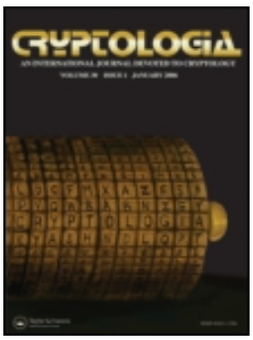

Taylor & Francis
Taylor & Francis Group

Cryptologia

ISSN: 0161-1194 (Print) 1558-1586 (Online) Journal homepage: http://www.tandfonline.com/loi/ucry20

Human factors and missed solutions to Enigma design weaknesses

Harold Thimbleby

To cite this article: Harold Thimbleby (2016) Human factors and missed solutions to Enigma design weaknesses, Cryptologia, 40:2, 177-202

To link to this article: http://dx.doi.org/10.1080/01611194.2015.1028680

CRYPTOLOGIA
2016, VOL. 40, NO. 2, 177–202
http://dx.doi.org/10.1080/01611194.2015.1028680

Taylor & Francis
Taylor & Francis Group

Human factors and missed solutions to Enigma design weaknesses

Harold Thimbleby

ABSTRACT

The German World War II Enigma suffered from design weaknesses that facilitated its large-scale decryption by the British throughout the war. The author shows that the main technical weaknesses (self-coding and reciprocal coding) could have been avoided using simple contemporary technology, and therefore the true cause of the weaknesses is not technological but must be sought elsewhere. Specifically, human factors issues resulted in the persistent failure to seek out more effective designs. Similar limitations seem to beset the literature on the period, which misunderstands the Enigma weaknesses and therefore inhibits broader thinking about design or realising the critical role of human factors engineering in cryptography.

KEYWORDS

Enigma; human factors; reciprocal coding weakness; self-coding weakness; situational awareness

1. Introduction

The Enigma cryptographic machine was patented by Alfred Scherbius in 1918 and was very widely used particularly during World War II, with the Germans confident of its security [2, 13]. Its use continued around the world well after the war, with the British keeping secret that it had been broken on an industrial scale.

There has been increasing interest in why the Germans were over-confident in the Enigma's security [21]. There was certainly systemic failure in the Axis intelligence system, but this article shows that some of the Enigma's weaknesses would have been avoidable had the Axis powers thought through facts that they "almost" knew but were evidently unwilling or unable to acknowledge. The Enigma and its operational procedures were continually reviewed throughout the war, so the persistent oversight to improve its design is surprising.

This article argues that the human factors of *design* are at least as important as the human factors of the *use* of a cryptographic, or indeed of any system. Any failures in design affect all use regardless of operator culture or training; design culture is critical and underpins the strengths and weaknesses of all use. Arguably the human factors of design should be the higher priority: Design can be undertaken in a managed, experimental, reflective environment, and

CONTACT Harold Thimbleby harold@thimbleby.net Swansea University, Swansea SA2 8PP, United Kingdom.

therefore can in principle be done optimally. Design errors can be fixed. In contrast, particularly in war, operators work in harsh conditions, and it is hard to work well. Use errors may be catastrophic.

One therefore expects and should design for use error, but more strategically, one should expect and should design for error in the design process itself. Errors happen primarily because we do not see them happening, and in design any unnoticed and unremarked errors will induce further (preventable) problems in use.

The purpose, then, of this article is to explore some of the design errors and wider misunderstandings in one system, the German Enigma, and to speculate on the human factors culture in which they occurred.

2. The role of human factors

The principles that a cryptographic system should be memorable, easy to use ("*usage facile*"), and not be stressful for its operators were first stated explicitly by Auguste Kerckhoffs in 1883 [14]. However, the scientific study and generalisation of these formative human factors ideas did not occur until much later.

Human factors is concerned with the human aspects of the operation of systems, often emphasising the possible failings of humans. Blaming the "human factor" or saying the cause of an incident was "human error" has become just misdirecting jargon to blame a human operator for a mistake. Properly understood, however, *human factors* is the scientific study of human behaviour with particular reference to work (e.g., operating machinery or flying aircraft): demanding tasks that involve attention and multitasking, phenomena that are known to lower human performance in predictable ways. Human error, for instance, can be induced by maintaining attention for a long period (causing fatigue) or multitasking (which induces omission errors). Human factors is not the study of human "foibles" but of the consistent and predictable factors of performance and behaviour that are common across individuals.

Human factors is particularly interesting because everyone (i.e., the people using, not using, or studying it) are humans and therefore subject to the same bias and error that is the very concern of the field. It is easy for people to ignore human factors precisely because they are human! Whereas bad engineering fails and cannot be ignored, people routinely make decisions without the benefit of human factors insights and therefore rarely wonder about trying to make decisions that could be or could have been better informed by human factors. In everyday life, errors rarely happen deliberately, but in war, errors are deliberately induced and manipulated by an enemy (sometimes as effectively as stage magicians misdirect us into seeing the impossible), so understanding human factors is crucial.

For example, *confirmation bias* is the tendency to preferentially look for evidence confirming an hypothesis; thus, if the Germans believed the Enigma

was secure, they would tend to selectively argue from evidence that supported their view and tend to ignore contradictory evidence, for instance dismissing it as coincidence. And this happened repeatedly. It should be noted that confirmation bias is unconscious and not done deliberately; in fact, it is very hard not to succumb to confirmation bias, since once it has happened, it is almost impossible to be aware of the poor justification for a decision.

- When German radio controllers in 1943 formed a view that British aircraft were at some location, they ordered German fighter planes to intercept them. Those German planes were then detected where the British were supposed to be, and misidentifying them reinforced the idea that British planes really were there, so more interceptors would be sent, further increasing the risk of misidentification [12, p. 381]. The uncritical reinforcement of the idea is important; humans seek confirmation for an idea, get some, and then are reassured, but fail to properly collect and assess evidence that might not confirm the hypothesis.
- British bombers thought that switching on their IFF (Identification Friend or Foe) transmissions protected bombers from detection. Bombers returning successfully had indeed used this technique, but thanks to confirmation bias, nobody wondered about the absence of data from the bombers that had failed to return. Later, Enigma decrypts showed bombers were being shot down *because* of their IFF transmissions [12, p. 389].
- After an increase in U-boat sinkings in 1941, Admiral Doenitz suspected the British were reading Enigma messages [13]. The German High Command proposed many alternative ideas, such as leaks or aerial reconnaissance, and "exculpated Enigma" [13, p. 260]. The profoundly worrying possibility that the Enigma was broken thus did not need hypothesising. In fact, the Allies very rarely took action on Enigma decrypts alone without first providing deceptive cover (e.g., a fly-over appearing to engage in aerial reconnaissance) so the Germans had evidence to confirm its unbroken security.
- In 2015, despite best efforts, this article at one time had 31 unnoticed incorrect spellings (25%) of Enigma as Engima. Until a spellchecker was used, the author never noticed because on each reading he read the words he expected to read. For human factors reasons, certain sorts of mistake are very hard to see. (The errors were not just caused by the author's sloppy proofreading but were also missed by reviewers and others because *all* humans find proofreading difficult.)

Winston Churchill, in his 1919 letter to Lord Drogheda [5], wrote that cryptanalysis "depends for its success more on a study of the psychology of the persons sending out the messages […] than on careful study and analysis for which there is no time." Probably, here Churchill meant studying the personality and relationships of the sender rather than what we now think of

as modern human factors, but his advice was certainly adopted; as Batey, who worked at Bletchley Park, put it, "we pursued [the] operator's subconscious" [1].

Human factors, then, attempts to understand the principles behind such phenomena so we can better anticipate problems and evaluate solutions. *Human factors engineering* goes up a level: How can we design and build (engineer) systems that enable humans to work effectively? Thus, in the examples above, confirmation bias might have been managed by formal teamwork (e.g., crew resource management), and unnoticed spelling errors, for instance, could have been handled by designing the word processor to automatically run a spellchecking tool — it would then have been impossible for the author to overlook using it!

Enigma operators worked under wartime pressures using complex procedures, the importance of which they may not have fully appreciated. Herivel [10] gave a famous example, the *Herivel Tip* or *Herivelismus*: Under pressure, tiredness, or even idleness, the operator might not change the Enigma's rotor settings as formally required, thus creating a detectable pattern in rotor settings over several messages that could be exploited by cryptanalysts. Thus, the Herivelismus describes a human factors weakness in the use of the Enigma. Had human factors engineering been applied (with any awareness or suspicion of this potential *use* problem for the operators) the Enigma might have been *engineered* to be better; for example to automatically spin rotors to some random setting, thus avoiding this human factors weakness in its use.

For many human factors reasons (functional fixedness, the Einstellung effect, tunnel vision, confirmation bias, and even pride) people are unable or reluctant to notice, let alone free themselves from assumptions that hold them back. It is hard to notice, correctly diagnose and then admit that anything can be improved — it is hard to see how it can be improved more than already achieved. In contrast, human factors engineering thinking therefore add targeted variation, teamwork and redundancy to the original "bare" or naïve design process to challenge predictable human factors from multiple perspectives.

This article is concerned with how the Enigma might have been designed to be more reliable, given the issues of human factors. We show that it could have been improved within the reach of contemporary technology and its limitations at the time. Human factors influences the designers of systems, not just system operators: The designers of the Enigma worked under wartime pressures and somehow failed to spot flaws in the Enigma's design. These design flaws then induced operational flaws the operators themselves could not have worked around. Additionally, in wartime, the enemy could be expected to mess with people's minds, exacerbating problems; in peacetime, an operator error might lead to an incident that directly exposes human factors issues, but in war the enemy might conceal the incident to encourage further errors or actively use hoaxes, subterfuges, deceptions, lies, and other means to misdirect operators. The Ultra project did this systematically [21].

Of course, the terms human factors and human factors engineering are modern, but whatever they are called, there is no evidence that the Enigma's operating procedures were subject to any human factor-style analysis (which might have identified operational failures such as the Herivelismus), but the design of the Enigma was repeatedly evaluated [21], and features like extra rotors and the Uhr [30] were introduced. Every time, though, intrinsic weaknesses were overlooked. This itself is a human factors issue.

This article will show that avoiding the Enigma's designed-in self-coding and reciprocal weaknesses would not have been difficult or costly. What was difficult, because of human factors, was to notice and acknowledge there was any problem to solve. This blindness to the unnecessary cryptographic weaknesses of the Enigma is familiar as *complication illusoire* (security by obscurity), a well-known fallacy where one mistakenly believes that adding complexity increases security. The very effort invested (cognitively, socially, physically, economically) in increasing complexity makes it hard to realise, let alone admit, that the effort may be ill-conceived, a recognised consequence of the human factors effect called *cognitive dissonance* [26]. Perhaps the electromechanical complexity of the Enigma, which was cutting-edge and perhaps relatively obscure technology for its day (the obscurity unwittingly enhanced by secrecy), was sufficient to be *complication illusoire*?

It is possible the Germans thought the Enigma design so simple it obviously had no flaws, but this alternate explanation of its unnoticed weaknesses is unlikely as its repeated evaluation was used to justify arguments about its security [15, 16, 20, and so on]. Indeed, the British cryptanalyst Gordon Welchman enjoyed the irony of decrypting Enigma messages, asserting its continuing security [32].

This article is written many years after the events it discusses. A well-known human factors problem is *hindsight bias*, which, for example, bedevils accident investigations [22, 23]. We now know more than the protagonists possibly could have, and in hindsight some of them made ill-informed decisions with outcomes they would have wanted to avoid. At the time, though, the decisions they made were not foreseen to lead inevitably to the consequences that they did; that is, in hindsight there was only one path taken, and it could have been chosen differently, but at the time there were many alternatives, and it was not so easy to choose as wisely as it may now seem. We say that the designers (the teams of people responsible for the design and its requirements) at the time lost *situational awareness*, meaning by focussing on what seemed to be the task at hand, the designers lost full awareness of the larger situation, namely the risks of their critical design choices; yet they probably did not even realise at the time those choices were critical. Even so, human factors makes it clear that to perform the design task at all, one must lose some situational awareness. It is a trade-off: If you worry about everything, you cannot get anything done. In particular, expertise by its nature is and requires a selective focussing on

technical details at the expense of "distractions." Modern human factors suggests design should involve multidisciplinary teams with authority to raise broader situational issues to help triage the distractions [23].

In this article, then, "design error" means that the potential problems of the chosen design were readily available, and that alternative designs that avoided those problems to varying degrees were also readily available at negligible marginal cost using reliable, readily-available contemporary technology. In other words, it is misleading to attribute the failure of the Enigma for its intended purposes to its reflector or any other specific technical details; it failed because of a failure in human factors.

Just as Woods and colleagues argue [33] that human error should not be treated as the *cause* of accidents but, more profitably, should be seen as a *symptom* of underlying problems, this article argues that the flawed design of the Enigma should not be seen as a cause of its weaknesses so much as a symptom of underlying problems. In hindsight, then, we can see those underlying problems included a lack of human factors applied to its design. There can, of course, be no useful criticism in lacking something that did not exist at the time, but today we can learn to apply human factors in new designs (for cryptography and more) to help ensure success, and we can also improve our historical discussions so that misunderstandings of the oversights in the Enigma design are not perpetuated.

3. The enigma and its reciprocal and self-coding weaknesses

The Enigma is a portable, suitcase-sized 26-letter keyboard, much like a typewriter, but with 26 lettered lamps instead of paper. Each time a key is pressed, some lamp lights, which gives the code for that key press. In addition, there is a plugboard (Steckerbrett) and some rotors (which rotate on each key press), together which change the configuration of the Enigma: Repeatedly pressing a single key makes lamps light apparently randomly as the rotors turn and permute the internal wiring. The Enigma is an electromechanical polyalphabetic cipher machine; an authoritative (previously classified) description of the Enigma was written by Turing [29].

The Enigma has a four position rotary switch: external power, off, internal power (dim lamps), and internal power (bright lamps). When using a fully charged battery, the "dim" mode would be used to prolong battery life, and for a partly-discharged battery, the "bright" mode would be used. We do not show this switch in the circuits shown in this article; it is not relevant to cryptographic security, but it is relevant to mention the switch in the sense that the operator was evidently able to use additional switches, a point we will return to later.

To send a message, the Enigma rotors and plugboard were set up using a complex process based on the key for the message, which was obtained from a daily codebook. An operator would then type the message and read aloud

the lamps as they lit on the Enigma, and another operator would transmit those letters using Morse code. To receive a message, another Enigma would be set up correspondingly with the same rotor settings and plugboard wiring, then the Morse code letters would be typed and the lit lamps read out so another operator could write down the message on paper as it was decoded letter by letter.

There are further operational details beyond the needs of the present article, including some relevant for human factors analysis of Enigma operation (e.g., to understand the mechanism of the Herivelismus).

3.1. The number of permutations

Imagine walking up to an Enigma, knowing nothing about its internal workings. How many ways might its wiring permute the key-lamp connections? Key A could light any of 26 lamps, key B could light any of the remaining 25 lamps, and so on; there are in principle $26 \times 25 \times 24 \ldots \times 1 = 26! \approx 4 \times 10^{26}$ potential ways to wire up an Enigma for each configuration. Furthermore, each time a key is pressed, the permutation will change to another from the same set of possibilities.

However, the Enigma is not designed like this. Instead, it provides a *reciprocal code* so that ciphertexts can be decoded with the same key as the plaintext has been encoded; thus, if A was coded as B then (in the same Enigma configuration) B would be reciprocally (de)coded as A. The reciprocal constraint eliminates many of the possible permutations: There are only $\approx 5 \times 10^{14}$ ways to connect the Enigma so it implements reciprocal codes.

Herivel argued that without reciprocal coding, the Enigma would have been useless [10, p. 89]. In fact, a simple switch (see below) could have provided a code/decode option, which would have retained the operational advantages of reciprocal coding Herivel wanted, but without reducing the number of permutations. It is worth emphasising that the number of permutations could have been nearly squared by a minor modification to the Enigma wiring if anybody had thought to design it that way.

If this design flaw had been fixed, the number of permutations would not just be increased (by a factor of $\approx 10^{12}$), but insecure patterns in the permutations would also have been removed, in particular, undermining the diagonal board optimisation of the automated Bombe decryption effort [32]. Breaking the Enigma would have been very much harder, "almost, if not completely, washed out" in Welchman's words (quoted in full below), and as all the hundreds of Bombes in use would have been impacted, this design improvement would have drastically curtailed the volume of Allied intelligence.

The Enigma also has a *self-coding weakness* so no letter can code as itself, hence removing all permutations containing any self-coding substitutions, A↔A, B↔B, and so on. Interestingly, self-coding is reciprocal, so the self-coding weakness eliminates some reciprocal weaknesses. There are only $\approx 8 \times 10^{12}$ permutations that are both reciprocal and have no self-coding.

The derivation of this last number can be understood by considering the rotor assembly (Figure 1), with its 26 connections to the rest of the Enigma. The rotor assembly, in any configuration, shorts pairs of connections (e.g., connecting Y↔Z). Let $p(n)$ be the number of pairs possible for n connections. Clearly, $p(2) = 1$ as there is only one way to short two connections. For n connections, pick one and it can be paired with any of the remaining $n-1$ connections, leaving $n-2$ connections to be paired up in $p(n-2)$ ways; hence $p(n) = (n-1) \times p(n-2)$. Now, $p(26)$ is easily expanded as $25 \times 23 \times 21 \ldots \times 1 \approx 8 \times 10^{12}$.

In short, by these two design weaknesses, the complexity of the Enigma code is reduced by a factor of $\approx 5 \times 10^{13}$ (i.e., $4 \times 10^{26} \div 8 \times 10^{12}$) or (put another way) reduced to less than the square root; either way, a dramatic reduction occurs. It must be emphasised that these two weaknesses in the Enigma design, although often confused, interrelated, and incorrectly explained, are independent problems; they are independent, both cryptographically and electrically (Figure 2).

The familiar huge numbers in the literature (e.g., 3×10^{114} in [21], etc.) arise for two main reasons. First, most calculations multiply key configurations: For example, the rotor choices and plugboard wiring are part of the key, but every permutation they introduce has already been counted in the foregoing analysis;

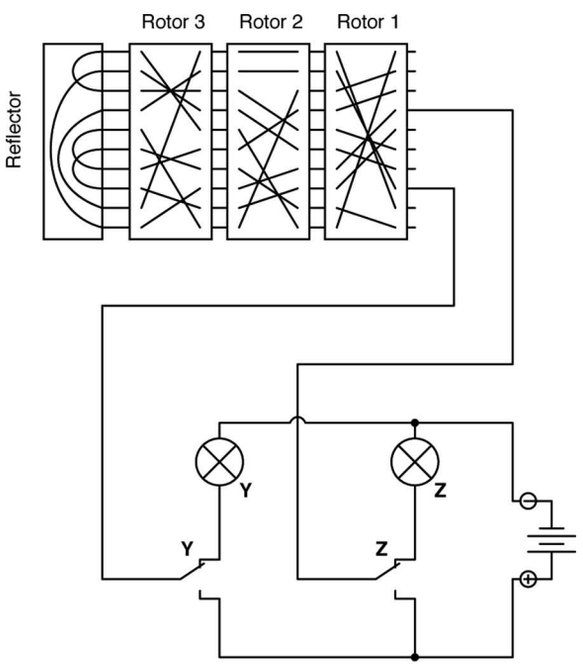

Figure 1. Enigma wiring without a plugboard. This circuit, showing only 2 keys instead of the full 26, is redrawn from a German Navy Enigma manual [13, p. 293]. (The wiring of the scrambler [the reflector and rotors] is purely indicative: The original diagram does not show details of its internal wiring. The actual scrambler has 26 connections to 26 letters, of course, not 10 as shown.)

	No self-coding weakness (letters can code as themselves)	Self-coding weakness (no letter codes as itself)
No reciprocal weakness (code/decode can be different)	A→A, B→C, C→D, D→B 24 permutations	A→B, B→C, C→D, D→A 9 permutations
Reciprocal weakness (code/decode identical)	A→B, B→A, C→C, D→D 10 permutations	A→B, B→A, C→D, D→C 3 permutations

Figure 2. Example codes representing all four combinations of self-coding weakness (or not) and reciprocal weakness (or not), illustrated with a four symbol alphabet, ABCD. The numbers are the numbers of permutations satisfying the combinations of design constraint (on an Enigma, every time a key is pressed the permutation changes to another within the constraints of the design). The four cases demonstrate that self-coding and reciprocal weaknesses are independent in principle; circuit diagrams in other figures in this article show that they are also electrically independent even when based on Enigma-style wiring.

many key configurations represent the same permutation. Secondly, on each key press the Enigma's permutation changes, and counting *message* permutations assuming messages are arbitrarily long, enough for n rotors to go through all of their 26^n positions (though some rotors "double stepped," hence reducing this number), rather than *letter* permutations, as here, increases the numbers further. Although it changes permutation on each key press, the Enigma cannot thereby increase the number of possible permutations!

However they are calculated, these large numbers are misleading as patterns (i.e., cryptanalytic knowledge) undermine them as a measure of security: A systematic cryptanalytic attack does not waste time trying so many permutations by brute force.

It seems very strange the Axis experts did not finish the line of thought: If it cannot be broken by brute force *therefore some other techniques may be used*, such as cribs or the dynamics of the rotor permutations (which the Bombes exploited [32]), to say nothing of stealing rotors or key setting books.

3.2. The enigma wiring

The basic wiring of the Enigma is shown in Figure 1. This circuit, including the layout, is taken directly from a German Navy manual [13, p. 293] and, interestingly, makes the electrical inevitability of the self-coding weakness obvious: If there were only two keys (and only two keys are shown in Figure 1), they can *only* code as each other. The scrambler (the reflector and set of rotors) either connects Y and Z in the diagram, or it does not. If it connects them, the coding is trivial, and if it does not connect, then nothing happens at all.

In other words, a two-key Enigma is useless, which might have been a clue that the 26-key Enigma had an underlying design flaw [27]. Figure 3 shows that for a two-key Enigma all the plugboard does is swap two wires that are anyway shorted by the rotor assembly: Even with a plugboard, it is still useless. As we shall later argue, the plugboard is wired into the wrong place, which compromises its potential.

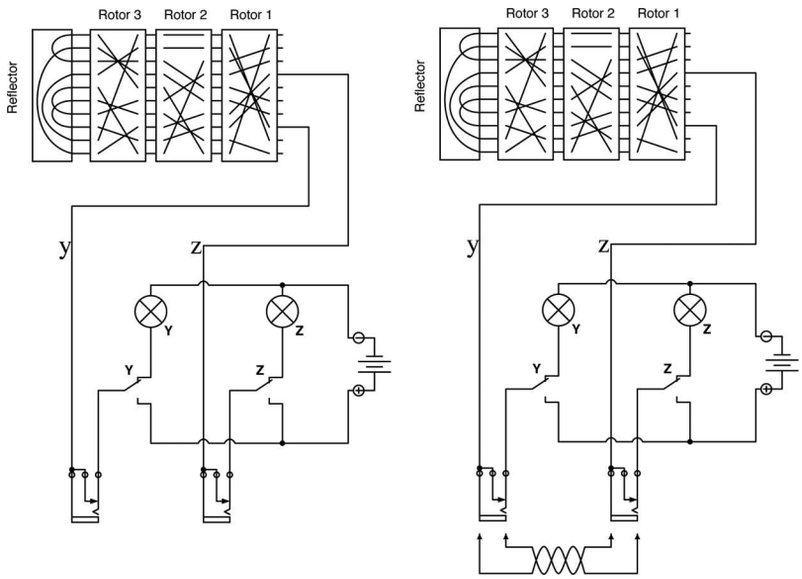

Figure 3. Original Enigma circuit with plugboard. When no plug is in a socket (left), each socket shorts its connections, so the wiring is unchanged. Connecting a cable (shown bottom right) effectively swaps over the wires marked *y* and *z*. The two-key Enigma's behaviour is *completely unchanged* whether or not a plugboard is used because swapping connected wires leaves them still connected. (The circuit shows a standard jack socket symbol electrically equivalent to the Enigma's original two-pin socket.)

Figure 1 does not show that the wiring from the keys to the rotors need not be direct; as Welchman pointed out [32, p. 210], the wiring from the keyboard (QWERTZ … by rows) was permuted into alphabetic order (ABCDEF …) as it connected to the rotors. In the 1930s, the German Army introduced the plugboard to allow an operator to change this permutation by rewiring. After the plugboard was set up by the operator (e.g., once a day), it would not be changed during a message.

The original plugboard cables were twin wires, and each cable effectively swapped lamp/key pairs of letters (e.g., if key Y was swapped with key Z, then lamp Y was also swapped with lamp Z), which obviously could not affect the weaknesses of the Enigma.

The Uhr was introduced in 1944 and replaced ten plugboard cables (offering $\approx 6 \times 10^{20}$ permutations) with 20 separate wires. It then provided a selection of just 40 prewired permutations, but it was no longer restricted to paired swaps, and (once set up) the permutations were more easily changed just by turning a 40-position rotary knob. The Uhr itself was not reciprocal, but electrically it was in the same place as and behaved much like adding another rotor, so it could not affect the weaknesses of the Enigma as a whole.

In addition to split rotors [2], other developments included a rewirable reflector also introduced in 1944 [31], which allowed arbitrary pairs of

connections (except J–Y, which remained fixed) and was rewired every ten days, effectively increasing the number of rotors in the field dramatically.

These various developments show the Germans tried to make the Enigma more secure, yet they did not impact its core weaknesses. The Germans were focussing more on increasing complexity than on reviewing fundamental assumptions. It is interesting to speculate what might have happened if basic human factors engineering questions had been asked: What assumptions had they, what was the evidence for these changes, would the increase in complexity of use encourage workarounds that would undermine any supposed benefits, and did they have a design checklist so they did not overlook critical design criteria [8]? (Confirmation bias implies, unless one takes careful precautions such as using a checklist, reasons why a solution may not work well will tend to be ignored.) In the event, both innovations were quickly compromised because operators had problems, and some messages were retransmitted (e.g., with and without the Uhr), which of course creates a serious security compromise.

3.3. An example of exploiting the self-coding weakness

It was routine to send dummy transmissions to camouflage changes in traffic volume before action, and in one case an Italian soldier sending a dummy message could not be bothered to code a random message but instead just pressed a single letter on his Enigma repeatedly. Mavis Batey (neé Lever), working at Bletchley Park, noticed the Enigma intercept had no letter Ls, and because of the self-coding weakness of the Enigma she realised the message was probably an idle transmission of just that letter, L. Testing this insight, she was thus able to reconstruct the key of the message, and hence was able to decrypt other messages [13, p. 139].

The following is from an Enigma operator's manual:

> **VI. Typing mistakes and failures.**
>
> 20. If mistakes happen during encryption by pressing the incorrect keys, omitting letters, or double-typing of keys, or through failures of the machine, (lamps not lighting up) the encryption needs to be repeated from the beginning of the message. In that case, the key in the windows must be reset to the initial state. To avoid repeating the whole encryption process, if a mistake happens towards the end, it suffices to press any key as often as letters have been encrypted before the mistake." [17]

These instructions to "press any key as often as letters have been encrypted before the mistake" seems like direct encouragement to repeat the mistake Mavis Batey exploited! Requiring messages to be transmitted without error puts pressure on the operators, and not only is this recommendation in the operator manual bad practice, but retransmitting *almost the same* message (i.e., the corrected message) under the same key is also ill-advised. This is to say nothing of encouraging further errors that must then be corrected, further increasing the risk of compromises. Obviously, some errors may be critical and

need fixing, but simple spelling mistakes would be far better left uncorrected to avoid retransmission.

Good human factors, as illustrated here, recognises that operators make errors. Bad human factors engineering then requires operators to correct errors insecurely. Good human factors engineering might have taken the instructions in the manual for the operator (which prove somebody thought about the problem) as a hint for a feature for the designer to think about: perhaps put a key on the Enigma to generate a random letter (e.g., by spinning a wheel).

3.4. Why were there weaknesses?

The Enigma has 26 keys and 26 lamps, and in a complicated way permutes key presses into lamps lighting. With contemporary technology, in its full generality this would require that 52 wires connect to a scrambling circuit (on the Enigma, the plugboard, rotors, and reflector) unless advantage was taken of technologies like Baudot, binary or multiple voltage representations to add more information than on/off to the wires can support. The later Baudot-based Geheimschreibers (Lorenz SZ40s, etc.) were too big and fragile to be as conveniently portable as the Enigma was, so other solutions would have been sought to the complexity.

The crucial electrical design decision in the Enigma was to use two-way switches for the keys, which connected a common wire to either the battery or to a lamp *but not to both*. When a key was not pressed, its lamp could come on using the same wire. This design decision reduced 52 wires (26 keys plus 26 lamps) to just 26 wires, shared in pairs of keys and lamps. All the rest of the circuitry could then be correspondingly simplified with half the number of wires, seemingly very cost-effective. Figure 4 illustrates this deceptively seductive design decision. Confirmation bias suggests that the Enigma design

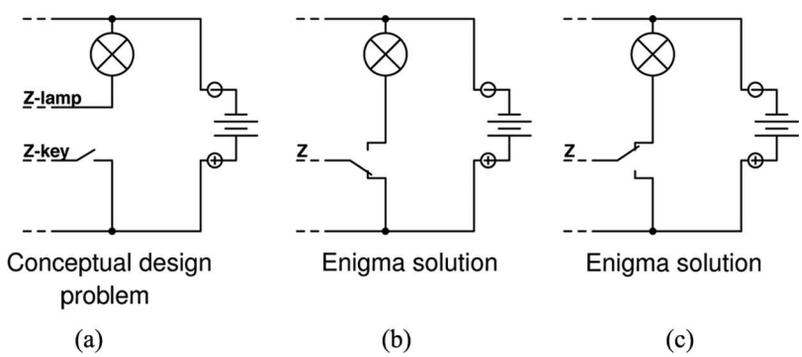

Figure 4. (a) For an arbitrary substitution cipher, letters (keys) and encoded letters (lamps) must be independent, so a letter's key and lamp each require separate connections, which means two wires per letter: 52 connections in all. (b, c) The Enigma design (e.g., Figures 1, 3, and 5) seductively halves the number of letter wires from 52 to 26. (Figure 10 shows how an Enigma might have been designed with 52 wires.)

thought process (whether Scherbius or the Axis powers) would be unlikely to look further for any potential problems in this design choice.

On the Enigma, if a key is pressed (letter Z is shown pressed in Figure 4b), one terminal of the corresponding lamp is necessarily disconnected, so it cannot light, hence the self-coding weakness. The reciprocal weakness also follows because once each key and its corresponding lamp share a connection, then the key and the lamp must behave the same way (key pressed and lamp lit share a single wire, so they cannot be independent). For any key "A":

- If key A is pressed, it disconnects lamp A, so lamp A *cannot* light, hence the self-coding weakness.
- If key A makes lamp B light (A → B), say, then as A and B are connected by a single wire, key B *must* make lamp A light (B → A), hence the reciprocal weakness.

The plugboard as wired on this common key/lamp connection (Figure 3) could not affect the weaknesses. Turing explained the Enigma using a diagram similar to the one shown in Figure 5, and he immediately noted that if the result of pressing a key $\mathcal{P}$ is that $\mathcal{G}$ lights, then pressing $\mathcal{G}$ in the same configuration lights $\mathcal{P}$ (reciprocal weakness) and pressing $\mathcal{G}$ can never light $\mathcal{G}$ (self-coding weakness). *This behaviour remains true regardless of the scrambling of connections*, including any scrambling introduced by the plugboard.

Since a wire (e.g., in the reflector) behaves the same in both directions, it might seem that the weaknesses caused by the key switches could be circumvented by connections using diodes (which allow current to flow in only one direction). However, this does not work, since if A → B (but not B → A because of the diode) and, say, B → Z, then the two diodes will conduct in series, hence

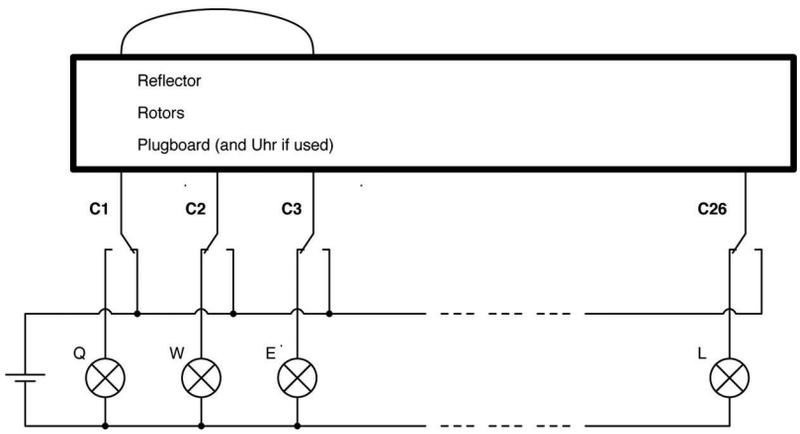

Figure 5. The German Enigma, based on an explanatory circuit from Turing [28, p. 2]. Using Turing's example, the key Q is pressed (shown by the C1 switch position), and supposing C1 and C3 are somehow connected through the scrambler, then lamp E lights. Turing glosses details of the keyboard–plugboard–rotor–reflector wiring, which is here signified by the rectangular block. (The keyboard and wiring is not in the modern QWERTY/QWERTZ order but enter the rotor assembly in alphabetical order [32, p. 210].)

A → Z *as well*, which would be incorrect. In short, it would be far easier to use 52 wires than work out ways of circumventing the problems caused by using only 26 wires! One infers that the Germans did not realise there were problems or that the cost of having 52 wires was excessive (though negligible compared to the cost of losing a war). The Enigma *looks like* it works well with 26 wires, and confirmation bias encourages the designers to believe it *really* works; why bother looking for problems when there are so many advantages?

3.5. The weaknesses discussed in the literature

It is an oft-repeated misconception that the original reflector made the Enigma reciprocal and forced the self-coding weakness. For example, in the authoritative 2014 biography of Gordon Welchman, we find the following:

> It was the reflector which delivered the reciprocal characteristic of the machine and thus also guaranteed that a letter could not be encrypted as itself [the self-coding weakness]. This was a weakness that the Germans were apparently prepared to accept as it provided the machine with the extremely useful property that the encrypting and decrypting processes were the same and one machine could do two tasks. This would have greatly simplified the training of their Enigma operators. [9, p. 203]

In fact, the self-coding weakness does not follow from the reciprocal characteristic. Rather, the problem is that a single wire for each letter connects to the rotors, thanks to the key switch, and it is this that ensures the self-coding weakness (Figure 4).

Even the training and ease of use issue is a misconception: Anybody capable of operating it would surely not have found the distinction between encrypting and decrypting a problem. Figure 6 shows "one machine could do two tasks" can be achieved without any more than a simple switch.

It seems that focussing on what a hypothetical, non-reciprocal Enigma's key settings should be for decrypting is baffling, and probably inhibited people both during the war and in the post-war literature from thinking of modifying the design so the same key could be used for both encrypting and decrypting processes.

Welchman himself wrote about the Enigma plugboard wiring [32, p. 168–169]:

> Suppose that, at any point in the war, the Germans had simply issued sets of single-ended connectors to replace the sets of double-ended ones. This would have meant that the upper socket corresponding to each letter, say X, would be connected to the lower socket of a specified letter, say Y. The lower socket corresponding to X would also be connected to the upper socket of some letter, but this letter need not be Y. […] our ability to accomplish the actual codebreaking would have been almost, if not completely, washed out […] the diagonal board would have been out of business.

This is incorrect. As we discussed above (section 3.4), no rewiring of the plugboard as Welchman described (which amounts to a variant of the Uhr) could have had this impact or affected the diagonal board. As Figure 5 makes

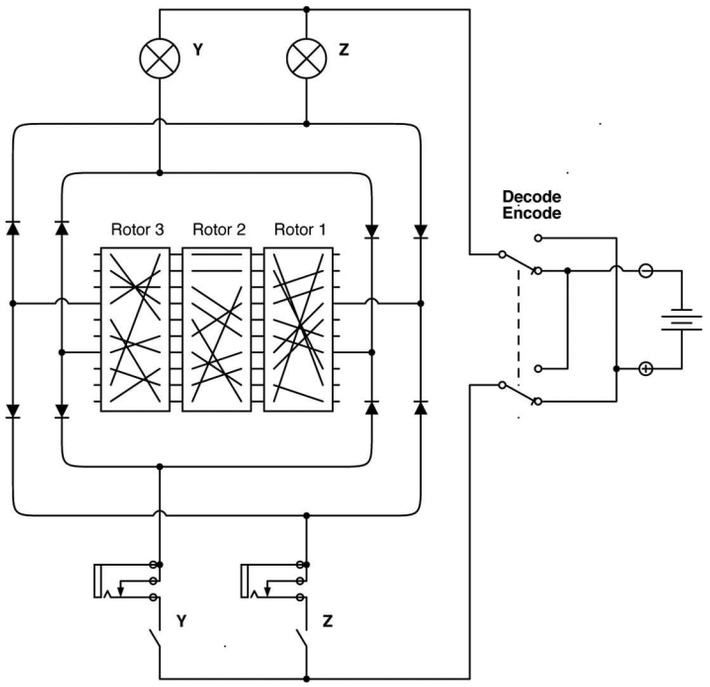

Figure 6. This elegant circuit uses original Enigma rotors and avoids both the self-coding and the reciprocal weaknesses (note the new location of the plugboard). As an alternative to diodes, either a more complex switch or a relay could be used, or even a rotor basket (see text). The point is that in principle the weaknesses can be avoided. (If one requires the extra security of a choice of reflector, either another rotor can be added or the techniques illustrated in Figure 10 may be used.)

clear, the plugboard, howsoever it might have been modified, plugged or wired (while it remains in the same part of the Enigma circuit) *cannot* achieve anything electrically more challenging than the restricted permutations possible using any variation of rotors and reflectors. Even automatically rotating, they hardly challenged the diagonal board. We know from his arguments with Thomas (Tommy) Flowers, the electronics lead on the Colossus, that Welchman was not good at electronics [7], but this misunderstanding is still a surprising claim from the inventor of the diagonal board.

Consider Figure 7, which is redrawn from F. L. Bauer's authoritative cryptography textbook [2, Figure 49, p. 107]. On the facing page, he said,

> The rotor solution, however, had the consequence no letter can be encrypted as itself. [2, p. 106]

This is mistaken: The problem is in the wiring of the switches, as is evident from the figure and as we explained above. The circuit in Figure 7 shows that pressing key e lights lamp M and that lamp E is disconnected, and therefore it *cannot* light (see also the caption of Figure 5, which puts the argument in Turing's words). The problem is due to the key switch wiring, not to the rotor solution.

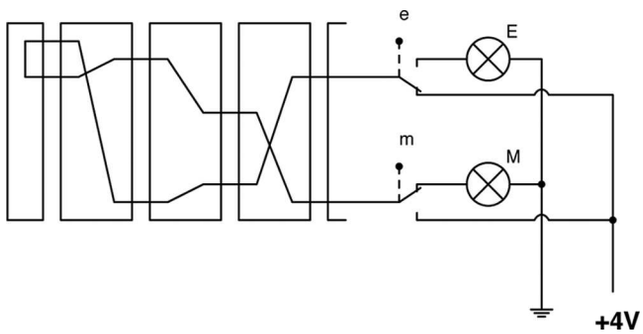

Figure 7. The Enigma with a reflector, three rotors, and no plugboard, redrawn from [2, Figure 49, p. 107]. The state shown in the diagram is key E pressed and lamp M lit.

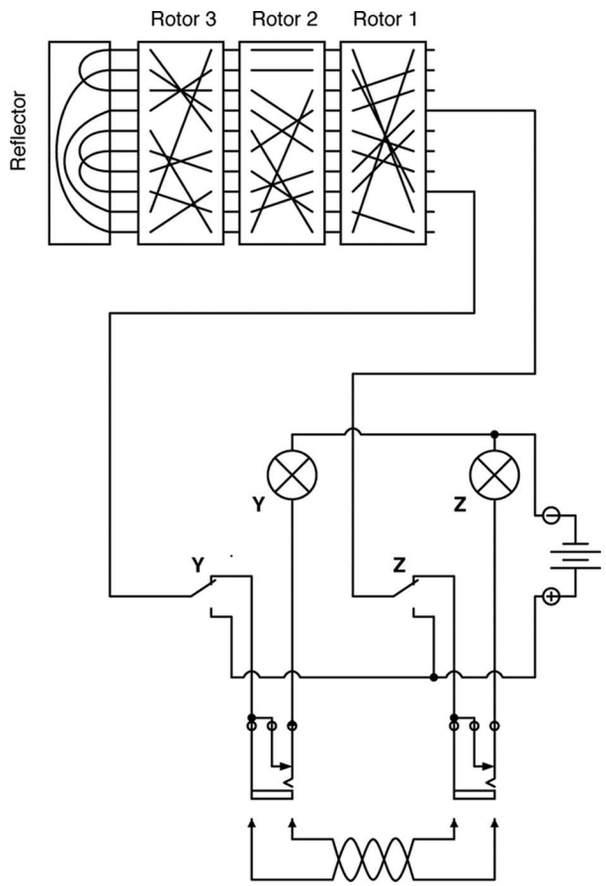

Figure 8. This simple variation of the Enigma, which has the plugboard connected directly to the lamps instead of to the key switch common wires, avoids the self-coding weakness (it swaps lamps alone, not lamps *and* keys as on the original). Had this design been considered, the obvious next step would be to make the plugboard wiring change with the rotor to stop the self-coding being fixed per message.

3.6. Possible alternative designs

All the following ideas avoid various weaknesses. Most retain a reflector: Again, the reflector has nothing to do with the problems (other than being a part of the circuit).

- Figure 8 shows that changing the wiring position of the plugboard avoids the self-coding weakness while retaining both the rotor and reciprocity (more precisely, it has an avoidable self-coding weakness, dependent on the plugboard wiring).
- Figure 9 shows that the original reflector can be used in a circuit that achieves reciprocal coding but without the critical self-coding weakness. The circuit relies on the special behaviour of neon lamps, described as early as 1921 [19], and widely available during the war.

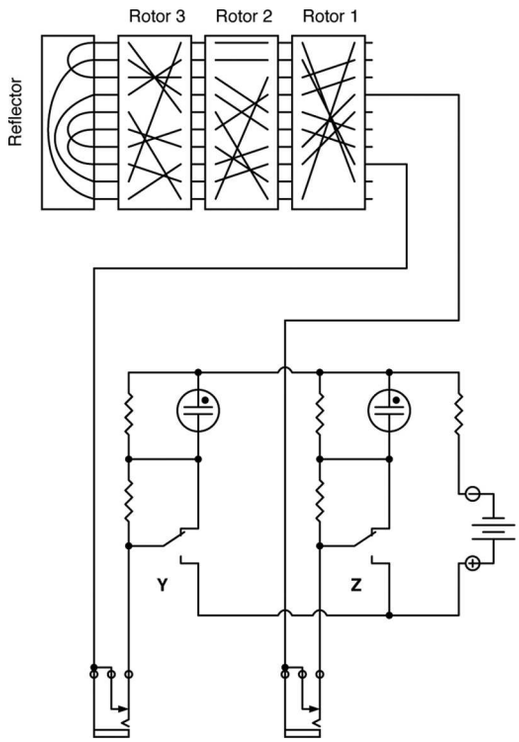

Figure 9. A simple alternative Enigma providing the original reciprocal coding but without the self-coding weakness. The design uses the original reflector, but some rotors have a missing connection (here, rotor 1 shows a missing connection). If key Y is pressed, neon Z lights except that with probability 1/13 (not 1/26 without making other changes). The missing connection in rotor 1 is selected, and neon Y lights instead. Note that although only one out of 26 pins in the rotor is directly disconnected, another wire is effectively disconnected in the return path from the reflector, hence two pins out of 26 are actually disconnected. The circuit works as follows: If key Y is pressed and connects to Z, neon Z ionises and clamps the voltage across the neon Y potential divider, so the voltage across neon Y is too low for it to ionise; otherwise, if key Y is pressed and does not connect to any other lamp, the voltage is not clamped and is enough for neon Y to ionise. (The battery must be perhaps 100 V instead of the original Enigma's 4 V.)

- An elegant solution is to use diodes (wired as 26 bridge rectifiers), as in Figure 6. Instead of physically reversing the rotors or needing a large multipole switch or relay, a simple switch reverses the battery polarity so the diodes then reverse all connections to the rotors. Note that diodes were widely available; a circuit for a homemade wireless set using one was published in 1935 [4]. Whether diodes would have been reliable enough in the field is another matter; the point is that if there is one solution to a problem, then others can be sought. For example, one or more relays could be used (26 two-pole relays, 13 four-pole relays, or one relay with 52 poles, the former choices being easier to field service).
- Figure 10 shows that using rotors with 52 connections rather than 26 avoids both weaknesses. It does not have the reciprocal weakness, but the reflector could be wired to ensure reciprocal coding if it was actually required; for example, if key A connects to lamp B, then a wire must be added in the reflector to connect key B to lamp A. No doubt this now optional pattern in the mechanism would raise concerns as it is obvious (at least in hindsight) that one does not want a cryptographic mechanism with such patterned constraints. Of course, in the original design the reciprocity was not optional, regardless of how the rotors and reflector were wired.

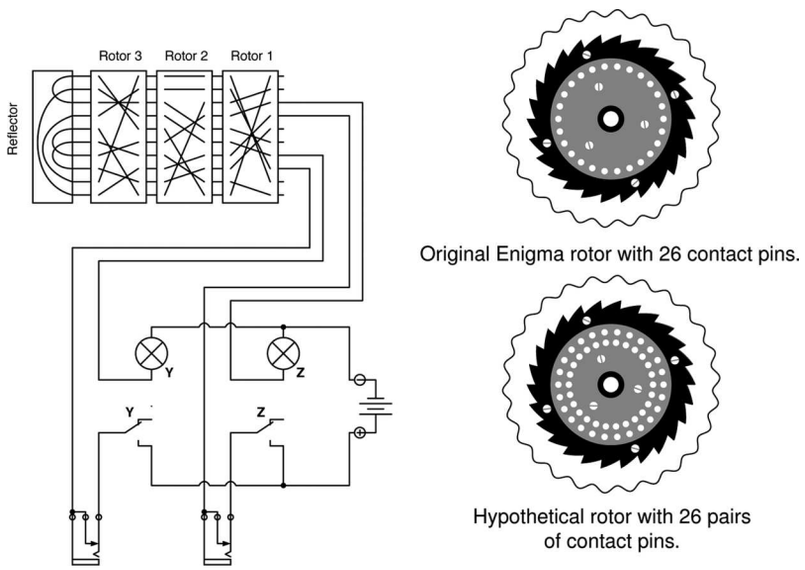

Figure 10. If the keys and lamps are separately wired to the rotors in pairs, the weaknesses are avoided despite using a reflector. The rotors still have 26 mechanical positions but with paired contacts, as illustrated (right). The reflector connects the outer ring to the inner ring of contacts, making a further permutation as it does so. The plugboard is only used once in the circuit, so its original reciprocal weakness is avoided. For completeness (not shown because it makes the diagram messy), this circuit needs either diodes as in Figure 6 or an encode/decode switch to swap the lamps and keys.

- In Figure 6, instead of doubling the rotor connections at one end, 26 normal connections can be made at each end. The rotors could then be in a basket (as on the British TypeX) to be removed as a piece, so when coding, the basket "points left," and when decoding the basket "points right." A removable basket would have had further advantages: It could have encouraged cryptographers to think of new operational procedures, not just changing one rotor at a time, but using *more* baskets, even changing them mid-message, and so forth. There could have been a single two-way multipole switch (or relay) to achieve the same effect as reversing the basket. This approach was used on the US SIGABA machine.

4. Broader discussion

It is possible that the precise electrical behaviour of the Enigma was pretty much a closed book to German cryptanalysts. Almost certainly, they would all have been mathematicians, not least because mathematicians need less space than engineers and are therefore cheaper [21, p. 189]. Whatever the explanation, the electrical problems never came to proper attention.

Contemporary Allied analysts, too, seem confused in their modern summaries of the issues. Welchman [32] said it would have been difficult to avoid the self-coding design fault, and Herivel [10, p. 89] said the reciprocal property was essential, specifically saying the Enigma would have been useless without it. Yet, avoiding self-coding is trivial, and reciprocity (the ease of use that the same *machine* can code and decode) can be achieved without the Enigma's reciprocal weakness (that the *code* itself is reciprocal), as we showed.

Herivel seems to focus so much on ease of *operational* use that he failed to realise (or if he realised he did not say) that adding a single switch to the *design* could have provided the ease of use he thought essential and at the same time would have avoided the reciprocal weakness. Given the depth of his overall analysis, it seems surprising he did not criticise the Germans for this oversight unless he, too, was blind to it (but, then, he is a mathematician providing a mathematician's analysis of the design, rather than an engineer providing an engineer's critique of it).

The modern, post-Ultra revelations, literature seems confused, too. There are many leading publications (e.g., [2, 9, 21]) that inaccurately describe or over-simplify the electrical properties of the Enigma (cf. Figure 7 from [2] for an example discussed above).

Interestingly, the UK TypeX and the US SIGABA designs were different, suggesting that the Axis powers could have had the same design insights but for human factors limitations, or perhaps that the overwhelming emphasis in the literature on the design of the Enigma and not its "competitors" has

perhaps trapped thinking into some of the Axis blindspots. More charitably, the explanations in the literature are "good enough," and a precise discussion of the Enigma's historical wiring properties is generally of little interest, but (as the present article emphasises) the clear understanding of the design details *at the right time* would have been critical.

4.1. Fundamental design problems

All design involves multiple and generally conflicting trade-offs. For example, the Enigma should be light and portable, but this design requirement must be traded against its physical robustness: The lighter it is, the more fragile it becomes. Two fundamental trade-offs seem to lie at the core of the problems we have discussed with the Enigma.

First, the Enigma was designed and used by humans. Human factors is concerned with relevant issues such as the ease, simplicity, and effectiveness of those human processes, and these are well known to include factors such as openness and clarity, clear objectives, and so on. On the other hand, the Enigma was used in a theatre of war, where security and secrecy were required, within a cautious framework of need to know; if anybody knows clear objectives, they are a greater risk to security. In short, the easier and more reliable a device is to design or use, the more it seems in conflict with the paranoid goals of security. As it happens, unnecessarily, this unavoidable tradeoff was tipped the wrong way (from the Axis point of view) by cultural problems and blindspots to thinking clearly.

Second, the wiring was simplified to halve the number of wires needed. This design choice would have made the Enigma lighter and simpler, and correspondingly eased its production in large numbers. While those are tactical gains, the design significantly reduced the number of permutations the Enigma supported from the theoretical maximum. Worse, the way the wiring was simplified introduced specific patterns that could be exploited cryptanalytically. In particular, the Enigma's weaknesses applied to every message and key, and thus could be exploited in *a one-time cryptanalytic effort*, for instance, preparing the Jeffreys Sheets or building the Bombe that could then be applied to *any* message.

The operational advantage of reciprocal coding could have been weighed against the security weakness it introduces, but of course this is an unasked or pointless question if reciprocal coding is (or is thought to be) unavoidable. Almost certainly, had the Germans thought through this, they would have preferred security against ease of use (the Enigma already required several trained operators, so issues of ease of use were arguably not that significant). Indeed, the ease of use of the Enigma encouraged more use (e.g., for social communications) than was prudent [21], so making it harder to use would have made it more secure!

4.2. But not just human factors engineering

Although the Enigma's design weaknesses and their persistence were critical, its design was not the only factor in its overall failure, as the following types of issue make clear:

- Operational failings (conventional human factors problems) regardless of the Enigma design itself, such as depths (identical messages retransmitted using different keys), cribs, recycling of keys, and patterned behaviours such as led to the Herivel Tip. Operational failings also arose from poor operational procedures (human factors at play at management levels), such as the counterproductive rules for selecting key settings (e.g., forbidding "predictable" choices of rotor, such as remaining in the same place two days running, but doing so reduces randomness and hence increases predictability [25, p. 164]).
- Although extra rotors were introduced during the war, their mechanical behaviour was never revised. It was a serious weakness that for most messages at least one rotor remains stationary during use.
- Although reviewing the variations on Enigma designs would be a complex diversion for this article, it is noteworthy that some pre-war models did not suffer from the reciprocal weakness, some models had more random rotor movement, and there was variation between the army, navy, and air force models. In other words, better designs (and the knowledge of a space of design options) were in principle available to encourage more careful review of the chosen designs.
- While examples of possible Enigma decryption were investigated (and generally absolved the Enigma and its operational security), the Germans (particularly Hitler) were obsessed with human fallibility and imagined betrayal from soldiers or spies [21]. An Axis signals director argued [21, p. 152] that since "the deciphering of the Enigma is out of the question," the problem *must* be with people (possibly spies) operating it! Ironically, as this article argues, the preventable weaknesses in the Enigma design were themselves due to human fallibility.
- When there was suspicion the Enigma might have been compromised, the Germans preferred alternate explanations of the incidents, such as the Allies' superior direction finding technology. Given the *acknowledged* superiority of Allied direction finding, this was a more comfortable explanation than exploring the denied weaknesses of the Enigma.
- The Allies' *automated* decryption technology was a generation ahead of the manual, electromechanical Enigma, and therefore the Axis estimates of the time to break it were unwittingly invalid.
- Various attempts to check the security of the Enigma [21] were brief attempts to decrypt particular messages. The Germans did not

re-examine the design. The attempts were *ad hoc* and brief (e.g., a few weeks), whereas the Allies' attempts were systematic and lasted the entire war.

- The Axis powers held the false conviction that had the Enigma been broken, there would have been direct evidence of that fact. As it happens, the Allies went to extraordinary lengths and deceptive ruses to conceal the sources of their decrypted information.
- As late as 1982, the U.S. NSA director wrote to Gordon Welchman telling him not to talk about the technical details of the Enigma [9, p. 165]. If the Allies were wilfully ignorant (i.e., insistent on compartmentalising Enigma secrets) decades after the war, perhaps there may also have been a *wilful* ignorance of design issues during the war, a type of thinking that would have been exacerbated by the Axis compartmentalisation.

An astonishing array of further human factors problems is described brilliantly in Ratcliff's *Delusions of Intelligence* [21]. The Axis powers, unlike the Allies, had very compartmentalised, hierarchical, and competitive signals intelligence units, so common problems were rarely spotted. Such problems created a vicious circle, in that failure of the intelligence system led to less respect for it and reduced resourcing, creating a spiral of ineffectiveness and loss of prestige. When some successful intelligence did occur, jealousies eclipsed it. This psychological set, including the groupthink where the blindspots were enculturated and impossible to speak out about, conspired to ensure thinking about the Enigma's weaknesses was never effective. The Axis culture was in complete contrast to the clear and open Allies' culture [32].

Given that the Allies had very similar cipher technologies, the problems cannot have been entirely unilateral. Everyone, on both sides, believed that physical security and secrecy was paramount. The Axis thought the Enigma statistics and their understanding of the technology was sound, and hence their improvements focussed on improving physical security and improving secrecy (in the keys and operational procedures) rather than improving the underlying technology [3, p. 336]. With asymmetric codes and a hindsight understanding of the Kerckhoffs principles [14], we now know these assumptions were false.

Yet, the Axis powers had reasons to convince themselves, including not questioning their assumptions, that they were right about the Enigma's design and hence they sustained no persistent or creative attack on (i.e., serious thinking) their own handiwork, whereas the Allies were motivated to find flaws and had no delusions about the value of doing so. As both Herivel and Welchman make very clear, the difference between success and failure was very narrow, often seeming a matter of luck, but human factors tipped the balance away from the Axis powers.

5. Conclusions

The Enigma's self-coding and reciprocal weakness could have been avoided using contemporary technology. Whether our circuits would have helped during the war is impossible to answer, but the *insight* that there were problems to think about (and as shown here, thinking about them would have found feasible solutions) might have been significant. Recognising any weakness in the design may have stimulated thinking about other weaknesses, in the design or the processes that led to the design, as well as in the design of the Enigma's operational protocols. Certainly some incidents (e.g., Mavis Batey's break) could have been avoided.

Even more powerfully, if the Axis powers had noticed they had been missing insights, they might have reflected on the human factors failings that induced those oversights. That sort of counter-factual insight might have had a far more profound impact than just fixing technical defects in the Enigma's design.

The Germans and others seemingly did not reason (or did not reason clearly enough) about Enigma circuits, despite having simple electrical problems staring them in the face. The Navy could have identified a problem early on, since their widely-distributed circuit diagram was completely dysfunctional as a cryptographic machine. The simplicity of Figure 1 (if anybody thinks about it) makes this evident. *Simple designs make problems evident*: a point also noted by Thimbleby during his design of a simplified Enigma [27].

The real Enigma was complex, and it simply grew in complexity rather than anyone questioning its underlying design assumptions; its implicit assumptions are hard to see in the maze of wiring needed for 26 keys. Hence, *complex designs disguise problems*, and crucially, they require more effort to build or even partially understand, and therefore encourage *cognitive dissonance*, as mentioned above: the failure to see or think about potential problems when effort has been expended in creating those very problems. Whereas cognitive dissonance is why we fail to admit or notice problems after the fact, implicit in our discussion has been the psychological "tunnel vision" that complexity in itself hinders people seeing potential problems in the first place: Design requires so much concentration that designers lose what elsewhere is called *situational awareness* [6] and lose sight of wider issues.

Stepping back a bit from the details, Peter Medawar has called science "the art of the soluble" [18], in other words, recognising and addressing the problems to be solved, but in the artificial sciences [24] there is a danger the complexity of the things we build obfuscates the problems we might otherwise be thinking about. Tony Hoare memorably expressed the choice:

> There are two ways of constructing a system, one way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies. The first method is far more difficult. [11]

In the present article, we have raised two types of simplicity, which might be called abstract and concrete. There is the abstract simplicity that helps one think more clearly, the simplicity to which Hoare referred, and which the Germans failed to utilise even from their own abstracted circuit diagrams of the Enigma. The other sort of simplicity occurs in the drive for concrete construction (and, for example, lower costs, as in reducing 52 wires to 26): If the second sort is prioritised, even implicitly, it leads to tunnel vision, and the first sort may never have a chance to support clear thinking.

In war, the deficiencies you cannot see you should assume will be seen and exploited by your enemies, and so it follows one ought to thoroughly explore simple systems first in order to encourage broader thinking. More generally, in any safety-critical design endeavour, one should surround oneself with clear thinkers with different psychological perspectives whose thinking is less likely to be drawn into one's own loss of situational awareness, cognitive dissonance, tunnel vision, and other human factors handicaps.

Alan Turing's pre-war 1936 insight into the universality of computers [28], then the Colossus, and other developments during and just after the war led to today's computers. Today, it is far easier to simulate an Enigma on a computer than to reproduce the original electromechanical construction. There are even open source projects, implemented on computers, that emulate physically realistic Enigmas, such as the Open Enigma project at www.openenigma.com. GCHQ, the U.K. intelligence agency and successor to Bletchley Park, has developed a similar app, Cryptoy, to introduce students to cryptography. It demonstrates shift, substitution, and Vigenère codes, and simulates an Enigma.

Of course, the modern computer's removal of physical limitations sidesteps the trade-offs discussed above, as well as the need to make a cryptographic device limited in any way like a real Enigma, but the fundamental human factors problems remain: Both design and use are still subject to human factors limitations. Computer bugs are all too familiar, and they persist for many of the same reasons the design flaws of the Enigma persisted.

Finally, Herivelismus [10] (see section 2) should be considered one of the landmarks of human factors, and the failure to design it out, correspondingly, should be considered a landmark failure of human factors engineering. That is, computer programs get too complex and confuse programmers in just the same way the complexity of the Engima was confusing: Programmers too often add complexity because it is easier to do so than thinking carefully.

About the author

Harold Thimbleby is Professor of Computer Science at Swansea University, Wales; he is Emeritus Professor of Geometry, Gresham College, London. He built an electromechanical Enigma in 2002 to illustrate his Gresham lecture

on cryptography, and he has been fascinated by the machine ever since. His research interest is human error, particularly in complex healthcare systems; the Enigma design is relevant because its failures make a provocative analogue to healthcare system design failures. See http://harold.thimbleby.net.

Acknowledgments

The author is grateful for comments from Ross Anderson, Paul Cairns, and Paul Curzon.

Funding

This work was funded by EPSRC Grants [EP/G059063, EP/K504002, EP/L019272].

References

[1] Batey, M. 2010. *Dilly: The man who broke enigma*. Biteback Publishing.
[2] Bauer, F. L. 1997. *Decrypted secrets*. Springer.
[3] Budiansky, S. 2000. *Battle of wits*. Penguin Books.
[4] Camm, F. J. 1935. My two-valve superhet. *Practical Wireless*, reprinted, 2007, 83 (8): 62–64.
[5] Churchill, W. 1919. Letter to Lord Drogheda, March 28. Reprinted in Denniston, R. 2012. *Thirty secret years: A. G. Denniston's work in signals intelligence 1914–1944*. Polperro Heritage Press.
[6] Endsley, M. R. 1995. Toward a theory of situation awareness in dynamic systems. *Human Factors*. 37 (1): 32–64.
[7] Gannon, P. 2006. *Colossus, Bletchley Park's greatest secret*. Atlantic Books.
[8] Gawande, A. 2011. *The checklist manifesto: How to get things right*. Profile Books.
[9] Greenberg, J. 2014. *Gordon Welchman: Bletchley Park's architect of ultra intelligence*. Frontline Books.
[10] Herivel, J. 2008. *Herivelismus and the German military enigma*. M. & M. Baldwin.
[11] Hoare, C. A. R. 1981. The emperor's old clothes. Turing Award Lecture. *Communications of the ACM*. 24 (2): 75–83.
[12] Jones, R. V. 2009. *Most secret war*. Penguin Books.
[13] Kahn, D. 1996. *Seizing the Enigma*. Arrow Books.
[14] Kerckhoffs, A. 1883. La Cryptographie Militaire (Military cryptography). *Journal des Sciences Militaires*. IX (Jan): 5–38 & IX (Feb): 161–191.
[15] Koot, H. 2002. Expert's opinion on the Enigma ciphering machine. *Cryptologia*. 26 (2): 101–102.
[16] Kruh, L., and C. Deavours 2002. The commercial Enigma beginnings of machine cryptography. *Cryptologia*. 26 (1): 1–16.
[17] Lord, B. 1998. *Gebrauchsanleitung für die Chiffriermaschine Enigma*, January 12, 1938, tr. Kaiser, K. www.ilord.com/enigma-manual.html.
[18] Medawar, P. 1994. *The art of the soluble: Creativity and originality in science*. Heinneman.
[19] Pearson, S. O., and H. St. G. Anson 1921. Demonstration of some electrical properties of neon-filled lamps. *Proceedings of the Physical Society of London*. 34: 204–212.
[20] Ratcliff, R. A. 2002. How statistics led the Germans to believe Enigma secure and why they were wrong: Neglecting the practical mathematics of cipher machines. *Cryptologia*. 27 (2): 119–131.

[21] Ratcliff, R. A. 2006. *Delusions of intelligence: Enigma, Ultra, and the end of secure ciphers*. Cambridge University Press.
[22] Reason, J. 1990. *Human error*. Cambridge University Press.
[23] Reason, J. 2008. *The human contribution: Unsafe acts, accidents and heroic recoveries*. Ashgate.
[24] Simon, H. A. 1996. *Sciences of the artificial*, 3rd ed. MIT Press.
[25] Singh, S. 2000. *The code book*. Fourth Estate.
[26] Tavris, C., and E. Aronson 2013. *Mistakes were made (but not by me): Why we justify foolish beliefs, bad decisions and hurtful acts*. Pinter & Martin Ltd.
[27] Thimbleby, H. 2003. The reduced enigma. *Computers & Security*. 22 (7): 624–642.
[28] Turing, A. M. 1937. On computable numbers, with an application to the entscheidungsproblem. *Proceedings London Mathematical Society*, s2–42 (1): 230–265; & 1938. On computable numbers, with an application to the entscheidungsproblem. A correction. *Proceedings London Mathematical Society*, s2-43 (1): 544–546.
[29] Turing, A. M. c1940. *Treatise on the Enigma*. 152pp, HW 25, National Archives, UK. www.alanturing.net/turing_archive/archive/b/B05/B05-001.html.
[30] Ulbricht, H. 1999. Enigma Uhr. *Cryptologia*. 23 (3): 193–205.
[31] Ulbricht, H. 2001. Uncle Dick and other horrors of the Enigma. *Journal of Intelligence History* 1 (1): 44–53.
[32] Welchman, G. 2001. *The Hut Six story*. M. & M. Baldwin.
[33] Woods, D. W., S. Dekker, R. Cook, L. Johannesen, and N. Sarter 2010. *Behind human error*, 2nd. ed. Ashgate.