Modern Breaking of Enigma Ciphertexts

Olaf Ostwald and Frode Weierud

Abstract "Breaking German Army Ciphers" is the title of a *Cryptologia* article from 2005, describing the lucky survival of several hundred authentic Enigma messages of World War II, and an account of a ciphertext-only cryptanalysis of a large number of these messages, leaving only a few, mostly short messages, unbroken. After reviewing the work done, and investigating the reasons for both lucky breaks and close misses, the modern ciphertext-only attack on Enigma messages is improved, especially on genuine ones with short lengths and/or many garbles. The difficulties of a proper measure for the candidate's closeness to a plaintext are clarified. The influence on the decryption process of an empty plugboard and one with only a few correct plugs is examined. The method is extended by a partial exhaustion of the plugboard combined with an optimized hillclimbing strategy. The newly designed software succeeds in breaking formerly unbroken messages.

Keywords Ciphertext-only, crib, cryptanalysis, Enigma, German Army, index of coincidence, plugboard, radio message, *Stecker*, trigram statistics, unicity distance, unsolved ciphers

Contact Frode Weierud. Email: frode.weierud@gmail.com. Address: Bjerkealleen 17, 1385 Asker, Norway.

1. Introduction

The *Cryptologia* article "Breaking German Army Ciphers" [18] begins with the words "This is the first report of an on-going cryptanalytical project." Indeed, the project carried on and the present article describes further advances and new results on breaking genuine Enigma ciphertexts. In the last ten or more years the work has been continued, studying the historical facts, investigating the characteristics of approximately 500 authentic Enigma messages, and designing several cryptologic software tools, in order to get a better understanding of the statistics of the Enigma ciphertexts and the techniques for their cryptanalysis. The aim was to improve the ciphertext-only attack, in order to eventually break formerly unbroken messages. As the work is still in progress, this article could be described as the second report of an on-going cryptanalytical project.

2. Historical Background

The electro-mechanical cipher machine Enigma (from Greek αίνιγμα for "riddle") was the backbone of the German armed forces' (*Wehrmacht*) cipher systems during World War II. It is operated like a typewriter, entering the plaintext via a keyboard. Each letter of the plaintext is enciphered individually [12, p. 1]. By pressing a letter key a switch is closed and current from an internal battery flows over the closed contact through the plugboard (*Steckerbrett*) into the cryptographically important "scrambler," which is formed by Enigma's rotor set (*Walzensatz*). Here the letter is permuted several times by three rotating wheels (*Walzen*). The current reaches the reflector (*Umkehrwalze* in German, abbreviated *UKW*), which is situated at the leftmost side of the rotor set. It feeds the current back through the three rotating wheels, which then passes the plugboard a second time. Finally the current reaches the lampboard and lights up a lamp. The illuminated lamp indicates the cipher letter that corresponds to the entered plaintext letter.

Enciphered messages were wirelessly transmitted in Morse code and subsequently deciphered by the intended receiver with the aid of the known secret key, which was changed daily

Postprint: The Version of Record of this manuscript has been published and is available in Cryptologia, January 2017, http://dx.doi.org/10.1080/01611194.2016.1238423

(*Tagesschlüssel*). The German Army Enigma regulations laid down that no transmitted ciphertext message was to exceed the length of 250 characters [14, p. 5]. The German key sheets (*Schlüsseltafeln*) determined three elements of the daily key, namely wheel order (*Walzenlage*) – i.e. arranging three wheels out of a set of five, ring setting (*Ringstellung*), and plugboard setting (*Steckerverbindungen*). The individually chosen start position of the three wheels (*Walzenstellung*) for each single message was the fourth element of the key.

The Enigma key space is the product of these four factors. While there are 5×4×3 or 60 possible wheel orders, out of the 26×26×26 wheel positions 26×26 are redundant, leaving 26×25×26 or 16,900 as relevant [15, p. 80]. Nevertheless, for the sake of convenience, all 26^3 wheel positions are considered here, as this eases the design of the software. While the ring setting of the left-hand wheel is completely irrelevant, and does not enhance the key space, 26×26 or 676 possible ring settings for the middle and right-hand wheels are possible. However, especially for short messages mostly no stepping of the left-hand wheel occurs. Hence the ring setting of the middle wheel is irrelevant and only the remaining 26 possible ring settings of the right-hand wheel have to be considered. In total, the first three factors of Enigma's key space give a product of $60 \times 26^3 \times 26$ or 27,418,560 "locations," as they shall be designated here.

Finally, as the fourth factor, the plugboard with its vast possibilities of connection options plays an important role, as it creates the lion's share of the magnitude of the key space. The German key sheets during the war generally specified ten *Stecker* cables (jumper cables), thus swapping 20 letters of the alphabet, while leaving the remaining six letters unswapped or "self-steckered." In total, for ten *Stecker* cables there exist 150,738,274,937,250 (more than 150 trillion) different ways of plugging (e.g., [2, p. 254]). Thus, the fourth factor of Enigma's key space is more than five million times as much as the above stated number of the locations.

In the same manner as for enciphering, each Enigma machine can also be used for deciphering. For that, the ciphertext is simply entered via the keyboard, and the lamps now indicate the corresponding plaintext [9]. During the war, the wireless German messages were intercepted by the British Y service, which sent them to the Government Code and Cypher School at Bletchley Park (BP). There, it was the task of the codebreakers of Hut Six, the organizational unit dealing with the cryptanalysis of Enigma messages originating from the German Army and German Air Force, to determine the keys and to recover the plaintexts. The technique used at BP was based on "cribs," i.e. known plaintext fragments [19, p. 78]. It was sufficient to break a single message a day for a specific net. After that, with the then recovered daily key for that net, it was easy to read all the others, often hundreds of messages of the same day. With the known *Tagesschlüssel* they could easily be deciphered in the same way as the intended German receiver did it.

3. Ciphertext-Only Cryptanalysis of Enigma

During World War II the German radio messages were broken with the aid of electromechanical devices such as the *Turing-Welchman-Bombe* [4] and a known-plaintext attack. In contrast to the historical techniques, nowadays a ciphertext-only attack becomes feasible with the aid of modern programmable computers, which were not available to the codebreakers at BP. In 1995, James Gillogly described in his *Cryptologia* article [8] a ciphertext-only attack based on an exhaustive search for all 60 wheel orders, 676 ring settings, and 17,576 wheel starting positions and a hillclimbing technique for evaluating the correct plugboard settings. The latter cannot be recovered exhaustively because of the vastness of the corresponding key space. But, as the plugboard is constant and does not vary during the enciphering (and deciphering) process, it may be treated similar to a monoalphabetic substitution and can be attacked by hillclimbing.

Gillogly first exhausted all possible 60 wheel orders and 26^3 or 17,576 wheel starting positions. For that, he left the ring settings constant, e.g. every ring at "A" or "01." Furthermore, he left the plugboard empty, which is a good first approximation of the correct plugboard, because it already holds the six self-steckered plugs. He deciphered the ciphertext with this partial key and investigated the resulting candidate texts. The advantage of his approach is to reduce the search space for exhaustion. It now comprises "only" 60×26^3 or 1,054,560 cases, which exactly matches the number of cases the *Bombes* at BP treated. For each of the candidate texts he determined the likeliness of being close to a plaintext by calculating the Index of Coincidence (IC) – this brilliant measure for general cryptanalysis invented by William Friedman as early as 1920 [7]. After that, Gillogly used some of the best key candidates for now optimizing the ring settings of the middle and right-hand rings, out of 26×26 or 676 different possibilities, via a second exhaustion phase. Finally, in a third phase, the plugboard settings were found via hillclimbing. His ingenious technique works by using trial plugs and leaving them steckered if the IC of the new candidate text increases. Otherwise the trial plug is removed and another is tested. For this last phase of the decryption process Gillogly switched from the IC to a trigram count as a new measure for the plaintext quality, as it was here more efficient than the IC. His technique proved efficient for Enigma messages with a length of about 300 letters and with keys using up to six plugs as well as for longer messages, then with up to ten plugs. His only success with ten plugs was for a message with a length of 1463 letters [8, p. 411].

In 2000 Heidi Williams, referencing Gillogly's work, made further investigations on alternative criteria for recognizing the plaintext quality of a candidate text and found the Sinkov statistic most useful [20, p. 16]. She demonstrated the successful decryption of a non-authentic English 500-letter message enciphered by an Enigma with ten plugs, leaving the six most frequent single characters in English texts, E, T, A, O, I, and N, self-steckered [20, p. 7], and claimed her success as an improvement on Gillogly's technique.

In 2005 a ciphertext-only attack on authentic Enigma ciphertexts, enciphered on the three-wheel, steckered *Wehrmacht* Enigma in 1941 and 1945, was published by Geoff Sullivan and Frode Weierud [18]. The ciphertexts from 1941 originate from the first phase of Operation Barbarossa, the German invasion of the Soviet Union, which began on 22 June 1941. More precisely, they are part of the secret communications of Army Group North (*Heeresgruppe Nord*, abbreviated *HG Nord*) in the first four months (22 June to 22 October 1941) of its advance through the Baltic countries toward Demyansk and Leningrad. They contain tactical information and correspondence, for instance between the SS Death's Head Tank Division (*SS-Totenkopf-Panzerdivision*, abbreviated *SST-Div*) and their commanding office of Tank Group 4 (*Panzergruppe 4*). The latter formed the highly mobile armored core troops of *HG Nord*, besides the 16th and the 18th Army, which mainly consisted of infantry divisions.

Based on the proposal of Gillogly, an improved ciphertext-only method was devised in order to break genuine messages with the authorized message limit of not more than 250 letters and the actually used number of ten plugs. For that, all 26^4 or 456,976 different possibilities for the starting positions of the three wheels, and the ring setting of the right-hand wheel, together with the 60 wheel orders, were exhausted, altogether a workspace of the said 60×26^4 or 27,418,560 locations. For finding the correct plugboard connections, after recognizing that the success rate of Gillogly's original method strongly depends on the self-steckered plugs, an innovative three-pass hillclimbing technique was utilized. The IC was used for finding the first few (e.g. four) *Steckers*. Then it was switched to bigram scoring, and, during the final phase, trigrams were used. Furthermore, it proved essential to use a well-suited text base. For that, common German plaintext is not the best solution. The *Wehrmacht* used special terminology, unusual abbreviations and some conventions, which are not used in common German, such as representing the frequent bigrams "ch" and "ck" by the single letter Q.

With the significantly improved ciphertext-only method, the first break of an authentic message of *HG Nord* was achieved on 17 March 2003 [18, p. 200]. It was radio message No. 25 (*Funkspruch Nr. 25*), transmitted by *SST-Div* on 13 July 1941 [18, p. 227]. In that article the first 5-letter group of the message was used for identification, a handy custom, which will be retained here. During the first years of the war, such as the year 1941, the first group of a message (in this case FHPQX) served as a discriminant (*Kenngruppe*). It was used to identify the correct system and daily key, i.e. it was not part of the ciphertext. This changed on 1 Sep 1943, when the German Army dropped the use of the *Kenngruppe* [16, p. 331]. The emended and translated plaintext reads “To Tank Group 4: SST Div stands since 12 July 1100 hours with vanguard at the accommodation space. Cannot enter, as 3rd Inf Div and 8th Tank Div are blocking and keeping place occupied. Div Cmdr.” (*An x Panz x Gruppe x Vier x Siegfried Siegfried Toni x Div x steht seit x Eins Zwo x Sieben x Eins Eins Null Null x Uhr mit Anfaengen am Unterkunftsraum x Kann niqt einflieszen x da x dritte x Inf x Div x und x aqte x Panz x Div x bloqieren und Ranm belegt halte x Div x Kdr x*).

Concerning this message, nearly everything is favorable with respect to an easy decryption. The message is long (214 letters). Usually, longer messages are easier to break than short messages, because their statistical characteristics generally fit better to the expected values. In 2005 “a preference for messages with around 180 letters, they were long enough to succeed and not so long as to give extended run times or encounter problems with a slow wheel turn-over” was observed [18, p. 203]. Actually, a left-wheel turnover occurs here directly after the first letter. But for the algorithm this is nearly identical to no turnover taking place, as only a single letter (i.e. the first letter of the text) is affected. This is quite similar to a single garble, and produces nearly no handicap for decryption. Furthermore this specific message contains almost no garbles (apart from *Ranm* and *halte*, which should read *Raum* and *halten*). Also the IC of the plaintext is rather high (6.76 %), while the mean value for German Army texts is approximately 6.1 %. All these characteristics make decryption rather easy and prove a lucky choice for the first successful break.

The message FHPQX is moreover a typical example for a ciphertext which possibly could have been broken at BP with the aid of a crib. Here for instance *“SiegfriedSiegfried”* or *“SiegfriedSiegfriedToni”* could have been used. The words *“Siegfried”* and *“Toni”* come from the spelling alphabet then commonly used by the *Wehrmacht* and served as substitutes for the letters S and T. Though BP had some difficulties intercepting German Army messages from the eastern front, because of the long distance and the low power of the German Army radio sets, they succeeded in receiving and decrypting several, especially from the beginning of 1942 and onwards. In 1941 BP knew about only three German Army keys from the eastern campaign. One was Vulture, a key for a complex network of army and army group communications on the eastern front and the Army High Command (*OKH*) in Berlin. This network carried detailed operational reports and the occasional high-level messages with appreciations and operational planning. The other two keys were Kestrel, an army/air co-operation key, and Kite, the Army's eastern supply key, called "*Oberquartiermeister Maschinenschlüssel A*" by the Germans [10, p. 69; 11, p. 346-347][1]. BP first broke this key on 2 January 1942 [11, p. 412]. When analyzing the content of the broken messages and looking at the amount of traffic on the different days we feel is highly unlikely this traffic came from the Army key Vulture. On the contrary we believe this traffic could come from Kite, the eastern supply key. If this is indeed the case then these messages were never broken and decrypted at BP. A software tool of the authors, simulating the *Turing-Welchman-Bombe*, produces the correct ten *Steckers* for FHPQX within seconds with the aid of the crib *“SiegfriedSiegfried.”* This experiment shows that these messages could easily have been broken at BP. Why Kite was not broken before January 1942 is not known, but questions of available traffic, expected intelligence value and availability of free Bombes and resources could have played important parts in a decision to perhaps delay the breaking of this traffic.

The ciphertext-only cryptanalysis of further messages of *HG Nord* succeeded in 2003 and the following years, after both the software algorithm and the database used for generating the bigram and trigram scores had been improved. Even short messages could be read in cases, where several messages of the same day were available. Here it was sufficient to recover the daily key via one of the longer messages, and after that all the other messages of the same day could be easily deciphered.

[1] Prior to August 1944 this key was called *Oberquartiermeister Maschineschlüssel Nr. II.*

4. Short Messages

A few messages of the *HG Nord* record nevertheless remained unbroken. This happened for instance, when only a single ciphertext was available for a specific day. Therefore, we decided to focus on short messages, i.e. ciphertext of lengths less than 80 letters down to roughly 30 letters. For that, obviously the existing known ciphertext-only techniques are insufficient and have to be improved. As a first step, a thorough study of several hundred authentic messages of *HG Nord* was performed. Their statistical characteristics were determined, and especially short messages were investigated.

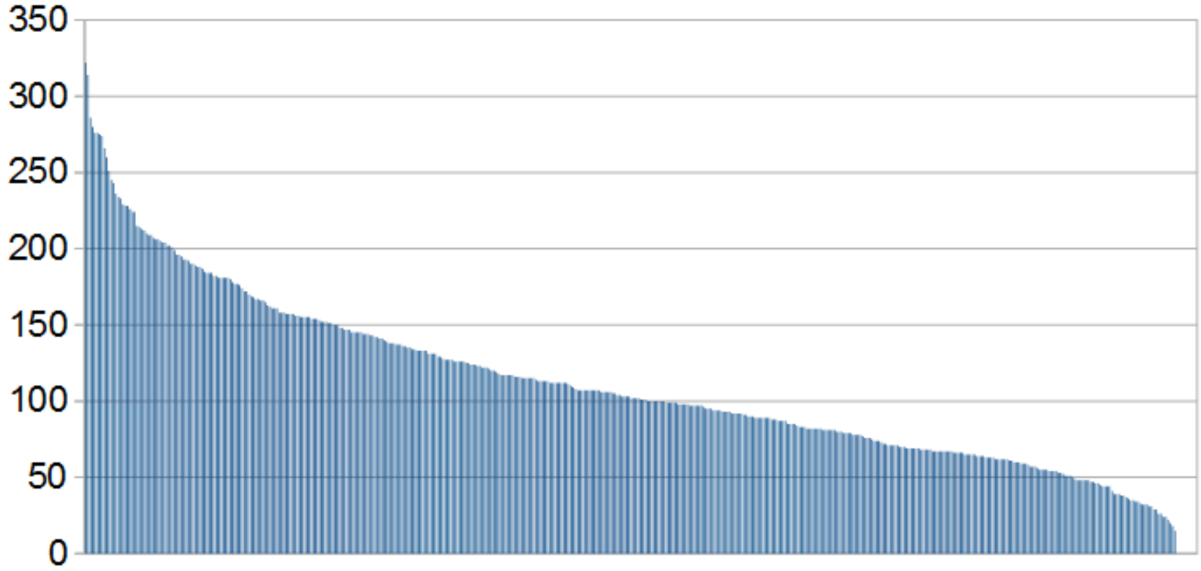


Figure 1. Ciphertext lengths of ca. 500 messages of *HG Nord.*

The message lengths vary. As shown in Figure 1 most of the approximately 500 messages have a length between 50 and 150 letters. The minimum message length is 15 and the maximum is 322. The mean message length is 126, the median 103, and the standard variation 55. Eleven messages or about 2 % violate the ordered maximum message length of 250. Approximately 30 % of the messages contain less than 80 letters.

As a matter of principle for any substitution cipher, there exists a certain ciphertext length for which messages shorter than that cannot be broken. This length is called the unicity distance. For example, a ciphertext, encrypted by a method as simple as the Caesar cipher, is theoretically unbreakable, if its length is one, i.e. only a single letter had been encrypted. In this case, the security level of a Caesar cipher is as high as for the theoretically unbreakable One Time Pad (OTP).

An Enigma message which contained only one letter must not be considered, as it surely can be stated unbreakable. This is true also for slightly longer messages, if they fall short of the unicity distance. We do not know it exactly for Enigma, but it is probably around 20 letters. As derived by Shannon [17, p. 660], and e.g. defined in [13, p. 246], the expected unicity distance of a cipher is approximately H/D, where H is the logarithm of the number of possible keys, and D is the plaintext redundancy (in bits/character). With H ≈ 72 bits for Enigma (corresponding to its key space of 150,738,274,937,250×26,364,000, assuming no left-wheel turnover) and D ≈ 3.1 bits for *HG Nord* texts, this results in a unicity distance of around 23 letters. This is quite similar to the unicity distance of a monoalphabetic substitution cipher, which is 24 letters [5, p. 54]. The shortest known plaintexts of *HG Nord*, with lengths of 18, 22, and 24, contain (slightly garbled) messages as “Give daily report” (*Tagesmmldung funken*), “Morning report cancelled” (*Morgenmeldung entfaelct*), and “Intermediate report cancelled” (*Zwisnenmeldcng entfaellt x*), and yield an IC of 4.6 %, 5.6 %, and 5.8 %.

These authentic plaintexts show a lower measure than many random texts with lengths between 18 and 24. The latter easily generate an IC as low as 2 % or as high as 8 % simply by accident. Therefore, the correct plaintexts can barely be perceived as being plaintexts because of their low ICs. Also the measure of bigrams and trigrams, or other criteria, does not help much. Because of the short text lengths and moreover because of garbles, virtually every criterion fails to yield a significantly higher value than for random texts. Assuming a text length of say 26 letters, then as an average value each letter of the alphabet occurs only once in the ciphertext. That is the reason why the statistics do not work well for short texts, especially for garbled ones. Thus, even if a complete exhaustion of the whole key space of Enigma, including all plugs, were possible, and for each single key the candidate text could be investigated, it would be virtually impossible to perceive the correct plaintext. Our statement is therefore, that Enigma ciphertexts with a length shorter than 20 letters are virtually unbreakable. This is true, especially when they contain garbles, and, as long as they are singular texts and no further information, e.g. cribs or the daily key, is given.

Furthermore, most texts with lengths up to say 20 to 30 letters proved to be practically unbreakable. Experiments with concocted plaintexts and keys showed that the turning point, where some short messages become breakable, is for a message length of about 24 letters. Of course, this is no general statement and there is no sharp limit. The breakability always depends on individual characteristics of both the plaintext and the key. As a test, the artificial plaintext *"EinsxEinsxVierxNullxNull"* (One, one, four, zero, zero) was enciphered using the key B432 rit VOR AH BO CG DP FL JQ KS MU TZ WY. The resulting ciphertext FVKFC DWRII CYFHV SKQOW QTTH (length = 24) was broken by the authors' software tool, which exhausted one wheel order in five hours, while running in eight parallel instances.

One of the shortest authentic messages that could possibly be broken is message no. 128 TZLPT of 8 July 1941 with a length of 27 letters. The ciphertext XPDBQ LJWFT ULSZC DKQPS WIMGB YS can be broken by the authors' software tool and the plaintext *"Wo Roem Eins Berta Staffel Frage"* is detected. In this case the plaintext is free of garbles. But, as there exist spurious solutions with a higher trigram count, the solution is overwritten. If needed, this problem can be solved by a special assessment stage for the candidate texts. This has to evaluate the occurrence of words, which we know are frequently used by *HG Nord*, for instance *"Berta", "Eins", "Frage", "Roem"*, etc. By this the real solution can be detected and retained in spite of spurious solutions with a higher trigram score.

The shortest authentic message that proved breakable with our software is AMERI, with a length of 32, and the second shortest authentic message is PFCXY, with a length of 36. The total key space for the latter had been exhausted in less than three days. Breaking becomes even more feasible, if the ciphertext length is greater than 40, as can be shown for other authentic messages of *HG Nord*. Curiously enough, some of these messages, with lengths between 40 and 80 letters, broke surprisingly easily, while others were nearly unbreakable. Typical examples for that are YYBRW (of 21 Aug 1941 with a length of 46 letters without the *Kenngruppe*), HODSN (6 Sep 1941, 48), BOTKB (14 Sep 1941, 69), and ABPQX (24 Sep 1941, 76). Unexpectedly the two shorter ones broke fairly easily, while the two longer ones proved extremely hard. An interesting question, the answer to which will help improve the attack, is: What is the reason for a failure or a lucky break?

It was detected that for short messages the expected statistical characteristics vary strongly and do not match the mean values, which are more or less fulfilled for long messages. A prominent example is the superb criterion, which generally is highly appreciated and proves an excellent measure for cryptanalytical attacks, namely Friedman's IC (e.g., [2, p. 77]). It was found that for short texts of *HG Nord* the IC varies significantly. While the plaintexts of PFCXY, YYBRW, and HODSN show a very high IC of 8.25 %, 7.05 %, and 6.91 %, on the other hand ABPQX and BOTKB yield an IC of only 4.95 % and 4.90 %. For example, the plaintext after deciphering BOTKB reads "*Nachschubdienste x Null Aqt Vier Nulf x Omytscikino x Omytscukino x Hartjenstein.*" Though the plaintext looks typical at a first glance, with a common message text and the usual garbles, the letter count yields an unexpected smooth histogram containing only 5 E, and 8

N, only 2 R, and 4 X, 5 S, and 6 I, resulting in an IC as low as 4.9 %, which is not much higher than that of a random text (3.8 %) or the ciphertext itself (4.0 %) and far less than that of a typical message plaintext (6.1 %) or a common German language text (7.6 %). Furthermore, it has to be considered that the standard variation σ of a random text of 69 letters is 0.4 %, thus its IC frequently happens to be in the range of 3.8 % ± 0.4 % or even 3.8 % ± 0.8 % when considering 2·σ. This means, it is likely that a random text of 69 letters happens to yield an IC of 4.6 %, which is very close to 4.9 % and much the same as the IC of the plaintext BOTKB. That is the reason why the IC is sometimes useless.

5. The Influence of the Plugboard

All the other parts of the key space but the plugs, namely wheel order, ring settings, and wheel positions, sum up to the said 27,418,560 locations. This number is small enough to let us carry out a full exhaustion with the aid of a commercially available PC, in the authors' case an Intel i7-3770 processor running at 3.4 GHz. But the number of 150,738,274,937,250 different possibilities for arranging ten plugs is far too huge for that, even with modern hardware. (The cables used for plugging were cross-over jumper cables with double-pole connectors at both ends, so "a plug" here always means the use of a double-pole cable which swaps two letters. As a consequence ten plugs swap twenty letters.) Just for illustration we will calculate the theoretical runtime for a full exhaustion of the plugboard: With the authors' PC a sample decipherment of a text with 50 letters needs about 100 ns. In other words, 10 million candidate texts can be generated per second. To produce all the roughly 150 trillion possible candidate texts for all possible plugboard arrangements, approximately 15 million seconds or about half a year would be needed, nota bene, this time would be required to exhaust the plugboard at each and every one of the 27,418,560 locations, and this does not even include any further processing.

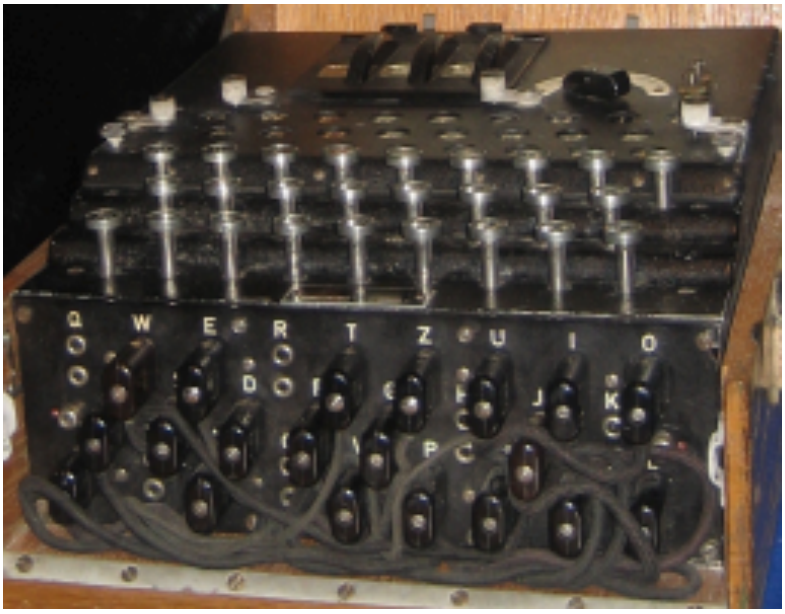


Figure 2. Enigma's plugboard with the usual number of ten plugs, leaving six letters unswapped or "self-steckered," offers 150,738,274,937,250 different selection options for arranging the plugs. (Source: Wikimedia Commons, accessed 5 April 2016, https://commons.wikimedia.org/wiki/File:DDayMuseumEnigmaMach.jpg)

Fortunately, a full exhaustion of the plugboard (Figure 2) is not needed to find the correct plug arrangement. As in breaking a monoalphabetic substitution cipher, the plugs can be searched via hillclimbing. For developing an optimal hillclimbing strategy for searching and, if at all possible, finding only correct plugs during the hillclimb, a thorough understanding of the influence of the plugboard on the candidate texts is essential. Assuming we stand at the correct location (wheel order, ring settings, and wheel positions), and then decode the ciphertext, a candidate text comes out, which would be the correct plaintext, if all ten plugs were correct. But, as they are unknown to the codebreaker, it is a good choice to start with an empty plugboard. The *Wehrmacht* strangely enough decided not to use the maximum number of possible plugs, namely 13, for plugging, but settled for only 10. That is the reason why an empty plugboard fundamentally already holds the six correctly self-steckered letters. The whole trick is to retain these, if at all possible, which is equivalent to finding only correct plugs. Incorrect plugs at most make cryptanalysis harder, and they have to be corrected earlier or later during hillclimbing, while, in rare cases, they might ease a break. Starting with some random plugs, therefore, is generally not a good idea, because the hillclimbing then starts with a handicap, and is comparable to a hypothetical case of defeating more than ten plugs.

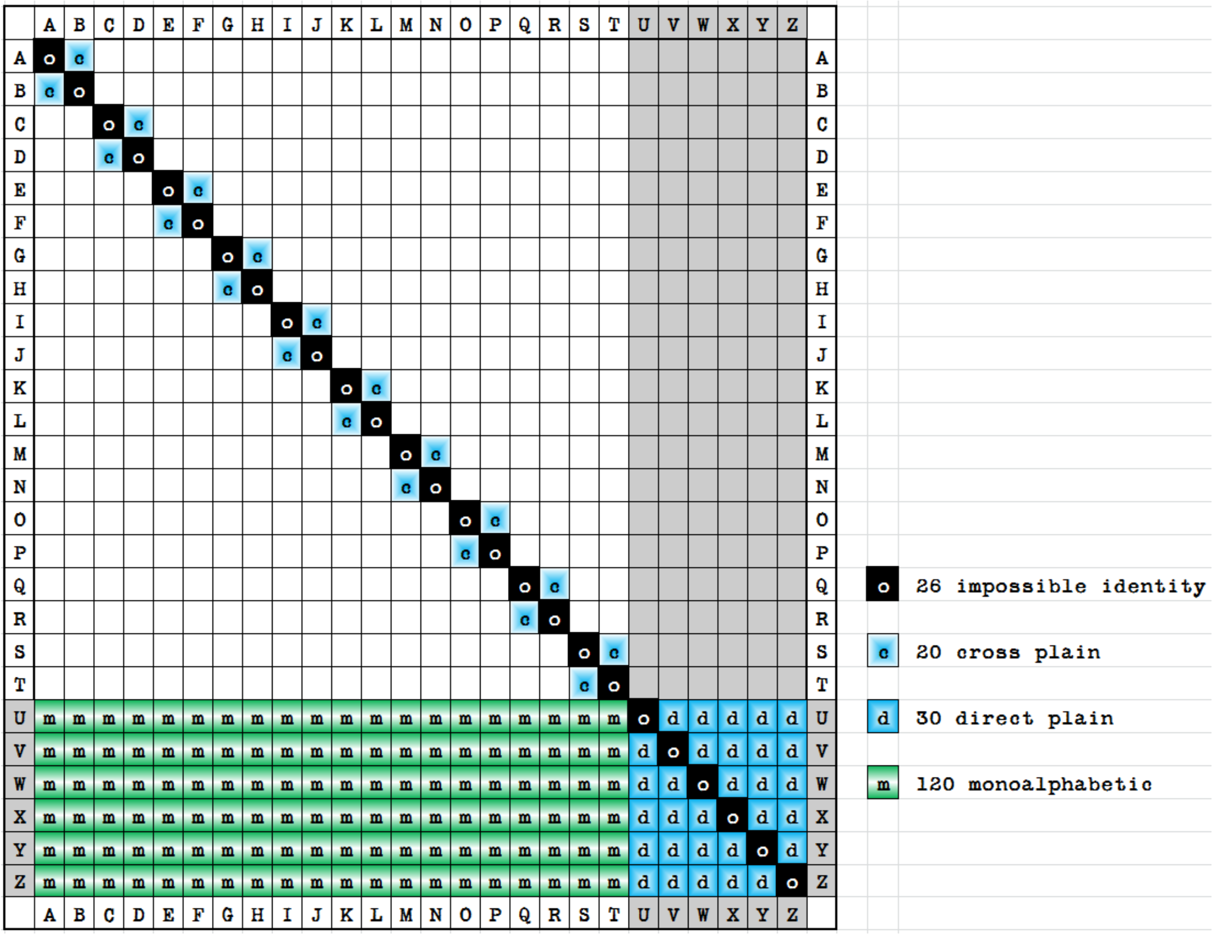
If the *Wehrmacht* had used the maximum number of 13 plugs instead of only 10, this would not have threatened the codebreaking capabilities at BP, because the *Bombe* was insensitive to the number of plugs. However, three further plugs affect the modern hillclimbing techniques, because the added plugs dim the brightness of the IC, as well as other measures for the plaintext. We therefore have the curious situation that messages with 13 plugs could have been broken during the war, whereas today, more than 70 years later with modern ciphertext-only techniques and without cribs, some of them are still hard to break, perhaps seemingly unbreakable.

To verify this statement, some experiments have been performed utilizing authentic ciphertexts and keys. The known plaintexts served to create modified ciphertexts. For that, the original keys were used, but now with all 26 letters plugged, including the six originally unplugged letters. For example, the key of FHPQX left the six letters B, C, F, J, P, and S unsteckered. As an experiment the authentic key was re-used, but with the additional plugging BC, FJ, and PS. The original plaintext, including garbles, was then re-enciphered with this new key with 13 plugs. It resulted in a new ciphertext. As the experiment showed, also the new ciphertext could be broken by the authors' software. In contrast to ten plugs, however, thirteen plugs need a higher sensitivity of the program. This results in a lower execution speed. While the authentic ciphertext of FHPQX with 10 plugs breaks within a time span of approximately ten minutes per wheel order (on a single core of the authors' PC), a break of the modified ciphertext with 13 plugs is significantly slower and needs about six hours per wheel order. If represented by a factor, one could say the break with 13 plugs is approximately 30 times harder than with 10 plugs.

Further experiments with other authentic texts showed more or less similar results. The said factor varies and sometimes, depending on the length of the text and the number of garbles, is significantly higher, maybe 300. This allows the following conclusion for a fictive scenario of 13 plugs used by the *Wehrmacht*. If a ciphertext with ten plugs is broken by a modern hillclimbing software after say one day of exhaustion time, then the needed time, if 13 plugs had been used, would have been in the region between one month and one year. In practice, this could make the critical difference, and could change an easily breakable ciphertext into a seemingly unbreakable one. To the authors' knowledge, the *Wehrmacht*, with few exceptions at the very end of the war, almost never used 13 plugs, thus this scenario is a fiction. Fortunately, we have to deal with "only" ten plugs as used by *HG Nord* during Operation Barbarossa.

First of all, it is useful to get a thorough understanding of the influence of an empty plugboard during decryption, as it forms the start for hillclimbing. It is essential to answer the question, what does a candidate text look like, when the ciphertext is decoded at the correct location but with an empty plugboard? The scrambler (i.e. the inner part of the Enigma, consisting of the three rotating wheels and the reflector) now reverses the effect of the same item of the Enigma used for encryption, and only the influence of the plugboard remains. For that, it is important to consider that the plugboard affects the text twice. In the first instance, during enciphering a plaintext letter is permuted by the plugboard, assuming a plug has been set for that specific letter. As usually ten plugs were inserted, out of the 26 letters of the alphabet 20 were swapped, and six remained unchanged. After that it is substituted by another letter by means of the scrambler and passes the plugboard a second time. Here it suffices to look at the scrambler as a whole, with its well-known characteristic of excluding the identity permutation. As a consequence, there exist 25 (and not 26) possibilities for permuting each of the 26 possible input letters.

In the logical scheme of the scrambler, i.e. the inner part of the Enigma, consisting of the three rotating wheels and the reflector, (Figure 3) the ciphertext letters enter from the left and the resulting output letters, forming the candidate text, leave it at the bottom. Here, for the sake of simplicity, the ten plugs AB, CD, EF, GH, IJ, KL, MN, OP, QR, and ST were assumed, leaving the six letters U, V, W, X, Y, and Z self-steckered. Because of the well-known characteristic of the scrambler that no letter ever encrypted to itself (black boxes with o-symbols in the diagonal), only 26 × 25 or 650 different cases have to be considered.



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

o 26 impossible identity

c 20 cross plain

d 30 direct plain

m 120 monoalphabetic

Figure 3. Logical scheme of the scrambler with the ciphertext letters entering from the left and the resulting output letters, forming the candidate text, leaving at the bottom. 30/650 cases *directly* yield a plaintext letter (d). (360/23)/650 *"accidentally"* convert to a plaintext letter (a fraction of the white cases). 20/650 yield a plaintext letter because of *cross-plugging* (c). 120/650 convert to the monoalphabetic substitute (m) of the plaintext letter.

Assuming the correct location of the scrambler and an empty plugboard, then two scenarios exist for deciphering a ciphertext letter. Firstly, in the 6 self-steckered cases out of the 26 possible cases (marked gray in the first column of Figure 3), a ciphertext letter at the input of the plugboard passes the empty plugboard correctly in the first instance. Afterward it is correctly permuted by the scrambler, and converted to the correct output letter of the scrambler, avoiding identity to the input letter. Now, for each of the 6 unsteckered input cases, 25 different output cases have to be considered. When passing the plugboard in the second instance, in 5 out of the 25 output cases, the output letter of the scrambler passes the plugboard correctly again (marked gray in the last row of Figure 3), yielding a correct plaintext letter. In the remaining 20 of the 25 output cases the monoalphabetic substitution of the correct plaintext letter comes out, because the letter is correct but misses the final permutation of the plugboard. Hereby, the plugs used for encryption give the involutoric substitution alphabet.

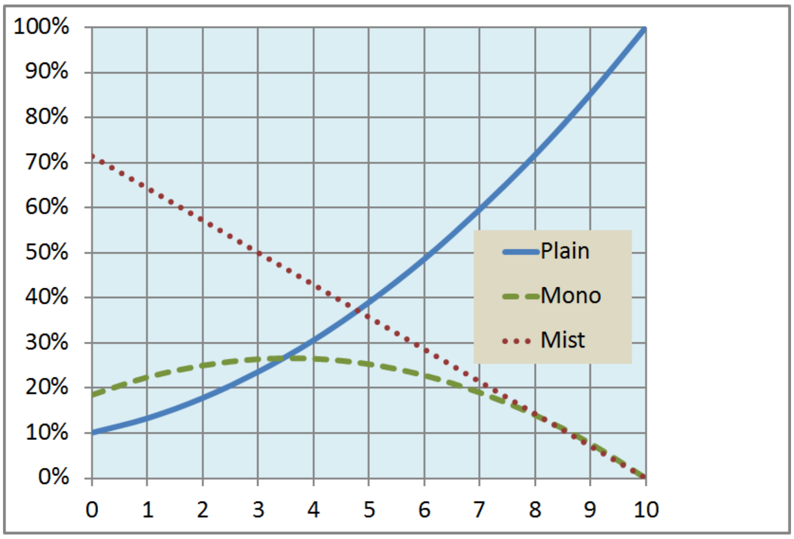
Secondly, in the 20 steckered input cases out of the 26 possible cases, a ciphertext letter passes the empty plugboard wrongly in the first instance. Afterward, though the scrambler is at its correct location, the wrongly swapped letter is permuted into a pseudorandom letter, again avoiding the identity permutation. These output letters then pass the plugboard in the second instance.

To sum up, at the correct location of the scrambler, although the plugboard during deciphering is empty, a correct plaintext letter can be produced. This happens when the ciphertext letter was self-steckered and the corresponding plaintext letter was self-steckered too. The chance for it is 30/650 or about 4.6 %. (The number 30 can be verified by counting the boxes marked ‘d’ in Figure 3.) Furthermore, it can happen that the ciphertext letter was steckered to the plaintext letter. In these cases, also with an empty plugboard, the correct plaintext letter comes out. The chance for this is 20/650 or approximately 3.1 %. (The number 20 can be verified by counting the boxes marked ‘c’ in Figure 3.) Additionally, because of the arbitrary wirings of the scrambler, and depending on the specific wirings and the location, a plaintext letter may be produced “accidentally.”

The Enigma is of course perfectly deterministic. Nevertheless sometimes lucky things seem to happen by accident. For instance, though rather unlikely but not impossible, the following can and will happen. Assuming the ciphertext letter was plugged, then by deciphering with an empty plugboard it wrongly passes it unswapped, and the wrong letter reaches the scrambler. Now, with a probability of 1 in 25 cases, it converts it to the correct plaintext letter “by accident.” Afterward, during its second pass through the empty plugboard, it remains unchanged and the correct plaintext letter remains.

Such an occasional conversion of a ciphertext letter to a correct plaintext letter happens in about 16/650 cases or 2.4 %. Besides the 30/650 “direct” plain letters and 20/650 “cross” plain letters, “accidental” plaintext letters can fundamentally only be produced if both the ciphertext and plaintext letters are steckered. So one has to observe not the whole square with its total size of 26×26 (see Figure 3), but the smaller white-coloured sub-square with the white edges (meaning plugged letters). Its size is 20×20. From this the 20 impossible identity cases (black boxes in the diagonal) and the 20 cross-plain cases (marked ‘c’) have to be subtracted, thus leaving 20×20−20−20 or 360 cases as candidates for “accidental” plain letters. The number 360 can be verified by counting the number of white boxes of the figure. For these 360 cases, the ciphertext letter is permuted by the scrambler into another letter, with the exception of the identity and the two letters that were originally exchanged by the scrambler of the enciphering Enigma. (The scramblers of the enciphering Enigma and the deciphering Enigma here with its empty plugboard are identical by definition.) Hence, from 26 possible output letters, there are three that are impossible, leaving 26−3 or 23 possible cases. From these 23 possibilities one will “accidentally” occur, yielding a chance of 1/23 for each of the letters to happen, one of it being a plaintext letter “by accident”. This results in a probability of 1/23 out of 360 cases to “luckily” produce a plaintext letter, and exactly 360/23 or 15.652... or the said about 16 out of the 650 total cases.

The three different mechanisms for producing a correct plaintext letter though the plugboard is empty – namely direct plain, cross plain, and accidental plain – sum up to 4.6 % + 3.1 % + 2.4 % or about 10.1 %. Moreover, in 120/650 or 18.5 % of the cases (the number 120 can be verified by counting the boxes marked ‘m’ of Figure 3.), a ciphertext letter is correctly converted during its first passage of the empty plugboard and by the scrambler, but fails to be swapped in the last instance by the plugboard. Thus it remains the monoalphabetic substitute of the plaintext letter. In the rest of the cases, i.e. in 100 % − 10.1 % − 18.5 % or 71.4 %, a letter is cryptographically strongly converted to a pseudorandom letter, which is virtually useless for the codebreaker and for hillclimbing, and figuratively remains in the "mist." To stress it again, if the *Wehrmacht* had decided to use 13 instead of 10 plugs, then the "fog" here would have been substantially thicker for our hillclimber. The following figure (Figure 4) and table (Table 1) illustrate the influence of the plugboard.



#	Plain	Mono	Mist
0	10.1 %	18.5 %	71.4 %
1	13.3 %	22.4 %	64.3 %
2	17.8 %	25.0 %	57.2 %
3	23.6 %	26.4 %	50.0 %
4	30.6 %	26.5 %	42.9 %
5	39.0 %	25.3 %	35.7 %
6	48.6 %	22.8 %	28.6 %
7	59.6 %	19.0 %	21.4 %
8	71.7 %	14.0 %	14.3 %
9	85.2 %	7.7 %	7.1 %
10	100 %	0	0

Figure 4. **Table 1.**

Average influence of the number of correct plugs on the percentage of plaintext letters, monoalphabetic substitutes, and pseudorandom letters of the candidate texts.

The above percentages are valid for an assumed flat histogram of the plaintext, which naturally for a real plaintext is rough. Therefore, the actual letter histogram has to be additionally considered when evaluating the specific numbers of plaintext letters for an individual text. As can be seen from Figure 4, generally, with an empty plugboard (0 plugs), the part of the pseudorandom letters (71 %) is dominating; leaving only small parts for plaintext (10 %) and monoalphabetic substitutes (19 %). In other words, the brightness of e.g. the plaintext trigrams is significantly dimmed. Moreover, because of the individual enciphering of each letter by the Enigma, contiguous trigrams are more sensitive to wrong plugs and can be easily ruined. On the contrary, a measure for the monoalphabeticity, such as the IC, is less affected and dimmed to about 19 %. What can be further seen, is that for 0 to 3 plugs, the monoalphabetic part remains greater than the plaintext part, though the latter significantly increases. That is the reason why Friedman's IC works so well when searching the first few plugs, and pure plaintext measures, e.g. a trigram scoring, do not. This situation obviously changes after the third correct plug has been found. While the mist part is still dominating, even for 4 correct plugs, it is continuously decreasing. The plain part in the end exceeds both the monoalphabetic part and the pseudorandom part. After four correct plugs have been found, the rest is easy.

6. Practical Aspects

For ciphertext-only attack on the unbroken ciphertexts of *HG Nord* the authors used a specially designed software algorithm. It is based on the suggestions of Gillogly, also using the IC for searching the first few *Steckers*, but with some modifications, especially concerning the hillclimbing technique. Tests have shown a relationship between the ICs of different plaintexts and our inability to break their ciphertexts. The messages BOTKB and ABPOX, both having plaintexts with low ICs, proved nearly unbreakable with the authors' software. On the contrary YYBRW and HODSN, in spite of their even shorter message lengths, broke easily. For instance YYBRW breaks on the authors' PC, utilizing a single CPU core only, in less than two hours per wheel order.

Running the same software in eight instances simultaneously, each dedicated to a different part of the key space, using hyper-threading and all four cores of the PC, the complete key space of 60 wheel orders is successfully exhausted in approximately 28 hours.

The reason for the relatively easy breaks was found in the accidentally high IC. Naturally, when using the IC as the principal measure for plaintext recognition, especially for the detection of the first few *Steckers*, plaintexts with a high IC will be recovered both faster and easier than plaintexts with a low IC. An example for the latter is BOTKB, with an IC not much higher than that of the ciphertext or an arbitrary random text. The outer program loops for exhaustion, i.e. change of wheel orders, wheel stepping and ring setting, meaning straightforward programming. But the search for the correct plugboard connections, especially the finding of the first three or four hopefully correct *Steckers*, is the really critical part of the algorithm. It is decisive that a suitable technique for an efficient and mostly error-free plug search is used. Furthermore, the optimization of the plugboard must be efficient and capable of eliminating possibly wrong plugs.

It is essential to choose a criterion for the recognition of the plaintext that works with all authentic plaintexts, independent of accidental variations of the statistics. As described, because of the observed fluctuations, the IC is not always a sufficiently reliable criterion. This unfortunately is also the case for other classical or newly developed criteria, be it the Chi-squared statistic for monograms or bigrams, the Sinkov statistic for monograms or bigrams, the IC for bigrams, or others. All of them sometimes work surprisingly well for rather long texts and, depending on the accidental statistics of the plaintext and the specific characteristics of the key (e.g. was the letter E steckered or not), sometimes really excellently. But unfortunately this cannot be stated generally. Often a criterion, which may be very efficient for one ciphertext, is absolutely inefficient for another, as illustrated by the example of the IC and BOTKB.

The authors found that this is also true for bigram scoring. Especially for short texts and even more for garbled texts, which occur regularly rather than rarely for authentic messages, accidental bigrams occur such as EN, ER, or RE, which enhance the bigram score significantly, while the correspondingly tried *Steckers* are wrong. This leads to a wrong hillclimbing path and the break fails. A very reliable criterion for the detection of ungarbled plaintext is hexagrams (6-grams). One of the authors used these successfully for a cryptanalytical challenge created by Dirk Rijmenants in 2007, the breaking of a transposition cipher called the “Crypto Box Challenge” [3]. In this case the plaintext was absolutely free of garbles and hexagrams proved very efficient and far superior to bigrams or trigrams. For transposition ciphers the monograms do not change and their numbers are identical for both ciphertext and plaintext. That is the reason why all monogram statistics are useless for decrypting a transposition cipher; also bigrams and trigrams are poor.

In the case of Enigma and the garbled messages of *HG Nord* hexagrams lose efficiency, because of the frequent garbles. The number of garbles varies strongly. While one third of the texts are free of garbles or have only one or two, many ciphertexts contain 5 % to 20 % garbles, some even up to 30 % or more. So a compromise has to be found between long n-grams, which are very efficient for detecting pure plaintext but sensitive to garbles and short n-grams, which are less sensitive to garbles, but also less effective in discriminating plaintext. The compromise was found to be trigrams.

Because of the described reasons the authors decided not to use the IC or bigrams as a measure for finding *Steckers* for short ciphertexts. While the IC proves useful in finding the first few correct plugs for longer messages (length greater than 80 letters), bigrams are not so reliable, neither for long nor for short texts and neither in finding the first nor the last plugs. In contrast to that, a trigram score is always useful in finding the last plugs for both short and long messages, garbled or not.

The dominant question is, what is a suitable, efficient and reliable criterion for finding the first few correct *Steckers* for short messages? The authors’ conclusion is, such a criterion simply does not exist. For short messages it is impossible to distinguish between a random text, or the original ciphertext, or the candidate text resulting from deciphering the ciphertext with the correct key but with an empty plugboard, and even with a plugboard with one or two correct plugs. Therefore it is almost impossible to verify the first few correct plugs. This is mostly true for short messages (length less than 80 letters), but dramatically changes for longer messages. The longer a message is, the easier it can be broken, the shorter the harder. As said, even a Caesar's cipher is unbreakable for a message of length one.

In 1995 Gillogly detected a possibly correct wheel order and wheel starting position, thus a first step in breaking an Enigma ciphertext, by the power of the IC, which for long texts is strong enough to "shine through" the empty plugboard even with wrong ring settings. But this is only true for very long texts. He used a text length of 647 letters, which is far beyond all observed authentic messages. Williams was successful with another monogram criterion and a modified technique. She used a length of 450 letters, which is also beyond the maximum length of 250 as ordered by the German regulations.

7. Partial Exhaustion of the Plugboard

If there exists no reliable measure for detecting the first few correct plugs, how can they be determined? The usual answer in cryptanalysis is, if no shortcut can be found, then use exhaustion. The obvious disadvantage is, it is time consuming; but the advantage is, by exhaustion it is absolutely sure to get the correct plug. As the focus here is on short messages, one can take their short length as a further advantage. The shorter the message and consequently the shorter the corresponding candidate text is, the quicker the trial decipherment and the calculation of the statistics can be done. Another general rule in cryptanalysis is, don't waste time, use the time to perform relevant tasks and don't waste it with senseless work.

It is not a good idea to start with random *Steckers*, e.g. plugging two or three or even ten plugs randomly before starting the hillclimb. Starting with a random plugboard may produce really impressive success in one of thousands or millions of trial decrypts, but generally it is simply a waste of time. That is the reason why in all other cases one should start with an empty plugboard, as this guarantees the six self-steckered letters to be correct, while a randomly chosen plugboard only reduces the chances for that.

Of course one has to exhaust the 60 wheel orders and the 26^3 start positions, as no shortcut to reduce this effort is known. Furthermore it is useful to exhaust the 26 possible ring positions of the right-hand ring, in order to take account of a stepping of the middle wheel. A stepping of the left-hand wheel however may be neglected, as it occurs only every 650 letters. For messages of a length of 80 letters or less a left-wheel turnover is rather rare, occurring in approximately 10 % of the messages. So the primary workspace is given by 60 wheel orders, 26^3 start positions, and 26 positions of the right-hand ring, resulting in 60×26^4 or 27,418,560 locations.

For each of these locations one can furthermore exhaust all possible first *Steckers* of the plugboard, which means starting with an empty plugboard and trying all possibilities for setting a single plug. The number of possibilities for such an exhaustion, here called a "solo," can be calculated the following way. A first plug can swap A with any other of the 25 letters of the alphabet, or B with any other of the remaining 24, or C with any other of the remaining 23, and so on to the last possibility of plugging Y and Z. The sum $25+24+23+\cdots+1$ gives 325 cases for the exhaustion of all possible first plugs. Multiplied with the above number one gets the total number of 27,418,560 × 325 or 8,911,032,000 cases. If one is willing to do this, one can be sure that for ten of these cases, many parts of the key, namely virtually everything with the exception of the last nine plugs is correct. That is an excellent starting point for a subsequent hillclimb.

An even more rigorous exhaustion would be to simultaneously plug all possible combinations of a first and a second plug, that is two plugs in parallel, here called a "duet." This needs 44,850 cases to be considered for the plugboard and after multiplication with 60×26^4 then 1,229,722,416,000 cases in total. For this scenario two out of ten plugs or exactly 45 combinations will give two correct *Steckers*. Theoretically one can enhance this to even more plugs that are exhausted in parallel. In the case of three plugs, called a "trio," one gets 3,453,450 cases for the plugboard and 120 of them being three correct plugs for the start of the hillclimbing. In the case of four plugs, called a "quartet," one gets 164,038,875 different starting plugboards and 210 of them having four correct plugs. But with the current hardware the effort becomes too high and the program too slow for the latter cases.

So one can think about speeding this up and again focus on relevant things and to avoid unnecessary work. For instance it is rather unnecessary to exhaust plugs for letters which do not exist. For a ciphertext length of say 52 letters, the mean value for the frequency of a specific letter is 2. In a random text of this length mostly three or four specific letters are missing, their frequency is 0. So, it is less efficient to test plugs, which contain these letters. Furthermore one knows in advance the language of the plaintext and some of its characteristics. A monogram letter count of several hundreds of decrypted messages of *HG Nord* yielded the following histogram for German *Wehrmacht* plaintext, as shown in Figure 5.

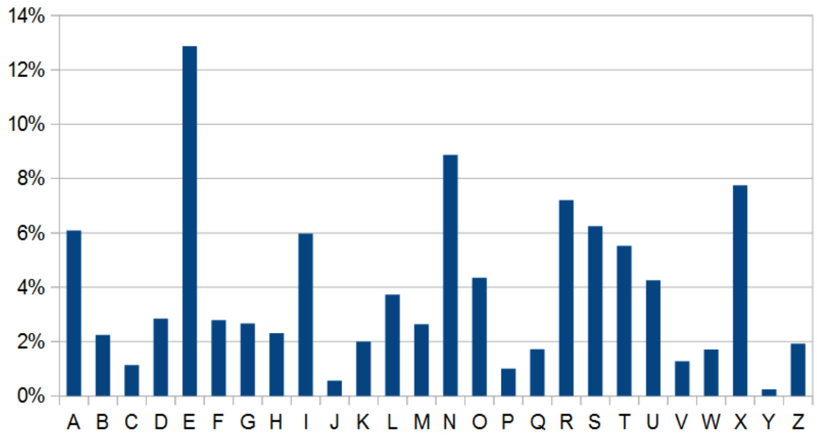


Figure 5. Plaintext histogram of ca. 500 messages of *HG Nord.*

This means E, N, X, and R (in this order) are the most frequent letters in *Wehrmacht* plaintexts, while C, P, J, and Y occur rather rarely. Y is the rarest letter with an observed frequency of less than a quarter percent in our *HG Nord* text base. That is why checking plugs which contain E, N, X, and R makes more sense and is more efficient than plugs with C, P, J, or Y. Using this, the above described "solo" technique with full exhaustion of all first plugs was modified, and instead only a partial exhaustion of the possible plugs for the most frequent letter E was performed. This technique, here called the "E-Stecker" method, includes the 25 possible partners of E plus the singular case of a self-steckered E, in total 26 cases. In this way the effort, compared to the "solo" technique, is reduced by more than a factor of ten and the speed is correspondingly increased.

An alternative and excellent practical compromise with regard to efficiency and speed is to exhaust not only all 26 possible plugs (including the self-steckered case) for the most frequent letter E, but additionally for other frequent letters such as N, X, R and so on. That increases the effort for exhaustion slightly but it remains less than for the full exhaustion of all 325 first plugs. The here called "R-Stecker" method exhausts all plugs which contain E, N, or R. The number of cases is 25+24+23+1 or 73. And the here called "I-Stecker" method exhausts all plugs, which contain E, N, R, X, S, or I. The number of cases is 136.

8. Example Decryptions

All the described techniques, beginning with a pure calculation of the IC for the candidate texts deciphered at all possible wheel orders and start positions with an empty plugboard, as proposed by Gillogly, over the different methods of hillclimbing and plug exhaustion, up to the extremely slow "quartet" method have been implemented in the authors' software and compared. A very efficient technique, which solves many short messages, is "I-Stecker."

Short messages from the data set *HG Nord*, which can be broken with our method, are AMERI and PFCXY with respective lengths of 32 and 36 letters. These cases were not first breaks, because we already knew the daily keys from breaking longer messages from the same days (27 Aug and 2 Sep 1941). Thus their breaks were not needed. They could simply be decoded with the aid of the given preamble (*Spruchkopf*), like how the intended German receiver would read them. Nevertheless, we were interested to see if they could have been broken without knowledge of the key and therefore we checked them experimentally. PFCXY broke fairly easy; AMERI was slightly harder but broke too. Thus both could have been broken in a reasonable time span on the authors' PC, in the order of two or three days, by exhausting the whole key space, even if they had been singular messages without any others from the same days.

An example of a singular and still unbroken short message of *HG Nord* is CFYZR. It is the only Enigma message known stemming from 14 July 1941. The keys of some other days of the month July 1941 are already known, including the key of 13 July, the day before. Knowing keys of adjacent days is extremely helpful, because the German authorities strangely avoided reusing wheel orders within a month, and moreover never used a wheel in the same place on two adjacent days ("non-clashing" rule [18, p. 204]). On 13 July the wheel order was 423. So, it is pretty safe to assume, that on 14 July, wheel no. 4 was not used as the left-hand wheel, wheel no. 2 was not the middle wheel, and wheel no. 3 not the right-hand wheel. This immediately reduces the workload from 60 to only 32 wheel orders. As further 8 wheel orders were known for eight other days of July 1941, only 24 wheel orders remained for exhaustion.

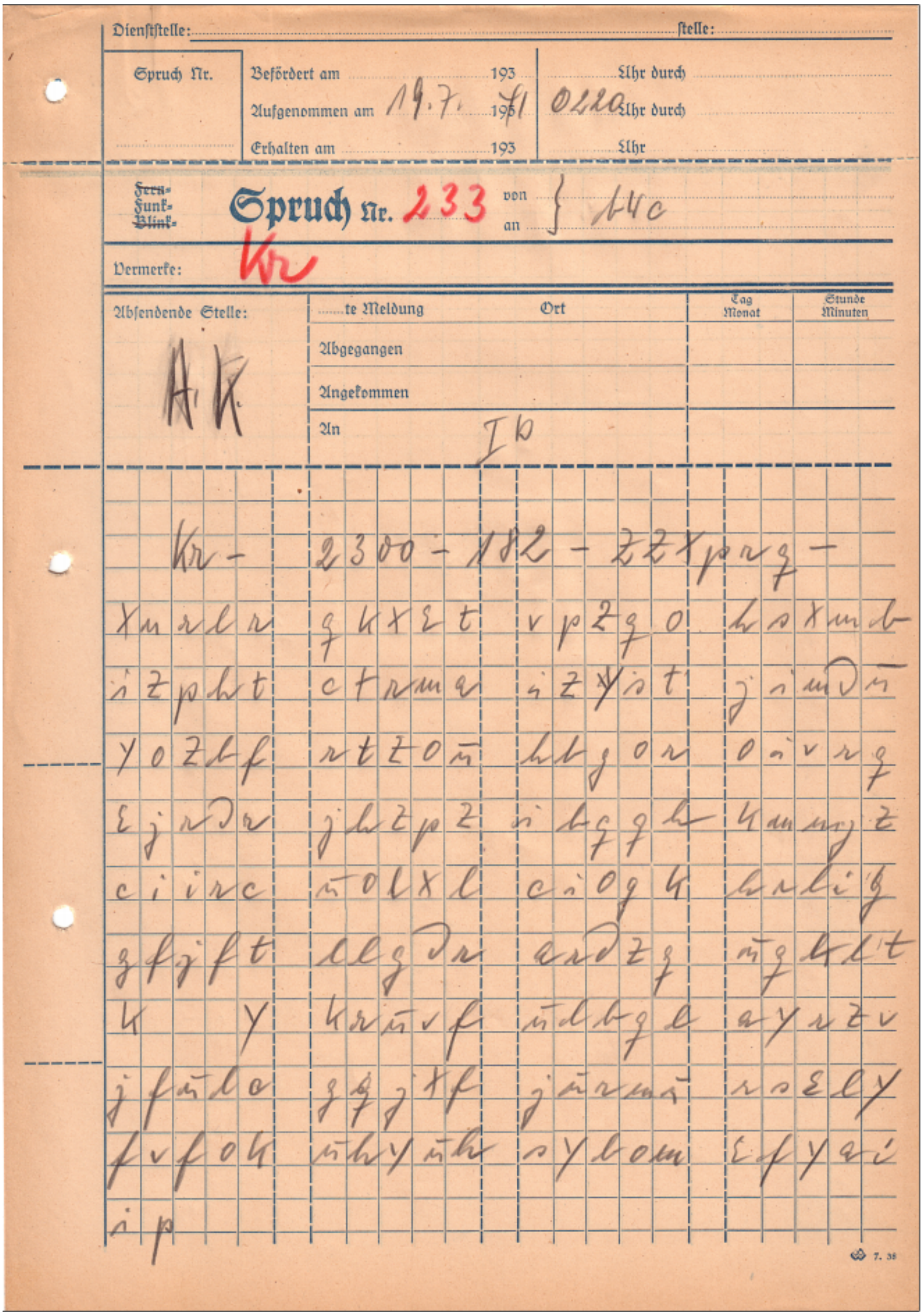
```
1005 - 77 - ULR AME -
CFYZR NFOSO IFKXN EMBCX
CWMSC MORVY WSVHF BZJHN
EMQFW ZQOLU IZBFF BSNKS
QXSHR DAMFR SESGJ JD
```

Figure 6. Message CFYZR of *HG Nord.*

As can be seen in the first line of Figure 6, the message has been sent at 10:05 hours (in the morning). It is stated to be 77 letters long, including the discriminant CFYZR. The six letters, ULR AME, are the enciphered message key, which is treated in the following way. With the Enigma already prepared with the daily key, i.e. correct wheel order, ring setting and plug connections, the three cipher wheels are set to the basic setting (*Grundstellung*) ULR. Then the three letters AME are typed on the keyboard and the letters of the three lit lamps are noted. These three letters make up the initial or start position for the message and the intended receiver must first set the three cipher wheels to this position before decoding the message. The codebreaker can also do this, if he already knows the daily key, but in this case, as it is the only known message of that day, the key was unknown.

Without the discriminant group the ciphertext of CFYZR is 72 letters long. This message was broken by the authors' program, using "I-Stecker," after less than five days, with each of the four cores of the PC working in parallel on different wheel orders, needing 28 hours runtime per wheel order. The key is given in the Appendix, which the reader can use together with a suitable Enigma simulator, e.g. Dirk Rijmenant's excellent Enigma simulator [6], to decipher the message, beginning from its first ciphertext group NFOSO. The emended and translated plaintext of CFYZR reads, "To Roman One B [Ib is the Chief of Supply] Quartermaster Tank Group Ostrov Ostrov barracks area" (*An Roem Eins Berta x Quartiermeispcr Panz x Gruppe x Ostrow Ostrow x Kasernengelzenme*).

A further advantage of an algorithm that is able to break short texts is that one may split long messages, which are suspected to comprise a stepping of the left-hand wheel. An example for that is XNRLR (177). Figure 7 shows the authentic message sheet of the *Funkspruch*. It was received on



Dienſtſtelle: ſtelle:

Spruch Nr.

Befördert am 193 Uhr durch
Aufgenommen am 19.7. 1941 0220 Uhr durch
Erhalten am 193 Uhr

~~Fern-~~
Funk-
~~Blink-~~
Spruch Nr. 233 von } A4c an

Vermerke: Kr

Abſendende Stelle: AK

....te Meldung	Ort	Tag Monat	Stunde Minuten
Abgegangen			
Angekommen			
An	Ib		

```
Kr -   2300 - 182 - zzxprg -
xnrlr  gkxet  vpzgo  hsxmb
izpht  ctrma  izyst  jimdu
yozbf  rtzou  hbgor  oivrg
ejrdr  jhzpz  ibggh  wamyz
ciirc  uolxl  ciogw  hrlig
gfjft  llgdr  ardzg  ugwlt
k   y  kruvf  udbgl  ayrzv
jfulo  ggjxf  jurmu  rsely
fvfow  uhyuh  sybom  efyai
ip
```

7. 38

Figure 7. Authentic Message Sheet (*Spruchzettel*) XNRLR of *HG Nord.* Though received on 19 July 1941 at 2:20 a.m., the message had been enciphered the day before at 11:00 p.m. (*2300*) with the daily key of 18 July 1941. The first two letters *'Kr'* of the preamble are the symbol for urgent (*Kriegsmeldung*).

19 July 1941 at 02:20 hours, but, as the preamble shows, has been enciphered at 23:00 the day before. Though it is rather long, nevertheless it did not break. So, a left-wheel turnover was suspected. As they occur systematically every 650 letters, they are rather uncommon, approximately in one of four ciphertexts of the said length. If the stepping of the left-hand wheel is at the beginning (as for FHPQX) or at the end it does not interfere much, as then only a few letters are garbled. But if the left-wheel turnover happens to be near the middle of the text, it is extremely damaging, as then one half of a possible decrypt is completely garbled. An effective instrument against this is to split the ciphertext into two halves and attack them separately. Then it is absolutely safe to assume one half is free of a left-wheel turnover. An obvious disadvantage is, the length is halved too, which then might be insufficient for the breaking tool.

For the given ciphertext, we took a first half (XNRLR1) with a length of 90 letters, and a second one (XNRLR2) with 92 letters, thus slightly overlapping the five center characters CIOQK. The first half of the ciphertext broke surprisingly easy, choosing the "E-Stecker" method with a speed of approximately four hours per wheel order and core, producing the raw decrypt *"Wie viel Klein Siegfried Gwosm Friedrich x Heinwigh Mun x insqrzaqt Eins Sieben Striq Ains Aqh x Siekjs rufm."* After that, it was found out that a stepping of the left-hand wheel occurs at position 81, hence actually right in the middle of the original unsplit text, and near the end of its first half. That is the reason why here the last seven letters of the XNRLR1 decrypt are garbled. After this breaking success, it was quite easy to reconstruct the complete message, though the plaintext contains no less than 30 or nearly 17 % garbles. The key is B425 agm QAY DM EP FL HI JR KY NQ OU SW TZ. The emended and translated plaintext reads "How much ammunition for heavy field howitzer shall be collected altogether on 17th to 18th July in Pskov? Who has been ordered, to collect it from there? Wireless response [requested]. The Quartermaster" (*Wie viel Klein Siegfried Gwosm Friedrich x Heinwigh Mun x insqrzaqt Eins Sieben Striq Ains Aqh x Sieben wilx Plesnau x abgeholt Frage x Wer p-- -efyhl Komma dohz atzuxolen Frago Vunkantwort x Der Qnaatyekvevster*).

9. Conclusion

Based on known methods for ciphertext-only cryptanalysis of Enigma and a data treasure of approximately 500 genuine radio messages, a thorough investigation of the hillclimbing strategies and the statistical characteristics of authentic plaintexts was accomplished. The message length, the number of garbles, the possible occurrence of a left-hand wheel turnover, and the actually used specific plugs affect a possible breaking success. The influence of an empty plugboard as well as one with only a few correct plugs was theoretically investigated and experimentally verified. The results explain the reasons for lucky breaks, close misses, and fatal failures. As a consequence, the hillclimbing strategy was improved, such that also strongly garbled and short Enigma messages down to the unicity distance can be successfully attacked. This results in the solution of formerly unbroken Enigma ciphertexts.

10. Appendix

For additional information, the ciphertexts of all authentic messages mentioned in this paper are listed here in their order of appearance within this paper. The original ciphertexts from the scans of the authentic message forms have been transcribed by the authors, and are given together with the recovered keys. Any suitable Enigma simulator, e.g. the one by Dirk Rijmenants [6] as previously mentioned, can be used to decipher the texts. Hereby it should be observed that the first 5-letter group is the discriminant (*Kenngruppe*) and not part of the ciphertext. Thus the deciphering always starts with the second group.

Taking the first message FHPQX as an example, the first letters of the ciphertext are FDZCJ. Before it may be entered into the simulator (or a real Enigma) the cipher machine has to be set to the correct key. The key here is given with a first group (e.g. 'B423') for the rotor set, indicating the reflector B, and the three rotating wheels, to be inserted from left to right (here using Roman numbers instead of the Arabic 423) wheel IV as the left-hand wheel, wheel II as the middle wheel, and wheel III as the right-hand wheel. The second group (here ‘gto’) indicates the ring setting (*Ringstellung*) for the three rotors. Some machines do not use letters for the rings, but numbers instead. Here, ‘a’ corresponds to ‘1’, ‘b’ to ‘2’, ‘c’ to ‘3’, and so on until ‘z’ to ‘26’. So, ‘gto’ can be substituted by ’07 20 15’.

Then the start position (here ‘SDV’) for the three rotors is given, which afterward can be checked through the windows of the Enigma. Again, some machines here use numbers instead of letters. And again the same relation is valid, namely ‘A’ corresponds to ‘1’, ‘B’ to ‘2’, ‘C’ to ‘3’, and so on until ‘Z’ to ‘26’. Instead of ‘SDV’ then ’19 04 22’ is the start position. Finally, the ten plugs (*Stecker*) are defined, here starting with the plug AD (meaning a cross-over plug between the letters A and D), then EH and so on, until at last the plug UW. Now the ciphertext, beginning with FDZCJ can be entered, and the plaintext lights up, starting with ANXPA, the first letters of *“An Panz. Gruppe Vier”* (To Tank Group 4).

The scans as well as the raw and emended plaintexts, their translation, and more detailed information will be made available on-line at www.cryptocellar.org.

Message no. 25 of 13 July 1941

```
B423 gto SDV AD EH GY IM KN LR OZ QV TX UW
FHPQX FDZCJ JDKVW PYFDW
POQZG TJQYY XAFRH SQESE
RKGJB WBYPE OOKFM MPOMK
QDDOL CPKHY PGUZY XBZYA
NYSAX IPXVQ CPJBF FFDRD
XFIJJ PPPEY ALCYK VLKXQ
HWIRZ ANGWU JBWVJ YCKES
MJQRY KQHCQ OKMMY WMCKV
LZJDV ZXRUM RMNWF DZBQG
XJQAP FFFZT AHJQZ PWQWN
IVZWU IJTHO YXGDC OJUW
```

Message no. 65 of 26 Aug 1941

```
B321 xbm DOF AE BT CF DK GJ HM IS LV OZ UX
EJRSB UNXXI SVILM HHKZP
JZU
```

Message no. 1 of 1 Oct 1941

```
B514 kbu DEI AG EL FN HU JV KM OP QR SW TX
PLVJH HBCZF WXKBE JDLUX
CODAA QV
```

Message no. 2 of 1 Oct 1941

```
B514 kbu ZAQ AG EL FN HU JV KM OP QR SW TX
OSMRV JMYDK APZMJ LRHTO
VJTMP JZVA
```

Message no. 128 of 8 July 1941

```
B432 pkf SWV CY EL FH GS IJ KQ MW PV RZ TU
TZLPT XPDBQ LJWFT ULSZC
DKQPS WIMGB YS
```

Message no. 71 of 27 Aug 1941

```
B132 les BEN AY BJ DG EH FQ IM KO LP NW RT
AMERI TDLYX LHUVK OGOTU
XNVRB PVICI BWTST YD
```

Message no. 15 of 2 Sep 1941

```
B432 rit VOR AH BO DP EX FN JQ KS LR MU TZ
PFCXY PSQDB CSFKH FJOMV
CJAUX TOTQB BPBWA CHZYX
H
```

Message no. 30 of 21 Aug 1941

```
B341 wgr TOR AC BE HW IP JZ KY LU OS QR VX
YYBRW CFVUA HZHPI WNUCX
TMJGX PMVWK FVHZJ TJGXM
SSDJY ESRCN X
```

Message no. 36 of 6 Sep 1941

```
B325 byj SAU AX BH ET FK GY IR JZ MS OU QW
HODSN ZLXAQ IZTGH JYEEC
HRVPU SGYHY IVKYI BVAZD
YNAPY NIDCU XRO
```

Message no. 46 of 14 Sep 1941

```
B243 ixm WAS AV BE CX FW GU HT IS JR LP NZ
BOTKB EXDFR WSTRG BVAJP
VAFKE BRSRC TIQEL DBHZX
OKLEB ADAXP LICYQ THTQC
FHTQX ANXDX KRVT
```

Message no. 94 of 24 Sep 1941

```
B231 szi DRI AQ BO CM DP EW FT HS JZ KX LU
ABPQX PWCQF EZLPX GENCL
BOXJF VWWPX OOGLR IPJKO
UIOTC TNSLZ DKYYJ QNTVC
TMPLU OAUNE SZVKX RCTMH
M
```

Message no. 203 of 14 July 1941

```
B531 lwb BER BT CH DR EW FU GK JO LV MS PZ
CFYZR NFOSO IFKXN EMBCX
CWMSC MORVY WSVHF BZJHN
EMQFW ZQOLU IZBFF BSNKS
QXSHR DAMFR SESGJ JD
```

Message no. 233 of 18 July 1941

```
B425 agm QAY DM EP FL HI JR KY NQ OU SW TZ
XNRLR QKXET VPZQO HSXMB
```

```
IZPHT CTRMA UZYST JIMDU
YOZBF RTZOU HBGOR OUVRQ
EJRDR JHZPZ IBQQH KMMJZ
CIIRC UOLXL CIOQK HRLIG
GFJFT LLGDR ARDZQ UQKLT
K---Y KRUVF ULBQL AYRZV
JFULC GQJXF JURMU RSELY
FVFOK UHYUH SYLOM EFYAI
IP
```

Acknowledgments

The authors gratefully appreciate the support of Michael van der Meulen, who gave us access to his collection of authentic German Enigma messages. His generous gift is the very foundation of the described codebreaking project and without it the project would never have been started. Very special thanks are due to Bernhard Richt for the always inspiring crypto talks and especially for his contributions in clarifying the exact number of the "accidental" plaintext letters. As always we are very grateful to our good friend and fellow researcher, Geoff Sullivan, whom we wanted to have on board as a co-author but who this time decided to support and rally us from the sideline. We sincerely thank Ralph Erskine for the thorough review and his most valuable comments.

About the Authors

Olaf Ostwald is a microwave engineer working with Rohde & Schwarz in Munich on the design of electronic measuring equipment. He is interested in historical cryptography and cryptanalysis, especially in the techniques of breaking the Enigma.

Frode Weierud is a retired electronics engineer formerly employed by the European Organization for Particle Physics (CERN) in Geneva. Cryptography has been his main interest for close to 50 years. His cryptological research is focused on cipher machines and cryptanalytical techniques combined with a deep interest in all historical aspects.

References

[1] Anon. 1945. *History of Hut 6, In Three Volumes*. UK National Archives, HW 43/70–72.

[2] Bauer, C. P. 2013. *Secret History: The Story of Cryptology.* Boca Raton: CRC Press.

[3] "Crypto Box Challenge." 2007. Crypto Box Challenge by Dirk Rijmenants, http://users.telenet.be/d.rijmenants/en/boxchallenge.htm (accessed 5 April 2016)

[4] Davies, D. W. 1999. "The Bombe A Remarkable Logic Machine," *Cryptologia*, 23(2):108–138.

[5] Deavours, C. A. 1977. "Unicity Points in Cryptanalysis," *Cryptologia*, 1(1):46–68.

[6] Enigma Simulator by Dirk Rijmenants, http://users.telenet.be/d.rijmenants/en/enigmasim.htm (accessed 5 April 2016)

[7] Friedman, W. F. 1920. *The Index of Coincidence and Its Applications in Cryptography.* Riverbank Laboratories, Publ. No. 22, Geneva, IL. Reprinted by Aegean Park Press, 1987. http://math.boisestate.edu/~liljanab/MATH509Spring2012/IndexCoincidence.pdf (accessed 5 April 2016)

[8] Gillogly, J. J. 1995. "Ciphertext-only Cryptanalysis of Enigma," *Cryptologia*, 19(4):321–413.

[9] Hamer, D. H., Sullivan, G., and Weierud, F. 1998. "Enigma Variations: An Extended Family of Machines," *Cryptologia*, 22(3):211–229. http://cryptocellar.org/pubs/enigvar.pdf (accessed 5 April 2016).

[10] Hinsley, F. H. et al. 1981. *British intelligence in the Second World War, Vol. 2.* London: Her

Majesty’s Stationery Office (HMSO).

[11] Jackson, J. ed. 2014. *Solving Enigma’s Secrets: The Official History of Bletchley Park’s Hut 6*. Redditch: BookTowerPublishing. (Edited version of [1]).

[12] Kruh, L. and Deavours, C. 2002. “The Commercial Enigma: Beginnings of Machine Cryptography,” *Cryptologia*, 26(1):1–16.

[13] Menezes, A. J., van Oorschot, P. C., and Vanstone, S. A. 1996. *Handbook of Applied Cryptography*, Boca Raton: CRC Press.

[14] Oberkommando der Wehrmacht. 1940. *Schlüsselanleitung zur Schlüsselmaschine Enigma*, H.Dv.g. 14, Reichsdruckerei, Berlin. http://www.ilord.com/enigma-manual1940-german.pdf (accessed 5 April 2016).

[15] Ostwald, O. and Weierud, F. 2016. “History and Modern Cryptanalysis of Enigma’s Pluggable Reflector,” *Cryptologia*, 40(1):70–91.

[16] Rijmenants, D. 2010. “Enigma Message Procedures Used by the Heer, Luftwaffe and Kriegsmarine,” *Cryptologia*, 34(4):329–339.

[17] Shannon, C. E. 1949. “Communication Theory of Secrecy Systems,” *Bell System Technical Journal*, 28(Oct):656–715. http://netlab.cs.ucla.edu/wiki/files/shannon1949.pdf (accessed 5 April 2016)

[18] Sullivan, G. and Weierud, F. 2005. “Breaking German Army Ciphers,” *Cryptologia*, 29(3):193–232. http://www.tandf.co.uk/journals/pdf/papers/ucry_06.pdf (accessed 5 April 2016)

[19] Welchman, G. 1982. *The Hut Six Story: Breaking the Enigma Codes*, London: Allen Lane.

[20] Williams, H. 2000. “Applying Statistical Language Recognition Techniques in the Ciphertext-Only Cryptanalysis of Enigma,” *Cryptologia*, 24(1):4–17.