prepare a full attack in time, an attack that indeed was just being launched at the very minute when elated US dive-bombers struck.) Yet, before Japan's entire 300-plane force was lost (along with the 4th carrier), just 16 of the final few planes sank the US carrier *Yorktown*, in retaliation, suggesting that had Japan's original full 300-airplanes been available to strike at the Allied fleet, the Battle of Midway would've ended quite differently.]

These chains are cyclic; not only does **d** as first letter correspond to **e** as fourth, but **q** (the last letter in the first chain) corresponds to **d**, the first in the chain. So I shall refer to them as cycles rather than chains. Note that **h** corresponds to itself; so does **b**.

A3 The corresponding results for second-to-fifth and third-to-sixth are

$$(\mathbf{axzlwiobyvspq})(\mathbf{hcurtmekdgjnf}) \tag{2}$$

and

$$(\mathbf{adfbhgrsywpk})(\mathbf{znmoieutjclx})(\mathbf{q})(\mathbf{v}) \tag{3}$$

Now we notice another striking feature. Each of these results contains two cycles of each length: (1) has two of length 8, two of length 4, and two of length 1; (2) has two of length 13; (3) has two of length 12 and two of length 1. To anyone familiar with the mathematical theory of permutations — and the Polish cryptographers were mathematicians — this rings a bell.

A4 The connection between cryptography and the theory of permutations is that an encoding is a permutation. If we write out the alphabet and under each letter write its encode, for example:

Original	**abcdefghijklmnopqrstuvwxyz**
Encode	**hiewdfvxgybuctramnskzqjlop**

then the encoding permutes the top row into the second row.

A5 There is a standard way of writing permutations. Here **a** permutes to **h**, so we write **ah**. Also **h** permutes to **x**, giving us **ahx** and so on. Continuing in this way, we write the above permutation in cycles as

$$(\mathbf{ahxluzp})(\mathbf{bigvqmcedwjyorntk})(\mathbf{f})(\mathbf{s})$$

We represent permutations by capital roman letters. The letter into which A permutes **x** is denoted by **x**A. If we apply permutation A and then apply permutation B to the permuted letters, the resulting permutation is denoted by AB and called the composite of A with B. For example, if A is **(abc)(de)(f)** and B is **(ac)(be)(df)**, then A sends **a** to **b** and B sends **b** to **e**, so AB sends **a** to **e** and one of its cycles starts **ae**. You can easily check that AB is **(aefdb)**. The result of applying AB to **x** is the same as applying B to **x**A, so **x**(AB) = (**x**A)B, and we can write the result unambiguously as **x**AB. Similarly (AB)C = A(BC) = ABC, the result of applying first A, then B, then C.

A6 If X is an encoding, the corresponding decoding is denoted by X^{-1}, it is the reverse permutation and its cycles are the cycles of X reversed. A permutation in which every cycle has length 2 is called a pairing. If X is a pairing, X^{-1} = X. The result in the theory of permutations that rang a bell for the Polish cryptographers is the theorem that the composite of two pairings must have an even number of cycles of each length (as do (1), (2) and (3)).

A7 We can see why this happens as follows. Suppose that we want XY, where X = **(ab)(cd)(ef)(gh)** and Y =**(ad)(bh)(cg)(ef)**, to take a reasonably compact example. In X, $\mathbf{a} \to b$; in Y, $\mathbf{b} \to h$; so in XY, $\mathbf{a} \to h$. Write this as

a h
 b

Similarly

h c and **c a**
 g **d**

bringing us back to where we started. Putting these together we have the set-up

$$\begin{array}{c} \mathbf{a\ h\ c\ [a]} \\ \mathbf{b\ g\ d} \end{array} \tag{4}$$

A8 The northwest-to-southeast pairs are pairs of X, the northeast-to-southwest pairs are pairs of Y, the top row **(ahc)** and the bottom row reversed **(dgb)** are cycles of XY, necessarily of the same length. We continue in the same way with any unused letters, in this case

e [**e**]
 f

each time obtaining two cycles of the same length. We deduce that each of (1), (2) and (3) is the composite of two pairings. This is what would happen if each of the six encodings is a pairing and the first six letters of each message are the first three letters repeated. We can see this as follows.

A9 Suppose that the first and fourth encodings are X and Y and that they are pairings. If the first letter in an encoded message is **a**, the original first letter is **a**X. If the original fourth letter is the same, the fourth letter in the encoded message is **a**XY. The same applies to any letter, so the first-to-fourth correspondence is XY, a composite of two pairings. This deduction ties in with what Polish Intelligence had discovered. The Germans had a machine, the Enigma, which encoded by pairings. It used three rotors, each of which could be set in 26 different positions, which could be denoted by letters. The recipient of a message would need to know the positions of the rotors, so the message would need to contain a trio of letters giving this information. It is a good bet that the first six letters of the message are this trio repeated.

A10 We now have the following problem: given XY, where X and Y are pairings, find X and Y. To take a simple example, what pairings will give

XY= $(\mathbf{ahc})(\mathbf{dgb})(\mathbf{e})(\mathbf{f})$?

The set-up labelled (4) above shows us how to find out. Write the second cycle of XY reversed under the first cycle. There are three ways to do this:

a h c [**a**] **a h c** [**a**] **a h c** [**a**]
g b d **g d b** **d b g**

And there is one way to write e under f. We get three solutions:

X= $(\mathbf{ab})(\mathbf{hg})(\mathbf{cd})(\mathbf{ef})$, Y= $(\mathbf{ad})(\mathbf{cg})(\mathbf{hb})(\mathbf{ef})$
X= $(\mathbf{ag})(\mathbf{hd})(\mathbf{cb})(\mathbf{ef})$, Y= $(\mathbf{ab})(\mathbf{cd})(\mathbf{hg})(\mathbf{e}f)$

and

X= $(\mathbf{ad})(\mathbf{hb})(\mathbf{cg})(\mathbf{ef})$, Y= $(\mathbf{ag})(\mathbf{c}b)(\mathbf{hd})(\mathbf{ef})$

Notice that the two cycles **(e), (f)** of length 1 imply that in every solution **e** and **f** encode into each other.

A11 It is easy to see that for (1) there are 32 solutions. The next step is to pick out the right solution, and to do the same for (2) and (3). To do this, the cryptographers used a fact that we found over and over again in cryptography: ask anyone to choose something "at random" and the result, though arbitrary, is usually far from random. The German operators did not choose the settings for the rotors randomly, but showed a strong predilection for trios like **aaa, bbb**, etc. or **abc, bcd** etc.

A12 Let the first six encodings be A, B, C, D, E, F, so that AD is (1), BE is (2) and CF is (3). From the two cycles of length 1 in (1), we see that in A and D the letters **b** and **h** encode into each other, so messages 5 and 6 start with **b**. Could either of them be **bbb**? Let us start with message 5. If we have

A	B	C	D	E	F
h	**q**	**x**	**h**	**a**	**z**
b	**b**	**b**	**b**	**b**	**b**

then **(qb)** must be a pair in B and **(ab)** in E. This is not possible; in (2) **b** is in the same cycle as **a** and **q**, so no way of writing the second cycle of (2) under the first will work. Now let us try message 6. If we have

A	B	C	D	E	F
h	**e**	**c**	**h**	**k**	**l**
b	**b**	**b**	**b**	**b**	**b**

then **(eb)** must be a pair in B and **(kb)** in E. There is a possible arrangement of (2), namely

a x z l w i o b y v s p q [**a**]
h f n j g d k e m t r u c

This gives

B = **(ah)(be)(cq)(di)(fx)(gw)(jl)(ko)(my)(nz)(pu)(rs)(tv)**
E = **(ac)(bk)(do)(ey)(fz)(gi)(hx)(jw)(ln)(mv)(pr)(qu)(st)**

Again **(cb)** must be a pair in C and **(lb)** in F, and again there is a solution.

A13 We can arrange (3) as

a d f b h g r s y w p k [**a**]
z x l c j t u e i o m n

giving

C = **(az)(bc)(dx)(es)(fl)(gt)(hj)(iy)(kn)(mp)(ow)(qv)(ru)**
F = **(an)(bl)(ch)(dz)(ey)(fx)(gj)(iw)(km)(op)(qv)(rt)(su)**

Now let us decipher the rotor-settings using B, C, D and F.

1 **.aa** 2 **.jo** 3 **.tt** 4 **.bc** 5 **bcd**
6 **bbb** 7 **.cc** 8 **.cb** 9 **.gh** 10 **.pp**
11 **.rs** 12 **.uv** 13 **.xx** 14 **.yz** 15 **.zz**

We have a classic case of non-random trios. Messages 1 and 4 start with the same letter; the trios are obviously **aaa** and **abc**. And 13 and 14 will be **xxx** and **xyz**. If so, **(ad)** and **(xe)** must be pairs in A and **(ae)** and **(xp)** in D. These work well. From (1) we obtain

a r o n j f m x [**a**]
d q y l b z p e

A14 What about the other cycle? Can message 3 be **ttt**? It can: we need **(ts)** in A and **(tw)** in D. The set-up

g k t u [**g**]
i w s c

gives these. We now have the first six encodings and all the rotor settings.

A15 This is not enough to enable us to read the coded messages, but it is a good start. In §B, we shall see how the Polish cryptographers used six successive encodings (actually they needed only four) to analyze the Enigma machine. Much has been written about the successful exploitation of the German Enigma codes by the Allies, starting with F.W. Winterbottom's *The Ultra Secret*. This breakthrough by the Polish cryptographers, on which the British exploitation was based, was the start of something really big.

B Analyzing the Enigma machine

B1 We saw in §A how Polish cryptographers attacked German messages encoded by a machine called Enigma. By purely cryptographic analysis of a day's messages they found six successive encodings produced by the machine. To follow the next steps we need to know how the machine was constructed. The military Enigma was modified from a commercial Enigma that anyone could buy.

B2 The Enigma has a keyboard like the one on a typewriter, except that it has only 26 letters: no numbers, no symbols, and no shift key. The keys are connected by wires to a *plugboard*, a board with 26 terminals on each face. Normally when a key — let us suppose that it is **x** — is depressed, a current flows from the key to terminal **x** on the front face of the plugboard and straight across to a terminal on the other face which I shall call rear terminal **x**. However, the operator has six wires, each with a plug on each end. If the two plugs on a wire are plugged into front terminals **x** and **y** they disconnect the two **x** terminals and the two **y** terminals, and connect the front **x** to the rear **y** and the front **y** to the rear **x**. If the current goes in at **x** it comes out at **y** and vice versa.

B3 From the rear terminals on the plugboard wires go to 26 terminals, which I will call entry terminals, arranged in a circle. A rotor is a thick disc with 26 terminals, arranged in a circle, on each face. Each terminal on one face is connected by a wire buried in the rotor to a terminal on the other face. When one rotor is placed in the machine, each terminal on the

front face contacts an entry terminal. The rear face of the rotor contacts the front face of a second rotor, which contacts similarly a third rotor. The rear face of the third rotor contacts a circle of 26 terminals forming what I shall call a reflector; these terminals are connected in pairs by wires.

B4 When a key is depressed the current goes through the plugboard to the entry terminals, through the first, second and third rotors, through the reflector, and back through the third, second and first rotors to the plugboard. From the front terminals of the plugboard wires go to 26 small light-bulbs each of which is labeled with a letter. The letter that lights up is the encode of the letter whose key is depressed.

B5 The current from key **x** enters the plugboard at front terminal **x**; if it goes through rotors, reflector and plugboard and back to terminal **y**, then current entering at terminal **y** would go through the same wires in the opposite direction to front plugboard terminal **x**. So if **x** encodes into **y** then **y** would encode into **x**: each encoding is a pairing, as described in part 1.

B6 The rotors can be removed from the machine and replaced in any order. There is a mechanism which makes the first rotor rotate through one twenty- sixth of a revolution each time a key is depressed. This moves each terminal of the rotor one place round the circle. The wires buried in the rotor are now in different positions relative to the entry terminals and so the machine produces a different encoding. At some point the movement of the first rotor makes the second rotor move on one place. Twenty-six moves later the second rotor moves again, and so on. The second rotor moves the third rotor in the same way. Anyone who wants to know what the machine actually looks like will find photographs in Gordon Welchman's *The Hut Six Story*.

B7 The first stage in reading the German messages is to reconstruct the wiring of the rotors and the reflector. On any one day the order in which the rotors were arranged in the machine and the connections in the plugboard remained unchanged. Choose four successive encodings, found as described in part 1, and hope that the second rotor does not move in the course of them. This will happen twenty-three times out of twenty-six, so the chances are good. The Polish cryptographers did find suitable encodings.

B8 Let the permutation produced by the plugboard be S. The rotors do not have letters engraved on the terminals, so let us imagine that the first rotor, in the position it occupies in the first of the four encodings, has engraved on each front terminal the letter of the entry terminal that contacts it, and has the same letter on the opposite rear terminal. Then the rotor produces a permutation of the alphabet, let us call it N. We similarly imagine letters on the terminals of the second and third rotors and the reflector. The second and third terminals and the reflector between them produce a permutation which I call Q.

B9 The permutation produced by going back through the first rotor is N^{-1}, the reverse of N. The permutation produced by going back through the plugboard is S^{-1} (which actually equals S). So the encoding is

$$\mathrm{SNQN}^{-1}\,\mathrm{S}^{-1}$$

For the next encoding the first rotor has moved one place. Let us see how this affects the permutation that the rotor produces. The entry terminals are arranged in alphabetical order, as the Polish cryptographers eventually guessed. (In the commercial Enigma they were arranged in German typewriter order **qwertzu . . .**) Let P be the permutation **(abc . . . xyz)**. The entry terminal after a letter α is αP. The rotor permutes it into αPN. When the rotor rotates one place it brings the terminal αP back one place to position α and the rear terminal αPN back to $\alpha\mathrm{PNP}^{-1}$. (When the Poles guessed that the entry terminals were arranged in alphabetical order they guessed that this order was in the opposite direction to the direction in which the rotors rotated.) The wire that connected αP to αPN now connects α to $\alpha\mathrm{PNP}^{-1}$, and the permutation now produced is PNP^{-1}.

B10 The reverse of a composite of permutations is the composite of the reverses in opposite order. That is, the reverse of XY is $Y^{-1}X^{-1}$. To see this, notice that if we compose XY with $Y^{-1}X^{-1}$ we obtain $XYY^{-1}X^{-1}$. But YY^{-1} is the permutation which leaves everything unchanged, so $XYY^{-1}X^{-1} = XX^{-1}$. (In fact, whenever a permutation and its inverse occur next to each other in a composite, they cancel each other.) So $XYY^{-1}X^{-1}$ leaves everything unchanged, which means that $Y^{-1}X^{-1}$ is the reverse of XY. If we abbreviate PP to P^2, PPP to P^3, and so on, and $P^{-1}P^{-1}$ to P^{-2} and so on, then the first three of the successive encodings, which I call A, B and C, are given by

$$\left.\begin{array}{lcl} A & = & SNQN^{-1}S^{-1} \\ B & = & SPNP^{-1}QPN^{-1}P^{-1}S^{-1} \\ C & = & SP^2NP^{-2}QP^2N^{-1}P^{-2}S^{-1} \end{array}\right\} \qquad (1)$$

We have to find the permutations S, N and Q which make these equations hold. At this time the cryptographers had a stroke of luck. The French Intelligence Service got hold of some operating instructions for the military Enigma which gave the plugboard connections for two months, and passed them on to the Poles.

B11 The permutation S is now known. If we set $U = S^{-1}AS$, then U is known and, from (1), $S^{-1}AS = NQN^{-1}$. If we set $V = P^{-1}S^{-1}BSP$ and $W = P^{-2}S^{-1}CSP^2$ then V and W are also known and

$$\left.\begin{array}{lcl} U & = & NQN^{-1} \\ V & = & NP^{-1}QPN^{-1} \\ W & = & NP^{-2}QP^2N^{-1} \end{array}\right\} \qquad (2)$$

Then

$$\begin{array}{lcl} UV & = & NQP^{-1}QPN^{-1} \\ VW & = & NP^{-1}QP^{-1}QP^2N^{-1} \end{array}$$

From the first of these, $QP^{-1}Q = N^{-1}UVNP^{-1}$; substituting this in the second gives

$$\begin{array}{lcl} VW & = & NP^{-1}N^{-1}UVNP^{-1}P^2N^{-1} \\ & = & NP^{-1}N^{-1}UVNPN^{-1} \end{array}$$

If we set $F = NPN^{-1}$ we have

$$VW = F^{-1}UVF$$

To find F, we use a well-known result in the theory of permutations: that to obtain $F^{-1}ZF$ from Z we write Z in cycles and replace each letter by the letter into which F permutes it. This gives the cycles of $F^{-1}ZF$. The reason is that two successive letters in a cycle of Z are α and αZ. We replace them by αF and αZF. The permutation that sends αF to αZF is $F^{-1}ZF$, because $\alpha FF^{-1}ZF = \alpha ZF$.

B12 Conversely, given permutations Y and Z, what permutations F will make $Y = F^{-1}ZF$? Unless Y and Z have the same number of cycles of each length no F will do. If they have, write the cycles of Y under the cycles of Z in all possible ways. Each way gives a solution. For example, if Z = **(ad)(bec)** and Y = **(be)(adc)** there are six possible arrangements. Two are

$$\begin{array}{ccc} (\mathbf{ad})(\mathbf{bec}) & & (\mathbf{ad})(\mathbf{bec}) \\ & \text{and} & \\ (\mathbf{be})(\mathbf{cad}) & & (\mathbf{be})(\mathbf{adc}) \end{array}$$

giving F = (**abcde**) and F = (**ab**)(**c**)(**de**). Each of the other arrangements gives a solution.

B13 If UV and VW do not have the same number of cycles of each length, something is wrong — perhaps the second rotor moved — but if they do we find the possibilities for F as above and select those that consist of a single cycle, because $F = NPN^{-1}$ and P consists of a single cycle. If there is more than one possibility we use our fourth successive encoding. Treating the second third and fourth encodings in the way that we treated the first three, we find another set of possible F's. The one and only F in both sets of solutions is the F that we want. (Only by incredibly bad luck could there be more than one.)

B14 Knowing NPN^{-1} and P we can apply the same method to find N. There will be 26 possible solutions. It does not matter which one we choose: they will all give the same encodings. We can see this as follows.

B15 If N and O are two possible solutions, then $NPN^{-1} = OPO^{-1}$, which means that $O^{-1}NPN^{-1}O = P$. So $N^{-1}O$ applied to the cycle of P must reproduce the cycle of P. The only way that this can happen is for $N^{-1}O$ to push each letter forward through the same number of places in the alphabet. Then $N^{-1}O = P^n$ for some number n, and $O = NP^n$. So if the actual rotor gives permutation N, the twenty-six solutions are NP^n for n from 0 to 25.

B16 If we take a rotor which gives the permutation N and twist the rear face through n twenty-sixths of a revolution while holding the front face still, the wire which led from α to αN now leads to αNP^n, so the permutation becomes NP^n. The twenty-six solutions are the actual rotor twisted in this way for n from 0 to 25. But if we twist the rear face of the rotor through n places we have only to twist the rest of the machine through n places to re-establish the connections and produce the same encoding. The same method applied to the second month for which the plugboard connections were known, when another rotor was in first place in the machine, gives the wiring of this rotor.

B17 For the next step there is no systematic method like the one I have been describing: cryptography is sometimes more an art than a science. The captured operating instructions included a sample message and its encoding at a stated setting of the rotors. Using these instructions and attempts to decode intercepted messages, the cryptographers, by making trials and adjustments, found the correct solutions for each of the first two rotors and the wiring of the third rotor and the reflector. They could now build a copy of the machine. To follow their next steps we need to know how the rotors were set in position.

B18 Each rotor has around its rim the twenty-six letters of the alphabet in order. These are not fixed to the rotor, but are on a thin metal ring that can slide round the rotor. When the rotor is put in the machine and the lid is closed one of the letters shows through a small window. The lug on each rotor that makes the next rotor turn is fixed to the ring, not to the rotor itself. The operating instructions tell the operator which rotor goes in which place in the machine, which letter on the ring is set against a zero mark on the rotor (one for each rotor) and which terminals on the plugboard are connected. It also specifies a trio of letters which I shall call the basic trio.

B19 The operator sets up his machine. To send a message he chooses a trio of letters which I shall call the message setting. He turns the rotors until the basic trio is showing, encodes the message setting twice, turns the rotors until the message setting is showing, and encodes the message. The recipient turns his rotors until the basic trio is showing, decodes the first six letters of the message (because Enigma encodings are pairings, encoding and decoding on the Enigma are the same) giving him a trio of letters repeated, turns his rotors until these letters are showing, and decodes the rest of the message. The ring setting and message setting between them determine the position of the rotor. Conversely, if we can discover the position of the rotor we can deduce its ring setting from the message setting found as described in §A.

B20 To find which rotor is in first place in the machine on any one day we look for a pair of messages in which the last two letters of the message setting are the same, for example **epz** and **kpz**. **k** is six places after **e** in the alphabet, so the second message setting is either 6 places after the first in the sequence of encodings or 20 places before it, according to when the first-place rotor makes the second rotor turn. If it turns between **e** and **k**, say at **h**, we have the sequence

kpz . . . **zpz a***pz* . . . **epz fpz gpz hpz cqz jqz kqz** . . .

and **kpz** comes before **epz**. If it turns before **e** or after **k** we have

. . . **epz fpz gpz hpz ipk jpz kpz** . . .

and **kpz** comes after **epz**. We write the second message under the first six places later, for example

```
p x l o w n z a b u v l . . .
            x a k u v h . . .
```

If **kpz** comes after **epz**, two letters in the same column will be encoded by the same permutation, so if they are the same they will both represent the same letter in the original message. It is a property of the German language that if we write one message under another, about one time in 12, on average, two letters in the same column will be the same. But if **kpz** comes before **epz** two letters in the same column will be encoded by different permutations and will be the same only one time in 26 on average. With long enough messages we can distinguish between the two, and we can confirm our result by writing the second message 20 places ahead of the first. We can now tell whether or not the turn takes place between **e** and **k**. If it doesn't, and for one of the rotors it does, we can rule that rotor out as first-place rotor. On any one day there will be several pairs of messages that we can use and we can usually rule out two of the three rotors.

B21 Except for the days on which the intelligence service found the plugboard connections the cryptographers did not know the permutation S. But they did know that there were only six wires for the plugboard, so that fourteen letters were left unchanged. We look at six successive encodings found as described in part 1 on a day on which we have found which rotor is in first place. We hope that the second rotor has not moved in the course of them. Let the permutation produced by the second and third rotors be Q. We know the permutation N produced by the first rotor in a certain position, but we do not know its position here. For the first of the six successive encodings it will produce a permutation $\mathrm{P}^x\mathrm{NP}^{-x}$ for some x. For the second encoding it will produce the permutation $\mathrm{P}^{x+1}\mathrm{NP}^{-x-1}$ and so on. If the encodings are A, B, C . . . F, then

$$\mathrm{A} = \mathrm{SP}^x\mathrm{NP}^{-x}\mathrm{QP}^x\mathrm{N}^{-1}\mathrm{P}^{-x}\mathrm{S}^{-1}$$

If S left all letters unchanged we would have

$$\mathrm{Q} = \mathrm{P}^x\mathrm{N}^{-1}\mathrm{P}^{-x}\mathrm{AP}^x\mathrm{NP}^{-x}$$

and similarly

$$\mathrm{Q} = \mathrm{P}^{x+1}\mathrm{N}^{-1}\mathrm{P}^{-x-1}\mathrm{BP}^{x+1}\mathrm{NP}^{-x-1}$$

and four other equations involving C, D, E and F. We compute the right-hand sides of each of these equations for each value of x from 0 to 25, and for one value of x all six of them would be the same: they would all be Q. Because S does not leave all letters unchanged this will not happen.

B22 Let us see what does happen. Call the permutation produced by the first-place rotor in the first of the six successive positions N_1. Then A = $\mathrm{SN}_1\mathrm{QN}_1^{-1}\mathrm{S}$. (Remember that S^{-1} = S.) The first of our right-hand sides is $\mathrm{N}_1^{-1}\mathrm{AN}_1$, i.e.

$$\mathrm{N}_1^{-1}\mathrm{SN}_1\mathrm{QN}_1^{-1}\mathrm{SN}_1$$

The cycles of $\mathrm{N}_1^{-1}\mathrm{SN}_1$, like those of S, are six pairs and fourteen singletons. The cycles of Q are thirteen pairs. Therefore at least one pair of Q must be made up of two of the singletons of $\mathrm{N}_1^{-1}\mathrm{SN}_1$, because the twelve letters in the six pairs of $\mathrm{N}_1^{-1}\mathrm{SN}_1$ cannot occur in more than twelve of the pairs of Q. At the other extreme, if these twelve letters between them form six of the pairs of Q, there will be seven pairs of Q made up of singletons of $\mathrm{N}_1^{-1}\mathrm{SN}_1$. On average there will be three or four. Let $\alpha\beta$ be such a pair. Then

$$\begin{aligned}\alpha\mathrm{N}_1^{-1}\mathrm{SN}_1\mathrm{QN}_1^{-1}\mathrm{SN}_1 &= \alpha\mathrm{QN}_1^{-1}\mathrm{SN}_1 && \text{because } \alpha \text{ is a singleton of } \mathrm{N}_1^{-1}\mathrm{SN}_1\\ &= \beta\mathrm{N}_1^{-1}\mathrm{SN}_1 && \text{because } \alpha\beta \text{ is a pair of Q}\\ &= \beta && \text{because } \beta \text{ is a singleton of } \mathrm{N}_1^{-1}\mathrm{SN}_1\end{aligned}$$

B23 So the pair of Q is also a pair of our first right-hand side. The same applies to the other five right-hand sides. So when we have found the right x a number of pairs of Q will be repeated in the right-hand sides. Conversely, one of the twenty-six sextets of right-hand sides will show repeated cycles, and this will be the one with the right value of x. This value of x gives us N_1, because $\mathrm{N}_1 = \mathrm{P}^x\mathrm{NP}^{-x}$. In an example that I made up for myself, twenty-five sextets showed no particular pattern, but one showed five occurrences of one pair, four of another, and three each of three others. (Three more pairs occurred twice, but these could be coincidences, so I ignored them.) The five multiply-repeated pairs are five of the pairs of Q.

B24 Three of the multiply-repeated pairs occurred in the first right-hand side, so each of the six letters involved is a singleton in $\mathrm{N}_1^{-1}\mathrm{SN}_1$. But if α is a singleton in $\mathrm{N}_1^{-1}\mathrm{SN}_1$, then $\alpha\mathrm{N}_1^{-1}\mathrm{SN}_1 = \alpha$, so $\alpha\mathrm{N}_1^{-1}\mathrm{S} = \alpha\mathrm{N}_1^{-1}$, and $\alpha\mathrm{N}_1^{-1}$ is a singleton in S. This gave me six of the singletons of S. Using the other five right-hand sides I found seven more of the singletons. Having found some of the pairs of Q and most of the singletons of S, we look for a pair of Q, call it $\alpha\beta$, for which $\alpha\mathrm{N}^{-1}$ is a singleton in S. Because A = $\mathrm{S}^{-1}\mathrm{N}_1\mathrm{QN}_1^{-1}\mathrm{S}$; applying $\mathrm{N}_1^{-1}\mathrm{S}$ to the cycles of Q gives the cycles of A. $\mathrm{N}_1^{-1}\mathrm{S}$ transforms α into $\alpha\mathrm{N}_1^{-1}\mathrm{S}$, which is $\alpha\mathrm{N}_1^{-1}$, because it is a singleton in S. We find the cycle of A that contains $\alpha\mathrm{N}_1^{-1}$; this must be the cycle that $\alpha\beta$ transforms into. Let the other letter in this cycle be γ. Then $\beta\mathrm{N}^{-1}\mathrm{S}=\gamma$, so γ and $\beta\mathrm{N}^{-1}$ form a pair in S. In this way (using all the right-hand sides) we find some of the pairs of S.

B25 If we have not found all the pairs of S we may have to use some ingenuity, varying from case to case, to complete the solution. In my example, I found four pairs and thirteen singletons of S, using between them all letters except **e**, **f**, **h**, **i** and **k**. Then S must transform each of these five letters into one of the five. I also knew that **ab** is a pair of Q and **a**N^{-1} = **e** and **b**N^{-1} = **f**. Then N^{-1}S transforms **ab** into **eSfS**. The only pair of A that is made up of the five relevant letters is **fh**. So **eSfS** must be **fh**. We cannot have **eS** = **f** and **fS** = **h**, because S = S^{-1}, so we must have **eS** = **h** and **fS** = **f**. So **f** is the remaining singleton, and **eh** is one of the two remaining pairs. The other pair can only be **ik**. In this way (or similar ways) S is found.

B26 Just before the war the Germans changed the method of encoding the message settings, introduced more rotors, and increased the number of plugboard connections. The cryptographers had to find new methods. These methods were later used by British cryptographers and are so well described by Gordon Welchman in *The Hut Six Story* that I need not describe them here.

C Italy and Japan

C1 The Enigma was not the only machine used for encoding by the axis powers. Another was the Hagelin, used by the Italian navy. Like the Enigma, it was based on a commercial machine. The Hagelin works by means of five wheels. Each time a letter is encoded, every wheel makes a fraction of a revolution; one wheel makes a complete revolution every 13 moves, one every 11 moves, the others every 9, 8 and 7 moves. I don't guarantee these particular figures — I am working from a forty-year old memory — but the general idea is that it will be a long time before the wheels repeat their alignment. If my figures are correct, this happens only after $13 \times 11 \times 9 \times 8 \times 7$ moves.

C2 The first wheel is fitted with 13 lugs, the second with 11, and so on. When the machine is set up, each lug could be set to be either active or inactive. When an active lug is in the operative position, the wheel pushes the letter being encoded a certain number of places forward in the alphabet. The number for each wheel is determined when the machine is set up. There is also a number, which we called the *slide,* which could be anywhere from 0 to 25, and was always operative. For example, the numbers allotted to the wheels might be 5, 9, 6, 7, 3 respectively and the slide might be 8. Suppose that when a letter was encoded only the first and fourth wheels had active lugs in the operative position. Then the letter would be pushed forward 5 + 7 + 8 places in the alphabet, so that **a** would encode into the 21st letter, **u**. Then the wheels move and a different set of lugs are in operative position for the next encoding.

C3 I don't know how the initial break was made. When I started to work on this code we had identified the machine, we knew that the set-up changed every month, and we had decoded a good many messages. The Italian navy did something that any reasonably competent cryptographer would regard as completely moronic: every message from the admiralty to the submarine command started *Da supermarina ad maricosom* and every message in the reverse direction started *Da maricosom ad supermarina.* (They used the Latin *ad* instead of the Italian *a* for "to". They also used *et* instead of *e* for "and".)

C4 Each month we had to discover the positions of the lugs, the slide, and the allocation of the numbers 3, 5, 6, 7, 9 to the wheels. (Again, I don't guarantee these particular numbers.) At midnight on the last day of each month two young but expert cryptographers came on duty, and well before the end of their shift, at 8 a.m., they had always found the new settings. They compared the standard addresses with the beginnings of intercepted messages. If the first letter of an intercepted message is **s**, then the first letter of the message, namely **d**, has been pushed forward 15 places. If the slide is 2 the wheels must have pushed the letter forward 13 places, and this can be made up from separate pushes 3, 5, 6, 7, 9 in only one way, namely 6 + 7. So whichever wheels had 6 and 7 allotted to them had active lugs in the operative position, and the others had inactive lugs there.

C5 The stereotyped beginnings were twenty-four letters long. The wheel that rotated once every 7 moves had the same lug in the operative position for the 1st, 8th, 15th and 22nd letters. If the pushes due to the wheels were respectively 13, 16, 6, 14, the possible combinations would be

$$\begin{gathered} 6+7 \\ 3+6+7 \text{ or } 7+9 \\ 6 \\ 3+5+6 \text{ or } 5+9 \end{gathered}$$

Then the push for the wheel cannot be 5, because that would mean that the lug is inactive in the first three but active in the fourth. Nor can it be 7. So it must be 3 (inactive), 6 (active) or 9 (inactive). Now we apply the same reasoning to the 2nd, 9th, 16th and 23rd letters, to see if we can eliminate any of these three. And so on. If we eliminate them all, we have the wrong slide, and start again with another. In practice, a wrong slide was eliminated quite

quickly. In this way, we find the push allotted to this wheel and the positions of the active lugs on it, and we find the slide. The rest is easy.

C6 Besides machine codes, manual codes were extensively used in the second world war. In the course of history many ingenious codes have been devised, and the interested reader can find them described in David Kahn's *The Codebreakers.* But not only is ingenuity no guarantee of security — the codebreaker may be more ingenious than the codemaker — but the more ingenious a code is, the harder it is to use. For large-scale use a code has to be straightforward enough for an ordinary signals clerk (who may be, especially in war-time, someone who is not particularly intelligent or well-educated) to use with the least possible risk of making a mistake. Mistakes in encoding are a godsend to the cryptographer. And the code has to be such that a mistake in transmission, or a word unread because of static or a weak signal, does not make the whole message unintelligible. As a result, by the time I started code-breaking almost, if not quite, all large-scale manual codes were of one type: the reciphered code-book.

C7 Early code-books were compiled by making a dictionary containing all the words likely to be used and numbering them, say from 00000 to 99999 (leaving gaps if there were fewer than 100,000 words). In order to deal with proper names and words that are not included, the book would contain code-groups for letters and syllables that can be used to spell them out. Messages sent in such a code are quite easy to decode, given enough material. Things are more difficult with what we called a "hat book". This is one in which the words are numbered in random order, not alphabetical order, as though they were thrown into a hat and withdrawn at random. Even this, surprisingly perhaps, is not secure. So the code-groups are reciphered. The simplest method of reciphering is by non-carrying addition. The encoder is provided with the code-book and with a key, consisting of groups of five digits, compiled as randomly as possible. If a code-group is, say, 92482 and the key-group is 79043 we add the digits of the key-group to the digits of the code-group separately, ignoring any tens digits that occur:

key-group	79043
code-group	<u>92482</u>
encoded group	61425

$$(7 + 9 = [1]6,\ 9 + 2 = [1]1 \text{ etc.})$$

C8 To avoid confusion with ordinary addition, non-carrying addition is performed from left to right. Eventually it was found better to subtract the code-group from the key-group; this makes decoding the same as encoding. This type of code is called a subtractor.

C9 Early on the key might consist of perhaps 20 groups, which would be repeated as often as required if a message were more than 20 groups long, but by the time I started a typical key would have several hundred or even several thousand groups. Each message would have to contain a group, which we called an *indicator*, telling the recipient whereabouts on the key encoding started, The longer the key the more secure the code. The extreme case is where it is possible to distribute so much key that no portion of it need be used more than once. One way to ensure this is to print the key on a perforated note-pad and instruct the operators to tear off each page once it has been used. We called this a *pad subtractor.* It is guaranteed secure. It is practical only for very limited amounts of traffic. We used it for messages containing information obtained by breaking enemy codes.

C10 Breaking subtractor codes depends on the fact that some words occur much more frequently than others. For example, on page 20 of *DIO* 3, which I happen to have open, there are about 350 words. One word (refraction) occurs 5 times, four words occur 3 times each, and seven words occur twice each. This is a far cry from the number of repeats that there would be if each of the several thousand words in one's working vocabulary occurred equally often. And military or naval prose is at least as stereotyped as astronomical prose.

C11 So we look for repeats. If, in a fair amount of traffic, two groups repeat, the chance that this has happened by accident is remote, there being one hundred thousand possible groups. The chance that it is the same code-group recovered by the same key-group is much higher. So we write one message under the other. We then look for other repeats and eventually, with a reasonable amount of traffic, we will have several messages overlapping. The number of messages that overlap is called the *depth*. We found that with a depth of seven or more we could usually start key-breaking.

C12 The users of a code aimed to distribute so much key that the enemy cryptographers wouldn't find much depth. This is often difficult and sometimes impossible. We knew, for instance, from the amount of traffic on our BAMS (British and allied merchant shipping) code, used for sending messages to convoys, that the Germans must surely be reading much of it most of the time.

C13 Suppose that six successive columns of four messages are as follows:

	1	2	3	4	5	6	
message A . . .	01384	92017	85318	79262	41047	33881	. . .
message B . . .	10789	92017	89064	30417	41047	19872	. . .
message C . . .	10789	13394	41037	30635	09910	30564	. . .
message D . . .	31480	10077	81032	12491	62324	07988	. . .

Messages A and B have been aligned by the repeats in columns 2 and 5. (The double repeat makes it practically certain that these are no coincidence.) Message C has been aligned by the repeat in column 1. Message D has been aligned by a repeat in some earlier or later column.

C14 The three repeated groups might derive from three different high-frequency code-groups. But two of them (or even all three) might derive from the same code-group. This happens often enough to be worth trying. So let's try it.

C15 Assume that the same code-group occurs in columns 2 and 5, messages A and B, and assume, quite arbitrarily, that it is 00000. (If it is actually xyzuv then every code-group and every key-group that we find will be xyzuv less than its actual value.) Then the key-groups for columns 2 and 5 are 92017 and 41047, so the code-group enciphered at C2, for instance is 89723:

key-group	92017
enciphered group	<u>13394</u>
code-group	89723

In fact, we have for the code-groups in the two columns

	2	5
message A	00000	00000
message B	00000	00000
message C	89723	42137
message D	82040	89723

The code-group 89723 has repeated. This confirms our guess. So we test all pairs of repeats in this way.

C16 The next step used the fact that the difference between the results of enciphering two code-groups by the same key-group is the same as the difference between the code-groups themselves. So we computed (using punched-card machinery) the differences between every pair of groups in the same column. We looked for repeats. For example, in column 2 the difference between groups C and D is 03327. In column 6 the difference between groups A and C is also 03327. So we see what happens if A6 = C2 = 89723 and C6 = D2 = 82040. For this to happen the key-group for column 6 must be 12504, giving

key	12504
A	89723
B	03732
C	82040
D	15626

C17 If we started with a depth of 7 there are three more groups in this column and in each of columns 2 and 5, which I have not displayed here. If any of these repeat we are on the right track. If not, this might be a dead end. We try all the repeats in this way. As I said, with a depth of 7 or more (and twenty or thirty columns) we could probably find all the key-groups, ending up with the equivalent of an unreciphered code-book. For this type of code the code-breakers were divided into two groups: the key-breakers — I was one — who did what I have just been describing, and the book-builders, who identified the code-groups once they had been stripped of their recipherment. The key-breakers were mostly mathematicians or chess-players, the book-builders were mostly linguists. They were not necessarily students of Italian, German and Japanese; the organizers took the viewpoint that if you can learn one language you can learn another, and in fact quite a lot of our book-builders were classicists.

C18 I don't know all the details of how code-books were built up, but I do know that the first code-group to be identified was usually the one for full-stop. It occurred frequently, with consecutive occurrences never very close together and never very far apart. Groups for "from" and "to" and addresses were also found fairly easily.

C19 When some code-groups had been identified we could use them to help in key-breaking. For example, if we suspected in a partly-decoded message that a certain group represented a number, we could try all the groups identified as numbers in that place, working out the key-group in each case and seeing what it brought up in the rest of the column.

C20 When the Italians became our co-belligerents, those of us who were working on Italian codes switched to Japanese. The code that I worked on was the naval attaché code. The code-book was ingenious. For each frequently-occurring word it had several alternatives; I seem to remember that it had six groups for full-stop. It also had groups which meant "message begins". The encoder placed one of these at the start of the message, divided the message in two, and sent the second part first. If the Italians had done this we would have found their machine code much more difficult, if not impossible, to break.

C21 The code-book did not have separate groups for spelling proper names. Instead, many groups had two meanings. For example, 41803 might normally mean "battleship", but after a group that meant "spelling starts" it might represent the syllable ENT. (The Japanese used roman letters for spelling.) Each succeeding group would have its alternative meaning until there came a group that meant "spelling ends". There were also code-groups that meant "next two groups are spellers" and "next three groups are spellers". By contrast with these clever devices the compiler of the code-book made a stupid mistake. He or she needed only 50,000 groups and chose them, not at random, but by having one of the first two digits of each code-group even and the other odd. This means that if we add the first two digits of each enciphered code-group together, code-groups enciphered by the same key-group are either all odd or all even. When two messages are arranged in depth they show the same sequence of odds and evens. Conversely, if two messages show the same sequences of twelve or more odds and evens, the chances are that they are using the same stretch of key. This gave us a useful method of getting more depth. It also enabled us to throw out some messages that we had set by repeats that were actually coincidences. Finally we discovered the indicator system. The last group but one in a message gave a position in the key (by page, row and column). The key-group there was added to the second group of the message to give the position in the key where the encoding started.

C22 The most interesting messages in this code were from the naval attaché in Madrid. Nearly all of them came from a spy in Algeciras giving the number of warships in harbour in Gibraltar. They presumably passed the information on to the Germans. (Just as our submarine tracking room kept a tag on their submarines, so their intelligence presumably kept a tag on our warships.) We could tell from these decodes how accurate their information was and how well our deceptive measures were working.

C23 These messages were very stereotyped. They enabled us to achieve a real *tour de force:* solving on a depth of 2. This is the minimum possible depth; a depth of 1 is both practically and theoretically impossible, being equivalent to a pad subtractor. A typical Madrid message might be:
message begins / Madrid naval attaché / from / X / to / Gibraltar harbour / in / battleship(s) / 1 / aircraft carrier(s) / 0 / cruisers / 2 / destroyers / 4 / submarines / 3 / small / ships / l0.

C24 In Japanese, words corresponding to prepositions like "from", "to" and "in" come after nouns; in fact, we called them postpositions. X is the addressee; I forget who it was. Two Madrid messages overlapped. Running "message begins" groups through one of them threw up "aircraft carrier" in the other. Trying "Madrid naval attaché" in the next column of the first message gave a small whole number in the second. And so on. After a while we were running types of warship through one message hoping to pick up numbers or types of warship in the other, and we finished by decoding both messages.

C25 Although the Japanese have a perfectly good syllabary (called hiragana) in which they can write their language, they prefer to use Chinese characters, reserving hiragana for postpositions, verb-endings, and similar items. (If you have travelled in Japan you have probably noticed that the names of railway stations and tram stops are written three times: once in characters, once in hiragana, and once in roman letters.) This posed a problem for us. The code-groups mostly represented Chinese characters, and it was essential to identify the character, not just its translation. This is standard practice in cryptography. For example, if the Germans were building up an English code-book and had identified the group for "order" they could not replace it by a German translation, because it might mean *Ordnung* ("in good order"), *Befehl* ("that's an order!"), or *Reihenfolge* ("alphabetical order"), and there are probably other meanings. We did not want to draw the Chinese characters. Most of us probably couldn't have made a decent job of it, and we would have had no practical way of indexing them. Anyone who has used a Japanese (or Chinese) character dictionary will know that it is slow and awkward.

C26 Every Chinese character used in Japanese (even those with only one meaning) has at least two entirely different pronunciations: the Japanese word that it represents and the Chinese word from which it is taken (or a rough approximation thereto). For example, the Japanese for "three" is *mitsu* and the Chinese is *san*. So in Japanese the character for "three" (three horizontal lines) is pronounced either *san* or *mitsu.* (This character can occur in proper names (is the Irish surname Twomey a fair analogy?) and is the first character of the name of the engineering firm Mitsubishi. It is also the first character of the name of the electronics firm Sanyo.)

C27 Again, the Japanese for "mountain" is *yama* and the Chinese is *shan*. So the character for mountain (a stylised diagram of three peaks) is pronounced *yama* or *san* in Japanese. That is why the mountain usually called Fujiyama is sometimes called Fuji san: it is just a different way of pronouncing the same character. Our solution was to identify each character by using both pronunciations. Some characters have more than two pronunciations; for them we chose two. So to us "three" was *san, mitsu;* "mountain" was *san, yama*; "north" was *hoku, kita,* and so on. I am using here the more-or-less phonetic (to an English speaker) spelling introduced by missionaries and used by Westerners. We actually used the spelling used by the Japanese themselves: *mitubisi* instead of *mitsubishi,* for example. Amusingly enough, when the Japanese introduced their system just before the war, *Jane's Fighting Ships* reported "the Japanese have renamed many of their warships".

C28 Even when we had pinned down a character our troubles were not over. Any one character can have several meanings, often quite different. We used to joke that every character had five meanings in Japanese: the one you want, the one you don't want, a religious one, an obscene one, and an obscure one. The dictionary that we used was made in Japan and its English was not always perfect. In particular, it translated one Japanese word as "a revolting lantern".

C29 I am not going to take up the question of how valuable our code-breaking was. A great deal has been written on this topic and opinions vary from rating it as a small but useful contribution, to the assessment of a Polish writer who described the theorem that the composite of two pairings has an even number of cycles of each length (which made possible the initial break into Enigma) as "the theorem that won the war".