The Lorenz SZ42 (”Tunny”) German Teleprinter Encryption Device

The Lorenz SZ42, codenamed Tunny, was a teleprinter encryption device used by Germany during WW2 for strategic communications. Its successful cryptanalysis at Bletchley Park (BP) provided the Allies with high-grade intelligence about several fronts, as well as for the preparations for the D-Day landings.

The story of Tunny's codebreaking and Colossus is well known, following the declassification of the General Report on Tunny in 2000 (Good et al., 1945), and the publication of several books (Reeds et al., 2015; Gannon, 2014; Copeland, 2010; Roberts, 2017; Mayo-Smith, 2014). The work on Colossus and other machines was carried out in the Newmanry, under the leadership of the mathematician Max Newman.

The work of the Testery, the other Tunny section at BP, is less known. Named after his commander, Major Ralph Tester, the Testery was responsible for the development and application of hand methods, that complemented the work of machines like Colossus. For some reason, the report on the Testery was not declassified until 2018 (Testery, 1945).

This documents describes the functioning of the device and the outline of historical cryptanalysis methods.

1 The Lorenz SZ42 (Tunny)

The history of the Lorenz SZ42 and the details of its design and functioning are documented in the references (Reeds et al., 2015; Gannon, 2014; Copeland, 2010). In this section, only a brief functional description is given.

The Lorenz SZ42 is a teleprinter encryption device. It encodes Baudot teleprinter symbols that consist of five impulses. Each impulse can have one of two states. It can be active, denoted as *cross* according to BP terminology, or **x**. Or it can be inactive, denoted as *dot* or •. The Baudot alphabet, as well as BP's notation for the Baudot symbols, is given in Table 1.

The Lorenz SZ42 functions as a Vernam device. It applies an XOR addition (denoted as $\oplus$) to encrypt plaintext Baudot symbols. The effect of the XOR operation on a pair of impulses a and b is described in Table 2. The XOR operation can also be applied to a pair of Baudot symbols with five impulses each. In that case, it is applied sequentially one impulse at a time. An example is given in Table 3. It should be noted that adding (using an XOR addition) a symbol to itself, results in the symbol ••••• which has only dots, as illustrated in Table 4.

The Lorenz SZ42 generates a keystream K of pseudo-random symbols and performs an XOR addition on a stream of plaintext P, producing the ciphertext Z, as described in Equation 1, the encryption formula.

$$Z = P \oplus K \tag{1}$$

Encryption and decryption are implemented identically. This is possible since adding (XOR) the keystream K to the ciphertext Z cancels out the effect of the keystream K originally added during encryption, as shown in Equation 2, the decryption formula.

$$Z \oplus K = (P \oplus K) \oplus K = P \oplus (K \oplus K) = P \tag{2}$$

As a result, two machines using identical settings can communicate properly, one side encrypting plaintext and transmitting ciphertext, the other receiving and decrypting the ciphertext.

The functioning of the Lorenz SZ42 is illustrated in Figure 1.

Symbol	**BP Notation**	**Meaning in Letter Shift**	**Meaning in Figure Shift**
●●●●●	/	null	
●●●●x	E	E	3
●●●x●	4	carriage return	
●●●xx	A	A	-
●●x●●	9	space	
●●x●x	S	S	'
●●xx●	I	I	8
●●xxx	U	U	7
●x●●●	3	line feed	
●x●●x	D	D	Who are you?
●x●x●	R	R	4
●x●xx	J	J	BELL
●xx●●	N	N	,
●xx●x	F	F	%
●xxx●	C	C	:
●xxxx	K	K	(
x●●●●	T	T	5
x●●●x	Z	Z	+
x●●x●	L	L	)
x●●xx	W	W	2
x●x●●	H	H	£
x●x●x	Y	Y	6
x●xx●	P	P	0
x●xxx	Q	Q	1
xx●●●	O	O	9
xx●●x	B	B	?
xx●x●	G	G	&
xx●xx	5 or +	figure shift	
xxx●●	M	M	.
xxx●x	X	X	/
xxxx●	V	V	;
xxxxx	8 or -	letter shift	

Table 1: The Baudot Teleprinter Alphabet

a	b	$a \oplus b$
●	●	●
●	x	x
x	●	x
x	x	●

Table 2: The XOR ($\oplus$) Operation

K	●xxxx
G	xx●x●
K $\oplus$ G	x●x●x

Table 3: XOR ($\oplus$) on the Symbols K and G

G	xx●x●
G	xx●x●
G $\oplus$ G	●●●●●

Table 4: XOR ($\oplus$) on the Same Symbol

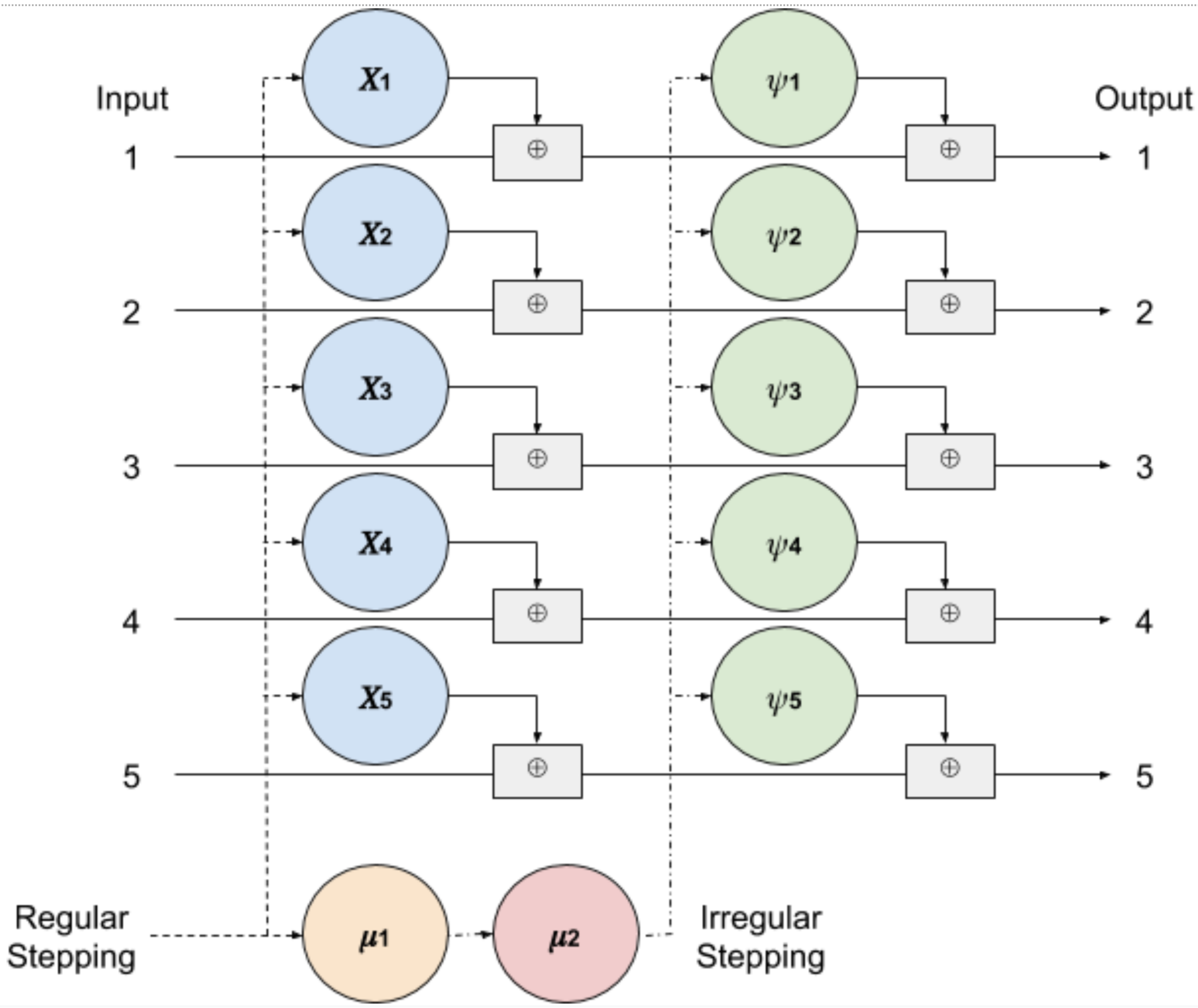


Figure 1: Tunny Lorenz SZ42 – Functional Diagram

The keystream K is generated by a set of twelve wheels, divided into three functional groups:

- **Five χ wheels, χ_1 to χ_5**: Those wheels have 41, 31, 29, 26, and 23 pins, respectively. Each pin can be set to either an active (cross) or an inactive (dot) state. The χ wheels regularly step after the encryption (or decryption) of each symbol. The stream of Baudot symbols generated by the five χ wheels is denoted as the χ *stream*.
- **Five ψ wheels, ψ_1 to ψ_5**: Those wheels have 43, 47, 51, 53, and 59 pins, respectively. Each pin can be set to either an active or an inactive state. Their stepping is governed by the motor wheels. Either all five ψ wheels step or none of them steps. The actual stream of symbols generated by the ψ wheels is denoted as the ψ' *stream*. It differs from a theoretical ψ *stream*, that would have been generated if the ψ wheels always stepped. The ψ' stream is an extended version of the ψ stream, with symbols duplicated at positions where the ψ wheels did not step.
- **Two motor or μ wheels, μ_1 and μ_2**: Wheel μ_1 has 61 pins, which govern the stepping of wheel μ_2. If the current pin of wheel μ_1 is active (cross), wheel μ_2 steps. Wheel μ_2 has 37 pins, and if its current pin is active, all five ψ wheels step. The single-impulse stream generated by wheel μ_2 is denoted as the *base motor stream*. In later models of the Lorenz SZ42, various *motor limitations* were introduced to reduce the number of *motor stops*, that is, positions where the ψ wheels are not stepping. A motor limitation forces the ψ wheels to move at positions where the base motor stream

is a dot, and the ψ wheels would otherwise not step. Motor limitations are governed by a combination of one or more impulses from the P, χ, and ψ' streams, at previous positions. The combined effect of the μ wheels (the base motor stream) and of the motor limitations is denoted as the *total motor stream*. A description of the various types of motor limitations may be found in (Reeds et al., 2015, Chapter 11B, p. 13), reproduced in the Appendix. As described in Section 2, most attacks against Tunny take advantage of skewed statistics at motor stop positions. Motor limitations are intended to reduce the number of motor stops, making cryptanalysis more challenging.

The keystream K consists of the (XOR) addition of two streams, χ, and ψ':

$$K = \chi \oplus \psi' \tag{3}$$

Therefore:

$$Z = P \oplus K = P \oplus \chi \oplus \psi' \tag{4}$$

We define D, also known as the *dechi stream* (or simply, the *dechi*), as:

$$D = Z \oplus \chi \tag{5}$$

The term dechi originates from the fact that we are removing χ from the ciphertext Z, by adding it so that the original contribution of χ cancels out:

$$D = Z \oplus \chi = P \oplus \chi \oplus \psi' \oplus \chi = P \oplus \psi'. \tag{6}$$

If we add ψ' to both sides of $D = P \oplus \psi'$, it also follows that $P = D \oplus \psi'$.

Note that they were restrictions on the settings of some wheel patterns, as shown in Figure 2 and Figure 3.

Wheel	Length	No. of crosses in χ	No. of crosses in $\Delta\chi$
1	41	20 or 21	20
2	31	15 or 16	16
3	29	14 or 15	14
4	26	13	12 or 14
5	23	11 or 12	12

Figure 2: Restrictions on χ wheel patterns (Source: (Reeds et al., 2015, Chapter 2, p. 52))

Additional restrictions are described in the simulator code.

2 Tunny Codebreaking Overview

A complete decryption of the machine and of the hand methods for the cryptanalysis of Tunny, as well as of the multitude of codebreaking scenarios the methods cover, is outside the scope of this paper and may be found in the Testery report and the GRT (Testery, 1945; Good et al., 1945). This section focuses on the main cryptanalytic scenarios.

The most challenging scenario is *breaking*, when the wheel patterns are unknown, there are no messages in depth (encrypted with the same key settings), and no crib is available. Historically, codebreaking for such a scenario included the following steps:

- The recovery by the Newmanry of the χ wheel patterns, using the *rectangling* method developed by Bill Tutte, and later performed with the help of Colossus (Reeds et al., 2015, p. 110-112). After the χ wheel patterns had been recovered, the dechi stream $D = Z \oplus \chi$ was produced by the Newmanry.
- The recovery by the Testery of the ψ' stream from the dechi stream D, using hand methods. From ψ', the ψ wheel patterns could be recovered.

For all values of d:	ψ_1	ψ_2	ψ_3	ψ_4	ψ_5
Length	43	47	51	53	59
No. of crosses in ψ	22	24	26	27	30

	Number of crosses in					
d	$\Delta\psi_1$	$\Delta\psi_2$	$\Delta\psi_3$	$\Delta\psi_4$	$\Delta\psi_5$	d
14	26	28	32	32	36	14
15	26	30	32	34	38	15
16	28	30	32	34	38	16
17	28	30	32 or 34	34	38	17
18	28	32	34	36	38	18
19	28	32	34	36	40	19
20	30	32	34 or 36	36	40	20
21	30	32	36	38	42	21
22	30	34	36	38	42	22
23	32	34	38	38	42	23
24	32	34	38	40	44	24
25	32	36	38	40	44	25
26	34	36	40	40	46	26
27	34	36 or 38	40	42	46	27
28	34	38	42	42	48	28

Figure 3: Restrictions on ψ wheel patterns (Source: (Reeds et al., 2015, Chapter 2, p. 54))

- The recovery of the motor wheel patterns, also by the Testery, from the ψ' stream.
- The decoding of the ciphertext (by the Testery).

For *setting*, when the wheel patterns are known, but the wheel starting positions are unknown for a specific ciphertext, the process was simpler. Historically, χ-setting was done by the Newmanry, and the settings for the ψ and motor wheels were recovered by the Testery.[1]

In case two or more messages in depth were available, their plaintexts could be recovered using linguistic methods, and using segments of plaintext, the keystream $K(= Z \oplus P)$ could also be extracted. From K, the wheel patterns were then recovered by the Testery.[2] A similar process was possible with the help of a long-enough crib.

But unless a crib is available, or plaintext can be extracted from depths, all attacks – for setting and breaking – rely on a major weakness of Tunny, which is described here.

We first introduce the notation Δ, or *differenced* stream. A differenced stream consists of adding (using XOR addition) to each element of an original (undifferenced) stream the value of the element right after it. Differencing can be applied to a single impulse, or to a stream of Baudot symbols, impulse by impulse. An important characteristic of a differenced stream is that if two consecutive symbols are identical, their differenced value is the symbol ●●●●● (all impulses inactive).

In Section 1, Equation 6, it was shown that the dechi stream $D = Z \oplus \chi = P \oplus \psi'$.

We analyze here the frequency distribution of the symbols in the dechi stream D. The ψ wheels may or may not step after each encryption (or decryption), but if they step, they all step together. When the wheels do not step (i.e., a motor stop), the corresponding symbol of ψ' is duplicated, and as a result, the corresponding $\Delta\psi'$ symbol has only dots (●●●●●). This means that at positions where there is a motor stop, $\Delta D = \Delta P$. Therefore ΔD at motor stops has the same frequency distribution as for ΔP.[3] Even though the symbols of ΔD are (roughly) randomly distributed at positions the ψ wheels step, overall, the frequency distribution of ΔD symbols is skewed toward the frequency distribution of ΔP symbols.

This important characteristic can be exploited for setting the χ wheels. While the plaintext for a given ciphertext is unknown, it is possible to compute the distribution of the *expected* differenced plaintext ΔP, using a corpus of the language (e.g., from prior decryptions). To set the χ wheels, we search for the χ wheel positions that result in the symbol distribution of $\Delta D = \Delta Z \oplus \Delta\chi$ being as close as possible to the

---

[1] For some motor limitations (or if no motor limitation was used), the setting of the ψ and motor wheels could also be performed using the more advanced models of Colossus.

[2] *Turingery*, a method for extracting the χ patterns from K, was developed by Alan Turing.

[3] During cryptanalysis, the positions where there is a motor stop and ψ wheels do not step are unknown.

expected frequency distribution of ΔP in the reference corpus. A similar methodology can be applied for χ breaking, to find the optimal χ patterns, so that the resulting ΔD best matches the expected distribution of ΔP in the reference corpus.

Due to the limits of WW2 technology, those techniques could only be applied to a pair of impulses at a time, e.g., impulses 1 and 2 (the so-called Δ_{1+2} method), rather than to all five impulses at the same time (Reeds et al., 2015, p. 110-112).

The same characteristic of ΔD can be used to recover ψ' from dechi, as described in Section 3.

3 Historical Manual Testery Methods

The main Testery methods are based on the characteristic of ΔD, as described in Section 2. Due to the ψ wheels often not stepping, there are numerous repetitions of consecutive symbols in ψ', and as a result a high frequency of ●●●●● symbols (all impulses inactive) in $\Delta\psi'$.

Due to security measures introduced by the Germans (Reeds et al., 2015, p. 306), there is also a high frequency of **xxxxx** symbols (all five impulses active) in $\Delta\psi'$, at positions where the ψ wheels step.[4] Furthermore, the frequency of $\Delta\psi'$ symbols with a majority of crosses (e.g., **●xxxx** or **●●xxx**) is significantly higher than the frequency of symbols with only one or two crosses (e.g., **●xx●●** or **●●x●●**). In addition, the probability for a ●●●●● symbol at positions where the ψ wheels are stepping is very low.

Historically, the work of the Testery started after receiving the dechi D, extracted from ciphertext by the Newmanry using mechanized methods. The Testery cryptanalysts tried various possible cribs P at different positions, examining the resulting (putative) $\Delta\psi' = \Delta D \oplus \Delta P$. A putative $\Delta\psi'$ mostly consisting of ●●●●● or **xxxxx** symbols, and the remaining symbols with a majority of crosses, was likely to indicate a correct crib guess. Still, there was always some probability for a wrong guess, especially if the crib was short. This process was labor-intensive and required extensive trial-and-error by the cryptanalysts, who had to memorize the full XOR addition table ($32 \cdot 32 = 1024$ elements) to mentally perform XOR additions (Roberts, 2017; Mayo-Smith, 2014).

For ψ setting, a machine named *Dragon* was developed to "drag" a crib over the whole dechi stream (Reeds et al., 2015, p. 346). For ψ breaking, there was no other choice but to test cribs manually.

After positioning a likely crib, the cryptanalyst would then try to extend it by testing additional symbols inserted before and after the crib, and checking the resulting new putative $\Delta\psi'$. With a long enough-crib and from the resulting ψ' segment ($\psi' = D \oplus P$), it was possible to recover the ψ patterns.

With modern computing, a more efficient process can be implemented.

```
Crib P:          89MANNERN89UND89FRAUEN5
Dechi D:         RCPDIIJ/IYZLBMRZQTSEUSX
ψ' :             YRRRRRRRRRYYYYYYGGIIIDI
Δψ' :            8////////8/////K/M//KK
Δψ crosses:      5000000005000004030044
```

Figure 4: Example of Crib Hit

Figure 4 shows an example of a particularly good crib hit. In this example, the elements of $\Delta\psi'$ have either no crosses, only crosses, or a majority of crosses (three or four).[5] In a more typical case, there will be less "good" symbols, and the ψ wheels are likely to step more often.

A manual attempt is then made to extend the most promising cribs, by guessing additional symbols at their beginning and at their end, so that the (longer) putative $\Delta\psi'$ still has good characteristics. With a solid knowledge of the language and of the traffic contents, it is possible to extend the crib further so that a long stretch of ψ' can be obtained. Then, by removing repeated consecutive symbols from ψ', it is

---

[4] To create a seemingly more random output Z as well as ΔZ, each pin on a given ψ wheel was more likely to be followed by a pin in the opposite state.

[5] As shown in Figure 1, in BP notation / represents the ●●●●● symbol, 8 represents **xxxx**, K represents **●xxxx**, and M represents **xxx●●**.

possible to obtain the (unextended) ψ stream and from it to extract the ψ wheel patterns. Historically, the Testery cryptanalysts would first recover the ψ patterns as described here, and finally, the motor wheel patterns.

4 Tunny Simulator

A Java-based simulator is available, that implements the various functions of Tunny. With this simulator, it is possible:

- To encrypt a plaintext.
- To decrypt a ciphertext.
- To run an virtual Colossus machine, to recover the starting position of the χ_1 and χ_2 wheels.

Usage:

To encrypt:

```
java -jar tunny.jar -e key-filename -i plaintext-filename
```

plaintext-filename contains the plaintext in British notation.

To decrypt:

```
java -jar tunny.jar -d key-filename -i ciphertext-filename
```

ciphertext-filename contains the ciphertext in British notation.

To run a virtual Colossus machine:

```
java -jar tunny.jar -c key-filename -i ciphertext-filename
```

key-filename must contain at least the first two lines of the key with the CHI1 and CHI2 wheel patterns.

References

Jack Copeland. 2010. *Colossus: The Secrets of Bletchley Park's Code-breaking Computers*. OUP Oxford.

Paul Gannon. 2014. *Colossus: Bletchley Park's Last Secret*. Atlantic Books Ltd.

Jack Good, Donald Michie, and Geoffrey Timms. 1945. *General Report on Tunny: With Emphasis on Statistical Methods*. Bletchley Park Report HW 25/4. Kew, London: U.K. National Archives.

Ian Mayo-Smith. 2014. *Eavesdropping on Adolph Hitler: Deciphering the Daily Messages in the Tunny cipher*. Four Pillars Media Group, Connecticut, USA.

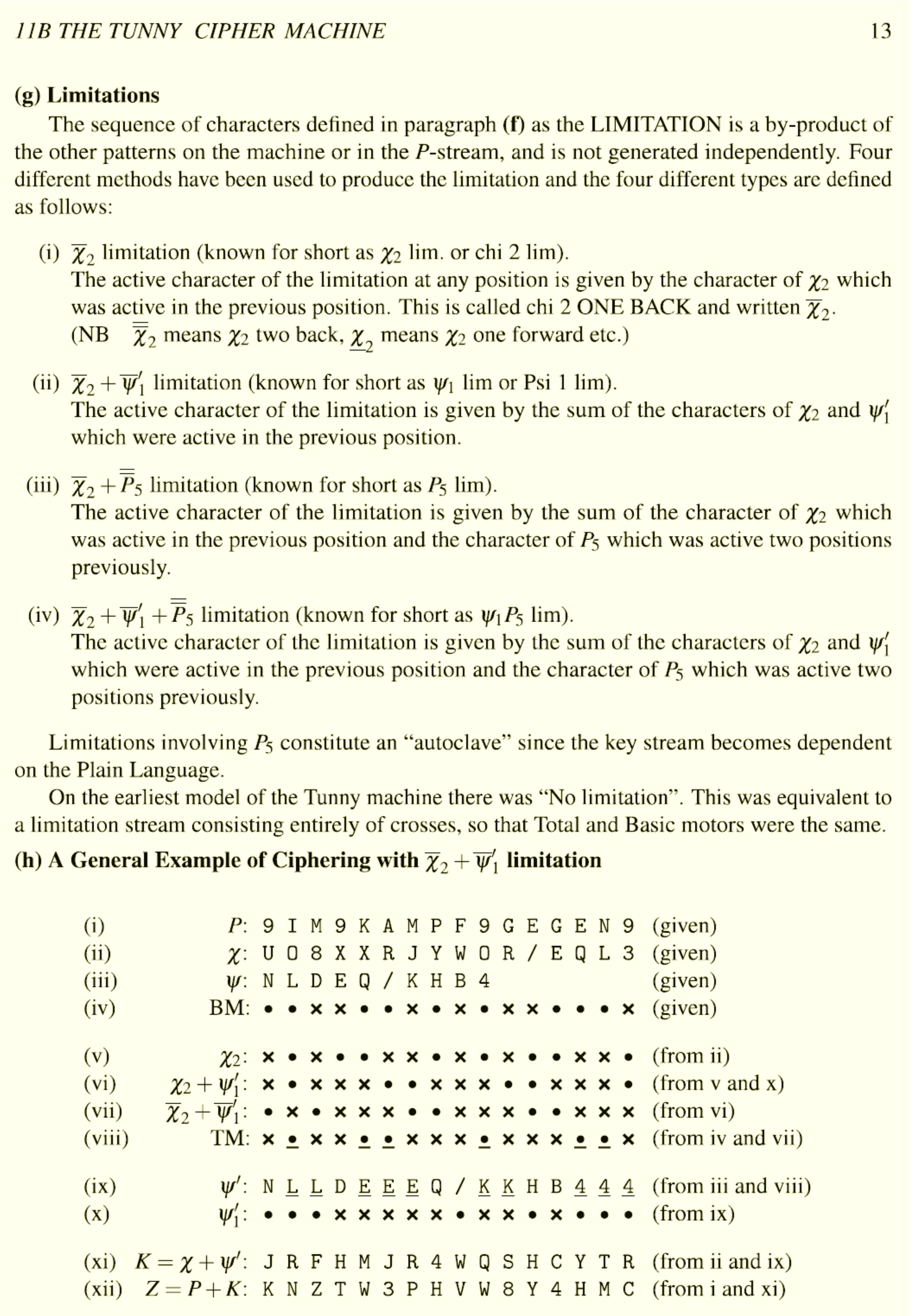
James Reeds, Whitfield Diffie, and J.V. Field. 2015. *Breaking Teleprinter Ciphers at Bletchley Park: An Edition of I.J. Good, D. Michie and G. Timms: General Report on Tunny with Emphasis on Statistical methods (1945)*. John Wiley & Sons.

Jerry Roberts. 2017. *Lorenz: Breaking Hitler's Top Secret Code at Bletchley Park*. The History Press, Stroud, Gloucestershire UK.

Sixta. 1945. *The Sixta History*. Bletchley Park Report HW 43/82. Kew, London: U.K. National Archives.

Testery. 1945. *Solution of German Teleprinter Cyphers (Testery) Linguistic Methods*. Bletchley Park Report HW 25/28. Kew, London: U.K. National Archives.

Appendix – Motor Limitations



11B THE TUNNY CIPHER MACHINE 13

(g) Limitations

The sequence of characters defined in paragraph **(f)** as the LIMITATION is a by-product of the other patterns on the machine or in the P-stream, and is not generated independently. Four different methods have been used to produce the limitation and the four different types are defined as follows:

(i) $\overline{\chi}_2$ limitation (known for short as χ_2 lim. or chi 2 lim).
The active character of the limitation at any position is given by the character of χ_2 which was active in the previous position. This is called chi 2 ONE BACK and written $\overline{\chi}_2$.
(NB $\overline{\overline{\chi}}_2$ means χ_2 two back, $\underline{\chi}_2$ means χ_2 one forward etc.)

(ii) $\overline{\chi}_2 + \overline{\psi}'_1$ limitation (known for short as ψ_1 lim or Psi 1 lim).
The active character of the limitation is given by the sum of the characters of χ_2 and ψ'_1 which were active in the previous position.

(iii) $\overline{\chi}_2 + \overline{\overline{P}}_5$ limitation (known for short as P_5 lim).
The active character of the limitation is given by the sum of the character of χ_2 which was active in the previous position and the character of P_5 which was active two positions previously.

(iv) $\overline{\chi}_2 + \overline{\psi}'_1 + \overline{\overline{P}}_5$ limitation (known for short as $\psi_1 P_5$ lim).
The active character of the limitation is given by the sum of the characters of χ_2 and ψ'_1 which were active in the previous position and the character of P_5 which was active two positions previously.

Limitations involving P_5 constitute an "autoclave" since the key stream becomes dependent on the Plain Language.

On the earliest model of the Tunny machine there was "No limitation". This was equivalent to a limitation stream consisting entirely of crosses, so that Total and Basic motors were the same.

(h) A General Example of Ciphering with $\overline{\chi}_2 + \overline{\psi}'_1$ limitation

(i)	P:	9	I	M	9	K	A	M	P	F	9	G	E	G	E	N	9	(given)
(ii)	χ:	U	0	8	X	X	R	J	Y	W	0	R	/	E	Q	L	3	(given)
(iii)	ψ:	N	L	D	E	Q	/	K	H	B	4							(given)
(iv)	BM:	•	•	×	×	•	•	×	•	×	•	×	×	•	•	•	×	(given)
(v)	χ_2:	×	•	×	•	•	×	×	•	×	•	×	•	•	×	×	•	(from ii)
(vi)	$\chi_2 + \psi'_1$:	×	•	×	×	×	•	•	×	×	×	•	•	×	×	×	•	(from v and x)
(vii)	$\overline{\chi}_2 + \overline{\psi}'_1$:	•	×	•	×	×	×	•	•	×	×	×	•	•	×	×	×	(from vi)
(viii)	TM:	×	$\underline{\bullet}$	×	×	$\underline{\bullet}$	$\underline{\bullet}$	×	×	×	$\underline{\bullet}$	×	×	×	$\underline{\bullet}$	$\underline{\bullet}$	×	(from iv and vii)
(ix)	ψ':	N	$\underline{\text{L}}$	L	D	$\underline{\text{E}}$	$\underline{\text{E}}$	E	Q	/	$\underline{\text{K}}$	K	H	B	$\underline{4}$	$\underline{4}$	4	(from iii and viii)
(x)	ψ'_1:	•	•	•	×	×	×	×	×	•	×	×	•	×	•	•	•	(from ix)
(xi)	$K = \chi + \psi'$:	J	R	F	H	M	J	R	4	W	Q	S	H	C	Y	T	R	(from ii and ix)
(xii)	$Z = P + K$:	K	N	Z	T	W	3	P	H	V	W	8	Y	4	H	M	C	(from i and xi)

Figure 5: Motor Limitations (Source: (Reeds et al., 2015, Chapter 11B, p. 13))