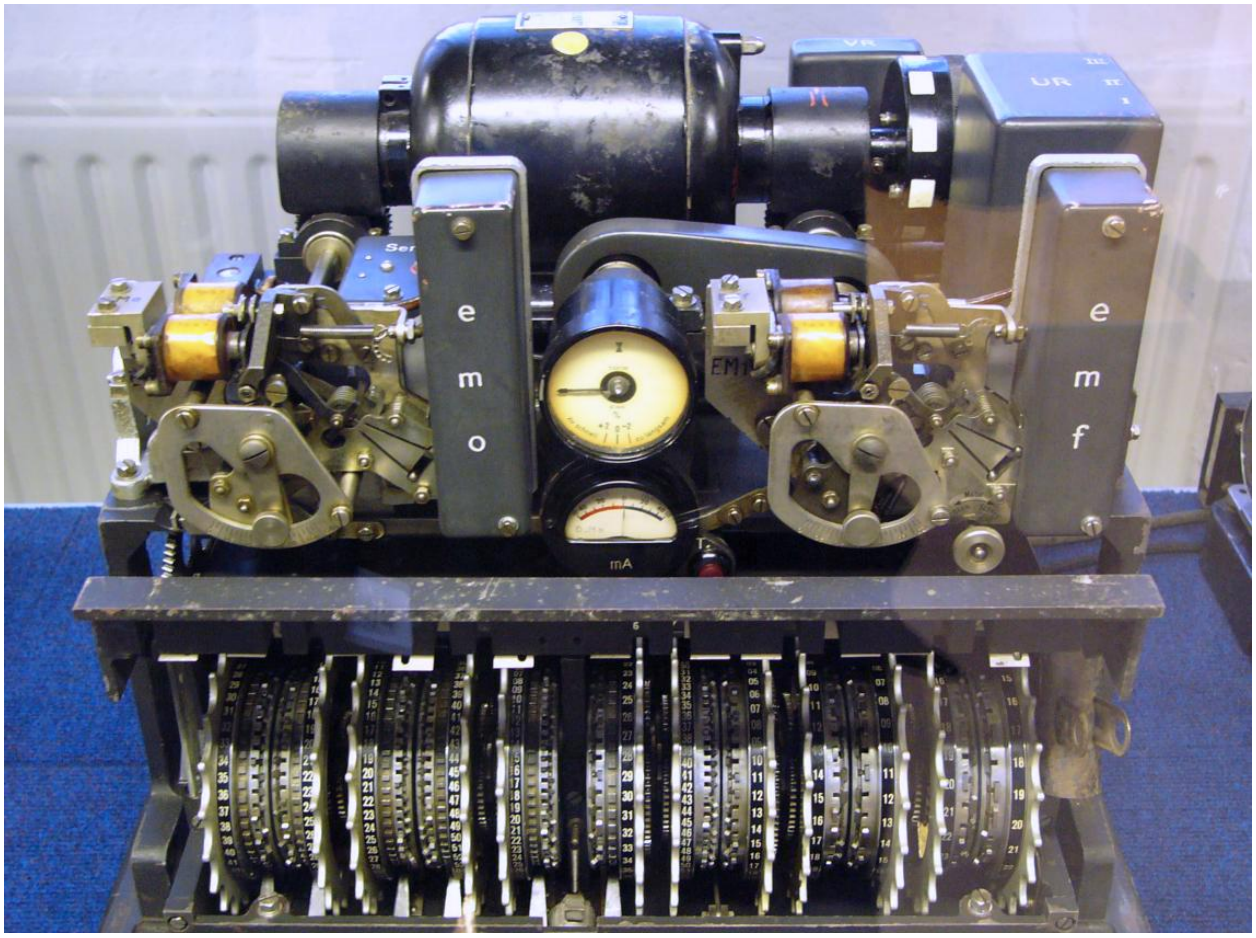
Literature Review: Lorenz Machine and its Implementation

Contributed by Swasthi P Rao, Muskan C Kothari

Centre of Information Security, Forensics and Cyber Resilience, PES University



1. Introduction

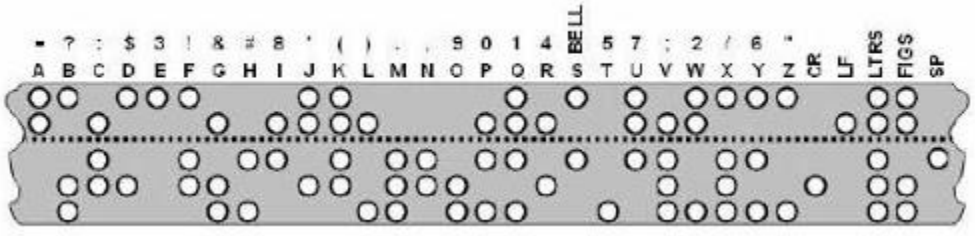
The Lorenz machine was a high security teleprinter cipher machine. Lorenz SZ40, SZ42a and SZ42b were German rotor stream cipher machines used by the German Army during World War II. The instruments implemented a Vernam stream cipher. The teleprinters use the 32-symbol Baudot code.

British cryptanalysts, who referred to encrypted German teleprinter traffic as Fish, dubbed the machine and its traffic Tunny (meaning tunafish) and deduced its logical structure three years before they saw such a machine, making cryptography advance several years at Bletchley Park.

The SZ machines were in-line attachments to standard teleprinters. The first version of the Lorenz machine was the SZ-40 followed by the SZ-42 which was brought into substantial use from mid-1942 onwards for high-level communications between the German High Command.
The Germans used mainly radio and teleprinters for their links, this allowed portability. They followed the then standard International Telegraph Alphabet No2 (ITA2). Although the radio signal was identified as teleprinter traffic, attempts at making any sense of the transmissions resulted in failure. Some were deciphered using hand methods before the process was partially automated, first with Robinson machines and then with the Colossus computers. The deciphered Lorenz messages made one of the most significant contributions to Allied victory in the War.

2. Working of the Lorenz Machine

2.1 Machine structure



This is the ITA2 code that was followed. Each letter was represented by 5 bits (vertically read in the picture). So, A was represented as 11000. These bits could be any combination of the two binary states 1 or 0 (a hole in the tape or no hole). If we calculate how many possibilities there are from 5 bits, we get 32, which is not enough

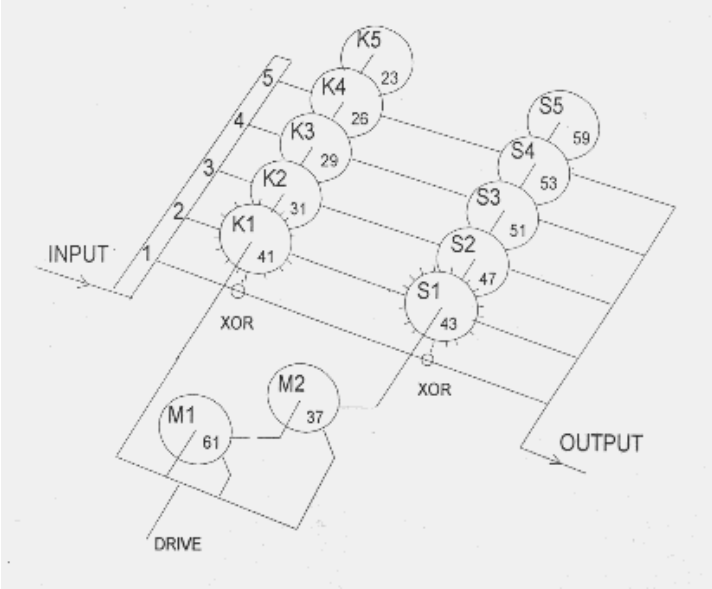
for a 26 letter alphabet, ten digits and punctuation. So two of the 32 letters are special: letter shift and figure shift. This almost doubles the number of symbols that can be generated and allows roughly 60 letters to be represented.

The Lorenz machine had 12 wheels. From the right, the first 5 wheels were called the K wheels (chi), followed by 2 M wheels (Mu) and the leftmost 5 wheels represented the S wheels(psi).

Each wheel had a certain number of cams which were either set or not set. If the cam was set, the input letter bit would be reversed. For example, the bit 0 when the cam is set would change to 1 and vice versa. Starting from K1 to K5, the number of cams were 41, 31, 29, 26, 23. In total, there were 501 which allowed 2 to the power of 501 settings. The other numbers can be seen from the picture inserted below.

These numbers are relatively prime which means that there wouldn't be any repeats until you had stepped:

41x31x29x26x23 = 22,041,682 steps for the K wheels, 43x47x51x53x59 = 322,303,017 for S wheels and 61x37 = 2,257 steps for the motor wheels.



The settings and start positions of the 12 wheels were distributed to the various German military units in the form of a Code Book. Setting a Lorenz machine from scratch was a long job. Each cam on each wheel had to be accurately set or not set, only after which the input text was processed. Several successful attempts at getting the codebooks for Enigma were recorded but there were almost no cases of the Lorenz codebooks being compromised by Bletchley Park. Shown below is a setting of an

actual cipher where cam (“0”=cam not set, “+”=cam set, will change cipher, number in brackets is number of cams on wheel), last number is number of +’s: . You can observe that the number of cams set (+) is always as near as possible to 50% of the total cams on that wheel, except the motor wheels, they are different. This is to try to ensure that the resulting cipher stream contains 50% 1’s and 50% 0’s and looks like a random stream of information, to deter cryptanalysis. Quite a brilliant move. The K wheels would then be set as 00101 (K1 to K5 vertically).

```
S1 0+0+0 0+0++ 0+0+0 +0+00 ++000 ++++0 0+0+0 +0++0 ++0                          [43] 22
S2 00+0+ 0++0+ 0+++0 0++0+ 0+00+ +0000 +0+0+ ++00+ 0++0+ 0+                    [47] 24
S3 0+00+ ++000 +0+00 0++00 +0+0+ +0+0+ ++00+ 0+00+ 0+0++ +0+0+ +              [51] 26
S4 +0++0 0++0+ 00+0+ 0+00+ +0000 +++00 +0+++ +0+0+ 0+0+0 +00+0 ++0            [53] 27
S5 +0+00 +0+0+ 0+00+ 000++ 000+0 +++0+ +0+0+ 0+00+ ++00+ 0+++0 0+0++ 0+0     [59] 29

M1 +0+00 00+++ 00+++ +0+++ +00++ ++0++ ++0++ ++000 ++++0 +++0+ ++00+ +++0+ + [61] 41
M2 ++000 +0++0 ++++0 +++00 ++000 +0+0+ 000++ +0                               [37] 20

K1 0++00 ++00+ +00++ 00++0 0++++ 00++0 0+00+ +00++ 0                          [41] 21
K2 000+0 +00+0 ++000 0+++0 0+0++ ++0++ 0                                      [31] 15
K3 +++00 +++00 +++00 +0000 +++0+ 00+0                                         [29] 14
K4 00+0+ +++00 +00++ 0000+ +++00 +                                            [26] 13
K5 ++000 +++0+ 00+++ 0+000 ++0                                                [23] 12
```

2.2 How does it work?

The incoming input letter (from keyboard or paper tape reader) is first processed by the K wheels. Suppose we start with say, K1, it will have a cam which will be either set or not (K1 has 41 cams). If the cam is set, the input bit is XOR’d with the cam position. (If the cam is not set then the bit is unchanged, otherwise it is inverted: 0 becomes 1 or 0 becomes 1).

This happens on all five K wheels in parallel (5 wheels for 5-bit input). The modified output from the K wheels then reaches the S wheels where the bits are again changed/not changed depending on the S wheels cam settings. In the unlikely case of all the cams on all the wheels being off (not effective), the output would be an unchanged copy of the input.

The K wheels move on one step for every letter received but the S wheels only move if commanded to do so by the motor wheels, M1 and M2. The motor wheels have their cams set such that they will move roughly 50% of the time, and stay still roughly 50% of the time. (“Time” meaning in step with the letters coming in.) The two motor wheels were intended to make the cipher more unpredictable and difficult to break.

The output from the S wheels is either the final enciphered or the deciphered output. Initially the wheel settings were only changed once a month. Once the wheels had been broken, only the starting positions had to be found and the traffic could be read until the end of the month. But as time went by the settings were changed more often, eventually every day. This pushed the cryptographers very hard.

The Lorenz codebooks and machine were of utmost security and there are no cases of any information regarding it to be compromised. It is astounding how the cryptanalysis broke this complex structure, the cracking of which the Germans considered impossible.

3. How did the first significant breakthrough come to happen?

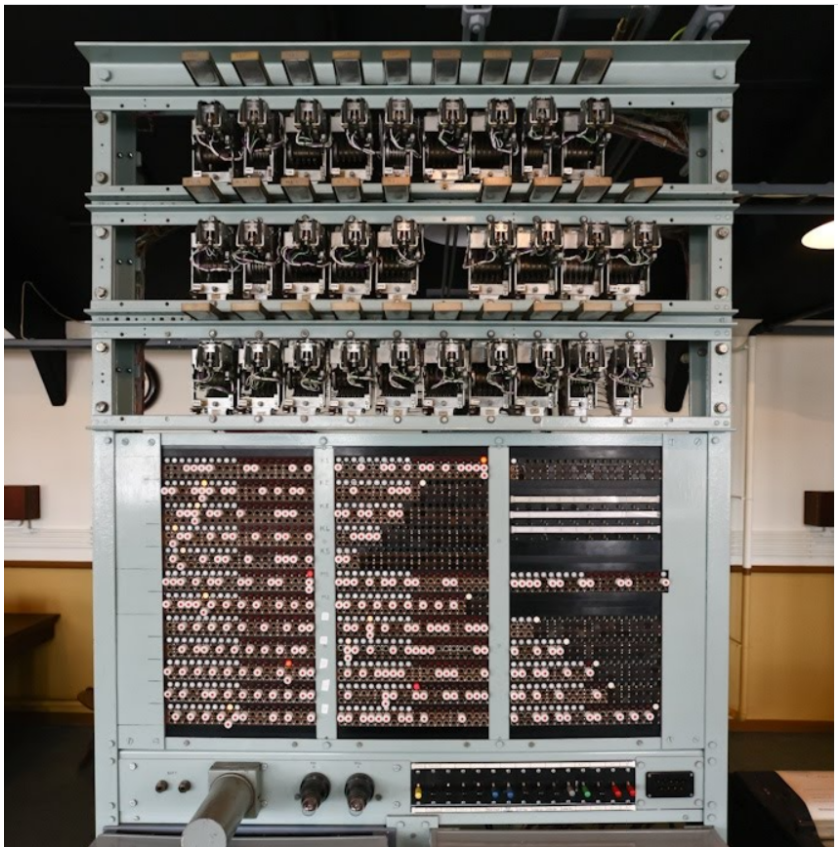
From June 1941 mistakes by the German operators resulted in a few short messages being received which used the same key - depths. But these were few. They gave the cryptographers a clue about the cipher being used but nothing else.
Both messages started with the twelve letters: HQIBPEXEZMUG, which referenced the starting positions of the 12 wheels. That message became known as simply ZMUG.

If the plain text messages had been identical then the cryptanalysts would be none the wiser, but they were different although some stretches were the same, but shifted due to abbreviations being used. This was a major mistake by the machine operator as this gave the british cryptanalysts two texts, deciphered, but mixed together.
The first text started (after the 12 letter identification) “SPRUCHNUMMER” and the second text “SPRUCHNR”.

The interceptors at Knockholt realised the possible importance of these two messages because the twelve letter indicators were the same. They were sent post-haste to John Tiltman at Bletchley Park, who worked out the structure of the machine from this first depth. Tiltman applied the same additive technique to this pair as he had to previous Depths. But this time he was able to get much further with working out the actual message texts because when he tried SPRUCHNUMMER at the start he immediately spotted that the second message was nearly identical to the first. Thus the combined errors of having the machines back to the same start position and the text being rekeyed with just slight differences enabled Tiltman to recover completely both texts. The second one was about 500 characters shorter than the first where the German operator had been saving his fingers. This fact also allowed Tiltman to assign the correct message to its original cipher text.

Now Tiltman could add together, character by character, the corresponding cipher and message texts revealing for the first time a long stretch of the obscuring character sequence being generated by this German cipher machine. He did not know how the machine did it, but he knew that this was what it was generating.

He drew out the pattern of the obtained key on a grid of squared paper. He was looking for possible repeat patterns in the structure of the data. After many failed attempts he spotted that the number 41 on his grid (say 41 wide) produced a noticeable pattern. He went on to deduce that the machine had 12 wheels and the length of each wheel, including how the two motor wheels operated. This was said to be the greatest intellectual achievement of the second World War.



When Bill Tutte had worked out the logical structure of the Lorenz machine, engineers set about building a functionally equivalent machine. Never having seen a Lorenz machine, the device they created looked entirely different and was electro-mechanical rather than purely mechanical. It was made mostly from telephone parts -- components with which the telephone engineers from Britain's General Post Office were most familiar.

4. Implementation with an input text

Let's take an example of an input text, both plaintext and ciphertext.
Plaintext: Bill Tutte
Key: 10101 (for simplicity)

	B	I	L	L	space	T	U	T	T	E
Plain:	10011	01100	01001	01001	00100	00001	11100	00001	00001	10000
Key:	10101	10101	10101	10101	10101	10101	10101	10101	10101	10101
Cipher:	00110	11001	11100	11100	10001	10100	01001	10100	10100	00101
	N	W	U	U	Z	S	L	S	S	H

After applying the XOR operation, the resultant cipher text turns out to be "NWUUZSLSSH". On further playing around with the 3 components of the machine, let's see what happens when we XOR the plaintext with the cipher text:

	B	I	L	L	space	T	U	T	T	E
Plain:	10011	01100	01001	01001	00100	00001	11100	00001	00001	10000
Cipher:	00110	11001	11100	11100	10001	10100	01001	10100	10100	00101
Key:	10101	10101	10101	10101	10101	10101	10101	10101	10101	10101

We retrieve the key! This way, if 2 components are known, the third can be found out quite easily. Now, what might be the possibility of knowing the plaintext and ciphertext?

Let's say we have a depth - 2 similar plaintexts with the same key. If you add the two ciphertexts together modulo 2, as long as the key is the same, then the key will "drop out". In other words, be cancelled leaving just the two messages merged together. Adding the two cipher texts (XOR) will result in all bits=0 (obviously).

This way, we find that the two texts come from the same key.

5. Implementation details

Our aim is to implement the Lorenz machine and perform cryptanalysis of ciphertexts with the same key as a working program. This program would use Python language and will be worked on Windows and Mac operating systems with the processor speed of 1.6 GHz each.

6. References

1. Virtual Lorenz SZ40/42, https://lorenz.virtualcolossus.co.uk/LorenzSZ/
2. Simplified Lorenz Cipher Toolkit, https://www.cimt.org.uk/resources/codes/lorenz/index.htm
3. The Lorenz Cipher and how Bletchley Park broke it by Tony Sale, https://www.codesandciphers.org.uk/lorenz/fish.htm
4. The Lorenz Cipher and How it was Broken at Bletchley Park by Charles Coultas, The National Museum of Computing, https://www.tnmoc.org/s/Breaking-Lorenz-III.pdf
5. Breaking teleprinter ciphers for at Bletchley Park, https://ieeexplore.ieee.org/book/712327
6. Lorenz Cipher Machine- Applied Cryptography, https://youtu.be/_yfl3KOVzDE
7. Colossus and the Breaking of Lorenz- National Museum of Computing https://artsandculture.google.com/exhibit/colossus-the-breaking-of-lorenz/ARkvJ5E5