Me, **Thorsten Butz,** in front of the Bletchley Park mansion in 2023.

"**Bletchley Park** is an English country house and estate in Bletchley, Milton Keynes (Buckinghamshire) that became the principal centre of Allied code-breaking during the Second World War. The **mansion** was constructed during the years following 1883 for the financier and politician Sir Herbert Leon in the Victorian Gothic, Tudor, and Dutch Baroque styles, on the site of older buildings of the same name."

Reference: https://en.wikipedia.org/wiki/Bletchley_Park

The picture was taken in April 2023.

PowerShell Conference Europe

PS ENIGMA

Thorsten Butz

PRAGUE23

Wednesday, 21 June 2023, 17:00 h
Track 3, PSConfEU

Cubex Centrum Praha
http://www.cubexcentrum.cz/
Na Strži 2097/63, 140 00 Praha 4-Nusle

Many thanks to our sponsors:

Thanks for **sponsoring** this event!

- https://chocolatey.org
- https://patchmypc.com
- https://www.centinosystems.com
- https://www.scriptrunner.com
- https://synedgy.com

about_Session

```
{
    "Title"   : "PSEnigma",
    "Speaker" : "Thorsten Butz",
    "Uri"     : "thorsten-butz.de",
    "Twitter" : "@thorstenbutz",
    "Podcast" : "slidingwindows.de"
}
```

@thorstenbutz

Thorsten Butz is a distinguished "Microsoft Certified Trainer" (MCT), consultant, book author and podcaster. He started his career in the late 1990s as a networking specialist and Unix enthusiast. He conducts Microsoft's server technologies since 2000. His desire for scripting and automation brought him at an early stage to his current focal point: the Windows PowerShell
Thorsten is the host of the "Sliding Windows" (sllidingwindows.de) podcast.

about_Mugs

(

)

@thorstenbutz

https://psconf.eu
https://powershell.video

- PSConfEU 2016 Hannover
- PSConfEU 2017 Hannover
- PSConfEU 2018 Hannover
- PSConfEU 2019 Hannover
- PSConfEU 2020 Online Event
- PSConfEU 2022Vienna
- PSConfEU 2023 Prague

Motivation

```
CLASS ENIGMA:
  WHEEL
  REFLECTOR
  POSITION
```

```
CLASS ENIGMA {
  WHEEL
  REFLECTOR
  POSITION
}
```

1 JavaScript
2 Python
3 Java
4 PHP
5 C#
..
18 PowerShell
19 Rust
..

Reference: https://redmonk.com/sogrady/2023/05/16/language-rankings-1-23

about_Classes

https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_classes

The python tutorial: 9. Classes

https://docs.python.org/3/tutorial/classes.html

Python for PowerShell-ers with Kyle Ruddy (YT)

https://youtu.be/T5Lq7hwfVHs

Previously on cryptography

Book recommendation

Simon Singh: The code book
https://simonsingh.net/books/the-code-book/
http://www.amazon.co.uk/gp/reader/1857028899/

German version:
Simon Singh: Geheime Botschaften. Die Kunst der Verschlüsselung von der Antike bis in die Zeiten des Internet.

One-time pad

1	2	3	4	5	6	7
A	B	C	D	E	F	G
8	9	10	11	12	13	14
H	I	J	K	L	M	N
15	16	17	18	19	20	21
O	P	Q	R	S	T	U
22	23	24	25	26		
V	W	X	Y	Z		

1	2	3	4	5	6	7
G	L	E	P	O	W	I
8	9	10	11	12	13	14
Y	A	M	B	F	J	Q
15	16	17	18	19	20	21
S	X	D	Z	K	V	R
22	23	24	25	26		
H	U	T	M	C		

One-time pad (WP)

"In cryptography, the one-time pad (OTP) is an encryption technique that cannot be cracked, but requires the use of a single-use pre-shared key that is larger than or equal to the size of the message being sent. In this technique, a plaintext is paired with a random secret key (also referred to as a one-time pad). Then, each bit or character of the plaintext is encrypted by combining it with the corresponding bit or character from the pad using modular addition."

Reference: https://en.wikipedia.org/wiki/One-time_pad

One-time pad, ASCII

65	66	67	68	69	70	71
A	B	C	D	E	F	G
72	73	74	75	76	77	78
H	I	J	K	L	M	N
79	80	81	82	83	84	85
O	P	Q	R	S	T	U
86	87	88	89	90		
V	W	X	Y	Z		

65	66	67	68	69	70	71
G	L	E	P	O	W	I
72	73	74	75	76	77	78
Y	A	M	B	F	J	Q
79	80	81	82	83	84	85
S	X	D	Z	K	V	R
86	87	88	89	90		
H	U	T	M	C		

```
[char]65
[char]90
$offset = 65

## Create a ONE-TIME-PAD
[string] $myRandomPad = [char[]] (Get-Random -InputObject (65..90) -
Count 26) -join ''
$myRandomPad.Length

$letter = 'A'
$letter -match '^[A-Z]$'  ## Any number of times => '^[A-Z]+$'

function enryptLetter {
    param (
        [byte][char]$letter,
        [string]$otp
    )
    $offset = 65
    $otp[$letter - $offset]
}
enryptLetter -letter 'A' -otp $myRandomPad
```

Demo A

Code snippets

- Demo files from the Praguie conference https://github.com/thorstenbutz/conferences/tree/master/2023.PSConf.e

- https://github.com/thorstenbutz/PSEnigma

The enigma

Recommended videos

How did the Enigma Machine work? (Jared Owen)
https://youtu.be/ybkkiGtJmkM

The Enigma Machine - Bletchley Park takes a closer look at how it works
https://youtu.be/3Ux03qPgYVY

Enigma M3 emulator

https://www.101computing.net/enigma/

Enigma M3

1918 Artur Scherbius patents the Enigma

1920s Commercial usage

1932 Marian Rejewski finds weaknesses in the cryptography of the Enigma

1939 Polish cryptographers hand their findings over to the UK

History of the Enigma

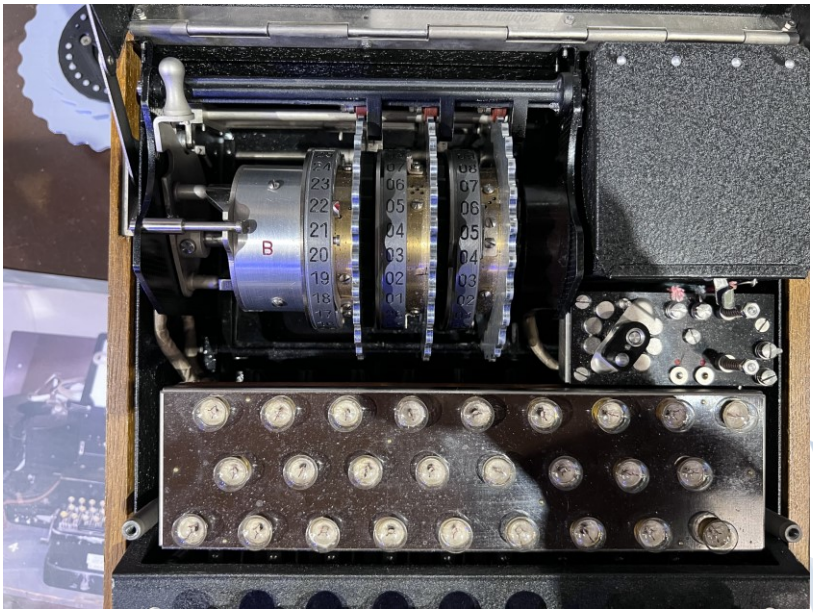
https://www.cryptomuseum.com/crypto/enigma/hist.htm

Enigma machine (WP)

https://en.wikipedia.org/wiki/Enigma_machine#

Enigma dictionary

Walze = rotor, wheel
Umkehrwalze (UKU) = reflector
Steckerbrett = plugboard
Verkabelung = wiring
(Übertrags) Kerbe = (turnover) notch
Walzensatz = set of rotors
Walzenlage = wheel order
Ringstellung = ring setting
Maschinenschlüssel = machine settings, (monthly) key list
Spruchschlüssel = message key

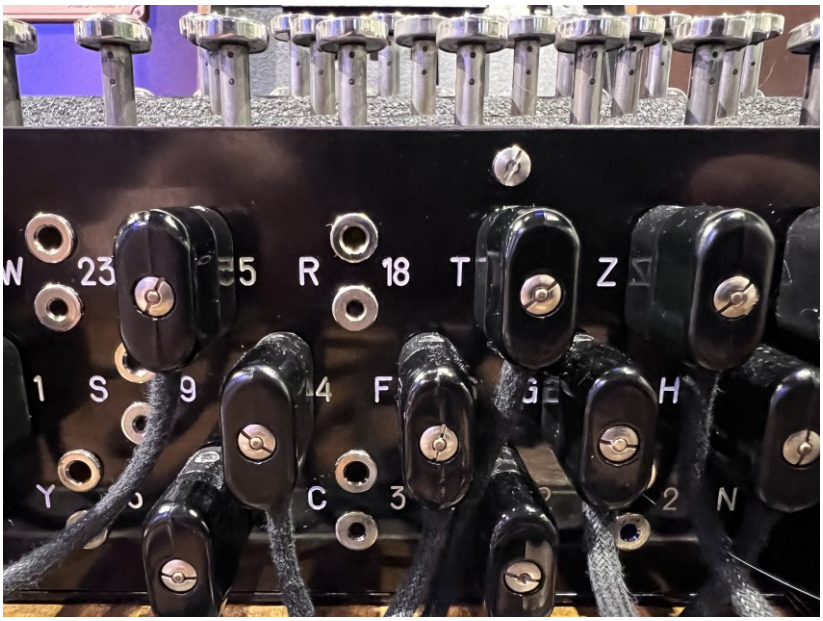


Illustration

- Reflector B
- 3 rotors out of 5

Illustration

- Set of rotors (2 remaining in the box out of 5)



Illustration

- Plugboard

Up to 10 cable pairs swap 2 letters with each other, changed daily.

Possible configurations

60 * 676 * 16.900 * 150.738.274.937.250
= 103.325.660.891.587.134.000.000
= **76 bits** (key length)[1]

[1] For Comparison: Data Encryption Standard (DES): 56 Bit

Howto calculate the "Possible configurations"?

a) Walzenlage / wheel order (selection and placement of rotors, 3 out of 5)
   5*4*3 = 60

b) Ringstellung / ring settings
   26^2 = 676

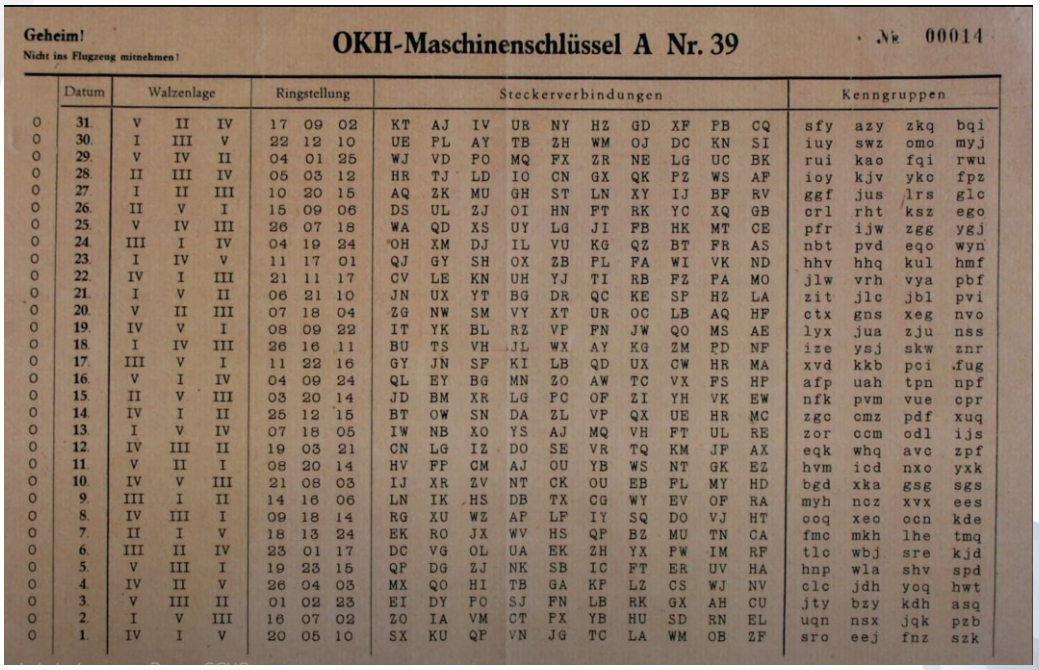
c) Walzenstellung / wheel settings
   In theory this value is: 26*26*26=26^3 = 17.576
   However, due to an anomaly [1] of the Enigma, the value is only:
   26*25*25 = 16.900

d) Steckerverbindungen mit 10 Paaren / plugboard pairing with 10 pairs
   150.738.274.937.250

Calculation of the plugboard pairing options with 10 pairs (d):

```
[bigint] (26*25/2)*(24*23/2)*(22*21/2)*(20*19/2)*(18*17/2)*
         (16*15/2)*(14*13/2)*(12*11/2)*(10*9/2)*(8*7/2) / 3628800
```

[1] Double stepping of the middle rotor
https://www.cryptomuseum.com/crypto/enigma/working.htm#double



Geheim!
Nicht ins Flugzeug mitnehmen!

OKH-Maschinenschlüssel A Nr. 39

№ 00014

	Datum	Walzenlage			Ringstellung			Steckerverbindungen										Kenngruppen			
0	31.	V	II	IV	17	09	02	KT	AJ	IV	UR	NY	HZ	GD	XF	PB	CQ	sfy	azy	zkq	bqi
0	30.	I	III	V	22	12	10	UE	PL	AY	TB	ZH	WM	OJ	DC	KN	SI	iuy	swz	omo	myj
0	29.	V	IV	II	04	01	25	WJ	VD	PO	MQ	FX	ZR	NE	LG	UC	BK	rui	kao	fqi	rwu
0	28.	II	III	IV	05	03	12	HR	TJ	LD	IO	CN	GX	QK	PZ	WS	AF	ioy	kjv	ykc	fpz
0	27.	I	II	III	10	20	15	AQ	ZK	MU	GH	ST	LN	XY	IJ	BF	RV	ggf	jus	lrs	glc
0	26.	II	V	I	15	09	06	DS	UL	ZJ	OI	HN	FT	RK	YC	XQ	GB	crl	rht	ksz	ego
0	25.	V	IV	III	26	07	18	WA	QD	XS	UY	LG	JI	FB	HK	MT	CE	pfr	ijw	zgg	ygj
0	24.	III	I	IV	04	19	24	OH	XM	DJ	IL	VU	KG	QZ	BT	FR	AS	nbt	pvd	eqo	wyn
0	23.	I	IV	V	11	17	01	QJ	GY	SH	OX	ZB	PL	FA	WI	VK	ND	hhv	hhq	kul	hmf
0	22.	IV	I	III	21	11	17	CV	LE	KN	UH	YJ	TI	RB	FZ	PA	MO	jlw	vrh	vya	pbf
0	21.	I	V	II	06	21	10	JN	UX	YT	BG	DR	QC	KE	SF	HZ	LA	zit	jlc	jbl	pvi
0	20.	V	II	III	07	18	04	ZG	NW	SM	VY	XT	UR	OC	LB	AQ	HF	ctx	gns	xeg	nvo
0	19.	IV	V	I	08	09	22	IT	YK	BL	RZ	VP	FN	JW	QO	MS	AE	lyx	jua	zju	nss
0	18.	I	IV	III	26	16	11	BU	TS	VH	JL	WX	AY	KG	ZM	PD	NF	ize	ysj	skw	znr
0	17.	III	V	I	11	22	16	GY	JN	SF	KI	LB	QD	UX	CW	HR	MA	xvd	kkb	pci	fug
0	16.	V	I	IV	04	09	24	QL	EY	BG	MN	ZO	AW	TC	VX	FS	HP	afp	uah	tpn	npf
0	15.	II	V	III	03	20	14	JD	BM	XR	LG	PC	OF	ZI	YH	VK	EW	nfk	pvm	vue	cpr
0	14.	IV	I	II	25	12	15	BT	OW	SN	DA	ZL	VP	QX	UE	HR	MC	zgc	cmz	pdf	xuq
0	13.	I	V	IV	07	18	05	IW	NB	XO	YS	AJ	MQ	VH	FT	UL	RE	zor	ccm	odl	ijs
0	12.	IV	III	II	19	03	21	CN	LG	IZ	DO	SE	VR	TQ	KM	JF	AX	eqk	whq	avc	zpf
0	11.	V	II	I	08	20	14	HV	FP	CM	AJ	OU	YB	WS	NT	GK	EZ	hvm	icd	nxo	yxk
0	10.	IV	V	III	21	08	03	IJ	XR	ZV	NT	CK	OU	EB	FL	MY	HD	bgd	xka	gsg	sgs
0	9.	III	I	II	14	16	06	LN	IK	HS	DB	TX	CG	WY	EV	OF	RA	myh	ncz	xvx	ees
0	8.	IV	III	I	09	18	14	RG	XU	WZ	AP	LF	IY	SQ	DO	VJ	HT	ooq	xeo	ocn	kde
0	7.	II	I	V	18	13	24	EK	RO	JX	WV	HS	QP	BZ	MU	TN	CA	fmc	mkh	lhe	tmq
0	6.	III	II	IV	23	01	17	DC	VG	OL	UA	EK	ZH	YX	FW	IM	RF	tlc	wbj	sre	kjd
0	5.	V	III	I	19	23	15	QP	DG	ZJ	NK	SB	IC	FT	ER	UV	HA	hnp	wla	shv	spd
0	4.	IV	II	V	26	04	03	MX	QO	HI	TB	GA	KP	LZ	CS	WJ	NV	clc	jdh	yoq	hwt
0	3.	V	III	II	01	02	23	EI	DY	PO	SJ	FN	LB	RK	GX	AH	CU	jty	bzy	kdh	asq
0	2.	I	V	III	16	07	02	ZO	IA	VM	CT	PX	YB	HU	SD	RN	EL	uqn	nsx	jqk	pzb
0	1.	IV	I	V	20	05	10	SX	KU	QP	VN	JG	TC	LA	WM	OB	ZF	sro	eej	fnz	szk

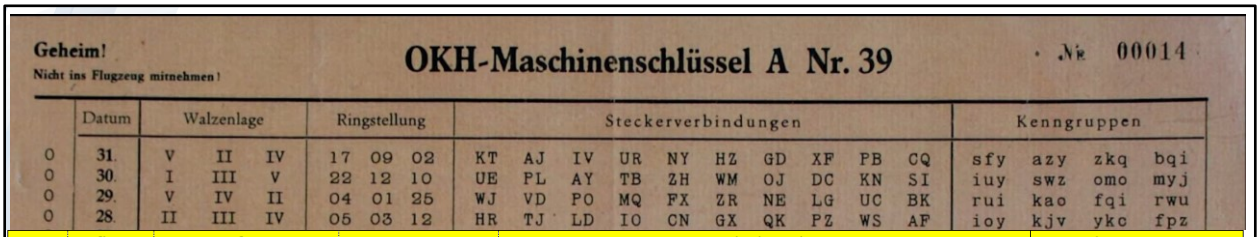
Enigma machine (WP)

"The security of the system depends on machine settings that were generally changed daily, based on secret key lists distributed in advance, and on other settings that were changed for each message. The receiving station would have to know and use the exact settings employed by the transmitting station to successfully decrypt a message."

Reference: https://en.wikipedia.org/wiki/Enigma_machine

Screenshot Maschinenschlüssel

The Enigma Machine - Bletchley Park takes a closer look at how it works
https://youtu.be/3Ux03qPgYVY



Geheim!
Nicht ins Flugzeug mitnehmen!

OKH-Maschinenschlüssel A Nr. 39

№ 00014

	Datum	Walzenlage			Ringstellung			Steckerverbindungen										Kenngruppen			
0	31.	V	II	IV	17	09	02	KT	AJ	IV	UR	NY	HZ	GD	XF	PB	CQ	sfy	azy	zkq	bqi
0	30.	I	III	V	22	12	10	UE	PL	AY	TB	ZH	WM	OJ	DC	KN	SI	iuy	swz	omo	myj
0	29.	V	IV	II	04	01	25	WJ	VD	PO	MQ	FX	ZR	NE	LG	UC	BK	rui	kao	fqi	rwu
0	28.	II	III	IV	05	03	12	HR	TJ	LD	IO	CN	GX	QK	PZ	WS	AF	ioy	kjv	yko	fpz

Date	Reflector	Set of rotors			Ring setting			Plugboard										Indicator groups			
Datum	UKW	Walzenlage			Ringstellung			Steckbrett										Kenngruppen			
31	B	V	II	IV	17	09	02	KT	AJ	IV	UR	NY	HZ	GD	XF	PB	CQ	afy	azy	zkq	bqi
30	B	I	III	IV	22	12	10	UE	FL	AY	TB	ZH	WM	OJ	DC	KN	SI	iuy	swt	omo	myi
29	B	V	IV	II	04	01	25	WJ	VD	PO	MQ	FX	ZR	NE	LG	UC	BK	rui	kao	fqi	rwu
28	B	II	III	IV	05	03	12	HR	TJ	LD	IO	CN	GX	QK	PZ	WS	AP	ioy	kv	yko	fpz
		left	middle	right																	

Original manual

https://www.cryptomuseum.com/crypto/enigma/files/schluessel_m.pdf

Example Maschinensschlüssel

https://cryptocellar.org/enigma/e-keys/luftwaffen-mschluessel-nr619.pdf

Demo B

Sample code

```
## Rotors and wiring (Walzen)
## https://en.wikipedia.org/wiki/Enigma_rotor_details
## https://de.wikipedia.org/wiki/Enigma-Rotors

[string[]] $rotors_r = # Wiring schema: rotors (Walzen) right side
'EKMFLGDQVZNTOWYHXUSPAIBRCJ', # I    (Enigma 1, 1930)
'AJDKSIRUXBLHWTMCQGZNPYFVOE', # II   (Enigma 1, 1930)
'BDFHJLCPRTXVZNYEIWGAKMUSQO', # III  (Enigma 1, 1930)
'ESOVPZJAYQUIRHXLNFTGKDCMWB', # IV   (Enigma M3/Heer, 1938)
'VZBRGITYUPSDNHLXAWMJQOFECK', # V    (Enigma M3/Heer, 1938)

## Reflectors (Umkehrwalzen)
[string[]] $reflectors =
'EJMZALYXVBWFCRQUONTSPIKHGD', # Reflector A
'YRUHQSLDPXNGOKMIEBFZCWVJAT', # Reflector B
'FVPJIAOYEDRZXWGCTKUQSBNMHL'  # Reflector C

## Turnover notch positions (Übertragskerben)
$notchPositions = "Q E V J Z ZM ZM ZM"
```

Operation Barbarossa, 1941

```
$cipherText = @'
SFBWD NJUSE GQOBH KRTAR EEZMW
KPPRB XOHDR OEQGB BGTQV PGVKB
VVGBI MHUSZ YDAJQ IROAX SSSNR
EHYGG RPISE ZBOVM QIEMM ZCYSG
QDGRE RVBIL EKXYQ IRGIR QNRDN
VRXCY YTNJR
'@
```

```
DREIG EHTLA NGSAM ABERS IQERV ORWAE
RTSXE INSSI EBENN ULLSE QSXUH RXROE
MXEIN SXINF RGTXD REIXA UFFLI EGERS
TRASZ EMITA NFANG XEINS SEQSX KMXKM
XOSTW XKAME NECXK
```

A real enigma message from WW II

- https://www.sarcnet.org/the-enigma-project.html
- http://wiki.franklinheath.co.uk/index.php/Enigma/Sample_Messages
- http://wiki.franklinheath.co.uk/index.php/Enigma/Sample_Decrypts

```
$testEnigma = [Enigma]::new()
$testEnigma.setup(2, (2,4,5), 'LSD', (2,21,12), 'AV BS CG DL FU HZ
IN KM OW RX')
$cipherText = @'
SFBWD NJUSE GQOBH KRTAR EEZMW
KPPRB XOHDR OEQGB BGTQV PGVKB
VVGBI MHUSZ YDAJQ IROAX SSSNR
EHYGG RPISE ZBOVM QIEMM ZCYSG
QDGRE RVBIL EKXYQ IRGIR QNRDN
VRXCY YTNJR
'@
$rawResult = translate -text $cipherText -e $testEnigma
$rawResult | groupify
```

Operation Barbarossa, 1941

```
$cipherText = @'
SFBWD NJUSE GQOBH KRTAR EEZMW
KPPRB XOHDR OEQGB BGTQV PGVKB
VVGBI MHUSZ YDAJQ IROAX SSSNR
EHYGG RPISE ZBOVM QIEMM ZCYSG
QDGRE RVBIL EKXYQ IRGIR QNRDN
VRXCY YTNJR
'@
```

```
DREIG EHTLA NGSAM ABERS IQERV ORWAE
RTSXE INSSI EBENN ULLSE QSXUH RXROE
MXEIN SXINF RGTXD REIXA UFFLI EGERS
TRASZ EMITA NFANG XEINS SEQSX KMXKM
XOSTW XKAME NECXK
```

During World War II the Enigma operators replaced white spaces with an X and CK and CH with a Q. Generally all messages were written down in chunks of five characters. A message was limited to 250 characters.

The Bletchley Park translated Enigma Instruction Manual
https://www.codesandciphers.org.uk/documents/egenproc/egenproc.pdf

```
AUFKL XABTE ILUNG XVONX KURTI NOWAX   # PART 1
KURTI NOWAX NORDW ESTLX SEBEZ XSEBE
ZXUAF FLIEG ERSTR ASZER IQTUN GXDUB
ROWKI XDUBR OWKIX OPOTS CHKAX OPOTS
CHKAX UMXEI NSAQT DREIN ULLXU HRANG
ETRET ENXAN GRIFF XINFX RGTX-

DREIG EHTLA NGSAM ABERS IQERV ORWAE   # PART 2
RTSXE INSSI EBENN ULLSE QSXUH RXROE
MXEIN SXINF RGTXD REIXA UFFLI EGERS
TRASZ EMITA NFANG XEINS SEQSX KMXKM
XOS2W XKAME NECXK
```

This real two-part message was sent on July 7th, 1941 from the Russian front and intercepted.

Reference: http://wiki.franklinheath.co.uk/index.php/Enigma/Sample_Messages

```
AUFKL ABTEILUNG VON KURTINOWA          # PART 1
KURTINOWA NORDWESTL SEBEZ SEBEZ
UAF FLIEGERSTR ASZERICHTUNG DUBROWKI
DUBROWKI OPOTSCHKA OPOTSCHKA UM
EINSACHTDREINULL UHRANGETRETEN
ANGRIFF INF RGT-

DREIGEHTLANGSAM ABERSICHERVORWAERTS    # PART 2
EINSSIEBENNULLSECHS UHR ROEM EINS
INFRGT DREI AUFFLIEGERSTRASZEMITANFANG
EINS SECHS KM KM OS2WKAMENEC K
```

The operators were instructed to **repeat** important elements, such as city names or locations: Kurtinowa, Sebez, Dubrowski, Opotschka

Numbers were written out. You can find these numbers above.

```
# A (PART 1)
EINSACHTDREINULL
EINS ACHT DREI NULL
1    8    3    0

# B (PART 2)
EINSSIEBENNULLSECHS
EINS SIEBEN NULL SECHS
1    7      0    6
```

Aufklärung Abteilung von Kurtinowa nordwestlich Sebez
[auf] Fliegerstraße in Richtung Dubrowki, Opotschka.
Um 18:30 Uhr angetreten Angriff.
Infanterie Regiment 3 geht langsam aber sicher vorwärts.
17:06 Uhr röm eins Infanterie-Regiment 3 auf
Fliegerstraße mit Anfang 16km ostwärts Kamenec.

Reconnaissance division from Kurtinowa north-west of Sebezh
on the flight corridor towards Dubrowki, Opochka.
Attack begun at 18:30 hours.
Infantry Regiment 3 goes slowly but surely forwards.
17:06 hours roman numeral one Infantry Regiment 3 on the
flight corridor starting 16 km east of Kamenec.

Enigma/Sample Decrypts

http://wiki.franklinheath.co.uk/index.php/Enigma/Sample_Decrypts

Bletchley Park

"**Bletchley Park**, British government cryptological establishment in operation during World War II. Bletchley Park was where Alan Turing and other agents of the Ultra intelligence project decoded the enemy's secret messages, most notably those that had been encrypted with the German Enigma and Tunny cipher machines. Experts have suggested that the Bletchley Park code breakers may have shortened the war by as much as two year"

Reference: https://www.britannica.com/place/Bletchley-Park

"The Bletchley Park site in Buckinghamshire (now in **Milton Keynes),** England, was about 50 miles (80 km) northwest of London, conveniently located near a railway line that served both Oxford and Cambridge universities. The property consisted of a Victorian manor house and 58 acres (23 hectares) of grounds. The British government acquired it in 1938 and made it a station of the Government Code and Cypher School (GC&CS), designated as Station X. At the start of the war in 1939, the station had only 200 workers, but by late 1944 it had a staff of nearly 9,000, working in three shifts around the clock. Experts at crossword-puzzle solving and chess, as well as mathematicians and scientists, were among those who were hired. About three-fourths of the workers were women."

Reference: https://www.britannica.com/place/Bletchley-Park

The "ultra" secret

1939-1945 Up to 10.000 people work in BP (aka Station X)

1945 End of World War II

1982 "The Hut Six Story" by Gordon Welchman is published

1992 Bletchley Park Trust established

2009 British government acknowledges the contribution of BP

Gordon Welchman

"William Gordon Welchman (15 June 1906 – 8 October 1985) was a British mathematician. During World War II, he worked at Britain's secret decryption centre at Bletchley Park, where he was one of the most important contributors.
After the war he moved to the US and worked on the design of military communications systems.
[..]
In 1982 his book The Hut Six Story was published, initially by McGraw-Hill in the US and by Allen Lane in Britain. The National Security Agency disapproved. The book was not banned, but as a result of it, Welchman lost his American and British security clearances, and therefore his consultancy with Mitre, and was forbidden to discuss either the book or his wartime work."

Gordon Welchman's book about Blechtley Park:
The Hut Six story: Breaking the Enigma codes.
Harmondsworth, England: Penguin Books. ISBN 0-14-00-5305-0.
"An early publication containing several misapprehensions that are corrected in an addendum in the 1997 edition."

Reference: https://en.wikipedia.org/wiki/Gordon_Welchman

Alan Turing's office

1912 Turing was born

1952 Prosecuted for homo-sexual acts, treated with chemical castration

1954 Turing dies

2014 Her Queen's pardon

2017 "Policy and crime act" (aka "The Alan Turing law") Homosexual acts that are no longer criminal offence

"**Alan Mathison Turing** (June 1912 – 7 June 1954) was an English mathematician, computer scientist, logician, cryptanalyst, philosopher, and theoretical biologist. Turing was highly influential in the development of theoretical computer science, providing a formalisation of the concepts of algorithm and computation with the Turing machine, which can be considered a model of a general-purpose computer. He is widely considered to be the father of theoretical computer science and artificial intelligence."

https://en.wikipedia.org/wiki/Alan_Turing

Illustration

The Polish Memorial in Bletchley Park

Cryptanalysis of the Enigma

"Three Polish mathematicians made breakthroughs in the mid-1930s, developing a machine (known as a Bomba) to help break the codes. Much of the early work on breaking Enigma focussed on repetition of the message key (specifically starting positions of the rotors) as well as several key phrases used in messages (known as “cribs”). Vital intelligence was passed to the Polish cryptanalysts and **Marian Rejewski** was able to deduce the internal wiring of the Enigma rotors, meaning the Polish could build a replica Enigma machine. They passed what they had achieved to Bletchley Park just before WWII began, but by this time Germany had upgraded its Enigma usage procedures. It is likely that the Polish codebreakers, after having escaped to Paris, made the first wartime break on 17 January 1940, with Turing present. The first team at Bletchley Park to break into an Enigma encrypted message was Gordon Welchman’s team in Hut 6 , with John Jeffreys overseeing use of the punched sheets utilised for the task."

Reference: https://bletchleypark.org.uk/our-story/alan-turing-faqs/

Read more

https://en.wikipedia.org/wiki/Cryptanalysis_of_the_Enigma
https://en.wikipedia.org/wiki/Marian_Rejewski

"I believe that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted"

Alan Turing, 1947

Reference

https://plato.stanford.edu/entries/turing-test/

PY vs PS

```
## PS
function str2num ([char[]] $text) {
    foreach ($letter in $text) {
        [byte] [char] $letter - 65
    }
}
## PY
def str2num(text):
  return [ord(letter)-65 for
  letter in text]
```

```
## PS
[string[]] $rotors_r =
'EKMFLGDQVZNTOWYHXUSPAIBRCJ', # I
'AJDKSIRUXBLHWTMCQGZNPYFVOE', # II
'BDFHJLCPRTXVZNYEIWGAKMUSQO', # III

[System.Collections.ArrayList] $alRotors_r = @()
foreach ($rotor in $rotors_r) {
  [void] $alRotors_r.Add((str2num $rotor))
}

## PY
rotors_r =
['EKMFLGDQVZNTOWYHXUSPAIBRCJ',  # I
 'AJDKSIRUXBLHWTMCQGZNPYFVOE',  # II
 'BDFHJLCPRTXVZNYEIWGAKMUSQO',  # III
rotors_r =
  [deque(str2num(rotor)) for rotor in rotors_r]
```

Kind regards

There are numerous implementations of the Enigma algorithms. I have primarily oriented myself to this code:

https://github.com/Gravitar64/
A-beautiful-code-in-Python/blob/master/Teil_44_Enigma.py

If you speak German (or just in case you want to learn it), I can highly recommend his YT channel with a big amount of instructional videos about Python.
https://www.youtube.com/@Gravitar/videos

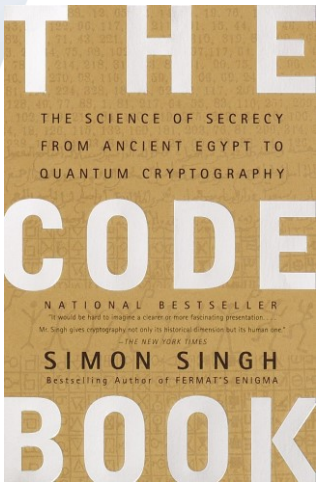
```
{
 "Venue"    : "PSConfEU 2023",
 "Title"    : "PSEnigma",
 "Speaker"  : "Thorsten Butz",
 "Uri"      : "thorsten-butz.de",
 "Twitter"  : "@thorstenbutz",
 "Podcast"  : "slidingwindows.de"
}
```

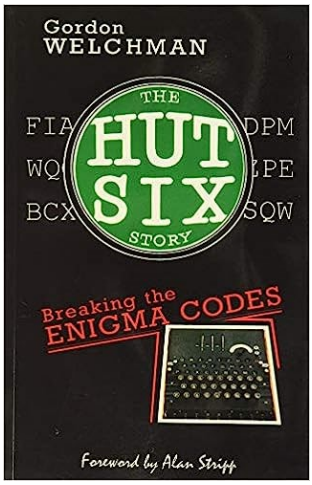
PSEnigma is published on Github (https://github.com/thorstenbutz/PSEnigma) and will also be available on the PowerShell Gallery.

Bletchley Park. once the top-secret home of the World War Two Codebreakers, is now a museum and vibrant heritage attraction open daily, managed by the Bletchley Park Trust. I highly recommend a visit on site, where you will also find the "The National Museum of Computing" (https://www.tnmoc.org).

The tickets are always valid one year, so you might consider a 2-days-trip to Milton Keynes near London (approx. 60 min by train from London Kings Cross).

Books





- Simon Singh: The code book (ISBN 0385495323)
- Gordon Welchman: The Hut Six Story: Breaking the Enigma Codes (ISBN 0-14-00-5305-0)