T52 Functional Description

Author: George Lasry
Last update: March 28, 2022

Table of Contents

Overview

The Siemens and Halske T52, codenamed “Sturgeon” by the Allies, is a family of German teleprinter encryption devices, used during WWII. It includes five main models: T52a/b, T52c, T52ca, T52d, and T52e. This document provides a functional description of the various models. It focuses on the details of the internal logic necessary for implementing a simulator or developing cryptanalytic attacks[1]. A full description of the physical T52 devices may be found in the references.

The Siemens and Halske T52

In the first section, the Baudot teleprinter alphabet is described, followed by a general overview of the design of the T52 devices. In the following sections, the functioning of each model is described in detail. An overview of the key setting procedures is also given. Finally, a list of references is provided.

The Baudot Alphabet

G

The T52 devices encrypt and decrypt teleprinter Baudot symbols. The Baudot alphabet consists of 32 symbols with five bits each. In addition to the 26 A-Z letters, it includes six control symbols, as follows:

---

[1] In some cases, an element presented here as a ‘unit’ may not have its equivalent as a physical unit. For example, some unit may be implemented by a set of wires. The focus of this document is to describe the functional logic, rather than the physical design of the devices.

- **SP** - for space
- **CR** - carriage return
- **LF** - Line feed (indicates a new line, sometimes preceded by CR)
- **FS** (or FIGS) - Figures shift mode, used for digits and other non-A-Z symbols
- **LS** (or LTRS) - Letter shift mode, to return the device to printing A-Z symbols
- **NUL** (of NULL) - A null symbol, usually not used in plaintexts

Bletchley Park and the Swedish FRA[2] used different notations for marking control symbols. Their notations, as well as the symbol values in letter and figure modes, are listed below.

[2] FRA: Försvarets Radioanstalt - the Swedish SIGINT organization.

			Letter Mode	Figure Mode	British Notation	Swedish Notation
00000	00	00	<NULL>		/	6
00001	01	01	E	3	E	E
00010	02	02	<CARRIAGE RETURN>		4	2
00011	03	03	A	-	A	A
00100	04	04	<SPACE>		9	5
00101	05	05	S	'	S	S
00110	06	06	I	8	I	I
00111	07	07	U	7	U	U
01000	08	08	<LINE FEED>		3	1
01001	09	09	D	ENQ	D	D
01010	10	0A	R	4	R	R
01011	11	0B	J	BELL	J	J
01100	12	0C	N	,	N	N
01101	13	0D	F	!	F	F
01110	14	0E	C	:	C	C
01111	15	0F	K	(	K	K
10000	16	10	T	5	T	T
10001	17	11	Z	+	Z	Z
10010	18	12	L	)	L	L
10011	19	13	W	2	W	W
10100	20	14	H	£	H	H
10101	21	15	Y	6	Y	Y
10110	22	16	P	0	P	P
10111	23	17	Q	1	Q	Q
11000	24	18	O	9	O	O
11001	25	19	B	?	B	B
11010	26	1A	G	&	G	G
11011	27	1B	<FIGURES SHIFT>		+	4
11100	28	1C	M	.	M	M
11101	29	1D	X	/	X	X
11110	30	1E	V	;	V	V
11111	31	1F	<LETTERS SHIFT>		8	3

The Baudot Alphabet and Notations

Example of Baudot encoding:

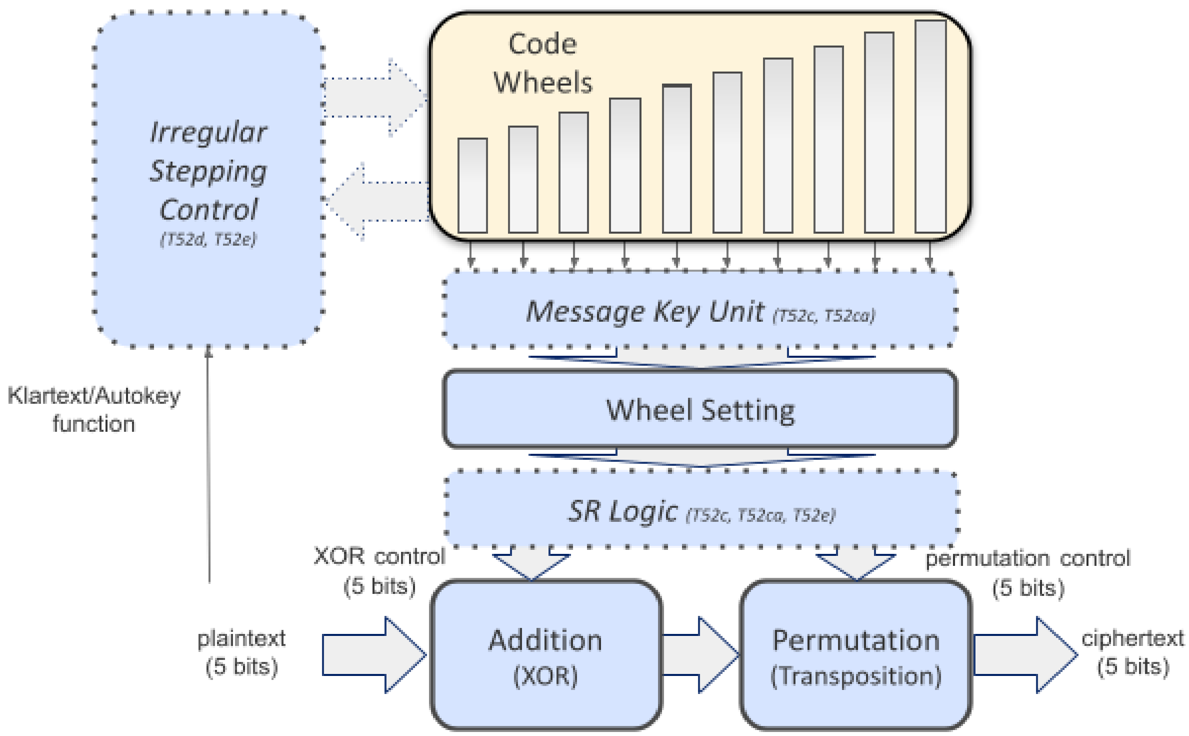
Raw plaintext	TO BE OR NOT TO BE? THIS IS THE QUESTION (SHAKESPEARE).

Baudot[3]	`TO BE OR NOT TO BE<?> THIS IS THE QUESTION <(>SHAKESPEARE<).`
British notation	`TO9BE9OR9NOT9TO9BE+B89THIS9IS9THE9QUESTION9+K8SHAKESPEARE+LM`
Swedish notation	`TO5BE5OR5NOT5TO5BE4B35THIS5IS5THE5QUESTION54K3SHAKESPEARE4LM`

T52 Design Overview

Overview

The functional design of the T52 family of devices is illustrated in the following diagram.



General Design of T52 - All Models

The elements in a solid border appear in all models. The elements in a dotted border appear only in some of the models.

All T52 models have an *XOR addition* and a *permutation* circuits, which together implement encryption (from left to right in the diagram) or decryption (right to left in the diagram). To encrypt

[3] The < sign represents here FS, and the > sign represents LS.

a 5-bit Baudot symbol, a 5-bit value (XOR control bits) is added using XOR (eXclusive OR)[4]. The resulting five bits are then permuted (or transposed) by the permutation circuit.

All T52 models have ten *code wheels* and a *wheel setting unit*, which routes the wheels' states according to their assigned functions. The T52d and T52e implement an irregular stepping of their wheels, while stepping is regular in all other models.

The T52c and T52ca have an additional *message key unit.* Finally, the T52c, the T52ca, and the T52e have an *SR logic* circuit (which differs per model).

Code Wheels

All models have the same set of 10 code wheels, labeled **`K, J, H, G, F, E, D, C, B, A`** from left to right (the letter **`I`** is omitted). Those wheels have 47, 53, 59, 61, 64, 65, 67, 69, 71, and 73 code pins, respectively. Those sizes are co-prime, ensuring that the sequence generated by the state of the ten wheels will not repeat itself in less than a period of 893,622,318,929,520,960 steps (about $2^{59.6}$). The patterns of the code wheels (the positions are active or inactive) are fixed[5], as described in the table below (**`x`** indicates an active position, **`.`** an inactive position):

[4] Each bit of the plaintext symbol is XORed with the respective bit of the XOR control. If the bits are equal (e.g. both are 1, or both are 0), then their XORed value is 0 (e.g. $1 \oplus 1 = 0$, where $\oplus$ denotes the XOR operation). If they are different (one is 1, and the other 0), then their XORed value is 1 (e.g. $1 \oplus 0 = 1$).
[5] Selmer and Weierud (see references) mention models with modified wheel patterns, mostly after WW2.

#	Size	Step Offset	Pattern
K	47	16	`.x....xxxxx...x..x.xxx..x.x.xxx..xxx.xxxx...xxx`
J	53	18	`.x.xx.xx..xx.x.xx...xxx...xxx..x.xxx.x....xxxx...x.xx`
H	59	20	`.x..xxxxx...xxxx.x..xxxx...xx.x...xx.x.xx...xxxxxx..xx.x.xx`
G	61	20	`.xxxx...xx..xx...x.xxxxx.x..x..xxx.x.xx.x.xx.x....xx.x..x.xxx`
F	64	22	`.x.x.xxxxx....x.xxxx..x.xx.xxxx.xx..xxx....x....xxx.xxxxxx...x.x`
E	65	22	`.x...x...x.xxxxxxx.x.xxxxxx..x.x..xx.x....x.xxxxx..xxx.x.x.xx...x`
D	67	23	`.xx.x.x...xxx.....xx.x....x..xxx..xxx..x.xxx...xx..x..x.xxxx..x...x`
C	69	23	`.xxxxx.x.xx..x.xx....xx.x....xxxx.xxx.xxx...x..x..xxx....xx....xx...x`
B	71	24	`.xxxx..xxxxxx...x...x.x..xx.xxxxx.....xxx...xx.xxx.....xx.x.x.xxx.....x`
A	73	25	`.xx.xxx...x..xx..x..xxxx..xxx.x.x.xxx..xx..xxx.xx.....xxx...xxxx.x......x`

While the wheels in the T52a/b, T52c, and T52ca step in a regular manner, the T52d and T52e feature irregular stepping, controlled by the state of other wheels (as well as by the last plaintext symbol when the Klartext/autokey feature is activated).

The state of each wheel - whether active or inactive - at the *current position* affects the encryption or decryption process. The state of wheels at some per-wheel offset (*step offset*) from the current position affects other wheels' stepping in models with irregular stepping (T52d and T52e).

Wheel Setting Unit

All models have a *wheel setting* unit, which allows the operator to select each wheel's function - the *wheel order*[6]. This unit routes the wheel state (active or inactive) at the current position to the selected element: either to one of the XOR or permutation control bits for T52a/b and T52d or to one of the inputs to the SR Logic of the T52c, T52ca or T52e.

[6] Although the order of the wheels cannot be physically changed, the logical effect of the wheel setting unit is to reorder the wheels so that their functions may be selected.

Message Key Unit

The *message key unit*, present only in the T52c and T52ca, allows the operator to change the function of the wheels (or their order) for each message being encrypted. The message key unit reroutes the state of the wheels (at their current positions) before entering the wheel settings unit (which further reroutes them). Therefore, the effective wheel order and their functions are controlled by both the message key and the wheel setting unit.

Historically, the wheel settings were set once every day and used for all messages during that day, while only the wheels' positions were changed for each message. Changing the wheels' effective order for each message (using the message key unit) reduces the risk of transmitting messages in-depth (i.e., encrypted using the same key settings).

SR Logic

In the T52a/b and T52d, each wheel is assigned (via the wheel setting unit) to either one of two functions:

- One of the five bits used for the XOR addition
- One of the five bits which control the permutation circuit

The *SR logic*, present only in T52c, the T52ca, and the T52e, eliminates this one-to-one relationship by combining several wheels' states to generate each of the five XOR addition bits and of the five permutation control bits.

XOR Addition and Permutation

To encrypt a plaintext symbol, five bits from either the SR logic output (T52c, T52ca, and T52e) or the wheel setting unit (T52a/b and T52d) are XORed with the five bits of the symbol. The resulting five bits are then permuted by the *permutation circuit* to produce the ciphertext symbol. The permutation circuit implements a set of five switches, also controlled by five bits from either the SR logic output (T52c, T52ca, and T52e) or the wheel setting unit (T52a/b and T52d). When decrypting a ciphertext symbol, the inverse permutation is first applied, and the same XOR addition is performed[7].

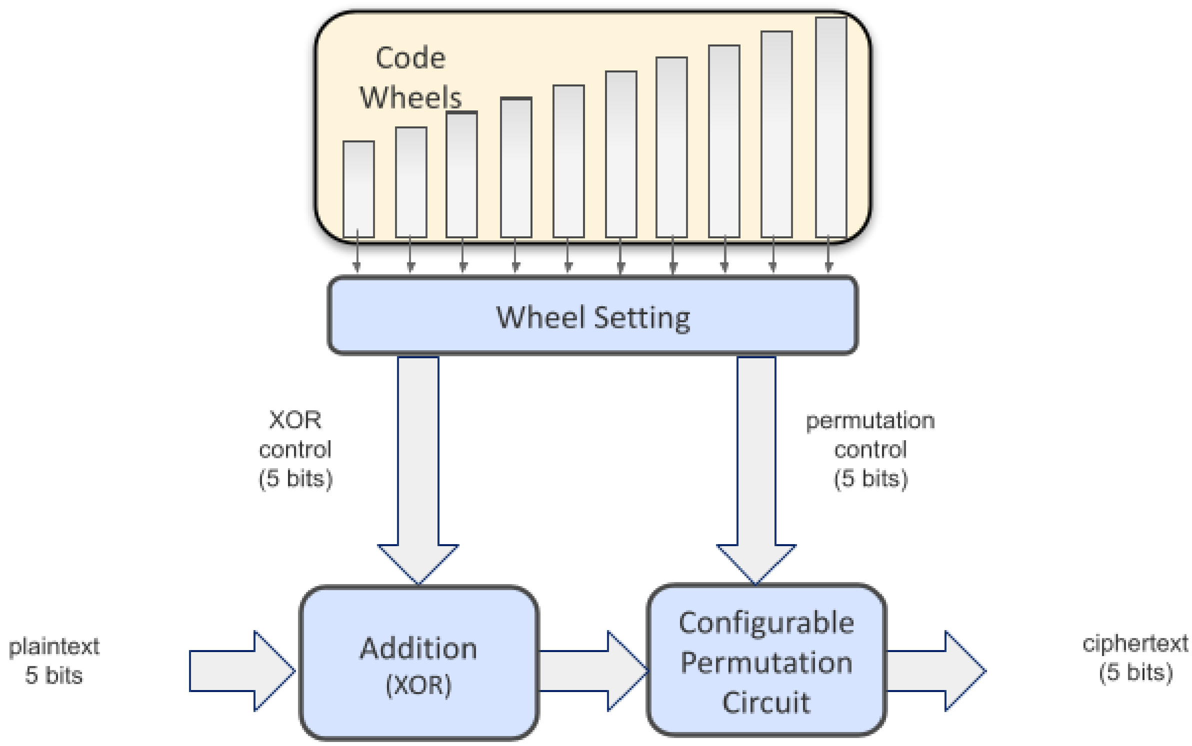
The T52c, T52ca, and T52e models have a fixed, non-configurable permutation circuit (described here). The permutation circuit of the T52a/b and T52d can be configured (described here).

[7] This cancels out the original XOR addition done when encrypting, as $y \oplus x \oplus x = y$ ($\oplus$ denotes the XOR operation).

T52a/b - Functional Description

Design

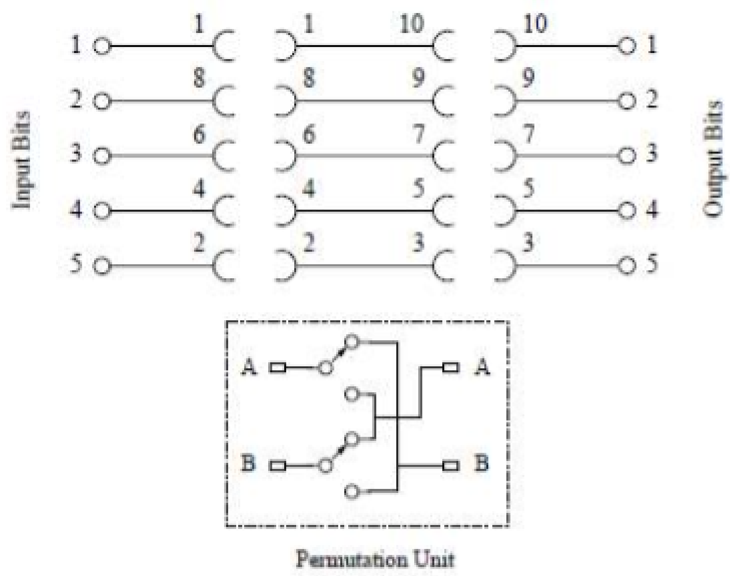
The T52a was the first model of the family. It was superseded by the T52b, which is functionally equivalent (with a modification for improving electrical noise suppression). Those two models are collectively referred to as T52a/b. Their design is illustrated in the following diagram:



T52a/b Design

Configurable Permutation Circuit

The T52a/b (and the T52d) features a configurable permutation circuit, with configurable switches. The circuit is described in the following diagram (Source: http://www.rutherfordjournal.org/article010106.html - Figure 16):



Programmable Permutation Circuit (T52a/b, T52d)

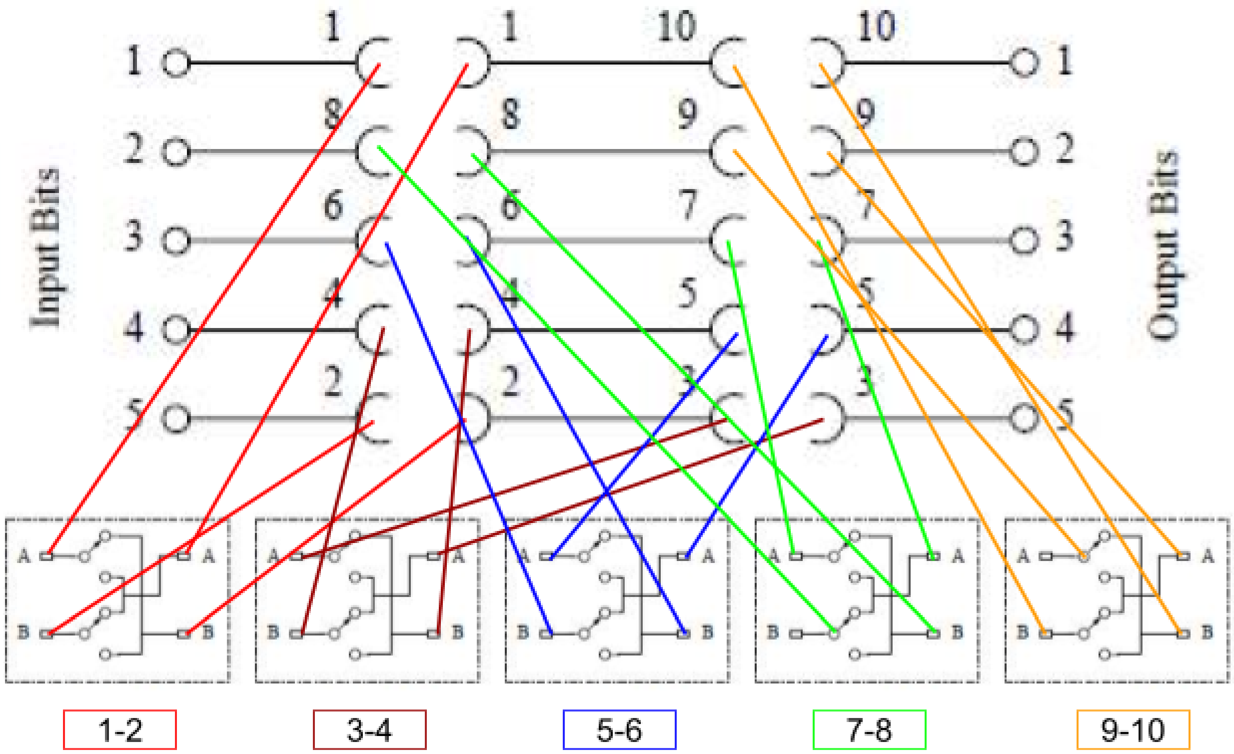
There are five permutation units or *permutation switches*. The configuration of a switch is denoted by a pair of numbers from 1 to 10, e.g., `1-2`. The configuration of the complete permutation circuit consists of five such pairs, where each number appears only once (for example, `1-2` and `2-4` are not compatible).

When the current state of the wheel assigned to control a permutation switch is 0 (inactive), the switch connects left side A to right side B, and left side B to right side A. When the state is active (1), no swapping occurs, and the switch connects left side A to right side A, and left side B to right side B.

We illustrate the effect of the configuration on the permutation circuit, with the following wheel settings:

```
I:II:III:IV:V:1-2:3-4:5-6:7-8:9-10
```

This setting indicates that wheels A to E control the five XOR addition bits, and wheels F to K control the five configurable permutation switches (`1-2, 3-4, 5-6, 7-8,` and `9-10`, respectively). The permutation circuit for this configuration is depicted in the following diagram:



Programmable Permutation Circuit for Wheel Settings:
`I:II:III:IV:V:1-2:3-4:5-6:7-8:-9-10`

This permutation circuit can be simplified as described below - **`x`** indicates a permutation switch (the XOR addition circuit is also shown):

```
      F      G      H      J      K          A         B      C      D      E
      I     II     III    IV      V         1-2       3-4    5-6    7-8    9-10
1 ---XOR-----|------|------|------|------\-------/-------------------------\-/---- 1
             |      |      |      |       \     /                           X
2 ----------XOR-----|------|------|--------\---/--------------------\-/----/-\---- 2
                    |      |      |         \ /                      X
3 -----------------XOR-----|------|----------X---------------\-/----/-\----------- 3
                           |      |         / \               X
4 ------------------------XOR-----|--------/---\------\-/----/-\------------------ 4
                                  |       /     \      X
5 -------------------------------XOR-----/-------\----/-\------------------------- 5
```

XOR Addition and Programmable Permutation Circuit for Wheel Settings: `I:II:III:IV:V:1-2:3-4:5-6:7-8:-9-10`

This configuration of permutation switches (**`1-2:3-4:5-6:7-8:9-10`**) is equivalent to the standard and fixed (non-programmable) permutation circuit, employed in the T52c, T52ca, and

T52e models. It was also used for T52a/b traffic in teleprinter lines traversing Sweden, from April 1942.

Similarly, wheel settings **`IV:3-6:I:1-8:III:4-7:V:5-10:II:2-9`** are implemented as follows:

```
     C      J      E      A      G       D        K          B       F        H
     I      II    III     IV     V      1-8      2-9        3-6     4-7      5-10
1 ---XOR-----|------|------|------|------\-/------------------------------\-----/---- 1
             |      |      |      |       X                                \   /
2 ----------XOR-----|------|------|------/-\----\-----/---------------------\-/------ 2
                    |      |      |              \   /                       X
3 -----------------XOR-----|------|---------------\-/------\---/----\-/------/-\------ 3
                           |      |                X        \ /      X      /   \
4 ------------------------XOR-----|---------------/-\--------X------/-\----/-----\---- 4
                                  |              /   \      / \
5 -------------------------------XOR------------/-----\----/---\---------------------- 5
```

XOR Addition and Programmable Permutation Circuit for Wheel Settings: `3-6:I:1-8:III:4-7:V:5-10:II:2-9`

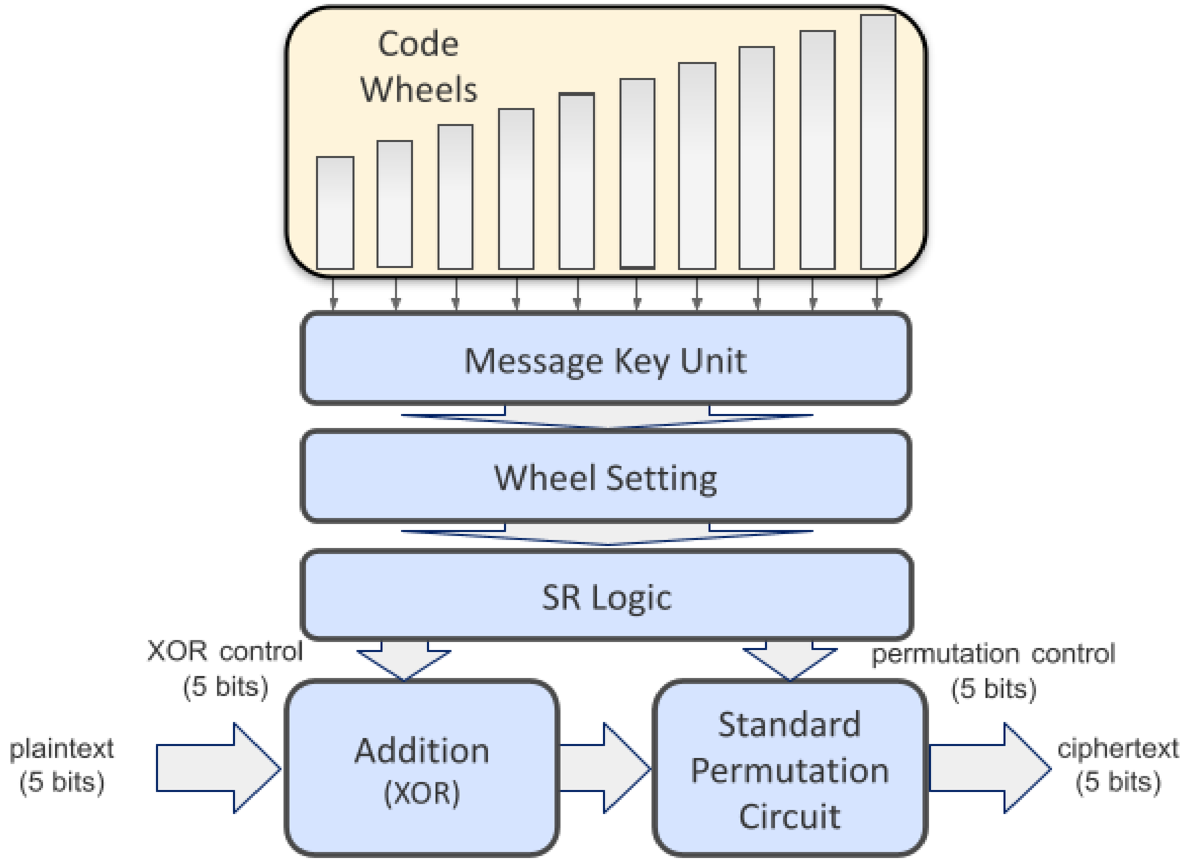
Note that not all switch configurations are valid. The configurations **`1-10`**, **`2-3`**, **`4-5`**, **`6-7`**, and **`8-9`** are degenerate, as they leave some of the input bits stuck, so that are never routed to any output bit[8].

T52c - Functional Description

Design

The design of the T52c model is described in the following diagram:

[8] A detailed analysis of the strength of the various configurations is given in http://www.rutherfordjournal.org/article010106.html.



T52c Design

Message Key Unit

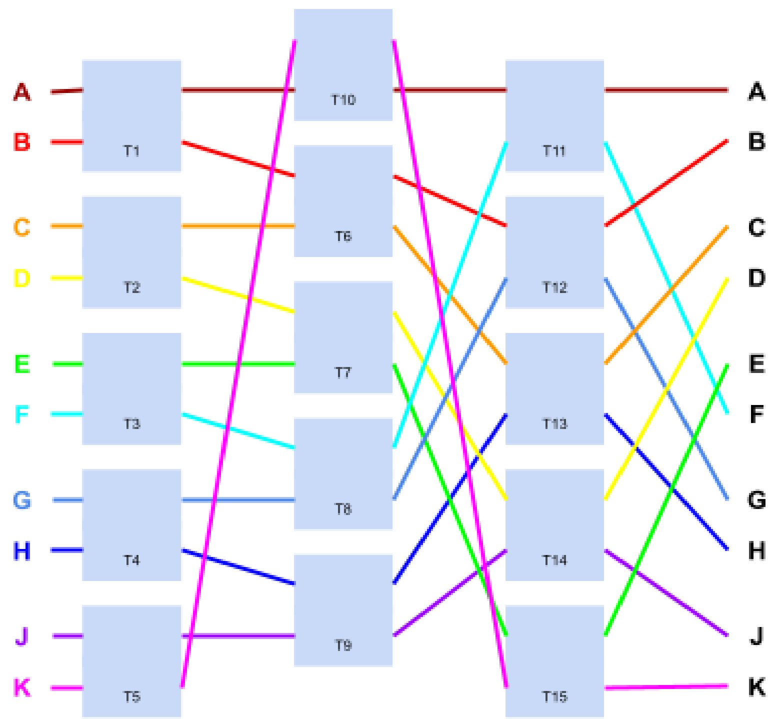
To reduce the risk for depths, e.g., several messages encrypted with the same key settings, the message key unit was introduced in the T52c. This unit reorders the wheels and changes their function for each message, in addition to (and before) the reordering performed by the wheel setting unit.

The message key unit is configured using five levers, each controlling three permutation switches, with a total of 15 switches (T1 to T15). Each lever can be set to one of 8 states, denoted as **`P`**, **`S`**, **`T`**, **`U`**, **`W`**, **`X`**, **`Y`**, and **`Z`**. A complete configuration of the message key unit consists of five letters from that set (with repetition allowed), such as **`PZXUS`** (the message key unit is in a neutral state and has no effect), or **`PSTUW`**. The effect of each setting on the levers and their switches is described in the table below:

		P	S	T	U	W	X	Y	Z
	T01				X	X	X		X
Lever 1	T06		X			X		X	X
	T11			X			X	X	X
	T02			X	X	X		X	
Lever 2	T07	X			X		X	X	
	T12		X			X	X	X	
	T03	X	X	X		X			
Lever 3	T08		X		X	X		X	
	T13			X	X	X			X
	T04	X		X				X	X
Lever 4	T09		X	X		X			X
	T14	X	X	X			X		
	T05	X				X	X	X	
Lever 5	T10	X		X			X		X
	T15	X			X			X	X

Message Key Unit Logic

For example, with **PZXUS**, none of the switches are activated, as described in the diagram below:

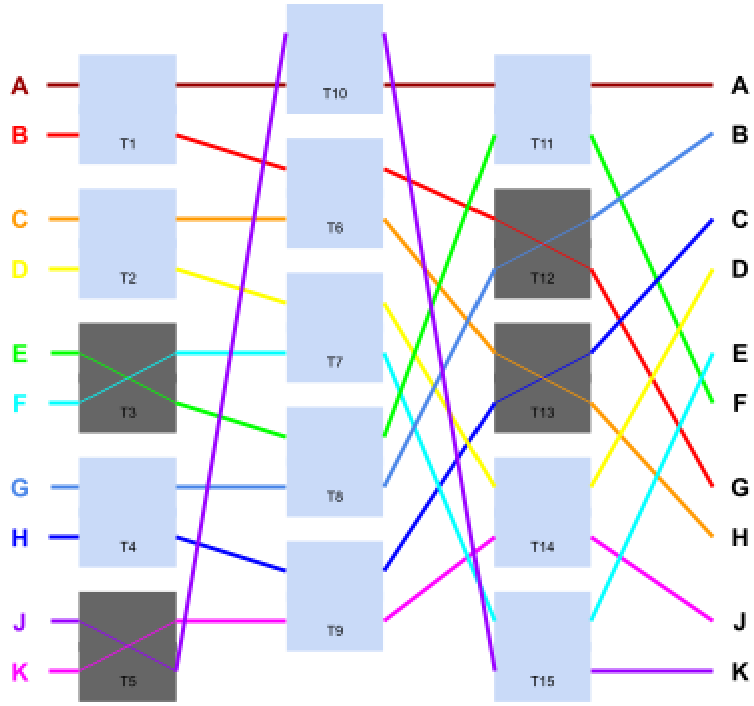


Message Key Unit Logic for PZXUS (Neutral)

With message key unit configuration **`PSTUW`:**

- **`P`** for Lever 1 does not activate any switch
- **`S`** for Lever 2 activates T12
- **`T`** for Lever 3 activates T3 and T13
- **`U`** for Lever 4 does not activate any switch
- **`W`** for Lever 5 activates T5

The message key unit circuit for this configuration is described in the diagram below:



Message Key Unit Logic for
PSTUW (T3, T5, T12, T13 activated)

SR Logic for T52c

With the T52a/b, each of the ten wheels was associated with one of the 10 bits that control the XOR addition (five bits) and the permutation (five bits), and vice-versa. To further complicate this association, additional logic was introduced in the T52c, the SR logic, which is also present in the T52ca and T52e (but is different for each model).

The SR logic for T52c is described below:

	1	3	5	7	9	I	II	III	IV	V
SR01	X	X				X		X		
SR02		X	X				X		X	
SR03			X	X				X		X
SR04				X	X	X			X	
SR05	X				X		X			X
SR06			X	X			X		X	
SR07		X	X					X		X
SR08	X	X					X			X
SR09	X				X	X		X		
SR10				X	X	X			X	

SR Logic for T52c

The inputs to the SR logic are marked as **1**, **3**, **5**, **7**, **9**, **I**, **II**, **III**, **IV**, **V.** For example, if the wheel settings are **I:1:3:III:II:7:V:9:IV:5** (and assuming the message key unit is set to the neutral **PZXUS** setting) then the inputs to the SR logic are mapped as follows:

SR Input:	1	3	5	7	9	I	II	III	IV	V
Wheel:	B	C	K	F	H	A	E	D	J	G

Each SR logic output implements an XOR addition (denoted with ⊕) on four inputs. For example:

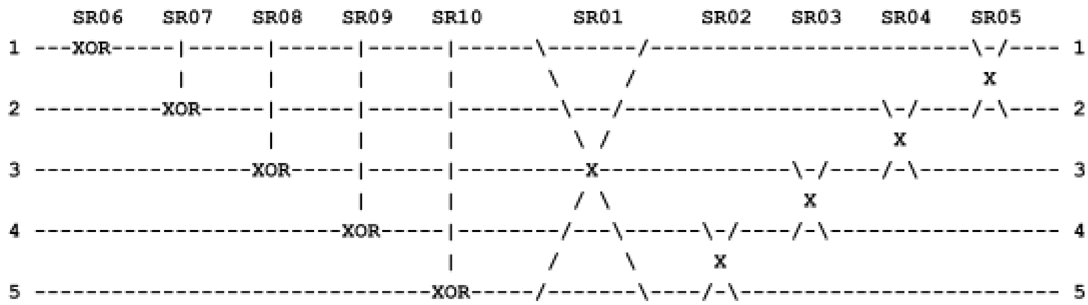
SR02 = [3] ⊕ [5] ⊕ [II] ⊕ [IV]

where **[3]**, **[5]**, **[II]**, and **[IV]** are the states of SR inputs **3**, **5**, **II**, and **IV**.

The function of the outputs (SR01 to SR10) is described in the next section.

Fixed Permutation Circuit

The XOR addition and the permutation circuits for T52c are as follows:



XOR Addition and Permutation
Circuit for T52c, T52ca, and T52e

SR logic outputs **`SR06`** to **`SR10`** control the five bits for XOR addition, and SR logic outputs **`SR01`** to **`SR05`** control the five permutation switches. A permutation switch is activated and will swap the two incoming inputs when the relevant SR output is inactive (0). For example, if **`SR01`** is inactive, and **`SR02`** to **`SR05`** are active, then input bits 1 and 5 are swapped (bits 2, 3, and 4 are unchanged).

The permutation circuit of the T52c (and of the T52ca and T52e) is fixed and cannot be configured, unlike the configurable permutation circuit of the T52a/b and T52d.

T52ca - Functional Description

Design

The T52ca model is identical to the T52c, except that the SR logic is different. It was modified as a result of an analysis of the T52c and its SR logic, which exposed several weaknesses[9]. All the other functions are identical to those of the T52c, including the message key unit and the fixed permutation circuit.

SR Logic for T52ca

The SR logic for T52ca is described below (see here for more information about the SR logic):

[9] See http://www.rutherfordjournal.org/article010106.html.

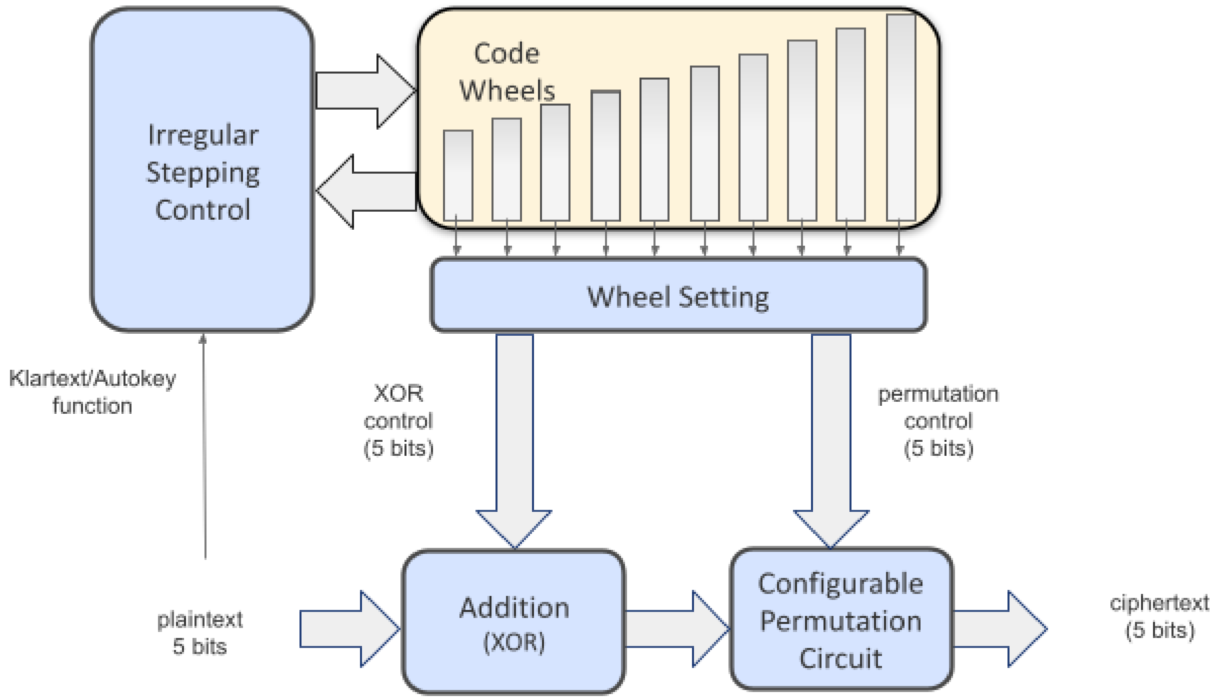
	1	3	5	7	9	I	II	III	IV	V
SR01				X	X	X			X	
SR02	X	X	X	X						
SR03				X		X	X			X
SR04	X				X		X		X	
SR05	X				X			X		X
SR06	X	X					X			X
SR07						X	X	X	X	
SR08		X	X		X				X	
SR09			X	X		X		X		
SR10		X	X					X		X

SR Logic for T52ca

T52d - Functional Description

Design

The design of the T52d is illustrated in the following diagram:



T52d Design

The T52d is similar in design to the T52a/b, except that it implements an irregular stepping of its wheels. As with the T52a/b, there is no SR logic, and there is no message key unit. Also, the permutation circuit is configurable (more details about the XOR addition and configurable permutation circuit).

Irregular Wheel Stepping

The wheel stepping logic is described in the following table:

Wheel	Without Klartext, steps if:	With Klartext, steps if:
A	¬f ∨ ¬e	b ∨ c ∨ z
B		¬c ∨ d ∨ z
C		e ∨ ¬d
D		¬f ∨ ¬e
E	¬g ∨ f	¬g ∨ f ∨ ¬z
F	h ∨ g	h ∨ g ∨ ¬z
G	¬j ∨ ¬h	¬j ∨ ¬h

H	**¬k ⋁ j**	**¬k ⋁ j**
J	**¬a ⋁ k**	**¬a ⋁ k**
K	**e ⋁ ¬d**	**a ⋁ ¬b**

The stepping of each wheel is governed by the state of two other wheels, at fixed offsets (or distance) from their current positions. For example, the code of wheel **A**, which affects the stepping of other wheels (such as **J**), is at an offset of +25 from the wheel's current position. It is denoted as **a**. Similarly, the code on wheel **K**, which affects the stepping of other wheels (such as **J**) and is denoted as **k**, is at offset +16 from the current position. The notation **¬a ⋁ k** in the table indicates that wheel **J** will step if either **a** is inactive, or if **k** is active[10].

When the Klartext (or autokey) function is activated, the stepping of some wheels is also affected by the state of the 3rd bit of the previous plaintext symbol, denoted as **z** in the table. The effect of the Klartext function is that pure depths are not possible, as the stepping is different for each plaintext. The drawback of the Klartext function is that any error in reception or in decryption will cause the wheels to be at wrong positions.

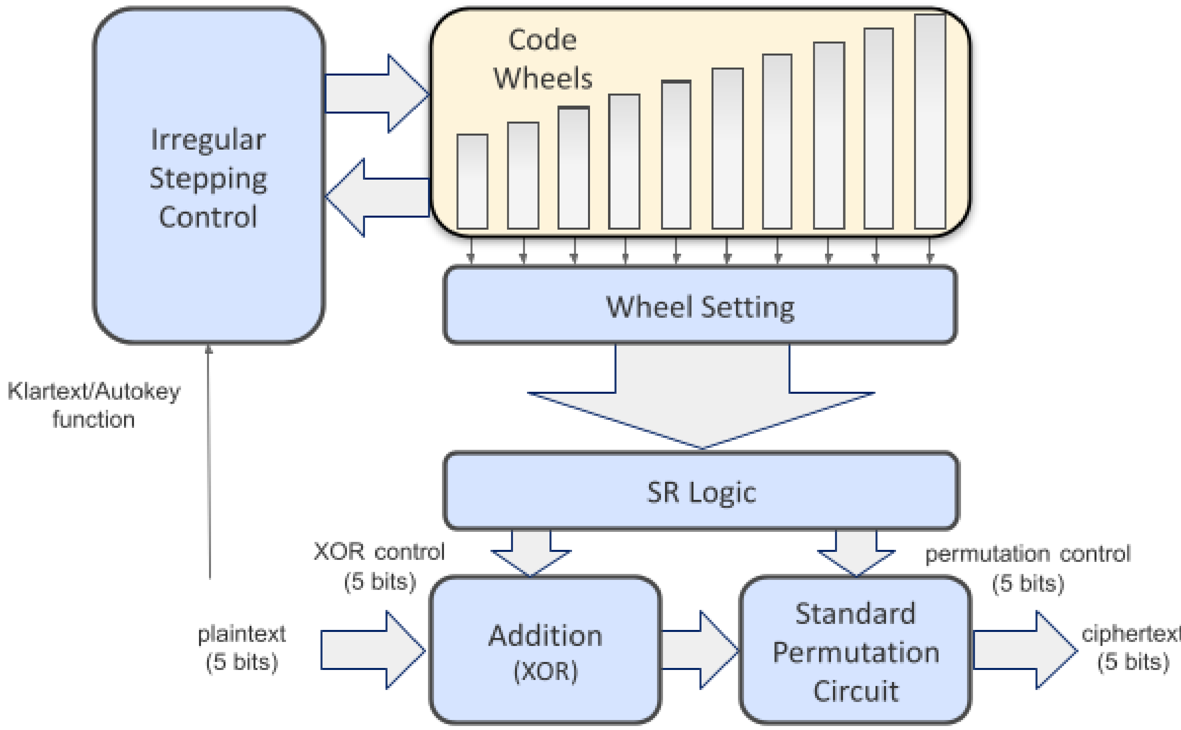
T52e - Functional Description

Design

The T52e model combines features from the T52c and T52ca, such as the addition of an SR logic (which is specific to the T52e) and the use of a non-configurable permutation circuit, with an irregular stepping mechanism identical to the stepping mechanism of the T52d. Unlike the T52c and T52ca, however, the T52e has no message key unit.

The design of the T52e is presented below:

[10] To avoid a situation in which none of the wheels step (all wheels are stuck forever at the same positions), most wheels affect at least one wheel when in active state (at the step offset), and at least one other wheel when in inactive state.



T52e Design

SR Logic for T52e

The SR logic for the T52e is presented below (see here for more details about the SR logic):

	1	3	5	7	9	I	II	III	IV	V
SR01	X	X						X		X
SR02						X	X	X	X	
SR03			X	X				X		X
SR04	X	X	X				X			
SR05	X			X	X		X			
SR06			X	X			X		X	
SR07		X	X		X					X
SR08	X	X				X			X	
SR09					X	X		X		X
SR10				X	X	X			X	

SR Logic for T52e

Key Setting Procedures

The key settings consist of two parts, the *inner key (Fernschreibgrundschlüssel)* and the *message key (Fernschreibspruchschlüssel)*. The inner key was typically set once per day, and guidelines mandated that the message key be different for each message. In practice, German operators often reused the same message key, providing the FRA and Bletchley Park with copious amounts of depth (i.e., messages encrypted with the same key).

The following table presents the main key settings scenarios used in WW2.

Relevant Models	Inner Key	Message Key
T52a/b	Wheel settings (order and functions) and positions of five wheels	Positions of the five remaining wheels
T52a/b, T52d, T52e	Wheel settings (order and functions)	Positions of all wheels
T52c, T52ca	Wheel settings (order and functions)	Positions of all wheels Message key unit settings (five letters)

References

The information in this document shall suffice for the implementation of a simulator. More information on the physical design of the devices, their historical uses and cryptanalysis, may be found in the references listed below.

The most comprehensive source of information on the T52:

- Weierud, Frode. "Bletchley Park's Sturgeon, the Fish that Laid No Eggs." *The Rutherford Journal: The New Zealand Journal for the History and Philosophy of Science and Technology* 1 (2005): 2005-2006. Link (Note: Figure 2 is erroneous).

Books:

- Beckman, Bengt. *Codebreakers: Arne Beurling and the Swedish crypto program during World War II*. American Mathematical Society, 2002 (a comprehensive account of the Swedish successes against the T52 during WW2).
- Ulfving, Lars and Weierud, Frode: The Geheimschreiber Secret: Arne Beurling and the Success of Swedish Signals Intelligence. In David Joyner (ed.), *Coding Theory and*

Cryptology: From Enigma and Geheimschreiber to Quantum Theory. New York: Springer Verlag (2000)
- Weierud, Frode: Sturgeon, The Fish BP Never Really Caught. In David Joyner (ed.), *Coding Theory and Cryptology: From Enigma and Geheimschreiber to Quantum Theory*. New York: Springer Verlag (2000)
- *A Swedish Success - Breaking the German Geheimschreiber during WW2* - FRA pamphlet. Link

Cryptologia Articles:

- Davies, Donald W.: *The Siemens and Halske T52e Cipher Machine*. Cryptologia 6(4) October (1982) 289-308
- Davies, Donald W.: *The Early Models of the Siemens and Halske T52 Cipher Machine*. Cryptologia 7(3) July (1983) 235-253
- Davies, Donald W.: *New Information on the History of the Siemens and Halske T52 Cipher Machine.* Cryptologia 18(2) April (1994) 141-146
- Mache, Wolfgang: *Geheimschreiber*. Cryptologia 10(4) October (1986) 230-242
- Mache, Wolfgang: *The Siemens Cipher Teletype in the History of Telecommunications*. Cryptologia 13(2) April (1989) 97-117
- Selmer, Ernst S.: *The Norwegian Modification of the Siemens and Halske T52e Cipher Machines. Cryptologia* 18(2) April (1994) 147-149

Archives:

- NRA (US): Captain Walter J. Fried's reports. National Archives and Records Administration (NARA) RG 457 NSA Historical Collection Box 880. Those reports contain a detailed description of Bletchley Park attacks against T52:
  - Report F-43 (IL 3530), 27 May 1944, Subject: Fish Notes
  - Report F-46 (IL 3601-A), 12 June 1944, Subject: Fish Notes
  - Report F-68 (IL 3639), 29 July 1944, Subject: Fish Notes
  - Report F-116 (IR 4054), 17 November 1944, Subject: Fish Notes
  - Report F-122 (IR 4069), 29 November 1944, Subject: Fish Notes

- FRA (Sweden): Reports providing details about the various models and a statistical attack developed against the T52a/b:
  - Carlbom, Lars: *Tyska Teletypechiffer (German teleprinter cipher).* In Swedish. Försvarets Radioanstalt (FRA), Stockholm, (31 January 1944). Source: Torbjörn Andersson. Link to original. Link to translation.
  - Carlbom, Lars: *Statistisk metod för forcering av Teletypechiffer av. AB - Typ. (Statistical Method for deciphering teleprinter cipher AB)*. In Swedish. Försvarets Radioanstalt (FRA), Stockholm, (21 March 1944). Source: Vik Anders, Torbjörn Andersson. Link to original. Link to translation.

- TICOM (US): Reports describing the attacks developed in Germany against the T52 models, and their assessment of the security of the various models:
    - Fricke, Dr. and Hüttenhain, Dr. Erich: *OKW/Chi Cryptanalytic Research on Enigma, Hagelin and Cipher Teleprinter Machines*. TICOM Report I-45. Link.
    - Hüttenhain, Dr. Erich, *Detailed interrogations of Dr. Huettenhain, formerly head of research section of OKW/CHI, 18th, 21st June 1945*, TICOM Report I-31. Link.