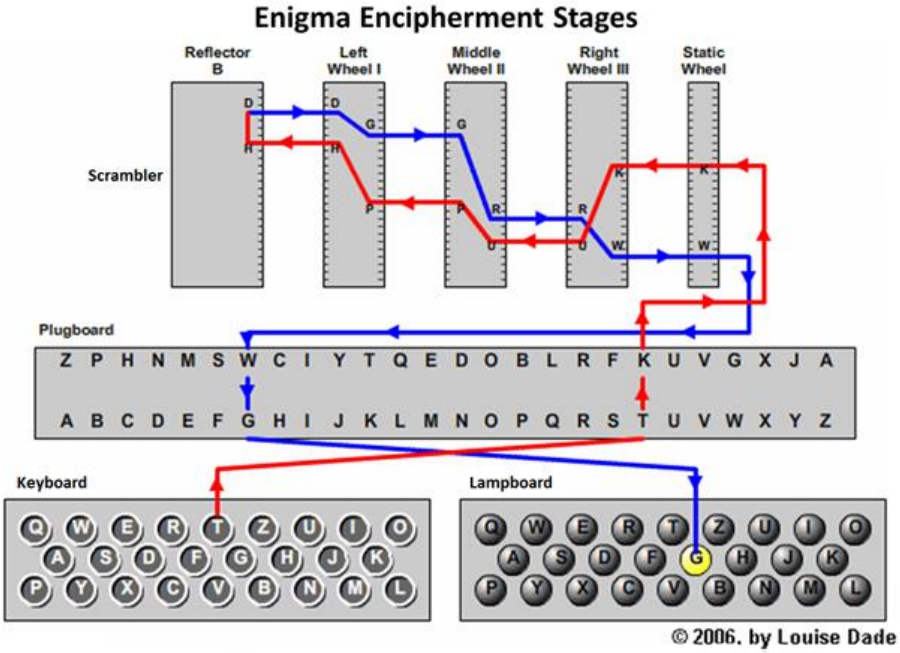
The Enigma Machine

Of the many encryption machines and schemes used during World War II, Enigma is probably the most famous. Invented by Arthur Scherbius, this supposedly unbreakable cipher turned out to be anything but. Allied cryptanalysts managed to break this code in the early days of the war, and the resultant intelligence, codenamed Ultra, proved instrumental in the Allied victory. The Enigma's mechanical components gave it its security, but a careful analysis of the patterns those components created will provide the insights needed to break Enigma.

How Did Enigma Work?

At its core, the Enigma consisted of the following components: a keyboard for entering a plaintext letter, a set of interchangeable scramblers that encrypts each plaintext letter into a corresponding ciphertext letter, a reflector that allows the Enigma machine to both encrypt and decrypt, a ring, a plugboard to further encrypt the ciphertext, and a display board, or lampboard, consisting of various lamps for indicating the ciphertext letter.



To encrypt a plaintext letter, an Enigma operator would press the relevant key on the keyboard. This sent a plaintext letter to the plugboard. The plugboard allowed the sender to swap some letters before they even entered the scrambler. Enigma operators had access to six cables, meaning that they could swap 6 pairs of letters. Later, the number of cables increased to 10, enabling operators to swap 10 pairs of letters.

This plugboard would send an electric pulse into the scramblers. Each scrambler has wiring entering it at 26 points. Within the scrambler, the wiring twists and turns before emerging at 26 different points on the other side.

Ordinarily, this combination of plugboard and scrambler would produce a regular monoalphabetic cipher that a dedicated cryptanalyst could crack in about an hour. However, the Enigma rotates the scrambler disk 1/26 of a rotation after each letter is encrypted! The rotation strengthened the cipher; but every 26 letters, the scrambler returns to the original configuration. This means that in any message longer than 26 characters, more than one letter will be encrypted with the same alphabet. At this point, the second scrambler comes in. The second scrambler will rotate only after the first scrambler has completed one revolution, thus raising the number of possible alphabets available to encrypt each letter from 26 to 676! The Enigma used three scramblers, all of which were removable and interchangeable. This meant that the scramblers alone meant that any one of 17,576 alphabets could encrypt a particular letter. This number would only increase later on, when the Germans raised the number of scramblers an operator could select from three to five.

Once the ciphertext had left the scramblers, it went to the ring and reflector.

Singh does not go into great detail on these subjects, only stating that the reflector allowed an Enigma machine to both encrypt and decrypt messages. However, all these components created so many possible alphabets to encrypt a letter that, if a dedicated cryptanalyst tested one alphabet every minute, it would take longer than the age of the universe to decrypt just one letter.

The Breaking of the Enigma-Part 1

The tale of Enigma's decryption begins with Polish cryptanalyst Marian Rejewski.

Rejewski was a member of the Polish cryptanalysis unit, the Biuro Szyfrów. Rejewski had at his disposal three vital items: a great deal of intercepted messages, all encrypted with Enigma; replica Enigma machines; and a clear outline of the protocols Enigma operators used to encrypt a message. These protocols would provide the first clear break into Enigma.

Because Enigma's strength relied upon the many different possible settings, it was vital for both the sender and receiver of a message to set their machines according to the same settings. To address this problem, the Germans distributed codebooks containing various configurations of the scramblers and plugboard. Each day, all Enigma operators would use a new page in the codebook.

Enigma M3 Code Book (UKW-B Reflector) - April 1940

Datum [Date]	*Walzenlage* [Rotors]	*Ringstellung* [Ring settings]	*Steckerverbindungen* [Plugboard settings]	*Grundstellung* [Initial rotor positions]
30	V III II	AKK	AO HI MU SN VX ZQ	FDV
29	IV III V	JHS	LW RH UQ VP YM ZA	OTO
28	IV I II	DIL	EM HL PZ RJ SV UQ	JJK
27	III I IV	ICC	AX CW FZ KT PO SQ	RXV
26	IV II III	ECW	GS JD MN OQ VF XH	GUB
25	V III I	MFO	DW GO HE UF YI ZJ	ZBY
24	V III I	UCO	GC JU KE MF OD XY	BDT
23	II V IV	RWQ	BN FK OS PW TA ZE	IYM
22	IV II I	TRK	BN DU JI OK TF XC	SFX
21	II V III	CTZ	AF BK GJ VQ XH YT	TQO
20	I V III	XOM	BX IS LY NF QO WA	DKV
19	IV V II	LDQ	CR FO LI NM PD XH	IAH
18	IV I III	NWL	HV IM JB OT QA UF	HSP
17	II IV III	HFZ	FE IB OQ VC YW ZM	GPZ
16	II I IV	UBJ	CO GV IH KD ML RB	PJU
15	I II IV	BCG	ES GD IZ JF LN YA	KFQ
14	II V IV	EAP	BT CO NE PK VY ZI	CCH
13	I V II	AOK	CA DZ HK LP RQ YV	DMF
12	III I II	CKU	CK IZ QT NP JY GW	VQN
11	II III I	BHN	FR LY OX IT BM GJ	XIO
10	I V II	QKP	AF HQ IJ OT PB YG	MSW
9	V I II	UTC	DE FT IP OB UC YL	EQL
8	V IV II	GDJ	GT HR JI OK QE UZ	PLE
7	I II III	WNM	HK CN IO FY JM LW	RAO
6	V I III	ETT	FT HC KD PM YO ZB	HXA
5	V I III	MHY	BZ HS JF MW NG PV	XXJ
4	IV V III	WXE	DG IN JT UC VB WZ	OFP
3	IV II III	LIQ	BJ HC PI RF UO ZQ	KTR
2	II I V	NQC	AV KZ MS QP XF YU	ZJR
1	V II I	IHQ	ET LD NP QS RA UW	UJJ

To add security, operators also used a message key. Unlike the day key, the message key was specifically generated for each message. They had the same plugboard settings and used the same scramblers and scrambler positions as the day key, but the scrambler orientations for the message key were different from the scrambler orientations for the day key. At the beginning of World War II, Enigma operators followed the following procedure to send a message:

1. Set the Enigma machine according to the day key.
2. Randomly generate a set of scrambler orientations for the message key.
3. Encipher the message key according to the day key.
4. Type the message key into the Enigma twice.
5. Change machine to message key setting.
6. Encrypt main message.

The message's receiver would do the following:

1. Set the machine according to the day key setting.
2. Decrypt the first 6 letters to recover the message key.
3. Set the machine according to the message key setting.
4. Decrypt the main message.

This protocol would allow Marian Rejewski to break into the Enigma. Rejewski knew that each batch of messages intercepted at the beginning of the day began with the same six letters of the repeated message key encrypted with the same day key. The first and fourth letters of a message would both be encryptions of the same letter. The 2nd and 5th letters would also be encryptions of the same letter, as would the 3rd and 6th. Furthermore, because these letters were encrypted using the day key and the message key, the scheme used to encrypt them would not change for the entire day.

Rejewski collected the intercepted messages sent during a specific day and tabulated the pairings of first and fourth letters, second and fifth letters, and third and sixth letters from the various message keys in the intercepts. The resultant table might look like this:

1st letter: ABCDEFGHIJKLMNOPQRSTUVWXYZ
4th letter: FQHPLWOGBMVRXUYCZITNJEASDK

(Singh, p. 151)

At this point, Rejewski had no idea of the day key or message keys, but he DID know that they resulted in that day's table of links. He began to look for patterns within these tables, eventually focusing on chains of letters. For example, in the above table, when A is the first letter, F is the fourth letter. When F is the first letter, W is the fourth letter. When W is the first letter, A is the fourth letter.

Rejewski then had to figure out how to determine the day key from these chains. He realized that “the number of links in the chains is purely a consequence of the scrambler settings.” (Singh, p. 153) Thus, instead of having to figure out which of the trillions of day keys was related to a specific set of chains, he only had to find which numbers of links within a set of chains corresponded to one of 105,456 scrambler settings. The problem had just become manageable.

Rejewski and his colleagues began checking each of the scrambler settings on their replica Enigmas, cataloguing the chain lengths each setting generated. This enabled Rejewski to crack Enigma. Every day, Rejewski would find the chain lengths for that day’s messages, then identify the scrambler settings that corresponded to those chain lengths. Once he had identified the scrambler settings, he would follow the following procedure:

1. Set the scramblers in a replica Enigma according to the newly discovered scrambler configuration.
2. Remove all cables.
3. Type a piece of intercepted ciphertext into the Enigma.
4. If it returns something that’s vaguely recognizable apart from swapped letters, such as “belrin,” it’s a reasonably safe bet that the swapped letters are connected on the plugboard.

Rejewski and his colleagues knew that checking all these possible settings by hand would be hideously inefficient, so they built devices known as bombes to mechanize the process of decipherment. These machines enabled the Poles to successfully decrypt Enigma until December 1938, when Enigma operators received two new scramblers. The receipt of the new scrambles meant that there were now sixty possible scrambler combinations, each requiring a separate bombe. The magnitude of the calculations was simply out of reach for the Biuro Szyfrów’s resources. The Polish decided to hand their research over to the Allies, and the task of cracking the Enigma fell to the cryptanalysts of Bletchley Park, including Alan Turing.

The Breaking of the Enigma-Part II

Turing knew that he had the resources necessary to keep using Rejewski's techniques, but that was not a good long-term strategy. Thanks to the Poles, Turing had access to a great deal of decrypted Enigma messages, as well as the day and message keys used to encrypt them. Turing started by relying on weaknesses in Enigma's operations that this information revealed.

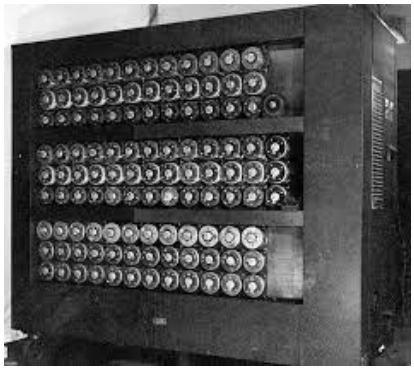
First, German Enigma operators would sometimes choose really obvious message keys-letters that were right next to each other on the keyboard, or their girlfriends' initials. The Germans also had two policies that seemed like good ideas in theory, but actually cut down on the number of possible keys and thus made the cipher less secure. The Germans did not permit any scrambler to remain in the same position for two days in a row. They also did not allow plugboard settings to include swaps between a letter and its neighbor in the alphabet. Finally, the Germans often sent messages following the same format. This let Bletchley Park predict the contents of a message depending on its origin. For example, the Germans sent a daily weather report after 6 AM. Therefore, a message intercepted at 6:05 was likely to contain the word wetter, and due to military protocol, the word would probably be in the same place. This provided Turing with a crib, or a section of ciphertext where the plaintext equivalent is known to the cryptanalyst.

Turing decided to take the same approach Rejewski had of finding the plugboard settings and scrambler settings separately. Turing's approach also relied on loops, connecting plaintext and ciphertext letters within a crib. For example, let us say that we have the crib wetter, which is encrypted in an intercepted message as ETJWPX. This allows us to say that w -> E, e -> T, t -> W is part of a loop. We label the first setting, the one that encrypted w to E, S. We label the setting that encrypts e to T as S+1. We label the setting that encrypts t to W as S+3. We can imagine three machines, each tasked with dealing with one element of our loop. The first machine would encipher w into E, the second machine would try to encipher e into T, and so on. These three machines would have identical settings, except the second machine would have its scramblers moved one place forward with respect to the first, to reflect the S+1 setting. The third would have its scramblers moved 3 places forward, to reflect s+3. Whatever changes we make in the plugboard or the scramblers in one machine we make on all the machines, but the relationships of the scrambler orientations will always stay the same.

We can then connect these three machines by running wires between the inputs and outputs, echoing the loop. When all three machines have the correct setting, the circuit these wires create is complete, illuminating a lightbulb. Furthermore, this arrangement also nullifies the effects of the plugboard, massively reducing the size of the problem. The electrical current enters our first machine and emerges at an unknown letter L1, then flows through the plugboard, yielding C1. C1 is connected to c1 in the second machine, and as it goes through the second plugboard it goes back to L1. Thus, the plugboards cancelled each other out.

Turing connected the 26 outputs of the first set of scramblers to the 26 outputs of the second set of scramblers, and so on. This created twenty-six electrical loops, each with a lightbulb

signifying completion of a circuit. These devices could check each of the possible scrambler configurations, with a lightbulb illuminating to signify a correct combination. Furthermore, because these devices are machines, they could change orientation once every second. This meant that it could take only 5 hours to find the scrambler setting.



There were two potential problems: first, the Enigma operated with 3 out of 5 available scramblers, yielding 60 possible combinations. If all 17576 combinations have been checked without yielding a successful combination, then you would have to try another scrambler arrangement. One might address this by running 60 sets of 3 machines running in parallel. Furthermore, even once Turing had figured out the scrambler settings, he still needed to identify the plugboard settings. To do this, he simply followed Rejewski's approach.

Bletchley Park turned these ideas into their very own bombes, which consisted of twelve sets of linked Enigma scramblers. At peak operating conditions, a bombe might find an Enigma key within an hour. From there, it was relatively trivial to identify the day key and thus identify all message traffic for that day.

Discussion

Overall, I found Singh's explanation of the algorithms used to create, code, and crack Enigma quite accessible. However, the presentation was also quite difficult to generalize to the sort of nonspecific algorithm that makes for easy application. I also wish that Singh had devoted some more space to the reflector and ring, as well as the relationship between the number of bombes and the ease of decipherment. I recognize that Singh only had so much space to devote to the Enigma, but such information would have greatly improved my analysis.

I believe that ultimately, the Enigma illustrates that the greatest vulnerability of any code, cipher, or security system, no matter how complex, is operator incompetence. The Enigma *could not* have been cracked without Marian Rejewski's knowledge of Enigma operating procedure. Even when that operating procedure changed, the repetitive structure and syntax of German communications and well-intentioned security precautions offered other avenues of attack. I think that any attempt to secure something must have the assistance of the security system's creators in order to succeed. After all, they have the greatest knowledge of that

system's vulnerabilities. Because they understand the vulnerabilities, they can best advise users on how to use the system in a way that doesn't expose those vulnerabilities.

Furthermore, I believe good security systems should be designed with an eye towards how technology might develop in the future. Developers could update the system whenever a new technological breakthrough occurs, ensuring that the system can defend against attacks utilizing that breakthrough. At the time of its introduction, Enigma was so fearsome because no available methods could break it in any reasonable timeframe. The bombes mechanized decipherment and negated that advantage.

Modern computers are immensely more powerful than the bombes. Deciphering Enigma might take a little skillful programming to set up the proper algorithms, but a modern computer could execute those algorithms in minutes.