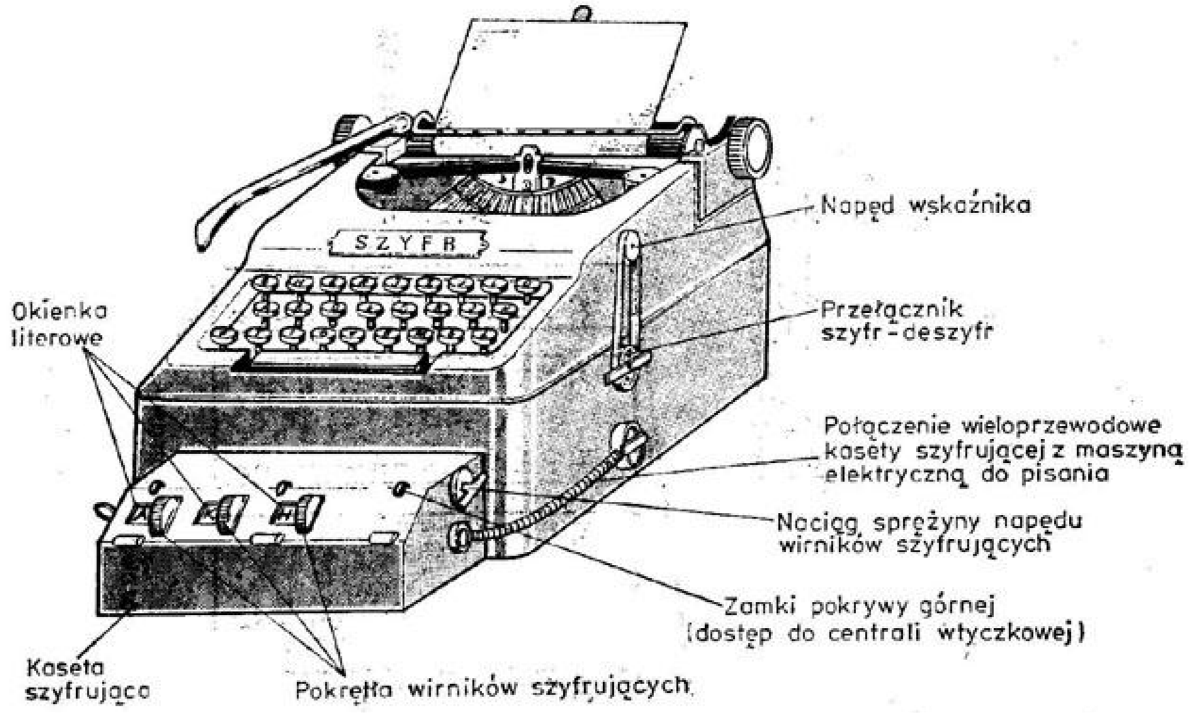
Introduction.

This article is about a WW II-era Polish Cryptomachine called **LACIDA**, named after its creators **LA**nger, **CI**ężki, and **DA**nilewicz. Not much is known about this machine, but what is known is quite interesting.

It is similar to Enigma, but it has six rotors, three of which turn during the encryption process, and three of which are fixed, and these rotors are of three different sizes. Unlike Enigma, there is no reflector or plugboard. There being no reflector, there is an explicit encryption mode and an explicit decryption mode - which is being used is selected by a switch. According to Wikipedia, Marian Rejewski held that this device was not very effective, and might have been easily broken **[1]**.

A closer look.



SZYFR
Napęd wskaźnika
Okienka literowe
Przełącznik szyfr - deszyfr
Połączenie wieloprzewodowe kasety szyfrującej z maszyną elektryczną do pisania
Naciąg sprężyny napędu wirników szyfrujących
Zamki pokrywy górnej (dostęp do centrali wtyczkowej)
Kaseta szyfrująca
Pokrętła wirników szyfrujących

Figure 1: A picture of LACIDA (from [2]).

Napęd wskaźnika.	**Display driver (to show "SZYFR"/"DESZFYR").**
Przełącnik szyfr-deszyfr.	**Cipher/Decipher switch.**
Połączenie wieloprzewodowe kasety szyfrującej z maszyną elektryczną do pisania.	**Multi-wire connection between the encryption system box and the electric typewriter.**
Naciąg sprężyny napędu wirników szyfrujących.	**Tension spring adjustment for the encryption rotor drive.**
Zamki pokrywy górnej (dostęp do centrali wtyczkowej).	**Top cover locks (access to the encryption system).**
Pokrętta wirników szyfrujących.	**Encryption rotors exposed for setting.**
Kaseta szyfrująca.	**Box containing the encryption system.**
Okienka literowe.	**Windows through which the turnable rotors can be seen.**

This author felt that it might be interesting to produce a simulator to replicate the behaviour of this machine; as this author's hardware design and construction skills are somewhat limited, this simulator would be entirely software-based, and its overall design would be determined by **[3]**.

The following problem was immediately encountered: none of the available documentation describes the wiring matrices for the rotors, although it was immediately apparent from **[2]** that the rotors have three different sizes **{24, 31, 35}** arranged as:

No:	**E/E**	**1**	**2**	**3**	**4**	**5**	**6**	**E/E**
	{24,	**24=24,**	**24=31,**	**31=31,**	**31=35,**	**35=35,**	**35=35,**	**35}**

Table 1. Rotor Layout and Sizes.

In the above table, the rotors **E/E** are the fixed entry/exit rotors;
rotors **1, 3, and 5** step during encryption/decryption;
rotors **2, 4, and 6** are fixed during encryption
(but can be preset by hand at the start of the encryption/decryption process.)

Given that the actual wiring is unknown, it was decided that the simplest solution was to generate the rotors' wiring matrices using a Random Number Generator. No attempt has been made to optimise the wiring.

Another interesting feature is that, unlike Enigma, there is no Reflector/Umkehrwalze, and therefore the device is non-reversible, and there has to be a Cipher/Decipher switch which determines the direction of current flow.

Referring to the above table:

during *encryption*, the current flow is from *left-to-right,* i.e. {1->8};
during *decryption,* the current flow is from *right-to-left,* i.e. {8->1}.

This leads to a keyboard quirk:

during *encryption*, only 24 keys (being ({A...Z} without {Q, V}) can be used, but 35 values ({A...Z}+{1...9}) can result from the encryption process;

during *decryption*, all 35 keys (being ({A...Z}+{1...9}) can be used, but only 24 values ({A...Z} without {Q, V}) can result from the decryption process.

There being no reflector, during encryption, there is theoretically a 1 in 35 chance that a character can encrypt to itself. This removes a famous Enigma design feature, generally considered to be a flaw, that no character could encrypt to itself.

An assumption is made that the three rotors which can rotate during en/de-cryption will make one step in the same direction of rotation before a character is en/de-crypted. This removes another Enigma problem, where for much of the operation of most versions of Enigma, the machine works as a one-rotor machine.

The Software Simulator.

The following points describe the simulator. Consider first the right-hand side of Figure 2 below:

The mode is "**SZYFR**".

The numbers on the line below represent the start positions of the rotors.

The keyboard is shown with only the keys valid for Encryption enabled; the keys valid only for Decryption are greyed-out, and disabled.

The 3 buttons below allow the switching of the simulator between Encryption mode ("**SZYFR**") and Decryption mode ("**DESZYFR**").

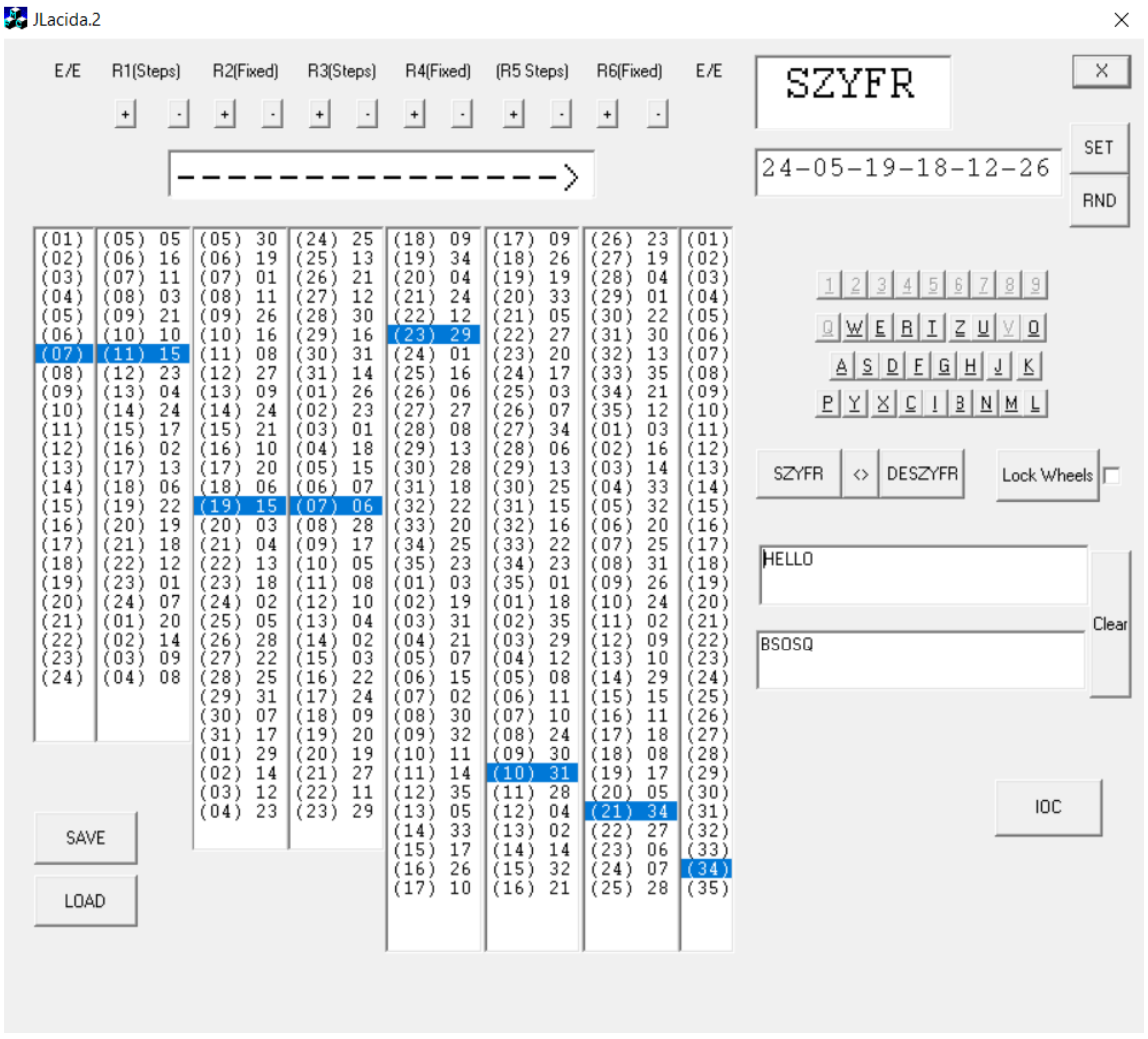
The plain text **"HELLO"** is shown encrypted into **"BSOSQ"**.

Consider next the left-hand side:

The 8 rotors, including the entry and exit connectors, are shown, with the 24-way wheels to the left, the 31-way wheels in the centre, and the 35-way wheels to the right. The current flow here is shown running from *left-to-right*, that is, from "O" to "Q", "O" being the last character encrypted, and "Q" being the result of that encryption.

The **[SAVE]** and **[LOAD]** buttons are used for the transfer of the rotor wirings and of their positions to another copy of the simulator.

There is a button **[IOC]** at the bottom of the picture which is reserved for the Index of Coincidence algorithm, which is described in more detail later in this article.



```
E/E   R1(Steps) R2(Fixed) R3(Steps) R4(Fixed) (R5 Steps) R6(Fixed) E/E

                 ----------------->

SZYFR

24-05-19-18-12-26

(01)  (05) 05  (05) 30  (24) 25  (18) 09  (17) 09  (26) 23  (01)
(02)  (06) 16  (06) 19  (25) 13  (19) 34  (18) 26  (27) 19  (02)
(03)  (07) 11  (07) 01  (26) 21  (20) 04  (19) 19  (28) 04  (03)
(04)  (08) 03  (08) 11  (27) 12  (21) 24  (20) 33  (29) 01  (04)
(05)  (09) 21  (09) 26  (28) 30  (22) 12  (21) 05  (30) 22  (05)
(06)  (10) 10  (10) 16  (29) 16  (23) 29  (22) 27  (31) 30  (06)
(07)  (11) 15  (11) 08  (30) 31  (24) 01  (23) 20  (32) 13  (07)
(08)  (12) 23  (12) 27  (31) 14  (25) 16  (24) 17  (33) 35  (08)
(09)  (13) 04  (13) 09  (01) 26  (26) 06  (25) 03  (34) 21  (09)
(10)  (14) 24  (14) 24  (02) 23  (27) 27  (26) 07  (35) 12  (10)
(11)  (15) 17  (15) 21  (03) 01  (28) 08  (27) 34  (01) 03  (11)
(12)  (16) 02  (16) 10  (04) 18  (29) 13  (28) 06  (02) 16  (12)
(13)  (17) 13  (17) 20  (05) 15  (30) 28  (29) 13  (03) 14  (13)
(14)  (18) 06  (18) 06  (06) 07  (31) 18  (30) 25  (04) 33  (14)
(15)  (19) 22  (19) 15  (07) 06  (32) 22  (31) 15  (05) 32  (15)
(16)  (20) 19  (20) 03  (08) 28  (33) 20  (32) 16  (06) 20  (16)
(17)  (21) 18  (21) 04  (09) 17  (34) 25  (33) 22  (07) 25  (17)
(18)  (22) 12  (22) 13  (10) 05  (35) 23  (34) 23  (08) 31  (18)
(19)  (23) 01  (23) 18  (11) 08  (01) 03  (35) 01  (09) 26  (19)
(20)  (24) 07  (24) 02  (12) 10  (02) 19  (01) 18  (10) 24  (20)
(21)  (01) 20  (25) 05  (13) 04  (03) 31  (02) 35  (11) 02  (21)
(22)  (02) 14  (26) 28  (14) 02  (04) 21  (03) 29  (12) 09  (22)
(23)  (03) 09  (27) 22  (15) 03  (05) 07  (04) 12  (13) 10  (23)
(24)  (04) 08  (28) 25  (16) 22  (06) 15  (05) 08  (14) 29  (24)
               (29) 31  (17) 24  (07) 02  (06) 11  (15) 15  (25)
               (30) 07  (18) 09  (08) 30  (07) 10  (16) 11  (26)
               (31) 17  (19) 20  (09) 32  (08) 24  (17) 18  (27)
               (01) 29  (20) 19  (10) 11  (09) 30  (18) 08  (28)
               (02) 14  (21) 27  (11) 14  (10) 31  (19) 17  (29)
               (03) 12  (22) 11  (12) 35  (11) 28  (20) 05  (30)
               (04) 23  (23) 29  (13) 05  (12) 04  (21) 34  (31)
                                 (14) 33  (13) 02  (22) 27  (32)
                                 (15) 17  (14) 14  (23) 06  (33)
                                 (16) 26  (15) 32  (24) 07  (34)
                                 (17) 10  (16) 21  (25) 28  (35)

HELLO

BSOSQ
```

Figure 2: The simulator in Encryption mode.

Next, let us consider the right-hand side of Figure 3 below:

The mode is "**DESZYFR**".

The numbers on the line below represent the start positions of the rotors; the same as in the previous "**SZYFR**" operation.

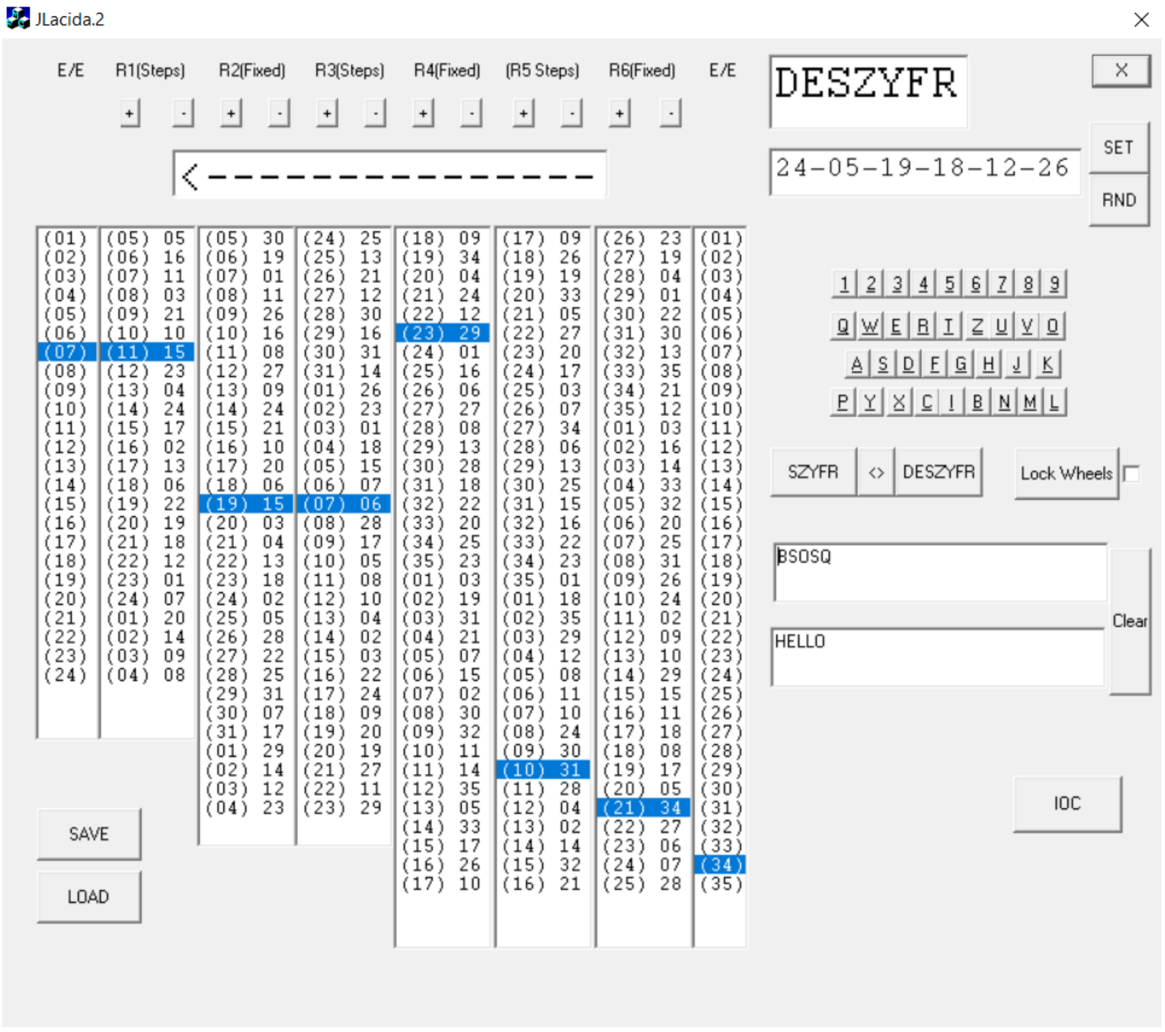
The keyboard is shown with all of the keys enabled.

As above, the 3 buttons below allow the switching of the simulator between Encryption mode ("**SZYFR**") and Decryption mode ("**DESZYFR**").

The encrypted text **"BSOSQ"** is shown decrypted into the plain text **"HELLO"**.

Consider next the left-hand side:

The 8 rotors, including the entry and exit connectors, are shown, with the 24-way wheels to the left, the 31-way wheels in the centre, and the 35-way wheels to the right. The current flow here is shown running from *right-to-left*, that is, from "Q" to "O", "Q" being the last character decrypted, and "O" being the result of that decryption.



DESZYFR

24-05-19-18-12-26

<-----------------

E/E	R1(Steps)	R2(Fixed)	R3(Steps)	R4(Fixed)	(R5 Steps)	R6(Fixed)	E/E
(01)	(05) 05	(05) 30	(24) 25	(18) 09	(17) 09	(26) 23	(01)
(02)	(06) 16	(06) 19	(25) 13	(19) 34	(18) 26	(27) 19	(02)
(03)	(07) 11	(07) 01	(26) 21	(20) 04	(19) 19	(28) 04	(03)
(04)	(08) 03	(08) 11	(27) 12	(21) 24	(20) 33	(29) 01	(04)
(05)	(09) 21	(09) 26	(28) 30	(22) 12	(21) 05	(30) 22	(05)
(06)	(10) 10	(10) 16	(29) 16	(23) 29	(22) 27	(31) 30	(06)
(07)	(11) 15	(11) 08	(30) 31	(24) 01	(23) 20	(32) 13	(07)
(08)	(12) 23	(12) 27	(31) 14	(25) 16	(24) 17	(33) 35	(08)
(09)	(13) 04	(13) 09	(01) 26	(26) 06	(25) 03	(34) 21	(09)
(10)	(14) 24	(14) 24	(02) 23	(27) 27	(26) 07	(35) 12	(10)
(11)	(15) 17	(15) 21	(03) 01	(28) 08	(27) 34	(01) 03	(11)
(12)	(16) 02	(16) 10	(04) 18	(29) 13	(28) 06	(02) 16	(12)
(13)	(17) 13	(17) 20	(05) 15	(30) 28	(29) 13	(03) 14	(13)
(14)	(18) 06	(18) 06	(06) 07	(31) 18	(30) 25	(04) 33	(14)
(15)	(19) 22	(19) 15	(07) 06	(32) 22	(31) 15	(05) 32	(15)
(16)	(20) 19	(20) 03	(08) 28	(33) 20	(32) 16	(06) 20	(16)
(17)	(21) 18	(21) 04	(09) 17	(34) 25	(33) 22	(07) 25	(17)
(18)	(22) 12	(22) 13	(10) 05	(35) 23	(34) 23	(08) 31	(18)
(19)	(23) 01	(23) 18	(11) 08	(01) 03	(35) 01	(09) 26	(19)
(20)	(24) 07	(24) 02	(12) 10	(02) 19	(01) 18	(10) 24	(20)
(21)	(01) 20	(25) 05	(13) 04	(03) 31	(02) 35	(11) 02	(21)
(22)	(02) 14	(26) 28	(14) 02	(04) 21	(03) 29	(12) 09	(22)
(23)	(03) 09	(27) 22	(15) 03	(05) 07	(04) 12	(13) 10	(23)
(24)	(04) 08	(28) 25	(16) 22	(06) 15	(05) 08	(14) 29	(24)
		(29) 31	(17) 24	(07) 02	(06) 11	(15) 15	(25)
		(30) 07	(18) 09	(08) 30	(07) 10	(16) 11	(26)
		(31) 17	(19) 20	(09) 32	(08) 24	(17) 18	(27)
		(01) 29	(20) 19	(10) 11	(09) 30	(18) 08	(28)
		(02) 14	(21) 27	(11) 14	(10) 31	(19) 17	(29)
		(03) 12	(22) 11	(12) 35	(11) 28	(20) 05	(30)
		(04) 23	(23) 29	(13) 05	(12) 04	(21) 34	(31)
				(14) 33	(13) 02	(22) 27	(32)
				(15) 17	(14) 14	(23) 06	(33)
				(16) 26	(15) 32	(24) 07	(34)
				(17) 10	(16) 21	(25) 28	(35)

BSOSQ

HELLO

Figure 3: The simulator in Decryption mode.

Polish alphabet.

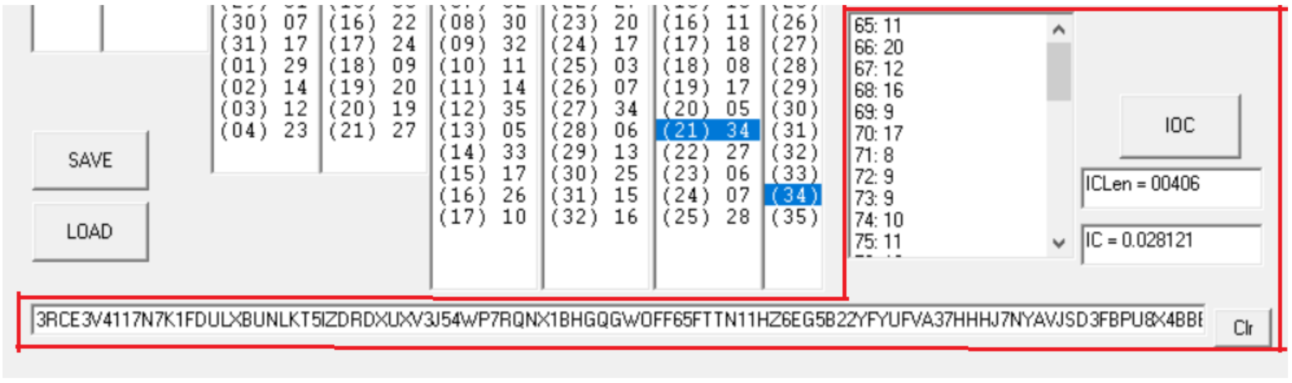
As many readers of this article will know, the Polish language uses 23 letters of the standard Latin (English) alphabet, being {A...Z} *without* {Q, V, X}. However, eight of those letters can have a diacritic applied, so that the full alphabet includes {Ą, Ć, Ę, Ł, Ń, Ó, Ś, Ź, Ż}. The question then arises as to how the Polish language could be used on this device, since, during encryption, only the letters {A...Z} *without* {Q, V} are available; there is no provision for a full Polish alphabet.

The possibility exists, inspired by the constructed language Esperanto, to use the spare letter "X" as an indicator, so for example, the words "**TOWAŻNAWIADOMOŚĆ**" ("this is an important message") *could* be rendered as "**TOWAZXNAWIADOMOSXCX**". It is *not* being suggested here that this protocol was actually used with this machine, but it's an inviting possibility.

Index of Coincidence.

The "Index of Coincidence" is a mathematical concept, which, amongst other things, can be used to determine the effectiveness of an encryption system. It is well-known, perhaps, that natural languages tend to use some letters more than others. English, for example, tends to use the letter "E" the most, followed by "T" and then "A". This is a somewhat non-mathematical description which can be reformulated as one usage of the Index of Coincidence (IOC) **[4]**.

In order to measure the behaviour of this simulator, an IOC calculator has been built into the simulator. Figure 4 below shows the result of running the IOC calculator while encrypting several hundred letter "A"s. The value "0.028121" shown is approximately 1/35, which is an appropriate value for a randomly distributed cryptotext using an alphabet of 35 characters. Similar values are achieved for other encryption sequences.



```
(30) 07 | (16) 22 | (08) 30 | (23) 20 | (16) 11 | (26)
(31) 17 | (17) 24 | (09) 32 | (24) 17 | (17) 18 | (27)
(01) 29 | (18) 09 | (10) 11 | (25) 03 | (18) 08 | (28)
(02) 14 | (19) 20 | (11) 14 | (26) 07 | (19) 17 | (29)
(03) 12 | (20) 19 | (12) 35 | (27) 34 | (20) 05 | (30)
(04) 23 | (21) 27 | (13) 05 | (28) 06 | (21) 34 | (31)
                  | (14) 33 | (29) 13 | (22) 27 | (32)
                  | (15) 17 | (30) 25 | (23) 06 | (33)
                  | (16) 26 | (31) 15 | (24) 07 | (34)
                  | (17) 10 | (32) 16 | (25) 28 | (35)
```

```
65: 11
66: 20
67: 12
68: 16
69: 9
70: 17
71: 8
72: 9
73: 9
74: 10
75: 11
```

```
ICLen = 00406
IC = 0.028121
```

```
3RCE3V4117N7K1FDULXBUNLKT5IZDRDXUXV3J54WP7RQNX1BHGQGW0FF65FTTN11HZ6EG5B22YFYUFVA37HHHJ7NYAVJSD3FBPU8X4BBE
```

Figure 4: The Index of Coincidence Calculator at Work.

About the Author.

Jerry McCarthy is a Volunteer Docent at The National Museum of Computing (TNMOC), Bletchley and an Occasional Speaker at the Instytut Józefa Piłsudskiego, Hammersmith, London.

[1] Available in English at https://en.wikipedia.org/wiki/Lacida ,
and in Polish at https://pl.wikipedia.org/wiki/Lacida .

[2] Found at https://armyradio.com/The_Enigma_Code_Breach.html labelled as "Lacida A Polish-made crypto machine used in diplo communications. Photo from 'Szyfr Enigmy - Metody Zlamania' book" by Krzysztof Gaj.

[3] Krzysztof Gaj, Polish Cipher Machine - Lacida, Cryptologia, Volume 16, Issue 1, January 1992. pp. 73-80.
Also
Krzysztof GAJ, The Polish Cipher Machine "Lacida", The Enigma Bulletin No. 1 - December 1990. pp 51-57.

[4] https://en.wikipedia.org/wiki/Index_of_coincidence