Unit 185
Lorenz SZ40 and SZ42

The *Lorenz SZ40* and *SZ42* formed a family of teleprinter cipher machines used by Germany in World War II. They were so named as an abbreviation of *Schlüssel-Zusatz* (“cipher add-on”). British cryptanalysts nicknamed them *Tunny*, which is their way of saying “tuna fish.” Like the T52, SZ40/42 used wheels with pins to modify the bits of baudot-encoded messages. Unlike T52, the pins of the wheels in SZ40/42 could be added or removed, and only XOR operations, but not permutations, were used. Look back at Table 181.4 for a definition of the XOR operation. A fun property of XOR is that

$$A \oplus A \;=\; 0$$

(Equation 182.5) for any bit (0 or 1) A. Therefore, if we use XOR to encrypt, then the same operation is used to decrypt. If we write P for the plaintext, C for the ciphertext, K for the key, and think of “$\oplus$” as acting on all the bits in parallel,

$$\begin{aligned} C &= P \oplus K \\ C \oplus K &= (P \oplus K) \oplus K \;=\; P \oplus (K \oplus K) \;=\; P \oplus 0 \;=\; P \end{aligned}$$

(Look back at Equation 182.6.) This property makes life easier for German cipher clerks. The rest of this unit will be spent on describing the various members of this family of machines.

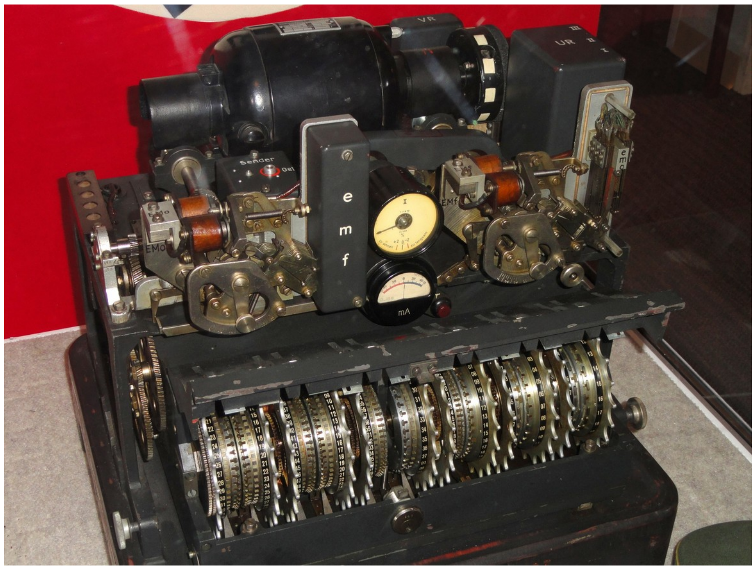

Figure 185.1: Lorenz SZ40 on display at the U.S. National Cryptologic Museum. Photo from U.S. government.
]

There was an initial model, the “old SZ40,” that was more of a prototype than anything else. It had a small production run and was used for only a short time. It has ten wheels, and all of them stepped after each letter was enciphered. They have fixed pins. They are grouped into five sets of two, so that the signal from two wheel affects one bit of the baudot data, i.e., each bit is XORed with the output of two wheels:

$$b_i \rightarrow b_i \oplus X_i \oplus Y_i$$

(i = 1, ..., 5). By giving the wheels sizes that are coprime, the period of the machine is lengthened.

The proper SZ40 has twelve wheels with variable pins. Pins can be flipped into place (“active,” denoted by × or 1) or out of place (“inactive,” denoted by · or 0). Like the old version, the proper SZ40 XORs the output of two wheels to each bit of data. However, only five of those ten wheels step after each letter is enciphered. The additional two wheel control how the remaining five wheels step.

Five of the wheels are called *Spaltencäsar* or *chi* (χ) wheels. The output of the first one is XORed into the first bit of the plaintext character, the output of the second into the second bit, etc. Another five of the wheels are called *Springcäsar* or *psi* (ψ) wheels. Again, the output of the first of these is XORed into the first bit of the data, etc., so that a bit is altered thus:

$$b_i \rightarrow b_i \oplus \chi_i \oplus \psi_i$$

The remaining two wheels are called *Vorgelege* or *mu* (μ) or *motor-control* wheels. More about them soon. Each of these twelve wheels has a different number of pins around its rim. See Table 185.1 for those numbers. A pin can be in active or inactive position. Only if a pin is in an active position can it conduct a signal out from the wheel. The wheels have numbers on their rims, like in other machines we have seen, to indicate which pin is being probed at the moment.

Stepping of the wheels occurs after each character is enciphered. All of the chi wheels advance by one position. If the output of μ_2 is a positive signal, then all of the psi wheels advance by one. If the output of μ_1 is a positive signal, then μ_2 steps. Finally, μ_1 steps every time. Got it? Take a look at Figure 185.2 for a sketch of the encryption process.

We need to discuss rules imposed by the creators of the Lorenz machines for the setting of active and inactive pins on the wheels. For each wheel, there is a prescribed number or range for the number of active pins; see Table 185.1. For the chi wheels, there can be no more than five active pins in a row, or five inactive in a row. Furthermore, the number of segments (sets of pins with the same setting all in a row) is prescribed also. For the psi wheels, the number of segments is determined by the number of active pins on μ_2; see Table 185.2. For μ_1 the constraints are that there can be no more than five inactive pins in a row, and no more than fifteen active pins in a row. For μ_2, there can be at most five inactive or six active pins in a row.

wheel	pins	active pins	segments
χ_1	41	20 or 21	20
χ_2	31	15 or 16	16
χ_3	29	14 or 15	14
χ_4	26	13	12 or 14
χ_5	23	11 or 12	12
μ_1	61	30-36, 38-50	
μ_2	37	9-23	
ψ_1	43	22	26-34
ψ_2	47	24	28-38
ψ_3	51	26	32-42
ψ_4	53	27	32-48
ψ_5	59	30	36-48

Table 185.1: Wheel sizes and key restrictions for the Lorenz SZ40 and SZ42. A "segment" is a group of contiguous pins in the same state. The number of allowed segments on chi wheels depends on the number of active pins on μ_2.

active pins on μ_2	allowed segments on psi wheels				
	ψ_1	ψ_2	ψ_3	ψ_4	ψ_5
9	34	38	42	42	48
10	34	36 or 38	40	42	46
11	34	36	40	40	46
12	32	36	38	40	44
13	32	34	38	40	44
14	32	34	38	38	42
15	30	34	36	38	42
16	30	32	36	38	42
17	30	32	34 or 36	36	40
18	28	32	34	36	40
19	28	32	34	36	38
20	28	30	32 or 34	34	38
21	28	30	32	34	38
22	26	30	32	34	38
23	26	28	32	32	36

Table 185.2: Key restrictions on the psi wheels in LZ40/42.

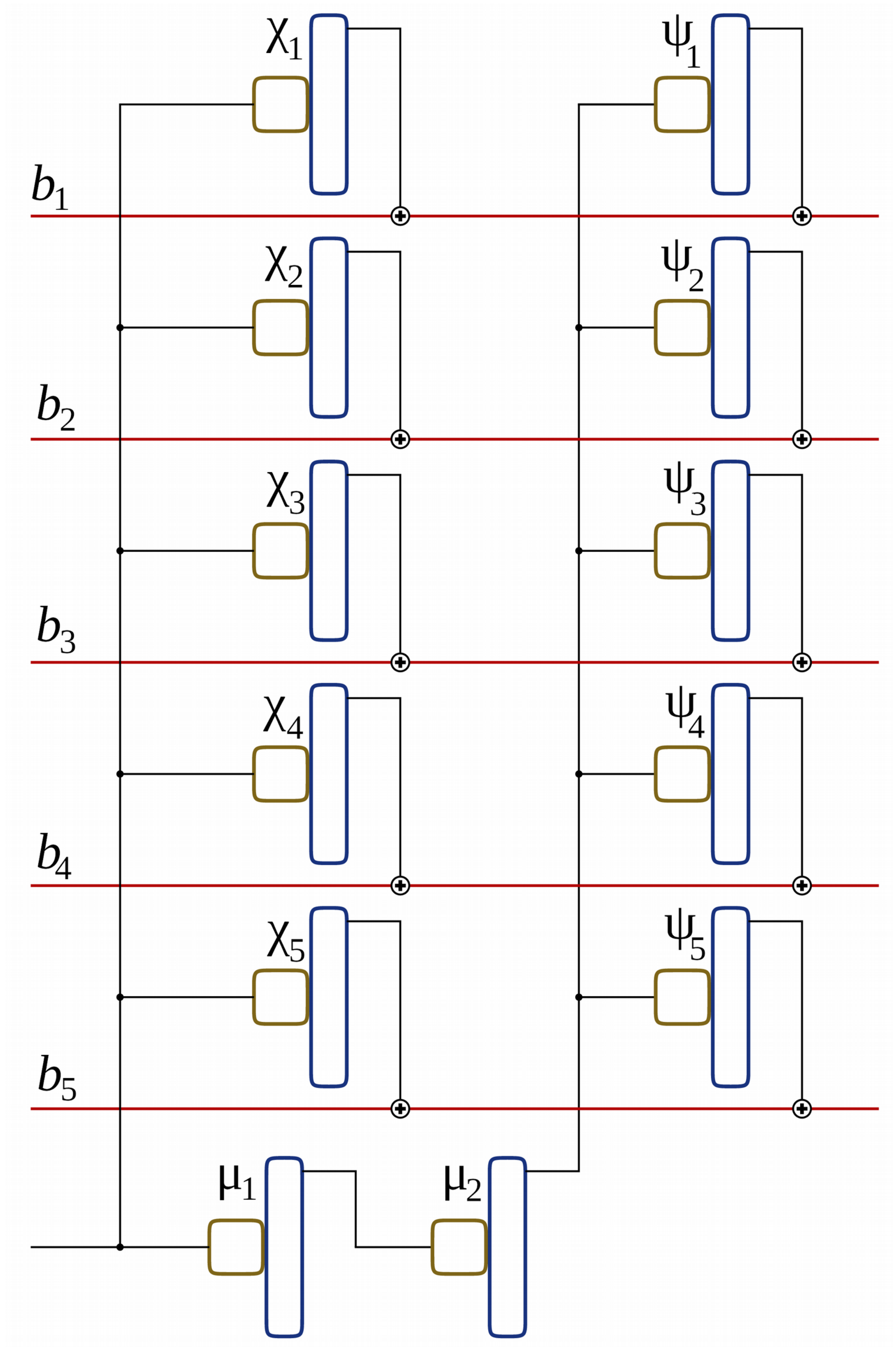

Figure 185.2: Simplified diagram of SZ40.

The key for the machine consists of the pin layout and starting position (counting from 1) of each of the twelve wheels. For example, this is a valid key:

```
χ1:   35   ·····×××··×·××××·××··×··×·×·×××····××××·×
χ2:   22   ××××···××·×·×·××··×··×××····×··
χ3:   17   ··××····××··×·××××·××·×·××···
χ4:    9   ××××·××·×·×·····×···××·××·
χ5:    5   ···×××·×·×··×·×××××·×··

ψ1:   32   ·×·×····××·×··×·×·×××·××··××·××·×·×·××·×··×
ψ2:   41   ·××··×·×·×·×·×··×·····××·×·××××·×··×·×·×××·×·××
ψ3:    3   ×·×··×·×××·×·×·×·×····×·×·××××·××·×·××·×··×·××··×··
ψ4:   10   ·×·×·×××··×××···×·×·×·×···×·×·××·×·××·××·×·××··×··×·×
ψ5:   23   ×·××·×··×·×·×·×·×·×××··×·×·××·××·×··×·×××·×·×···××··×···××·

μ1:   59   ··××···×··×××··××··×·××××····××··×··××··××·×·×××····×·××××××
μ2:   12   ××××··×····×·×···×··×××·×··×·×···××··
```

Notice that for χ_3 and χ_5 we have the correct numbers of segments, due to wrapping around the wheel.

Suppose we want to encipher a message with the example key above. Start with a plaintext:

```
THIS MESSAGE WAS ENCRYPTED WITH LORENZ SZ40
```

The zeroth step is to encode the message in Baudot, and list the bits for each character:

```
THIS9MESSAGE9WAS9ENCRYPTED9WITH9LORENZ9SZ+RP8
0 0 0 1 0 0 1 1 1 1 0 1 0 1 1 1 0 1 0 0 0 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1 0 1 0 1 1 1 0 0 1
0 0 1 0 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 1 1 0 1 0 0 0 0 1 1 0 0 0 1 0 1 0 0 0 0 0 0 1 1 1 1
0 1 1 1 1 1 0 1 1 0 0 0 1 0 0 1 1 0 1 1 0 1 1 0 0 0 1 0 1 0 1 1 0 0 0 0 1 0 1 1 0 0 0 1 1
0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 1 1 0 1
1 1 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0 1 1 0 1 1 0 0 0 1 0 0 1 1 0 1 1
```

The machine is in the state χ = 01111, ψ = 01100, μ = 11. To encipher the first letter, we xor the bits of `T` with the bits of χ and ψ:

```
T    0 0 0 0 1
χ    0 1 1 1 1
ψ    0 1 1 0 0
─────────────
3    0 0 0 1 0
```

Next, the chi wheels advance to 36, 23, 18, 10, 6. Since μ_2 outputs 1, the psi wheels also advance, to 33, 42, 4, 44, 24. Since μ_1 outputs 1, μ_2 advances to 13. Lastly, μ_1 advances to 60. The encipherment of the next letter is

```
H    0 0 1 0 1
χ    1 1 1 0 1
ψ    1 1 0 1 1
─────────────
O    0 0 0 1 1
```

Now, all chi wheels advance, as usual, but because μ_2 is at position 13 which outputs 0, the psi wheels do *not* advance. Since μ_1 outputs 1 at its current position, μ_2 advances. Then finally μ_1 may advance. The next letter is enciphered:

$$\begin{array}{ll} \texttt{I} & 0\,1\,1\,0\,0 \\ \chi & 1\,1\,0\,1\,0 \\ \psi & 1\,1\,0\,1\,1 \\ \hline \texttt{P} & 0\,1\,1\,0\,1 \end{array}$$

The full ciphertext is

```
3OPOD9NKZK/+/3SVKYRH8OMRONZGHIYSNXGMRFPEFFXDN
```

In Unit 151 we mentioned that we must never reuse the key in a one-time pad. The keystream of the Lorenz SZ40 ($\chi \oplus \psi$) is like the key of an OTP. It is not truly random, but appears to be. But the problem arises when two messages are enciphered with the same key. Suppose we have two such messages, P_1 and P_2. Encipher them by XORing bit-by-bit with the *keystream* K:

$$C_1 = P_1 \oplus K$$
$$C_2 = P_2 \oplus K$$

But when an adversary adds the two ciphertexts together,

$$C_1 \oplus C_2 = (P_1 \oplus K) \oplus (P_2 \oplus K) = P_1 \oplus P_2 \oplus K \oplus K = P_1 \oplus P_2 \oplus 0 = P_1 \oplus P_2$$

The result does not include the key at all! With only a few such messages, the plaintexts can be disentangled, and the keystream can be recovered. Such messages are called *messages in depth*, and they must be avoided for security.

To combat the problem of lazy cipher clerks reusing keys and producing messages in depth, new models, the SZ42A and SZ42B, were introduced with features to modify the control of the stepping of psi wheels, and some of them were dependent on the plaintext. By using plaintext information in the encryption process, different messages are encrypted to different ciphertexts, even with the same key. There are four options (see Table 185.3), two of which incorporate a bit from the plaintext. These are called *limitations* and they cause the psi wheels to advance even when the output of μ_2 is 0. All of them involve the output of χ_2 from the encipherment of the previous character. This bit is called “chi2 one back” and is notated with an overline. In addition, one or two bits can be incorporated: the output of ψ_1 one character previous (“psi1 one back”) and/or the fifth bit of the plaintext character two letters back (“P5 two back”) (denoted with a double overline). The one, two, or three bits are XORed together and then inverted (XORed with 1). If the result is 1, then the psi wheels advance. If the result is 0, then the output of μ_2 determines whether they advance. So that you can check your emulation of the machine, here are the encipherments of our example message with each of the four limitations added to the key. With chi2 1 back,

```
3OPODW9TCT9/E3KHGKH4W/W/LBYVSQMWDWWYW+JLVZVRK
```

With psi1 1 back,

```
3OPOVN9TCJ9/E3KHS/JY9RBG3IJ/CXNHOJKDES9W8C/GT
```

With P5 two back,

```
3OPOVNWTCT9/E3KHS/JY9RBGLIJ/T3W+TLCYW+JLZCVQI
```

And with psi1 one back P5 two back,

```
3OPODW9UBU89W+NDO89RWXWT3MH8CXNHOJK3MRYMTT/HZ
```

There was a third version of the SZ42. The SZ42C used output of the chi wheels to control the stepping of the psi wheels. The two motor-control wheels were removed. This model never made it to production.

name	notation / limit bit
chi2 one back	$1 \oplus \overline{\chi}_2$
psi1 one back	$1 \oplus \overline{\chi}_2 \oplus \overline{\psi}_1$
P5 two back	$1 \oplus \overline{\chi}_2 \oplus \overline{\overline{P}}_5$
psi1 one back P5 two back	$1 \oplus \overline{\chi}_2 \oplus \overline{\psi}_1 \oplus \overline{\overline{P}}_5$

Table 185.3: Limitations on stepping of the psi wheels in Lorenz SZ42.

Reading and references

Orr Huettenhain and Dr. Fricke, OKW/Chi Cryptanalytic Research on Enigma, Hagelin, and Cipher Teleprinter Machines, TICOM report I-45, 1945, https://drive.google.com/file/d/0B7sNVKDp-yiJOWYxZWFmNDgtODUyMS00Y2FiLTthkNWItYmQ5N2JmMzEyMzIz

Irving John Good, Donald Michie, and Geoffrey Timms, *Breaking Teleprinter Ciphers at Bletchley Park: General Report on Tunny with Emphasis on Statistical Methods (1945)*, Wiley, 2015, ISBN 9780470465899 and 9781119061601, https://doi.org/10.1002/9781119061601

George Lasry, Solving a Tunny Challenge with Computerized “Testery” Methods, Proceedings of the 3rd International Conference on Historical Cryptology, HistoCrypt 2020, 96-105, https://www.ep.liu.se/ecp/171/013/ecp2020_171_013.pdf

Wolfgang Mache, “Geheimschreiber,” *Cryptologia* 10:4 (1986) 230-242, https://doi.org/10.1080/0161-118691861065

Programming tasks

1. Make a function that checks the validity of a key. This is a project in itself.

2. Implement the Lorenz SZ40.

3. Add the stepping-control features of SZ42A and SZ42B to your implementation.

Exercises

1. Decipher this text that was encrypted with SZ40. Use the key from the SZ40 example above.

```
3OPODXJICJ3BC8YKRYHVICKVS4+XPOBF+44FUYP4TLUJLSUWKQGWZMINNCPD4B9Y
YT8Q4KEVI9Z+9PB+NUOUZF43DPJ+ZNYGW8UL/ZK8EHDJ4Y3ZLF3Z4WSPMENB9SKA
QKSFDZGS4GR9QTGZHX+PNMDALX9PX4NPGDJEVUIP9UVY4A8Q3IAAW3CR3T9XYGYV
DKKOBVP+CLWSRJBV9CORTW/E3+HDERUMEFDQQKYPF3IRWHC3K+VUGCBEH/DPS4HD
QUFYOLTGDAPVJPPEZJ8MA+Z3HRHGVK8GYYV9DQTBETMDGJPGLY4D+XVPUJ
```

2. Decipher this text that was encrypted with SZ42. Use the key from the SZ40 example above, but add the limitation P5 two back.

```
ADY4V+KBRM9/WO+WITARUFPKYWM3GUAPEA8TYVCX9+SLCB9AKFW+XKJ4ATFJ+/9R
U9M+Y9ZH3VR44NECFJYNOBATSYYXCRGMDIWT+PQDZX34VNT+LZT8ZZIEDU4RDHYU
ZJDMZJDPJU9VDBBUKHKD/ROY84ONHYRE9C8+IK+HDHPRWHR/IMJRKOFHMO8A3D//
RIOKCYXDJRJUEG/SCKUBC8Y9I/84B8V9UFKYXN4F9HWUNTBW/9XKWUFZXLWE4994
N/4RTOOWZW9YDGMALQUICFMQYC/ZCXRDLBLBJWTHVA4QQE/LDQ4AL9DOYUHD839C
/I+O9RIGFOKVO/SYEUE83L9TWGNPXCXUZIC4LTPF
```

3. These five messages were encryped by an SZ40 with the same key. See if you can recover (most of) the plaintexts and the keystream.

```
SPBMKROMZCNARFNBOQ9RKOUJPQO+Y4BWB4L4JJQ/DBSDGNNCABTHN3WUJFGNQNK+
TQMLFPVHV+OKNF4NKZIVMFO9YI3RFJWBLNA8RASTWOWR+LSRAKRIRMK/FXHSUGYF
RM33RUAQFKKDPKRRNLHIGHXTVHBKDO+

WBXUXPD4YSBAOEJYXLQCK4LCNLEJWTOEYN/3USYDHBGO/DATOYLMJ/VMFA8OM9SJ
LPCEMJO4TVEFC8R3BNLWOYNU9QIWSNK9QOKWFIKQP9DFWIEXLK8HTVWCBTROH9Y9
V8LW/H/DGDXDJ9UW

DCYSJRYXYY+JI4N/JJOPFA9XLJFCTK8UPOUKKJL/9+KFCCAUJFUSKO3+4F/YTJFJ
UL/HT8GMQCRKR3TNEFMWKFD+383WFQ9QUNJRADWONYZ++SUZ8KWJLJYYYQPIAAY8
T43H/D93

/3OIJPQERNOHE3NGNQOW3S9NRJKY/XTFQRIFDRY3MVTDCJACJFR4EEZGAHWTCDYJ
LAPDFSJBC+DUEKJYE9DZWF/4HOKVSPDNTU8R/QKOWZLC9CCFYIJLQWVZCL9HX+9

SDKSPG8AFU9/LSJ/BYOXKQFDYJ4IZ4NITNT8L3IS9HVF94DWDSB9Y/LBXOCTRMM8
YXLXSBUHH++KBCRVZYZPT3Q4NQI3/IU/BTUPQWR/ZVF3TILSGKR8IBLNWERIT8QK
RCSPJAA8+Q
```