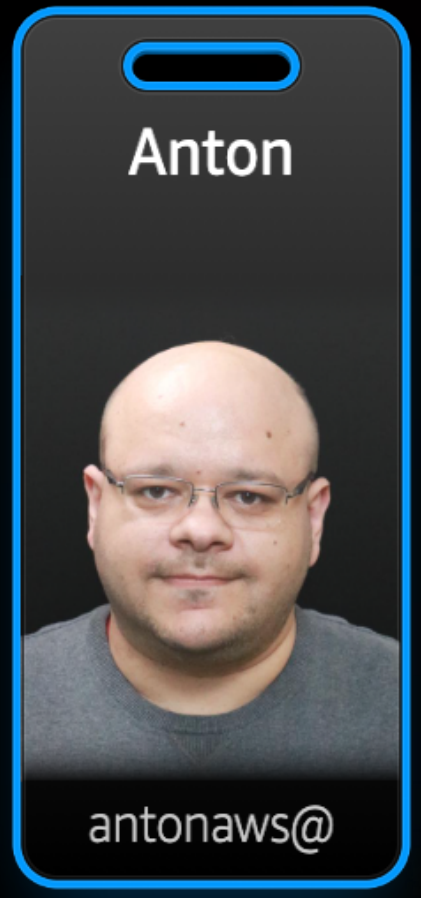
A visual journey to reverse-engineering the Enigma Machine

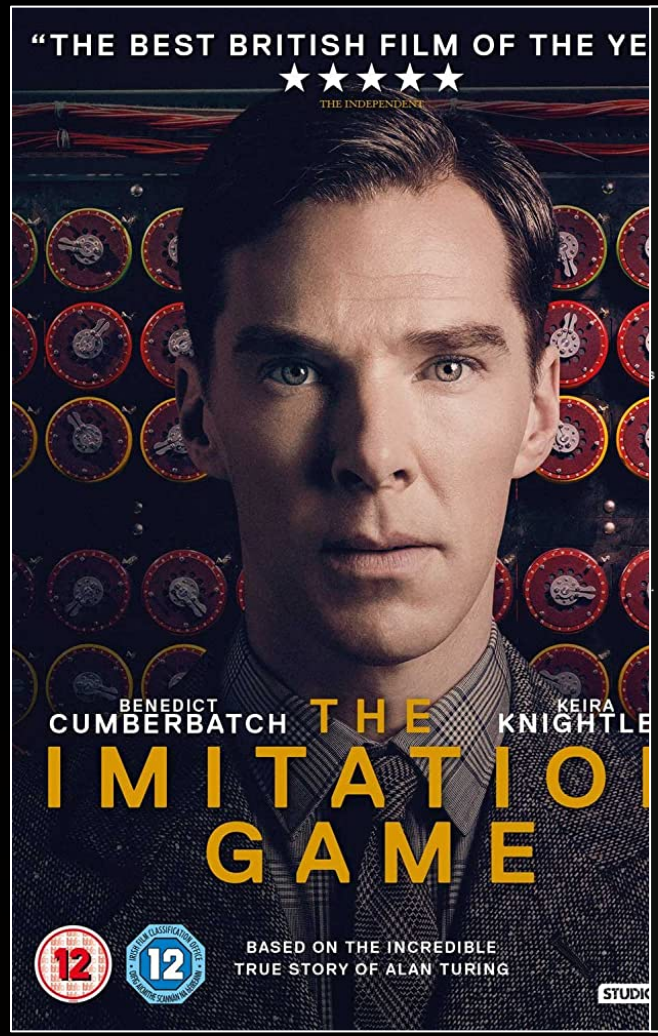
Anton Aleksandrov
Principal Solutions Architect
AWS Serverless

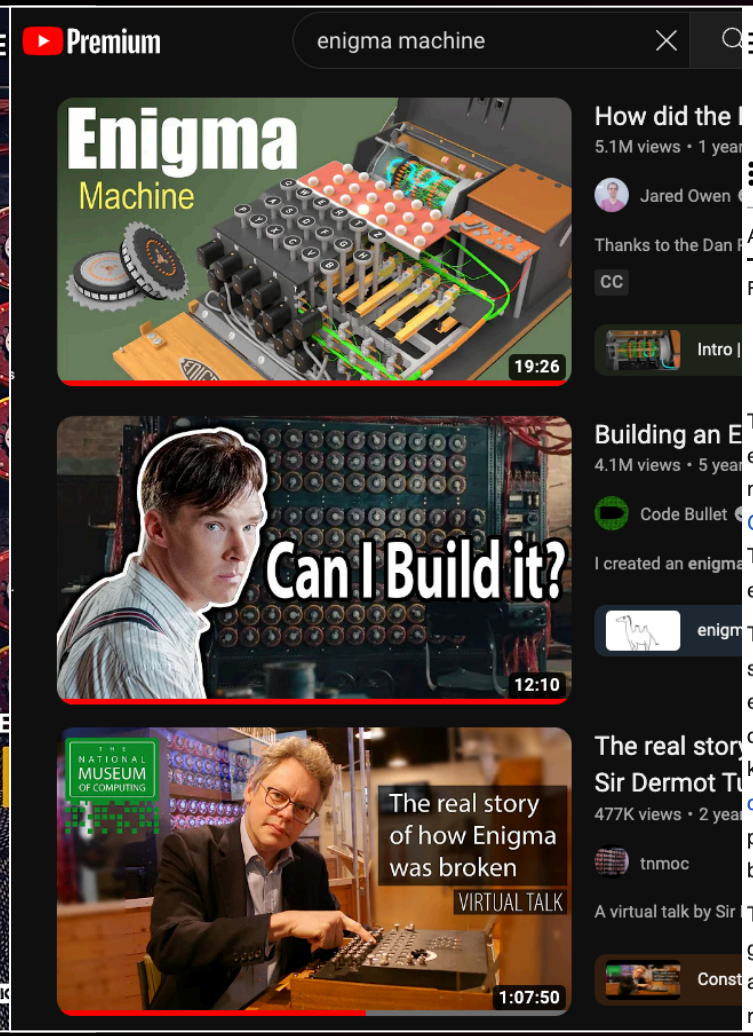
Who am I

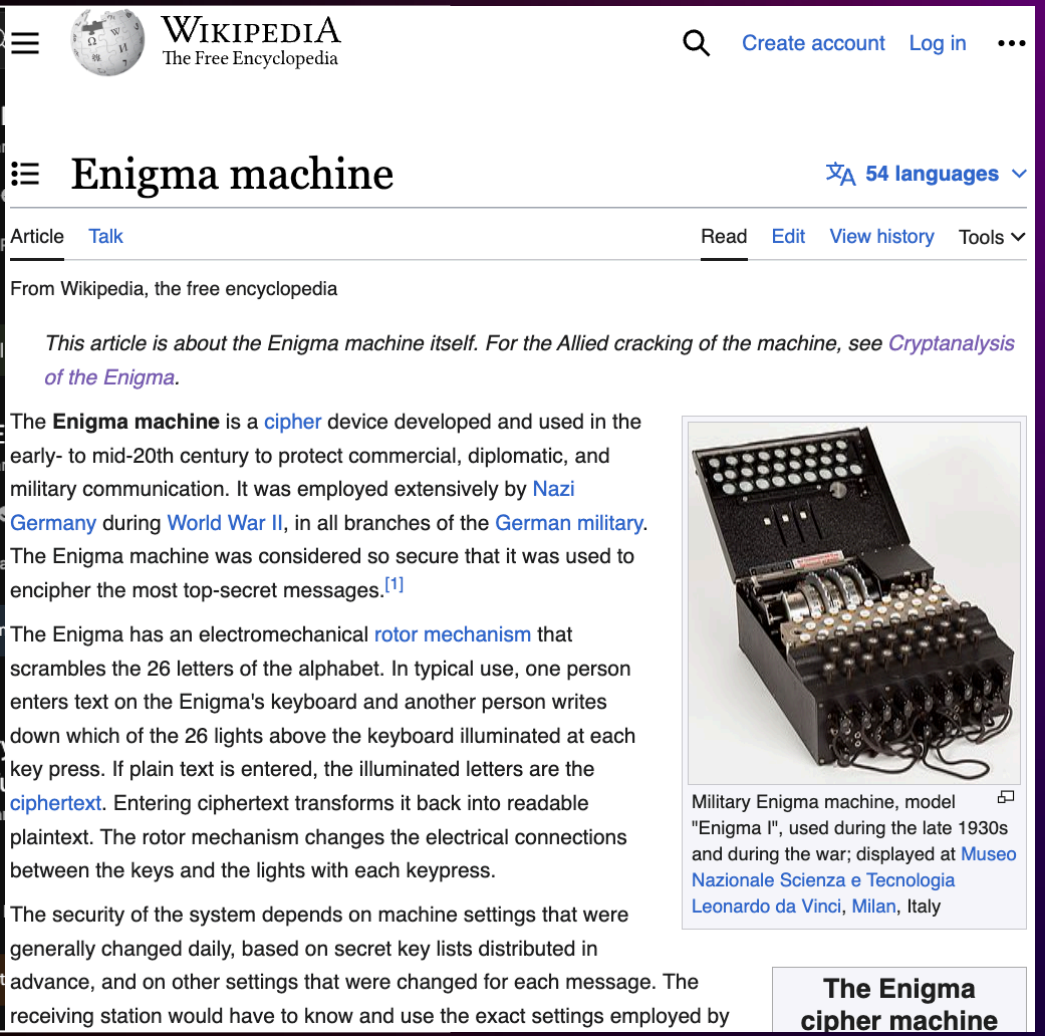


- Principal Solutions Architect, Serverless
- 20+ years of software/cloud architecture
- Ex-Chief Architect of Cloud Security Services
- Security geek
- A husband, a father, a foodie
- Star Wars > Star Trek (sorry and 🖖)
- Basically your classic definition of a nerd

How it all began

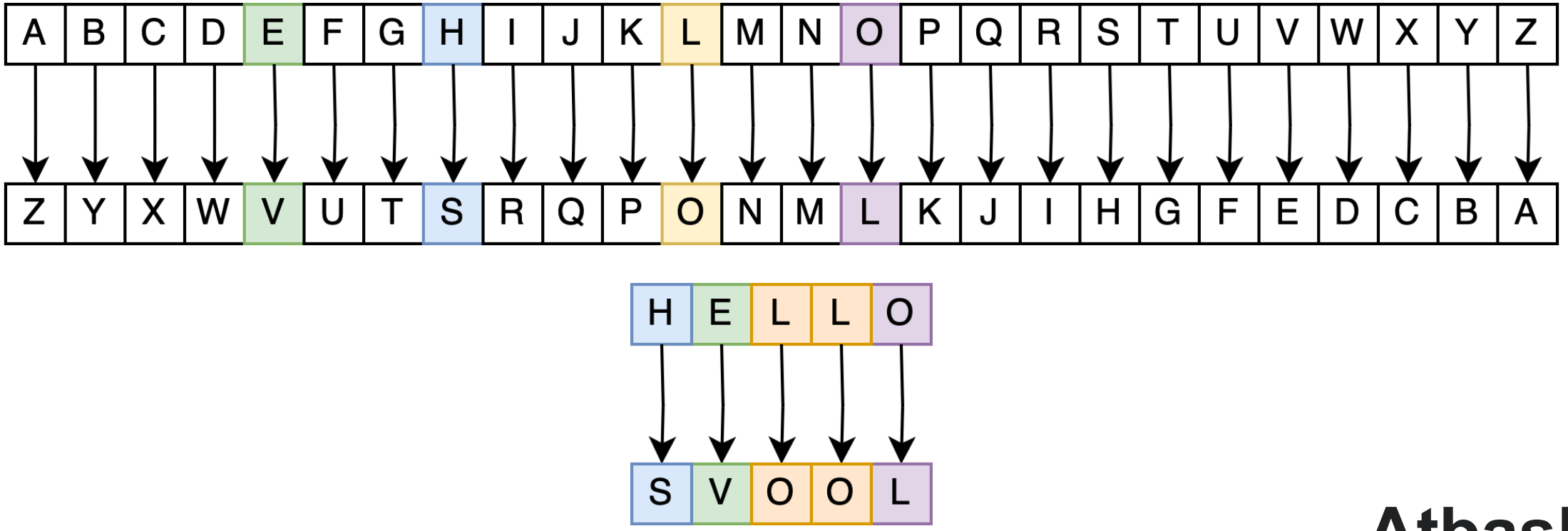






Early ciphers – biblical times

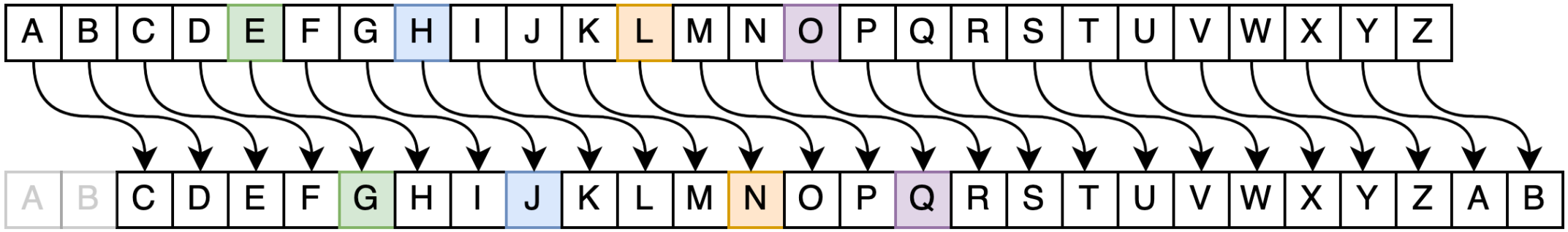
Substitution Cipher

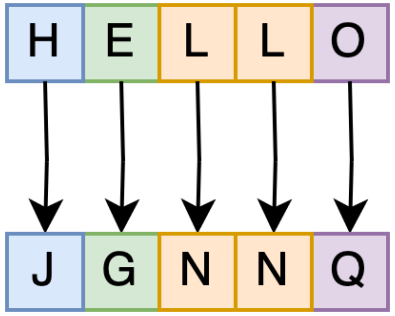


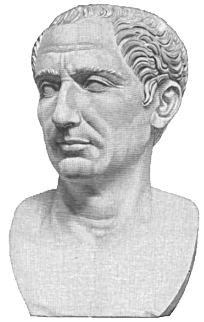
Atbash

Early ciphers – approximately 50 BC

Substitution Cipher

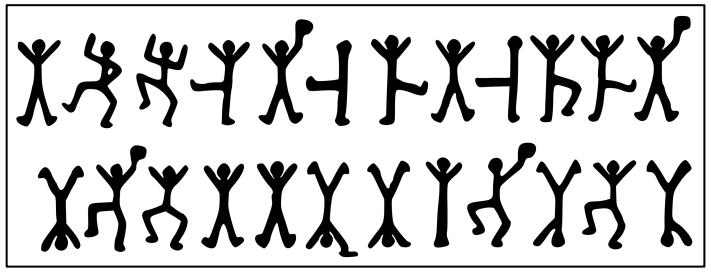




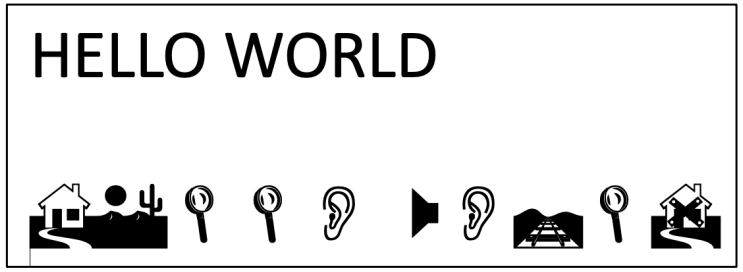


Caesar cipher

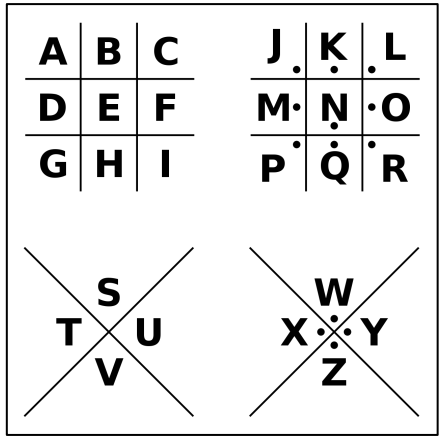
Ciphers, ciphers, ciphers…



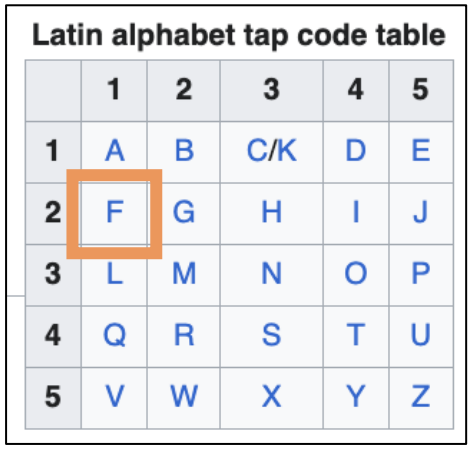
Dancing Men
(Sherlock Holmes)



Webdings font



Pigpen cipher



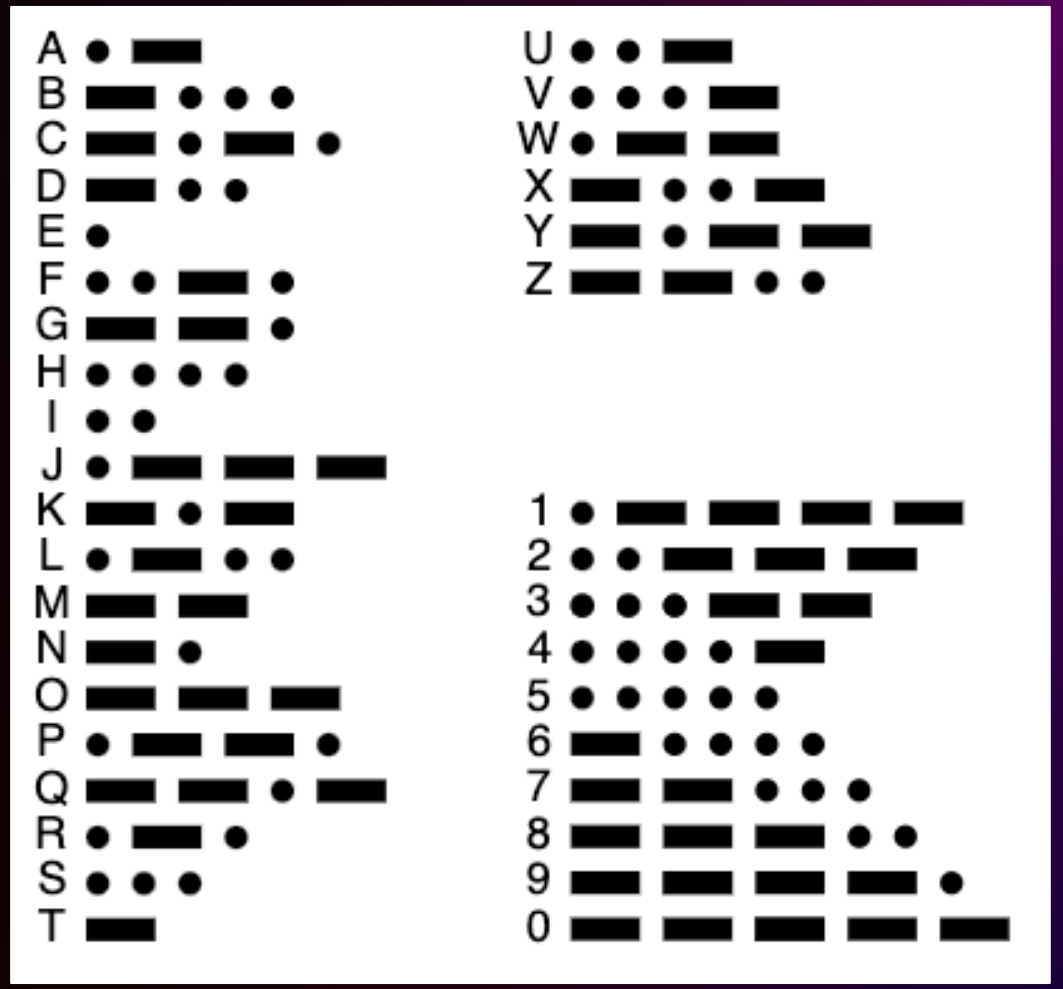
Latin alphabet tap code table

	1	2	3	4	5
1	A	B	C/K	D	E
2	F	G	H	I	J
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Tap cipher

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

Morse Code

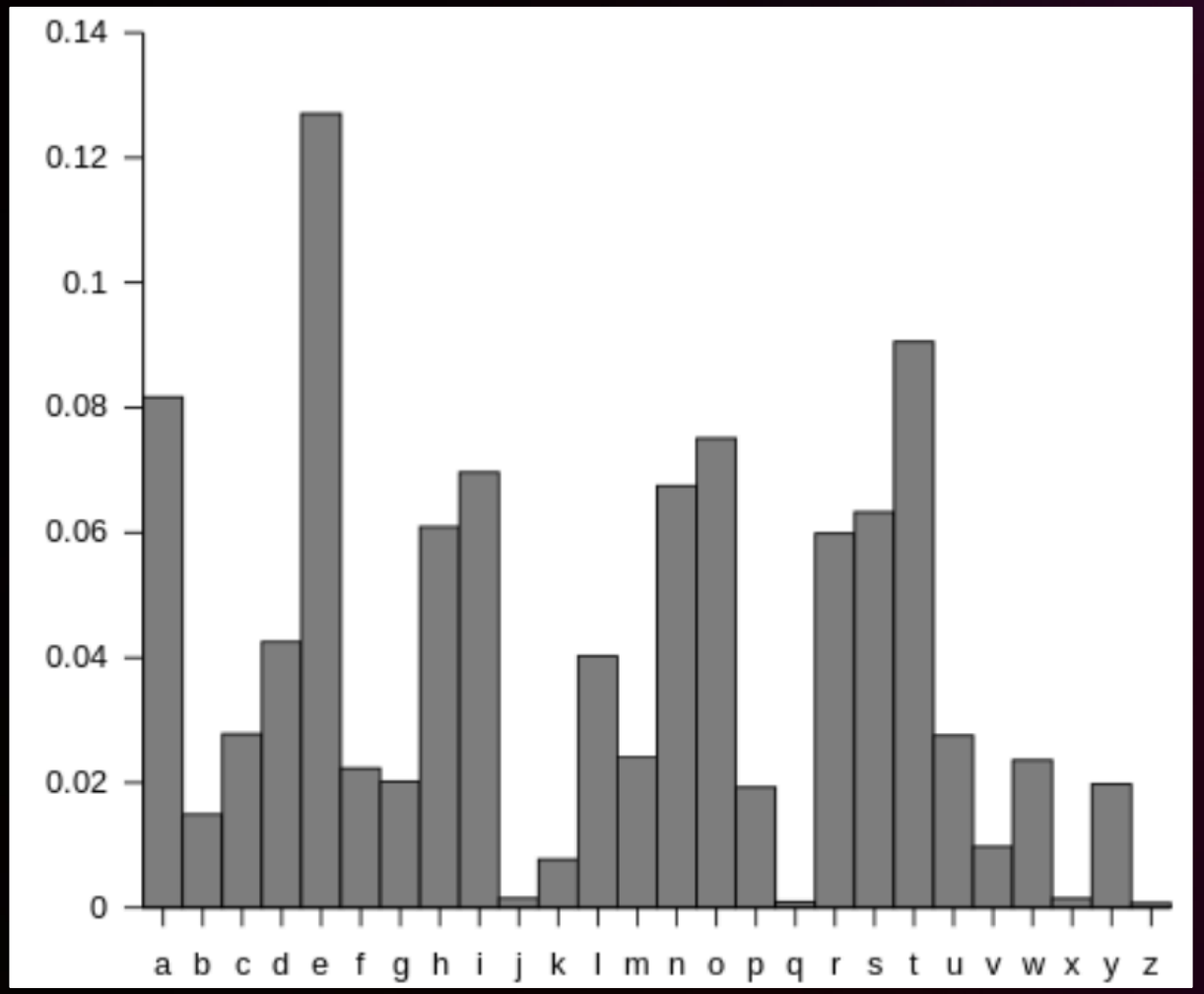


https://en.wikipedia.org/wiki/Samuel_Morse
https://en.wikipedia.org/wiki/Morse_code

The ”Slack” of our great-grandparents

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

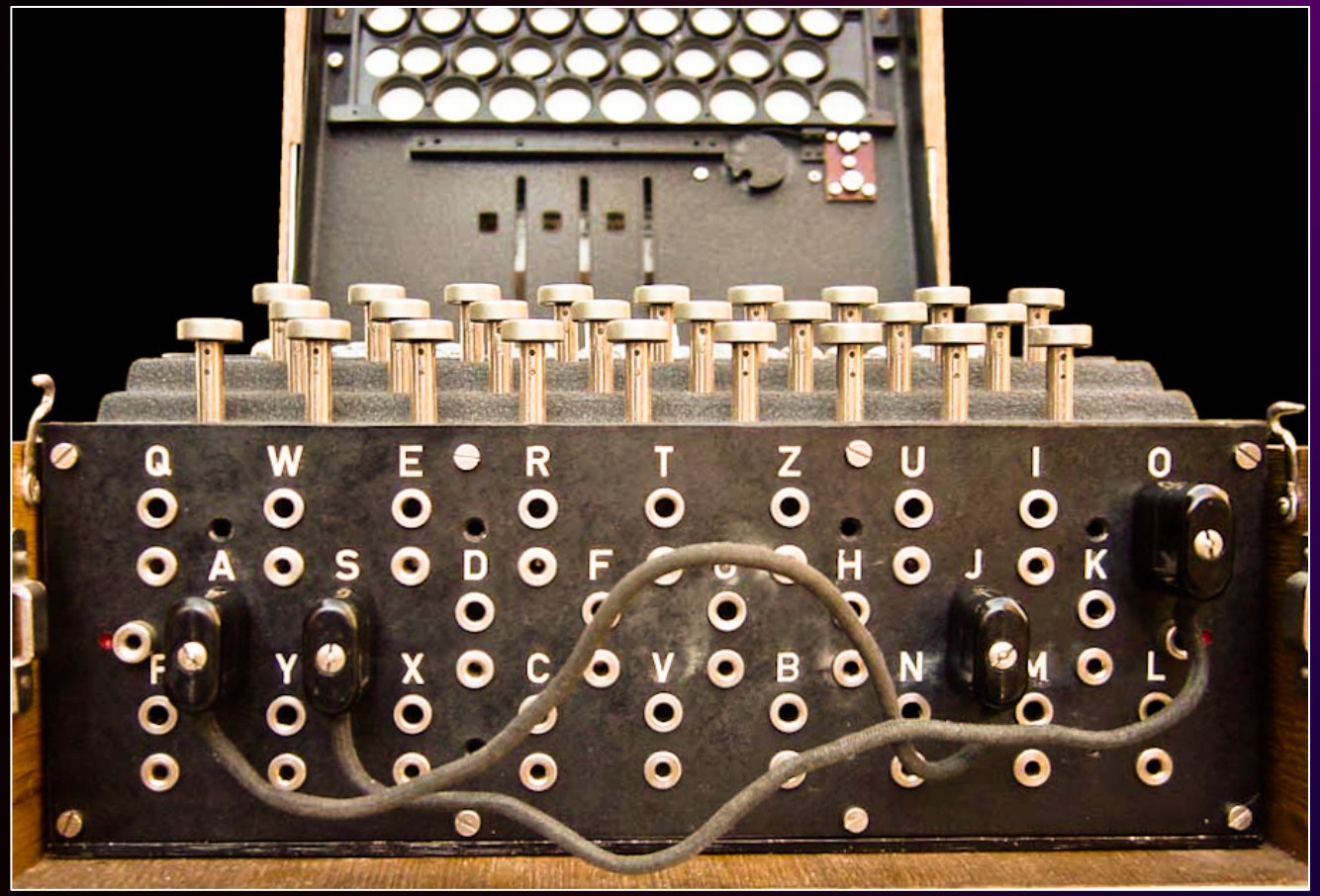
Cracking the substitution cipher



A typical distribution of letters in English language text.

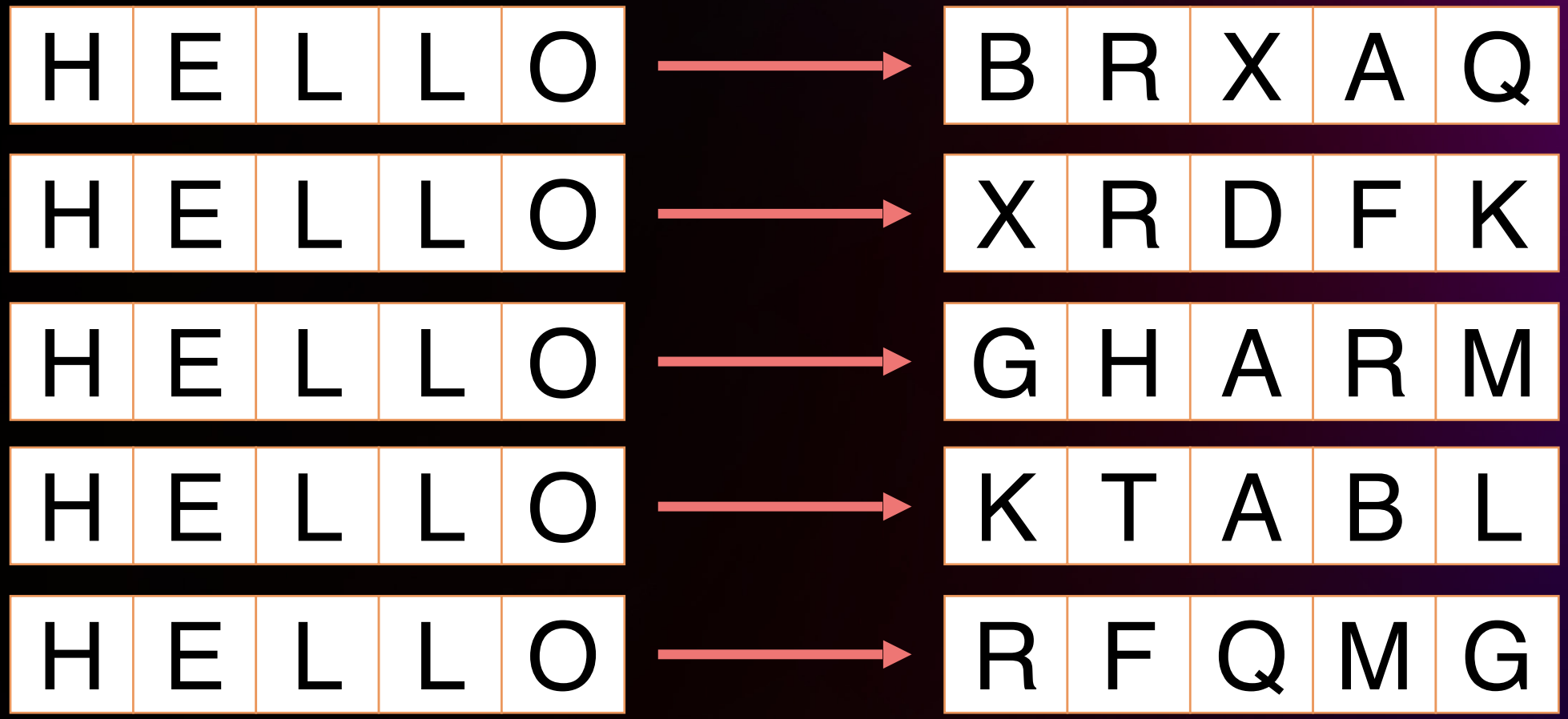
https://en.wikipedia.org/wiki/Frequency_analysis

The Enigma Machine

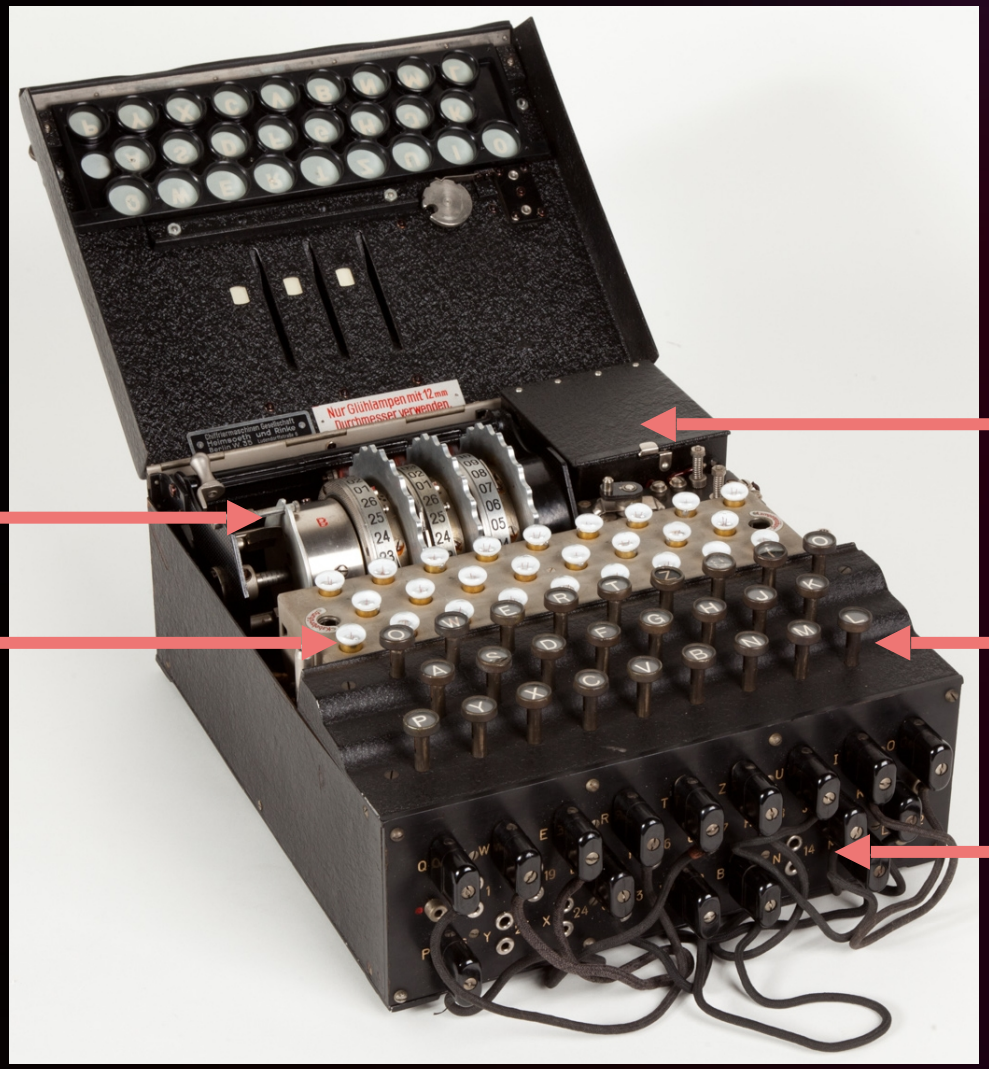


https://en.wikipedia.org/wiki/Enigma_machine

The Enigma Machine



The Enigma Machine



Rotors and reflector

Output lamps

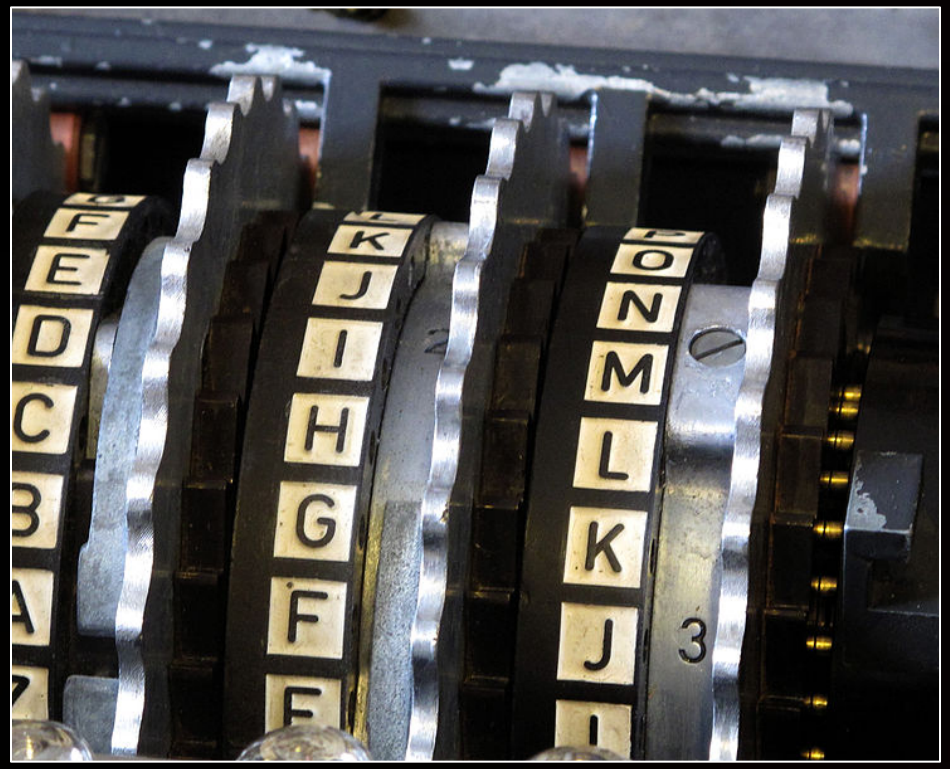
Battery

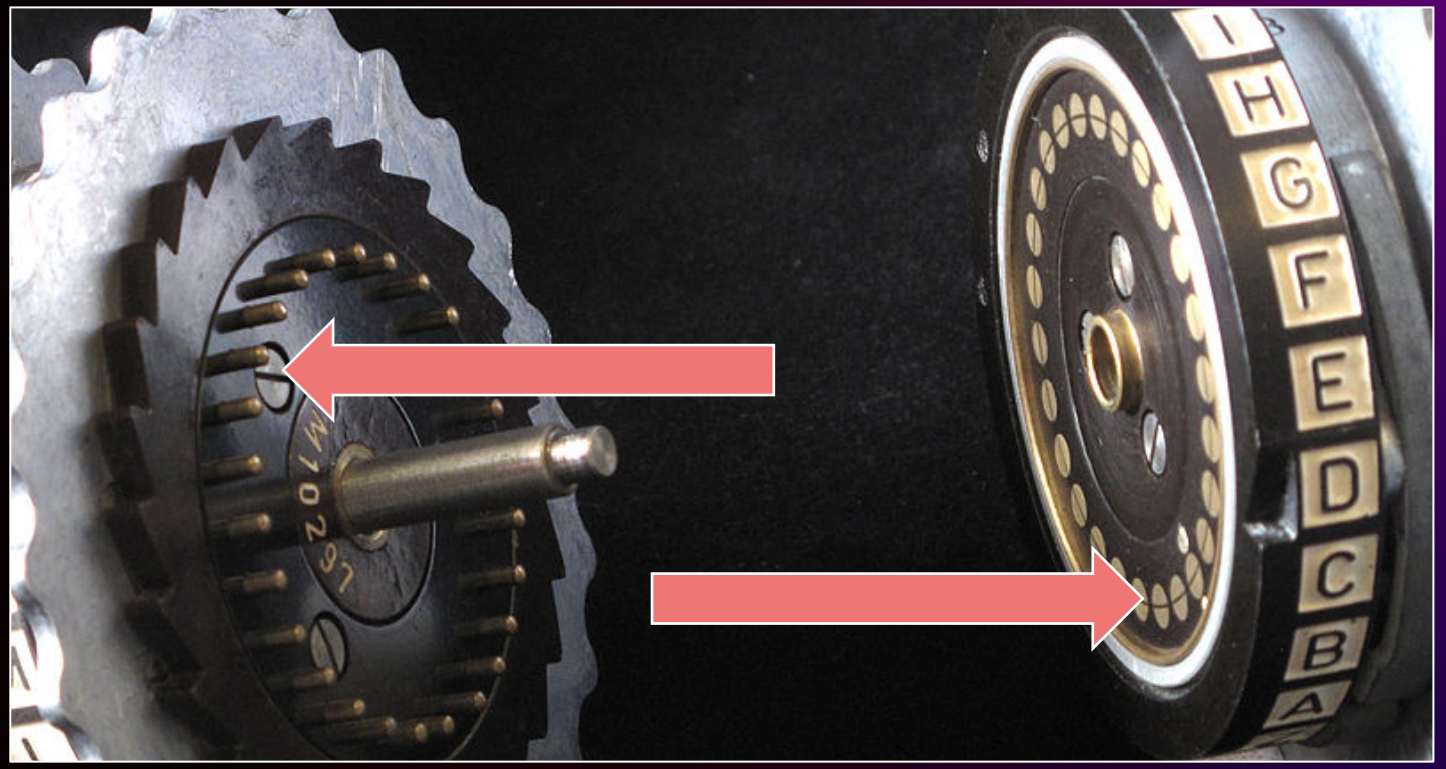
Input keys

Plugboard

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

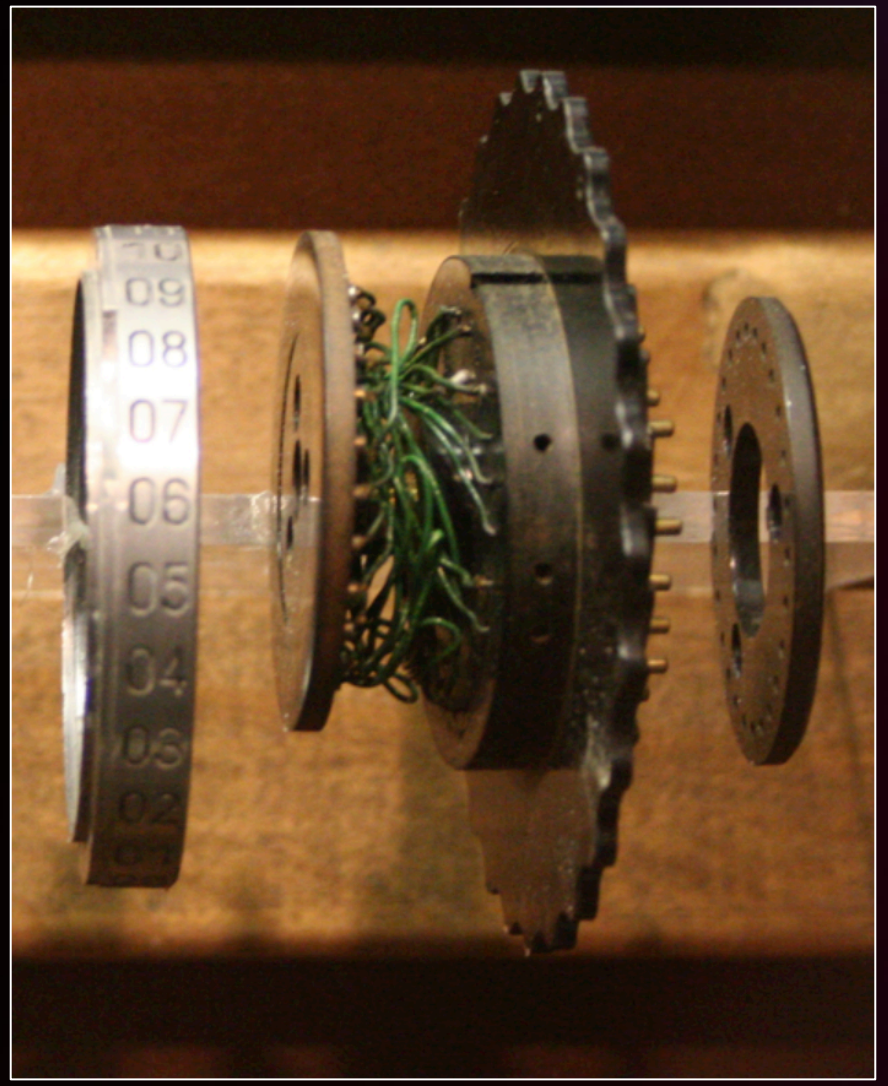
Rotors





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

Rotors

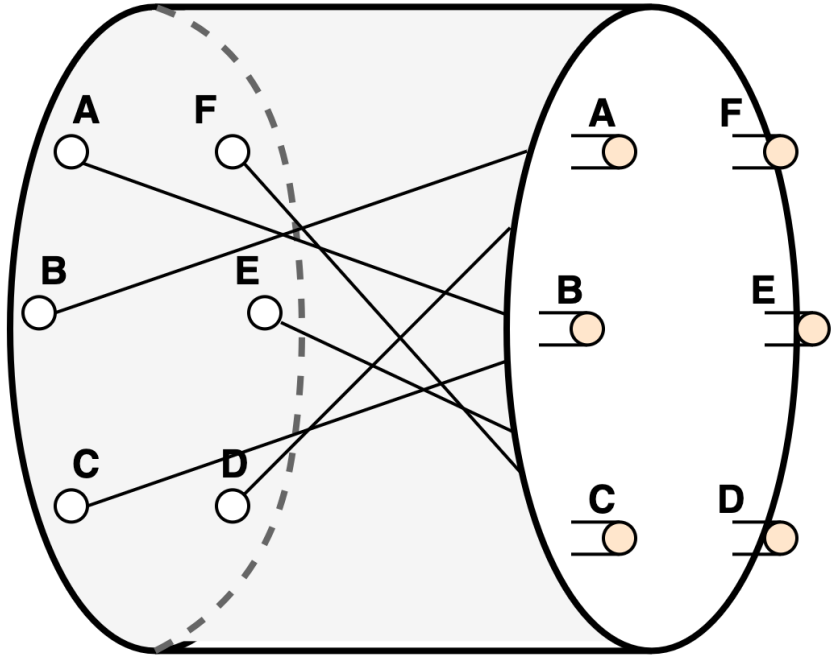


https://en.wikipedia.org/wiki/Enigma_rotor_details

Rotors

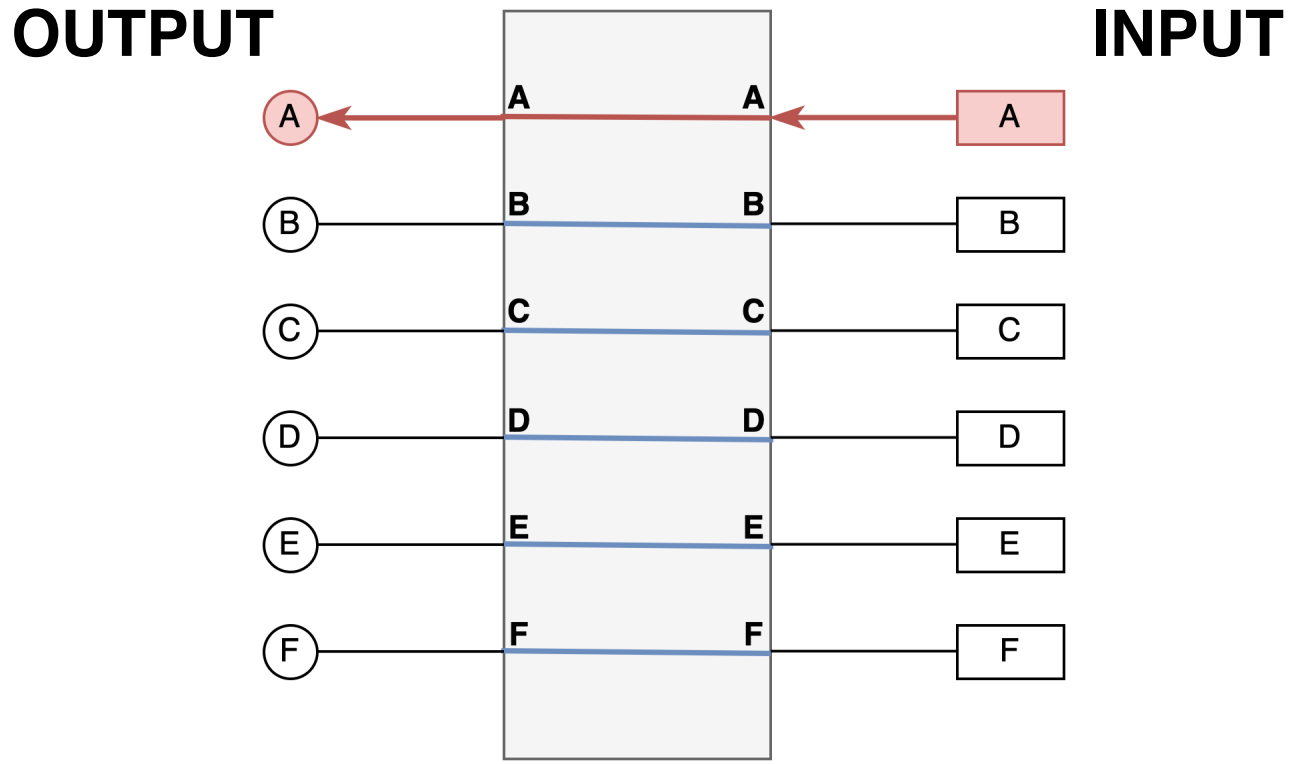
OUTPUT

INPUT

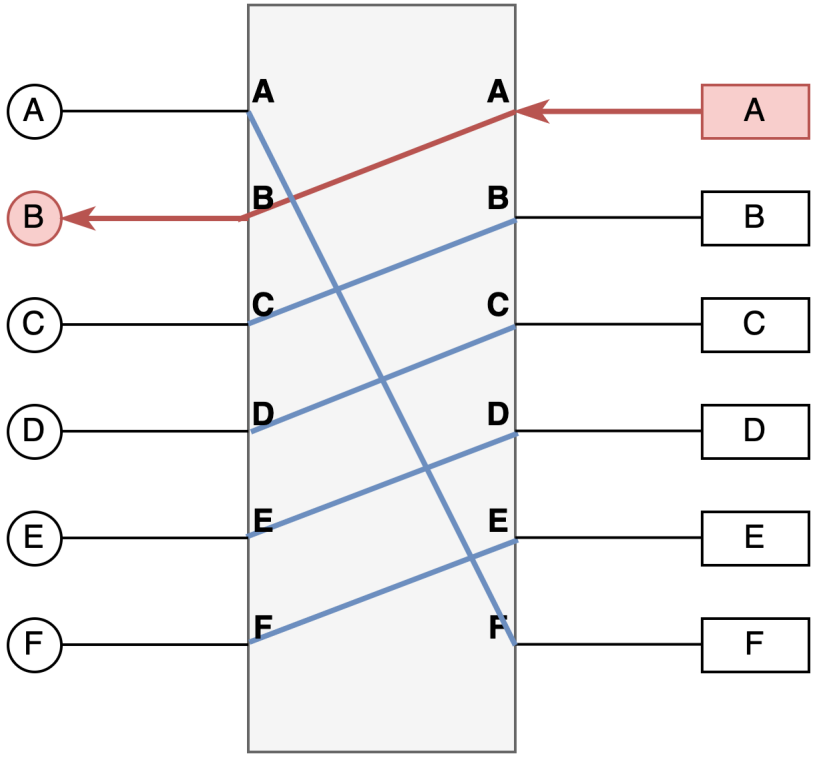


A substitution cipher!

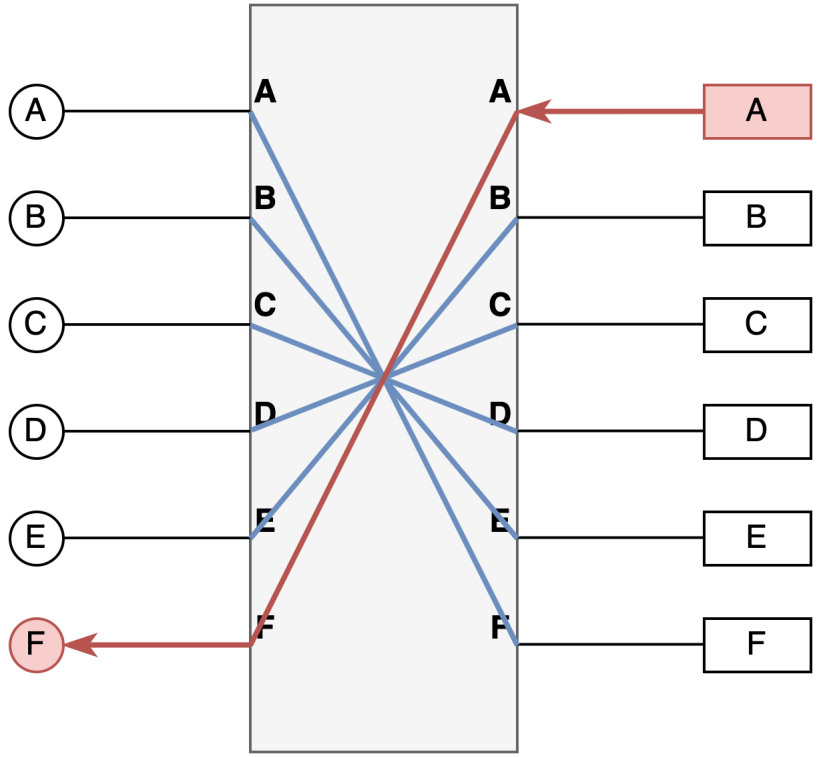
Rotors



Rotors

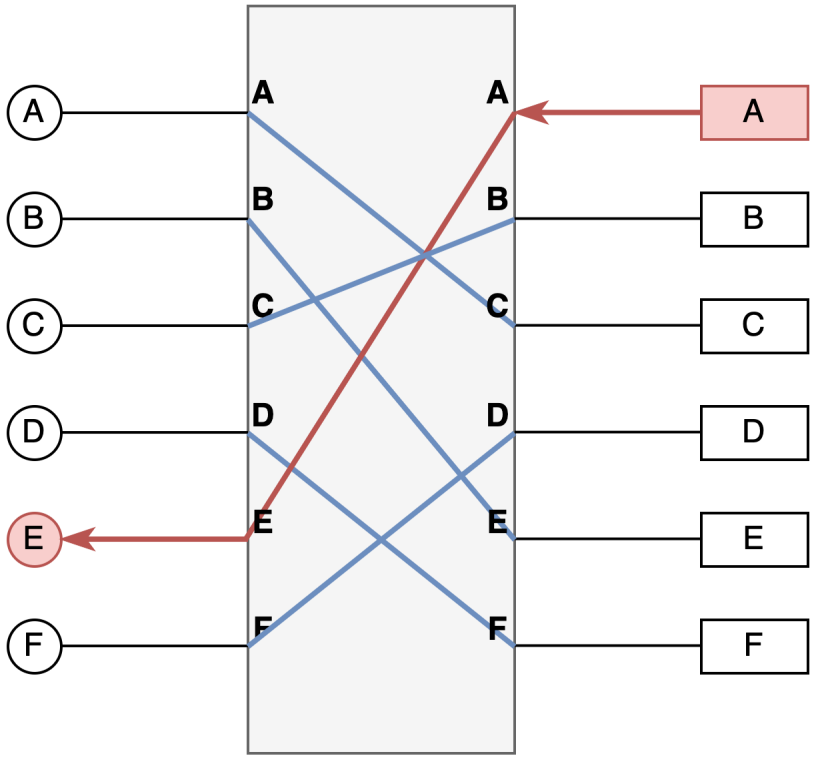


Caesar cipher

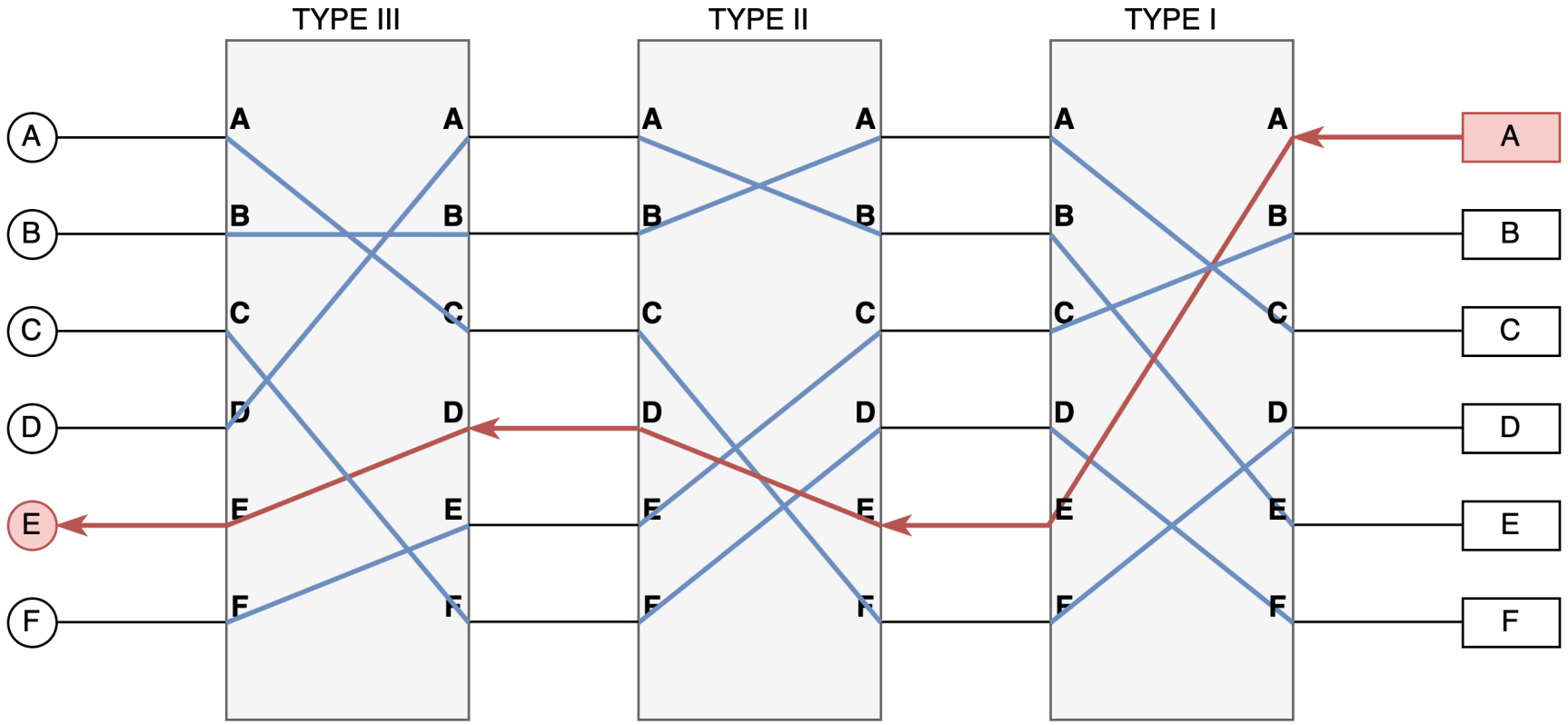


Atbash cipher

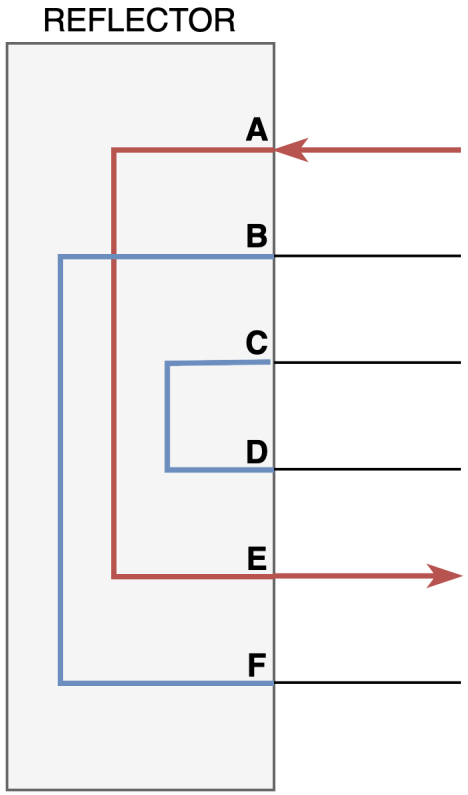
Rotors



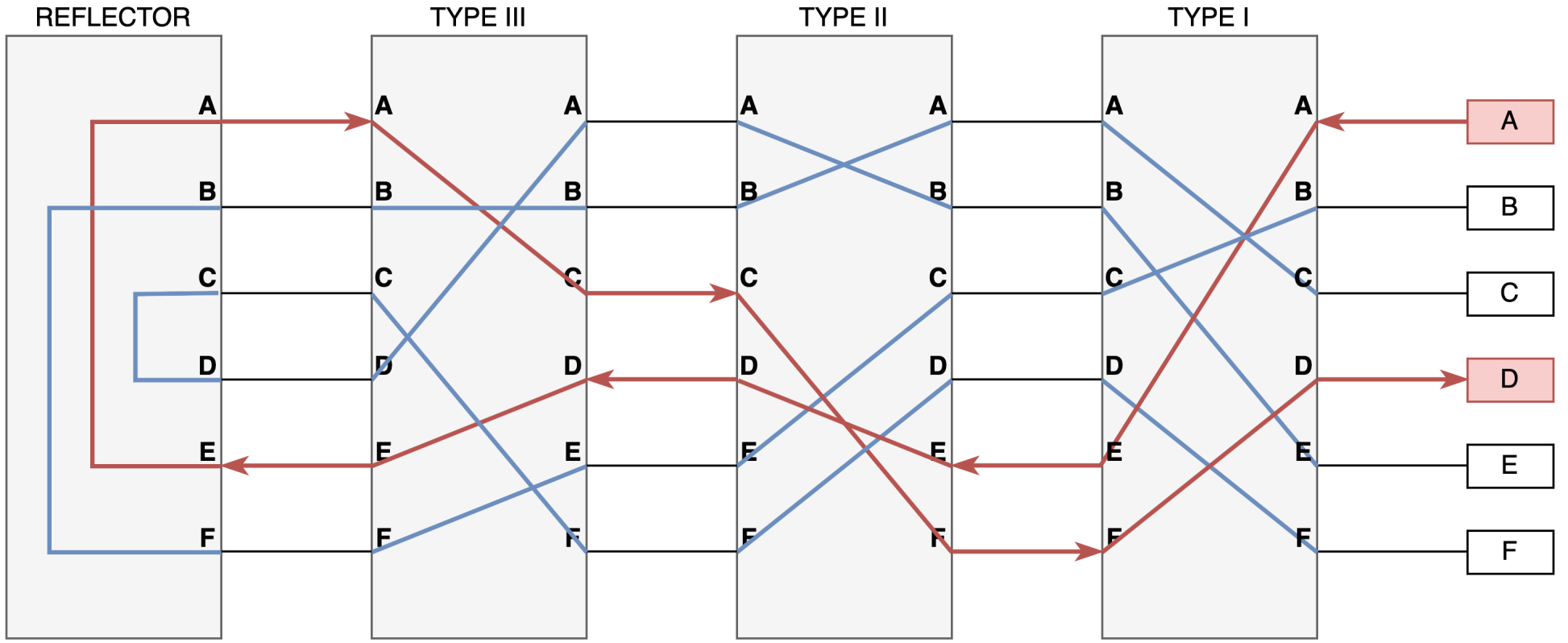
Different rotor types



Reflector

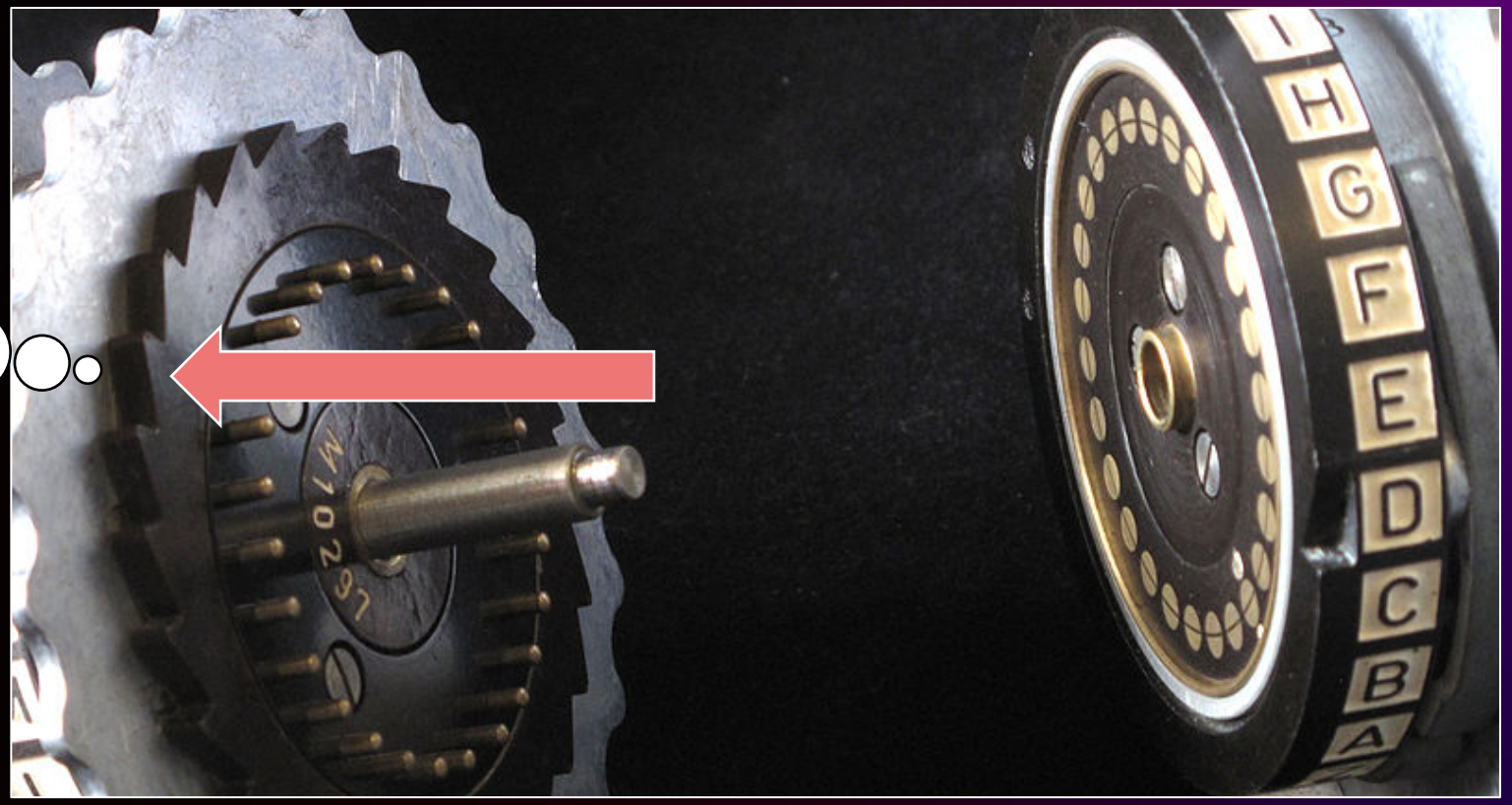


Rotors + reflector



Rotors

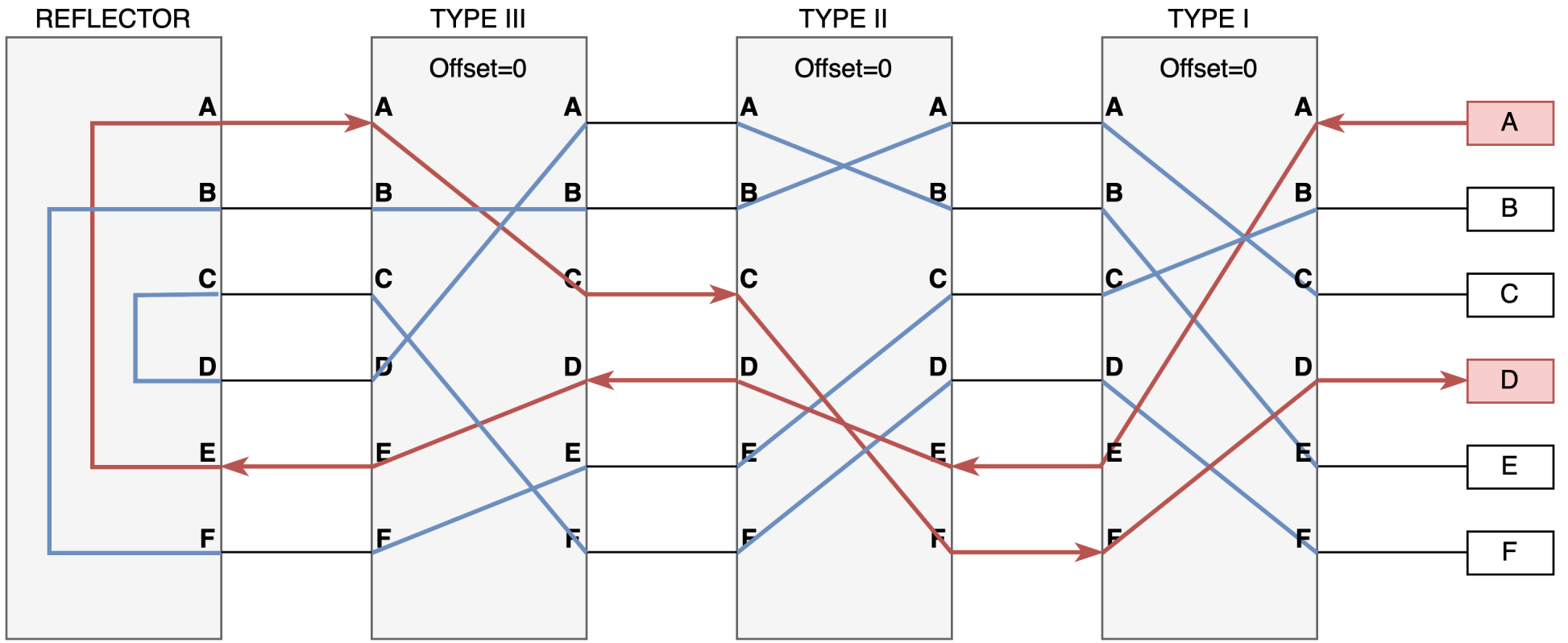
I wonder what these gear teeth are for…



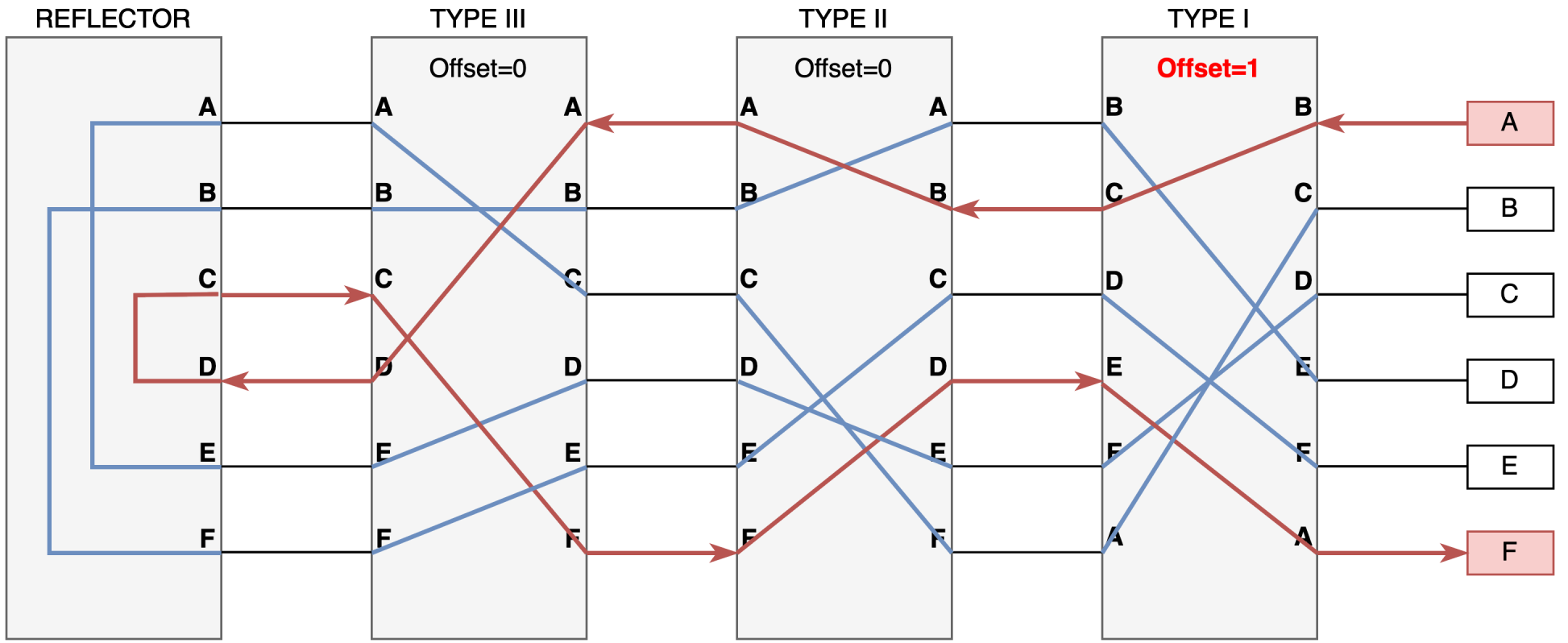
https://en.wikipedia.org/wiki/Enigma_machine

Tally counter

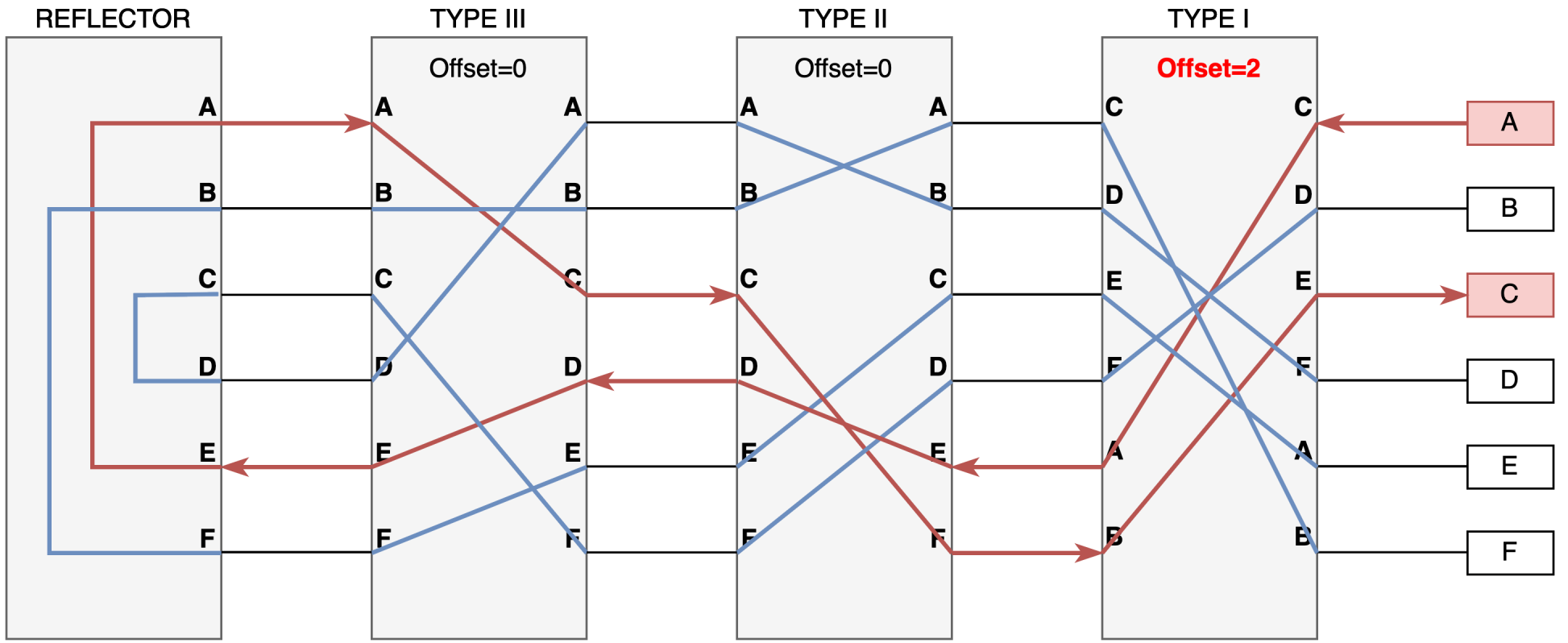
Rotor shift


REFLECTOR
TYPE III
Offset=0
TYPE II
Offset=0
TYPE I
Offset=0
A
B
C
D
E
F
A
B
C
D
E
F

Rotor shift


REFLECTOR
TYPE III
Offset=0
TYPE II
Offset=0
TYPE I
Offset=1
A
B
C
D
E
F

Rotor shift


REFLECTOR
TYPE III
Offset=0
TYPE II
Offset=0
TYPE I
Offset=2
A
B
C
D
E
F

Statistical Analysis

- Three rotors
- Each rotor has 26 positions

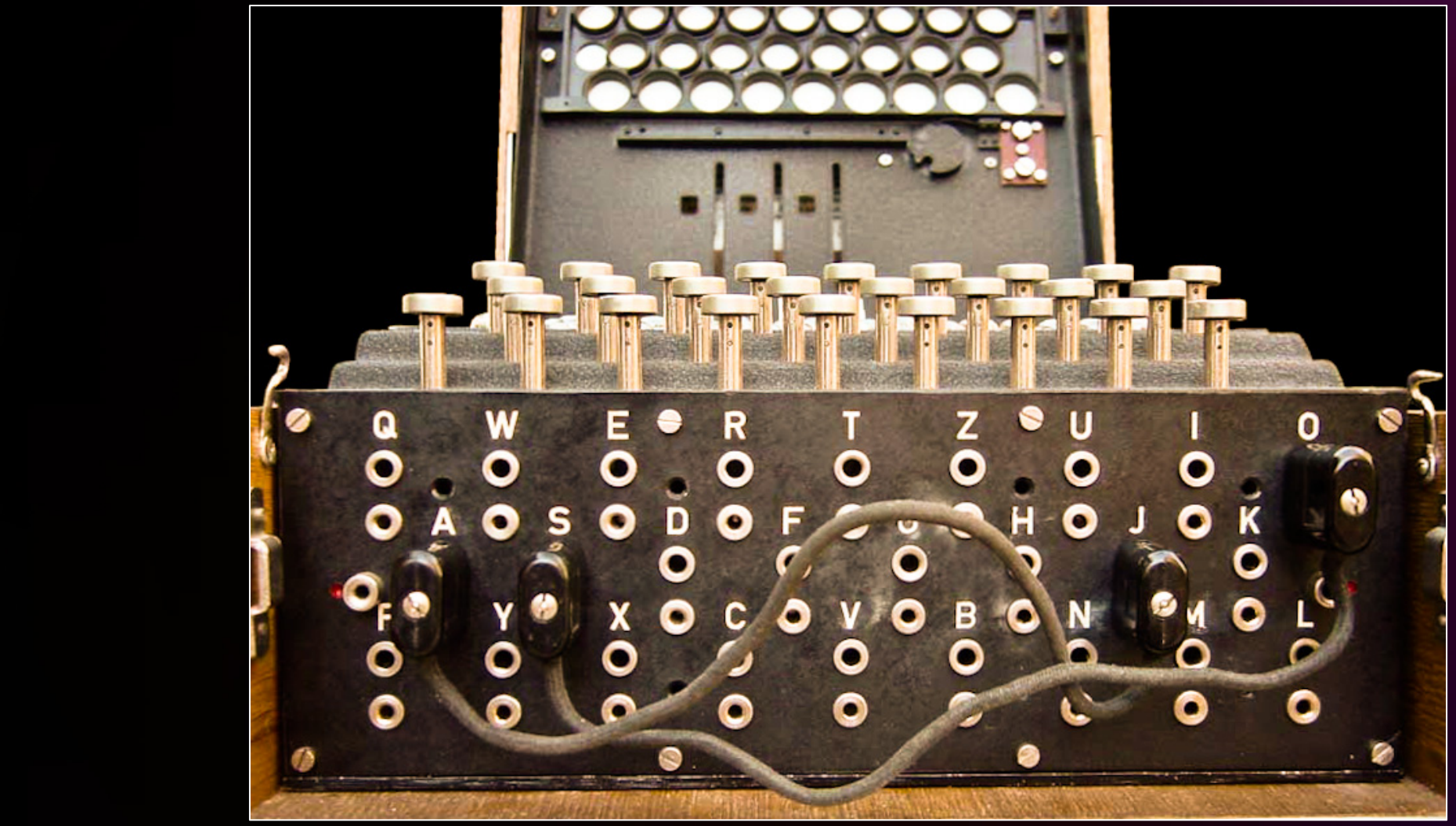
$$3! \cdot 26^3 = \mathbf{105{,}456}\ \textbf{configurations}$$

Statistical Analysis

- Three rotors **from a set of five**
- Each rotor has 26 positions

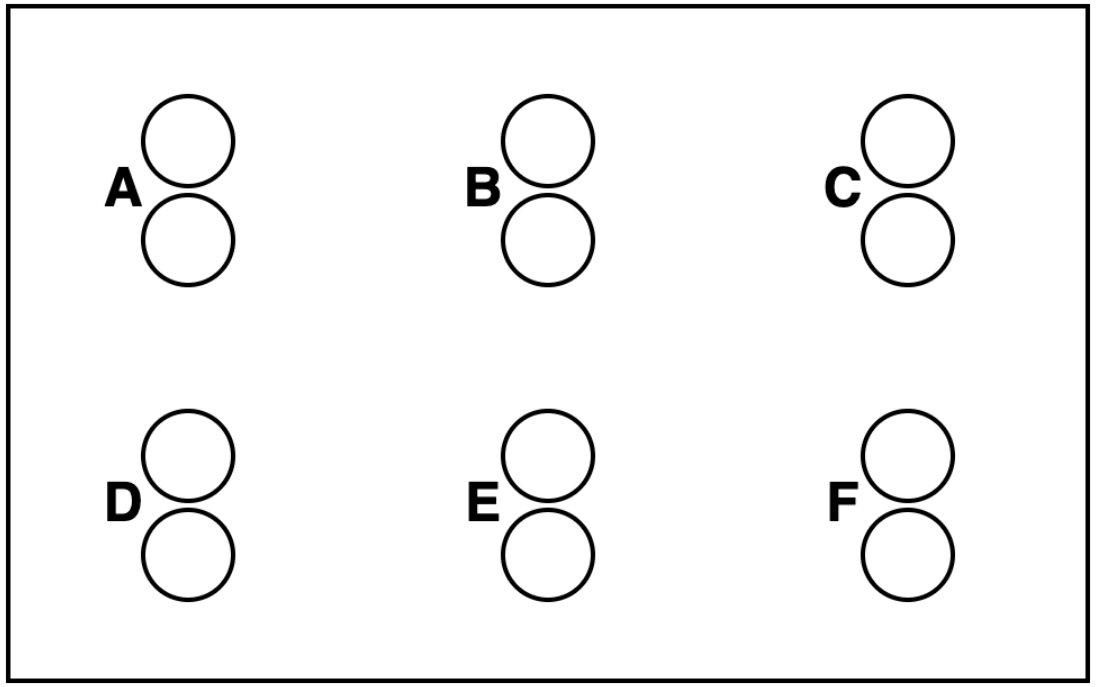
$$\frac{5!}{(5-3)!} \cdot 26^3 = 1{,}054{,}560 \text{ configurations}$$

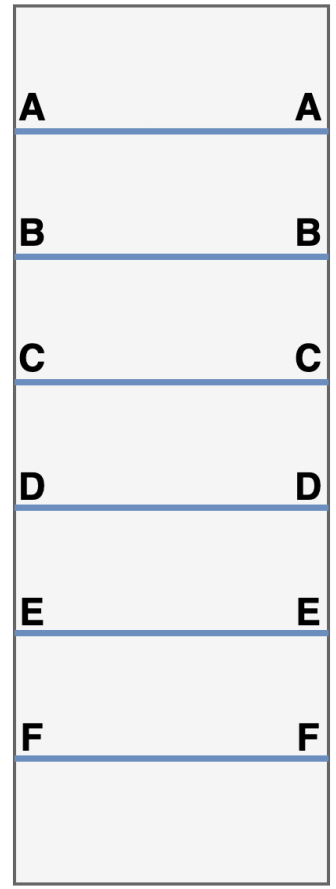
Plugboard



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

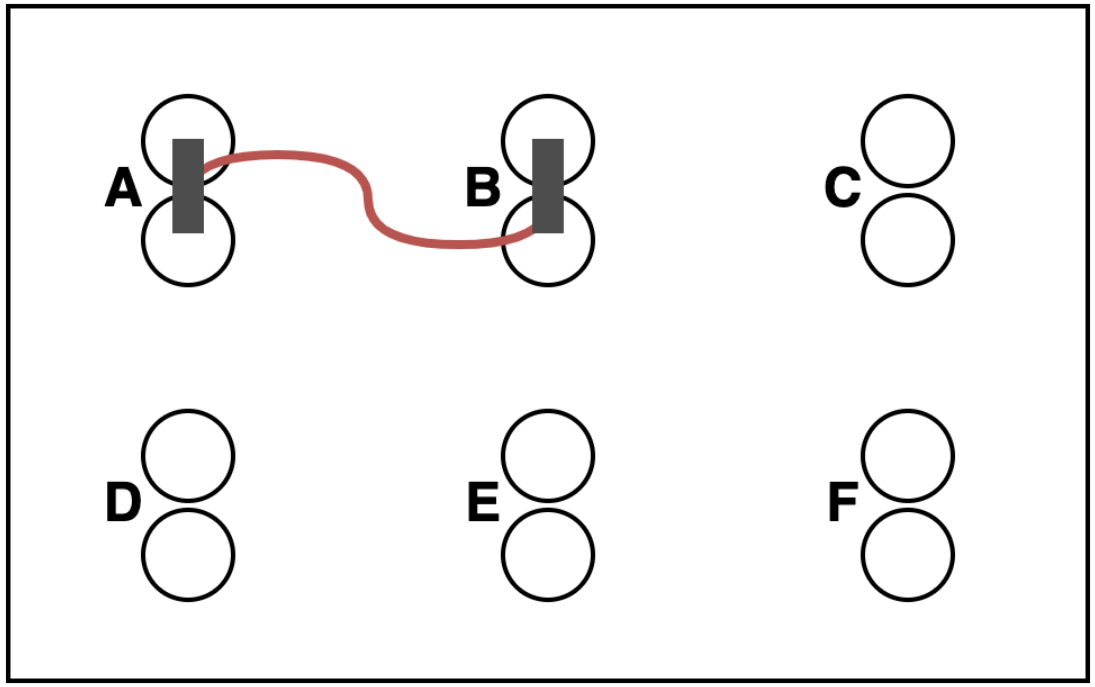
Plugboard

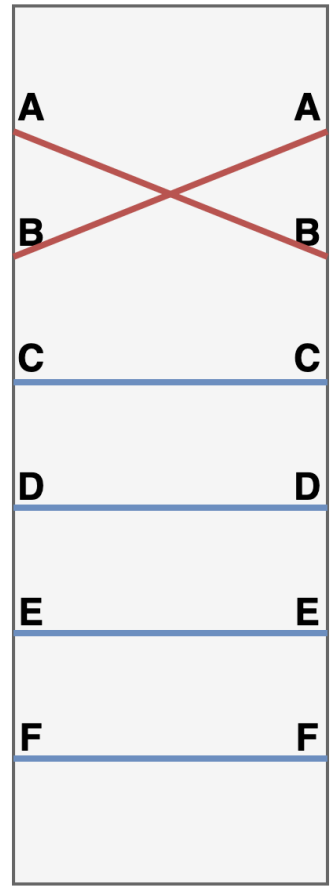




()

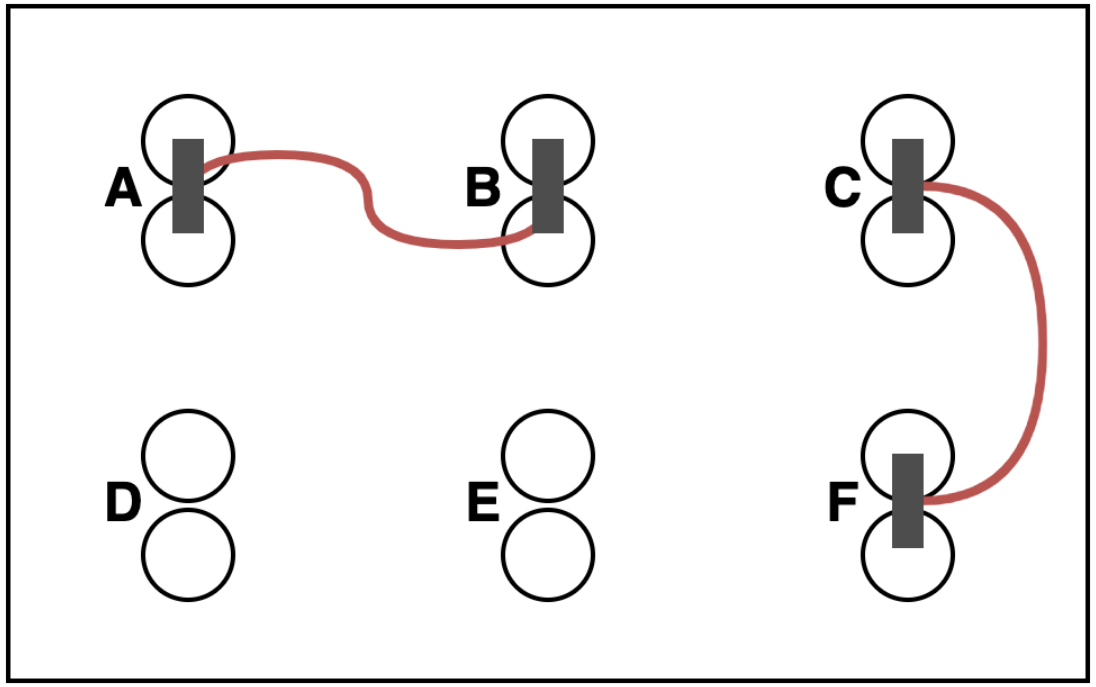
Plugboard

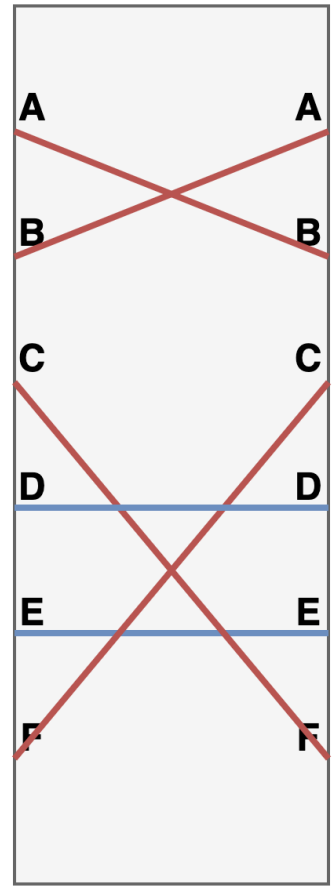




(AB)

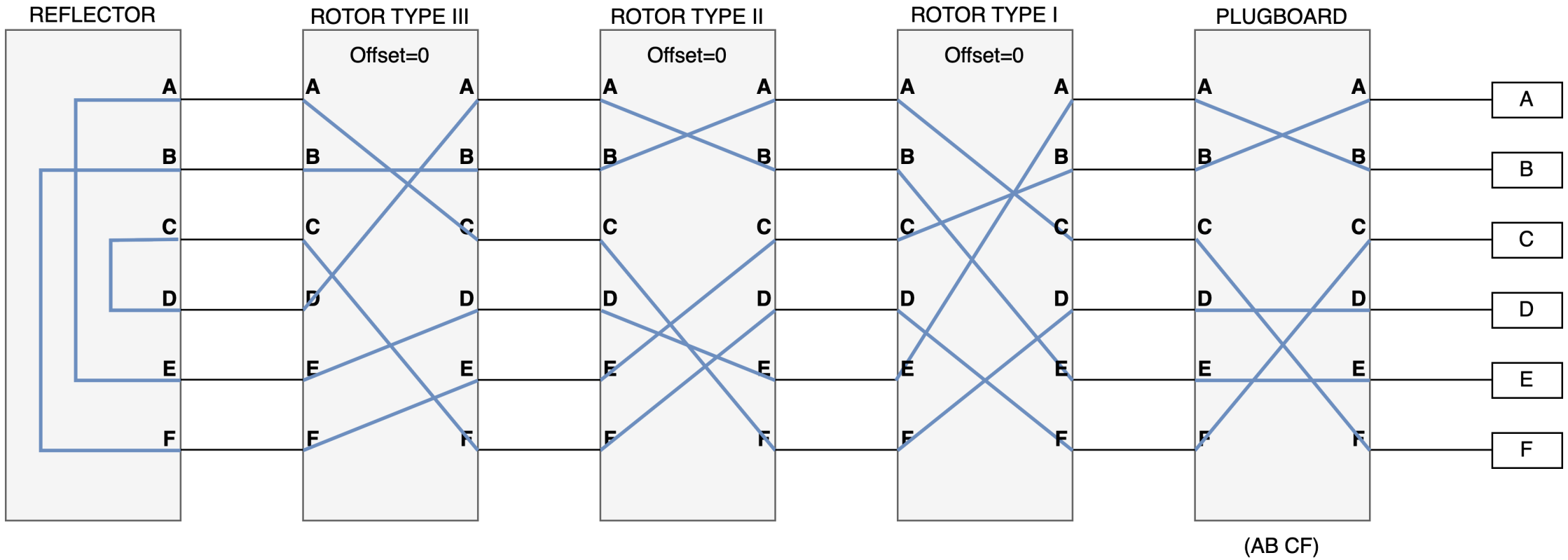
Plugboard



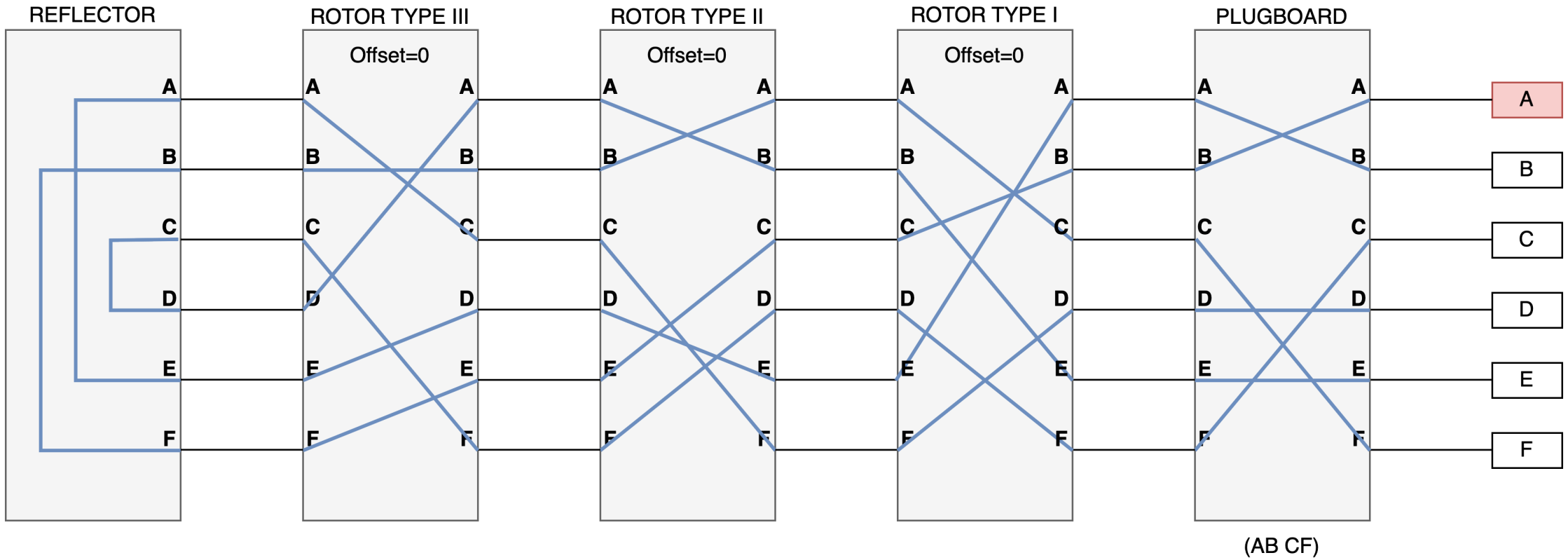


(AB CF)

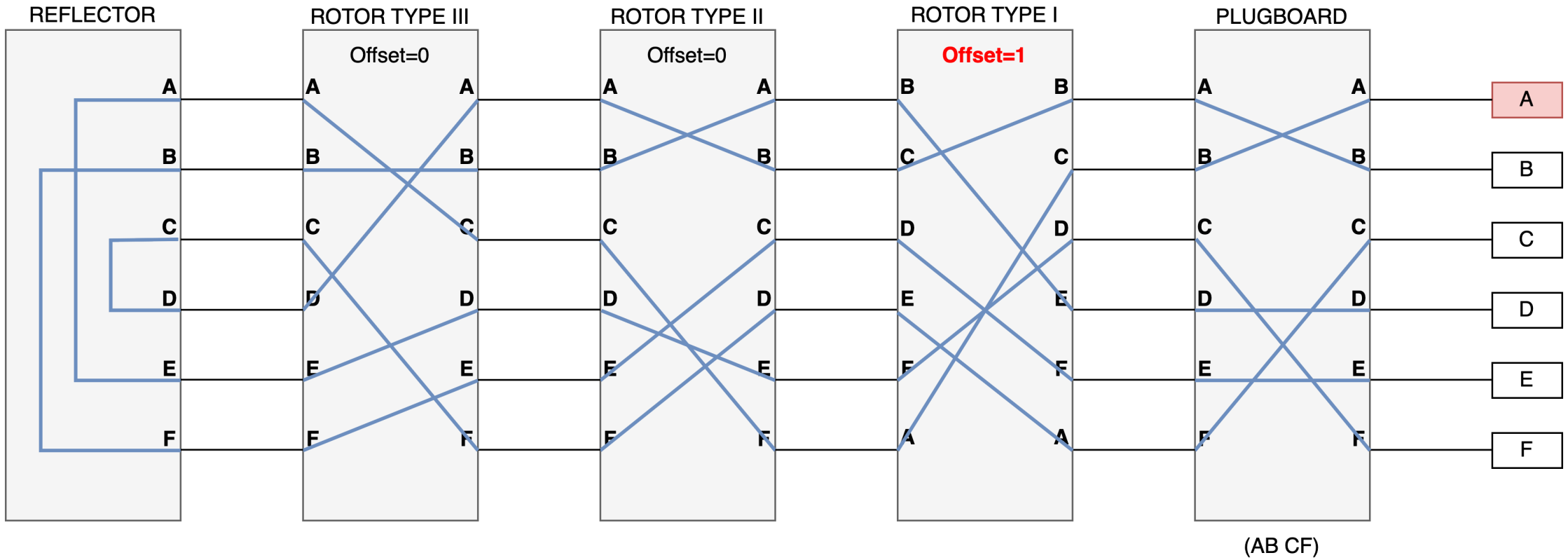
The Enigma Machine E2E



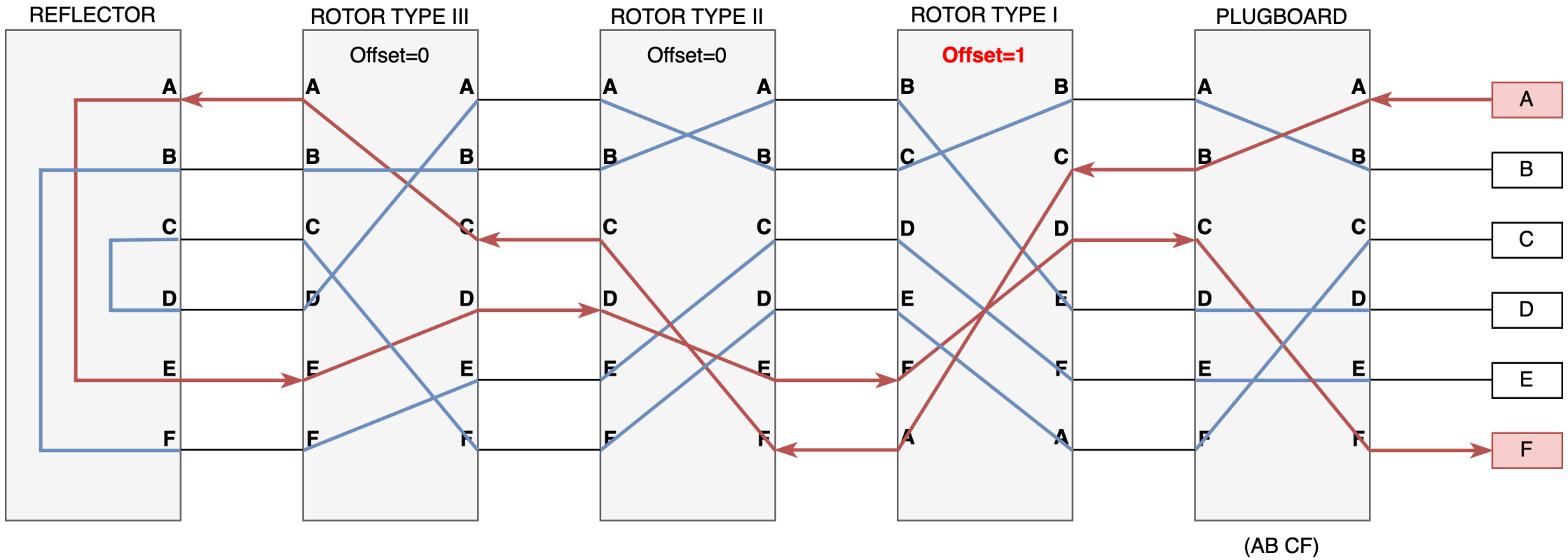
The Enigma Machine E2E



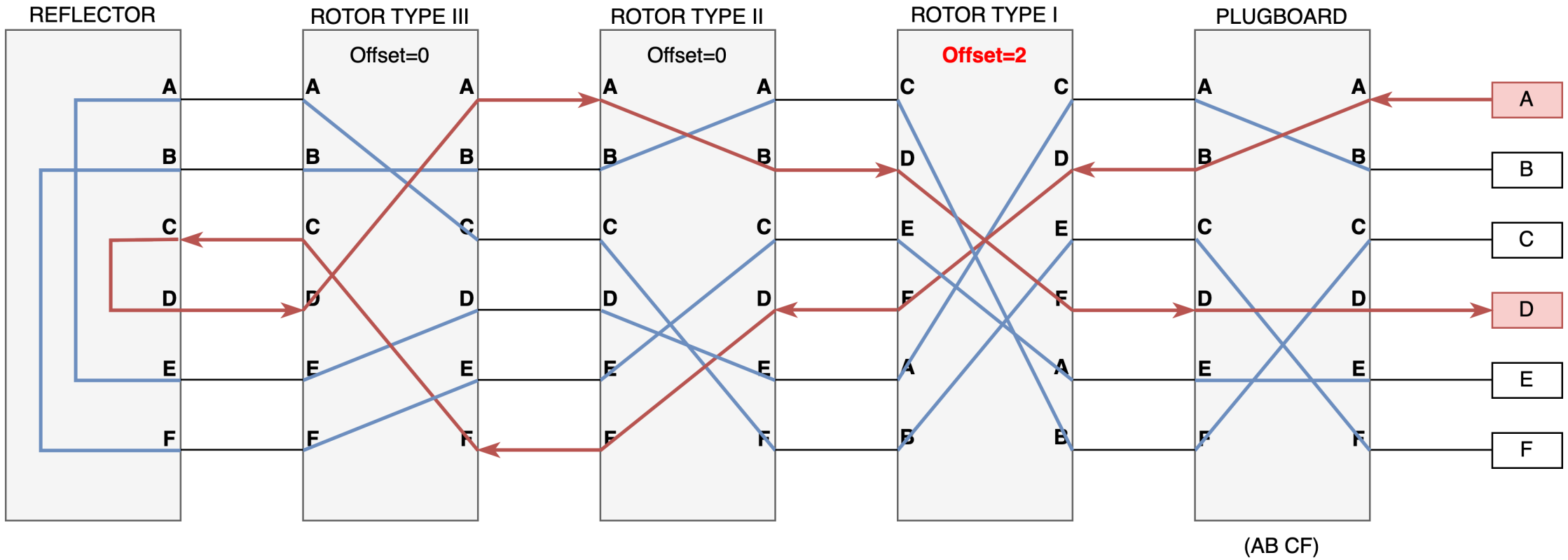
The Enigma Machine E2E



The Enigma Machine E2E



The Enigma Machine E2E



Statistical Analysis

- Three rotors from a set of five
- Each rotor has 26 positions
- **Plugboard with 10 pairs of letters connected**

$$\frac{26!}{(26-20)! \cdot 20^{10} \cdot 10!} =$$

150,738,274,937,250

Trillion

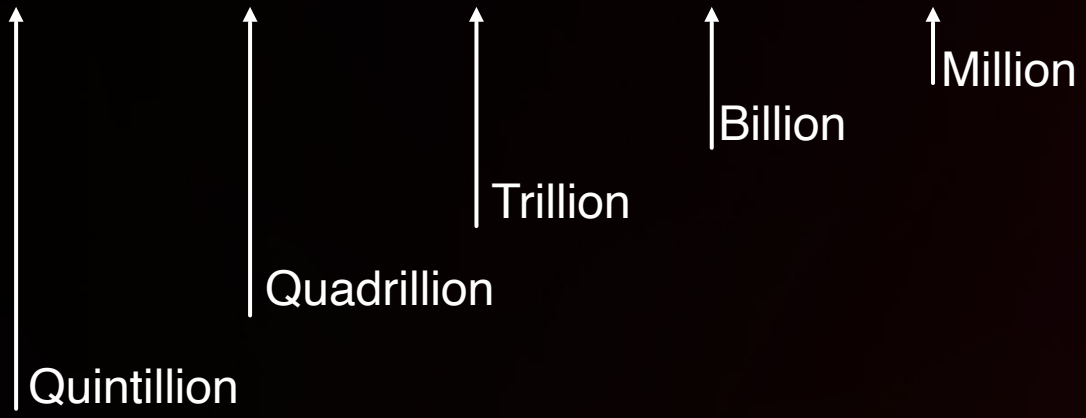
Billion

Million

Statistical Analysis

- Three rotors from a set of five
- Each rotor has 26 positions
- Plugboard with 10 pairs of letters connected

158,962,555,217,826,360,000 settings

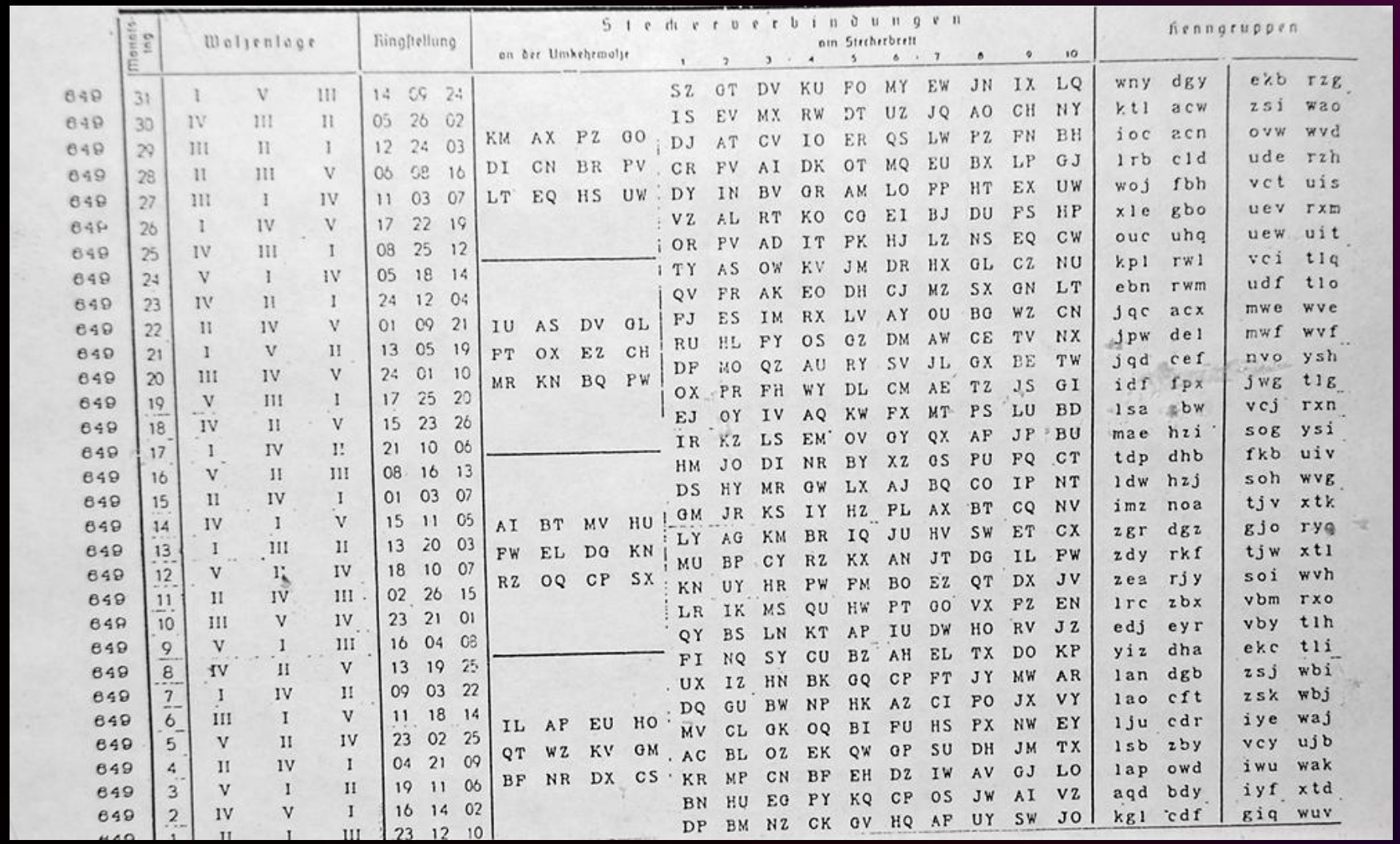


2^{67}

The Enigma Machine

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

The Enigma Machine



	Monats-tag	Walzenlage			Ringstellung			Steckerverbindungen an der Umkehrwalze				Steckerverbindungen am Steckerbrett										Kenngruppen			
												1	2	3	4	5	6	7	8	9	10				
649	31	I	V	III	14	09	24					SZ	GT	DV	KU	FO	MY	EW	JN	IX	LQ	wny	dgy	ekb	rzg
649	30	IV	III	II	05	26	02					IS	EV	MX	RW	DT	UZ	JQ	AO	CH	NY	ktl	acw	zsi	wao
649	29	III	II	I	12	24	03	KM	AX	PZ	GO	DJ	AT	CV	IO	ER	QS	LW	PZ	FN	BH	ioc	acn	ovw	wvd
649	28	II	III	V	06	08	16	DI	CN	BR	PV	CR	PV	AI	DK	OT	MQ	EU	BX	LP	GJ	lrb	cld	ude	rzh
649	27	III	I	IV	11	03	07	LT	EQ	HS	UW	DY	IN	BV	GR	AM	LO	FP	HT	EX	UW	woj	fbh	vct	uis
649	26	I	IV	V	17	22	19					VZ	AL	RT	KO	CG	EI	BJ	DU	FS	HP	xle	gbo	uev	rxm
649	25	IV	III	I	08	25	12					OR	PV	AD	IT	FK	HJ	LZ	NS	EQ	CW	ouc	uhq	uew	uit
649	24	V	I	IV	05	18	14					TY	AS	OW	KV	JM	DR	HX	GL	CZ	NU	kpl	rwl	vci	tlq
649	23	IV	II	I	24	12	04					QV	FR	AK	EO	DH	CJ	MZ	SX	GN	LT	ebn	rwm	udf	tlo
649	22	II	IV	V	01	09	21	IU	AS	DV	GL	FJ	ES	IM	RX	LV	AY	OU	BG	WZ	CN	jqc	acx	mwe	wve
649	21	I	V	II	13	05	19	PT	OX	EZ	CH	RU	HL	PY	OS	GZ	DM	AW	CE	TV	NX	jpw	del	mwf	wvf
649	20	III	IV	V	24	01	10	MR	KN	BQ	PW	DP	MO	QZ	AU	RY	SV	JL	GX	BE	TW	jqd	cef	nvo	ysh
649	19	V	III	I	17	25	20					OX	PR	FH	WY	DL	CM	AE	TZ	JS	GI	idf	fpx	jwg	tlg
649	18	IV	II	V	15	23	26					EJ	OY	IV	AQ	KW	FX	MT	PS	LU	BD	lsa	bw	vcj	rxn
649	17	I	IV	II	21	10	06					IR	KZ	LS	EM	OV	GY	QX	AF	JP	BU	mae	hzi	sog	ysi
649	16	V	II	III	08	16	13					HM	JO	DI	NR	BY	XZ	GS	PU	FQ	CT	tdp	dhb	fkb	uiv
649	15	II	IV	I	01	03	07					DS	HY	MR	GW	LX	AJ	BQ	CO	IP	NT	ldw	hzj	soh	wvg
649	14	IV	I	V	15	11	05	AI	BT	MV	HU	GM	JR	KS	IY	HZ	PL	AX	BT	CQ	NV	imz	noa	tjv	xtk
649	13	I	III	II	13	20	03	FW	EL	DG	KN	LY	AG	KM	BR	IQ	JU	HV	SW	ET	CX	zgr	dgz	gjo	ryq
649	12	V	II	IV	18	10	07	RZ	OQ	CP	SX	MU	BP	CY	RZ	KX	AN	JT	DG	IL	PW	zdy	rkf	tjw	xtl
649	11	II	IV	III	02	26	15					KN	UY	HR	PW	FM	BO	EZ	QT	DX	JV	zea	rjy	soi	wvh
649	10	III	V	IV	23	21	01					LR	IK	MS	QU	HW	PT	GO	VX	PZ	EN	lrc	zbx	vbm	rxo
649	9	V	I	III	16	04	08					QY	BS	LN	KT	AP	IU	DW	HO	RV	JZ	edj	eyr	vby	tlh
649	8	IV	II	V	13	19	25					FI	NQ	SY	CU	BZ	AH	EL	TX	DO	KP	yiz	dha	ekc	tli
649	7	I	IV	II	09	03	22					UX	IZ	HN	BK	GQ	CP	FT	JY	MW	AR	lan	dgb	zsj	wbi
649	6	III	I	V	11	18	14					DQ	GU	BW	NP	HK	AZ	CI	PO	JX	VY	lao	cft	zsk	wbj
649	5	V	II	IV	23	02	25	IL	AP	EU	HO	MV	CL	GK	OQ	BI	FU	HS	PX	NW	EY	lju	cdr	iye	waj
649	4	II	IV	I	04	21	09	QT	WZ	KV	GM	AC	BL	OZ	EK	QW	GP	SU	DH	JM	TX	lsb	zby	vcy	ujb
649	3	V	I	II	19	11	06	BF	NR	DX	CS	KR	MP	CN	BF	EH	DZ	IW	AV	GJ	LO	lap	owd	iwu	wak
649	2	IV	V	I	16	14	02					BN	HU	EG	PY	KQ	CP	OS	JW	AI	VZ	aqd	bdy	iyf	xtd
649	1	II	I	III	23	12	10					DP	BM	NZ	CK	GV	HQ	AF	UY	SW	JO	kgl	cdf	giq	wuv

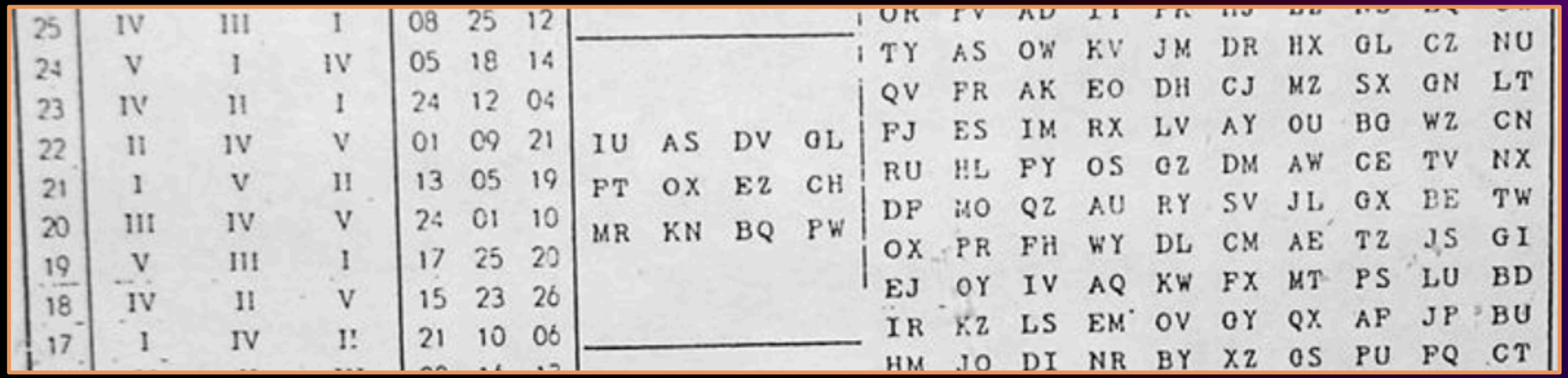
https://en.wikipedia.org/wiki/Enigma_machine

The Enigma Machine

Day of the month

Rotor offsets

Plugboard



25	IV	III	I	08	25	12	IU	AS	DV	GL	TY	AS	OW	KV	JM	DR	HX	GL	CZ	NU
24	V	I	IV	05	18	14					QV	FR	AK	EO	DH	CJ	MZ	SX	GN	LT
23	IV	II	I	24	12	04					FJ	ES	IM	RX	LV	AY	OU	BG	WZ	CN
22	II	IV	V	01	09	21					RU	HL	PY	OS	GZ	DM	AW	CE	TV	NX
21	I	V	II	13	05	19	PT	OX	EZ	CH	DP	MO	QZ	AU	RY	SV	JL	GX	BE	TW
20	III	IV	V	24	01	10	MR	KN	BQ	PW	OX	PR	FH	WY	DL	CM	AE	TZ	JS	GI
19	V	III	I	17	25	20					EJ	OY	IV	AQ	KW	FX	MT	PS	LU	BD
18	IV	II	V	15	23	26					IR	KZ	LS	EM	OV	GY	QX	AP	JP	BU
17	I	IV	II	21	10	06					HM	JO	DI	NR	BY	XZ	GS	PU	FQ	CT

Rotor types and order

Configurable reflector

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

The Enigma Machine

Patented Jan. 24, 1928. **1,657,411**

UNITED STATES PATENT OFFICE.

ARTHUR SCHERBIUS, OF BERLIN-WILMERSDORF, GERMANY, ASSIGNOR, BY MESNE ASSIGNMENTS, TO CHIFFRIERMASCHINEN AKTIENGESELLSCHAFT, OF BERLIN, GERMANY, A CORPORATION OF GERMANY.

CIPHERING MACHINE.

Application filed February 6, 1923, Serial No. 617,352, and in Germany February 11, 1922.

It has already been proposed to use for ciphering of a clear text and for deciphering machines which either type the ciphered letters in a similar manner to that of a typewriting machine or which produce a ciphered perforated cable tape or operate an indicating device. The operation of machines of this type is based for instance on the interchanging of the closed circuits between the keys marked with the letters of the alphabet and the type levers or the levers of a perforator for cable tapes each time after the sending of one or more of a determined number of letters. As soon as with two machines of this type this interchange, which is per se irregular, is effected in exactly the same manner, a telegram which has been ciphered with the aid of one machine can be deciphered with the aid of a corresponding machine. A condition is however that the number of letters counted from the same starting position has remained the same. At the sending of telegrams, especially with wireless telegraphy, one must however count upon the accidental omission of certain letters or groups of letters. The machine which is used in such a case for deciphering is thus unsynchronized, so that not only the letters which have been omitted but also all the succeeding text cannot be deciphered any more.

According to the invention this defect is avoided or at least restricted greatly by providing on the ciphering machine a device by means of which finishing of a series of letters of determined length is signalized every time to the operator of the machine so that he can mark the beginning of the new series of letters in the ciphered text. It is thus possible to compare and if necessary to correct the position of the deciphering machine after every series of letters. The termination of the series of letters is preferably signalized by the sounding of a bell or by the lighting up of an incandescent lamp. It would be better still if, after the termination of a determined series of letters, the machine is automatically stopped entirely or partly or thrown out of operation so that it is impossible to continue the typing. The mechanism which effects the interchange of the letters may for instance be stopped. The beginning of the new row of letters may then be indicated for instance by repeated sending of the same letter. The device which serves for counting the length of the row of letters may continue to operate, if this should be desirable for any reason, if the same letter is repeated each time for a determined number of times. It would be still better to reverse the machine in such a manner that it types clear text, the mechanism which effects the interchange of letters being stopped, wherefrom results the advantage that in the clear text an easily recognizable message can be given and that after the sending of this message the ciphering can be continued with the machine which during this sending has not been adjusted for sending code. Such a message can consist for instance of a check member, or if desired the number of letters which have been sent up to this moment. For each series of letters a new key indication might be selected on the machine which key indication for safety's sake would be sent in the clear text several times. Service regulations might further be inserted. Clear text might further be signalized by special signs, for instance by spaced type.

In order to make the invention clearly understood I shall proceed to describe the same with reference to the accompanying drawing wherein:

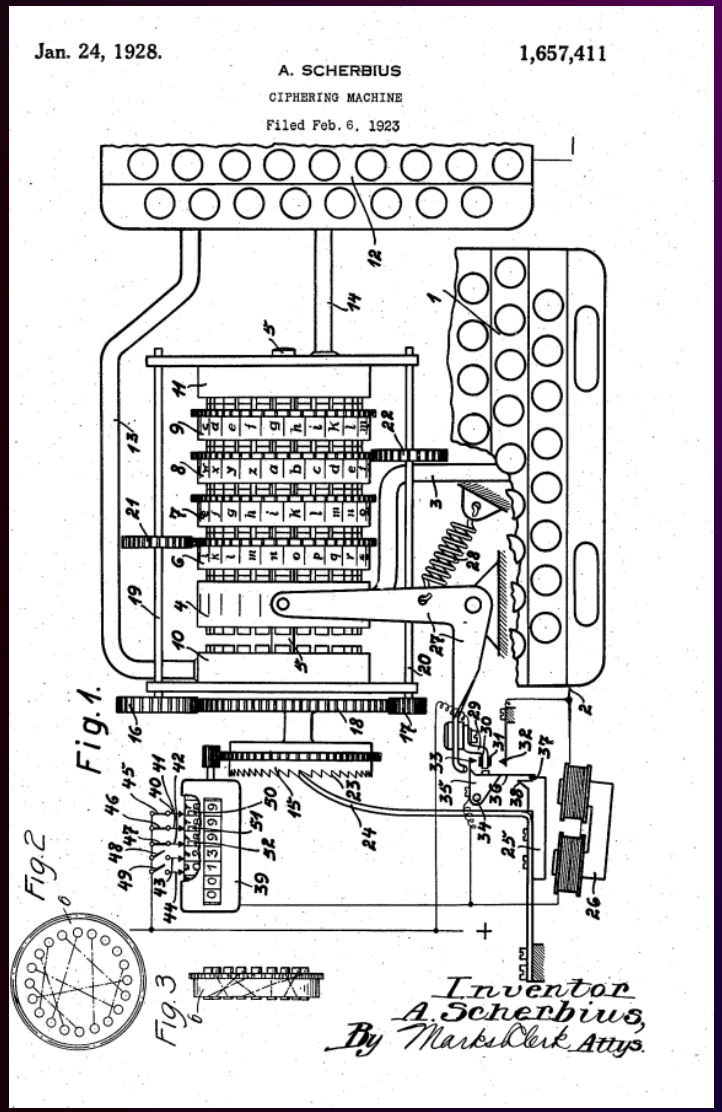
Fig. 1 shows by way of example a ciphering machine according to this invention.

Figure 2 is an edge elevation of one of the rotatable contact drums showing the irregular connection of the contact points.

Figure 3 is a front elevation of the drum.

Each key of the key board 1 of the typewriting machine connects by means of a contact operated at the depression of the key the lead 2 with one of the leads contained in cable 3 belonging to this key. The several leads corresponding with the keys are insulated from one another and united to form a cable 3 which is in contact with a drum 4. Upon the end faces of this drum as many contacts are arranged in a circle as the typewriting machine has keys and the several leads of the cables 3 are connected with these contacts in such a manner that every two opposite contacts are connected with the same lead. The drum 4 is mounted on a shaft 5 so that it may be displaced in longitudinal direction but cannot revolve with said shaft. Upon the

US1657411A



Cracking the Enigma Machine

Marian Rejewski

The Polish mathematician who, in 1932, first broke Enigma and, in July 1939, helped to educate the French and British about Polish methods of Enigma decryption

Alan Turing

English mathematician and computer scientist who devised techniques cracking the Enigma machine ciphers

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

Cracking the Enigma Machine

I don’t see any servers here…

Is it….
SERVERLESS !?

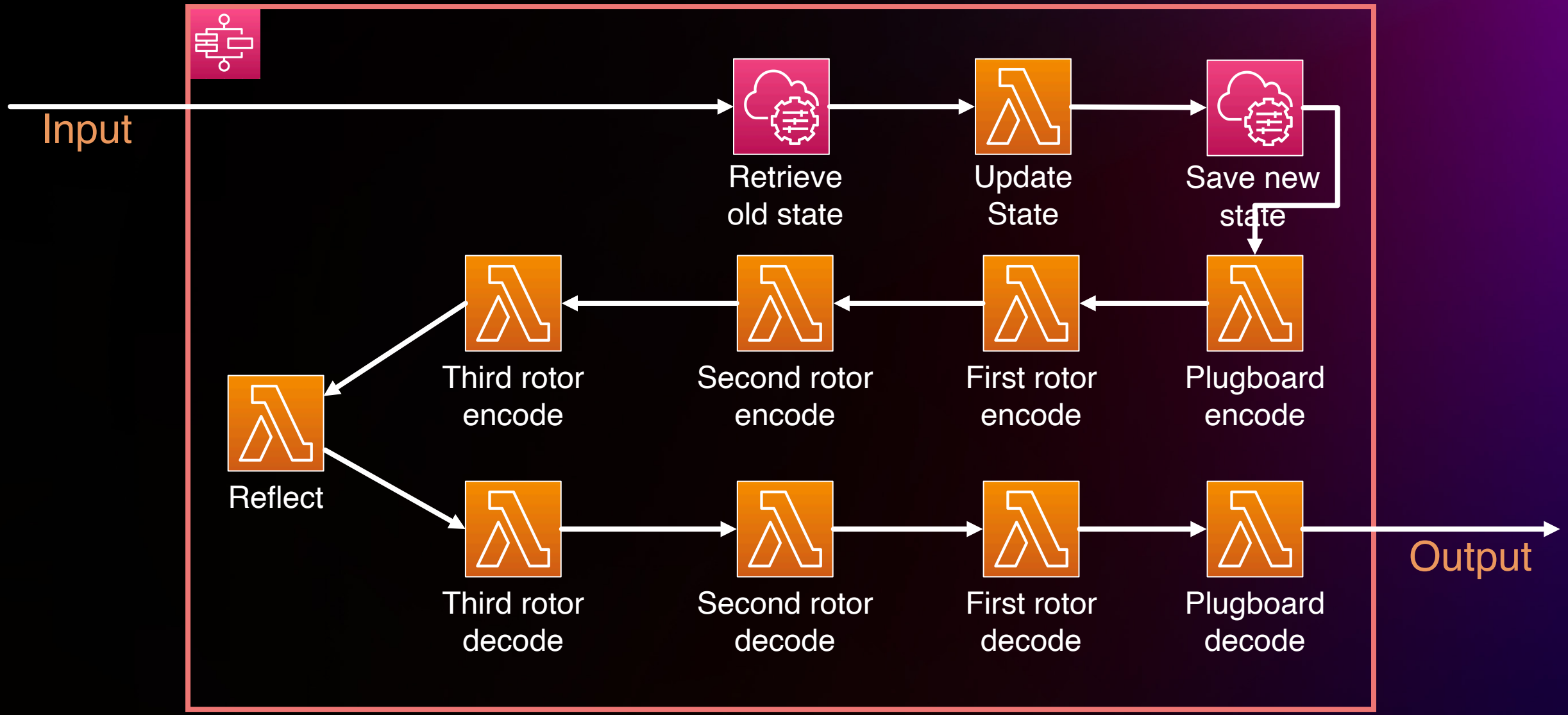
https://en.wikipedia.org/wiki/Enigma_machine

Reimplementing the Enigma Machine

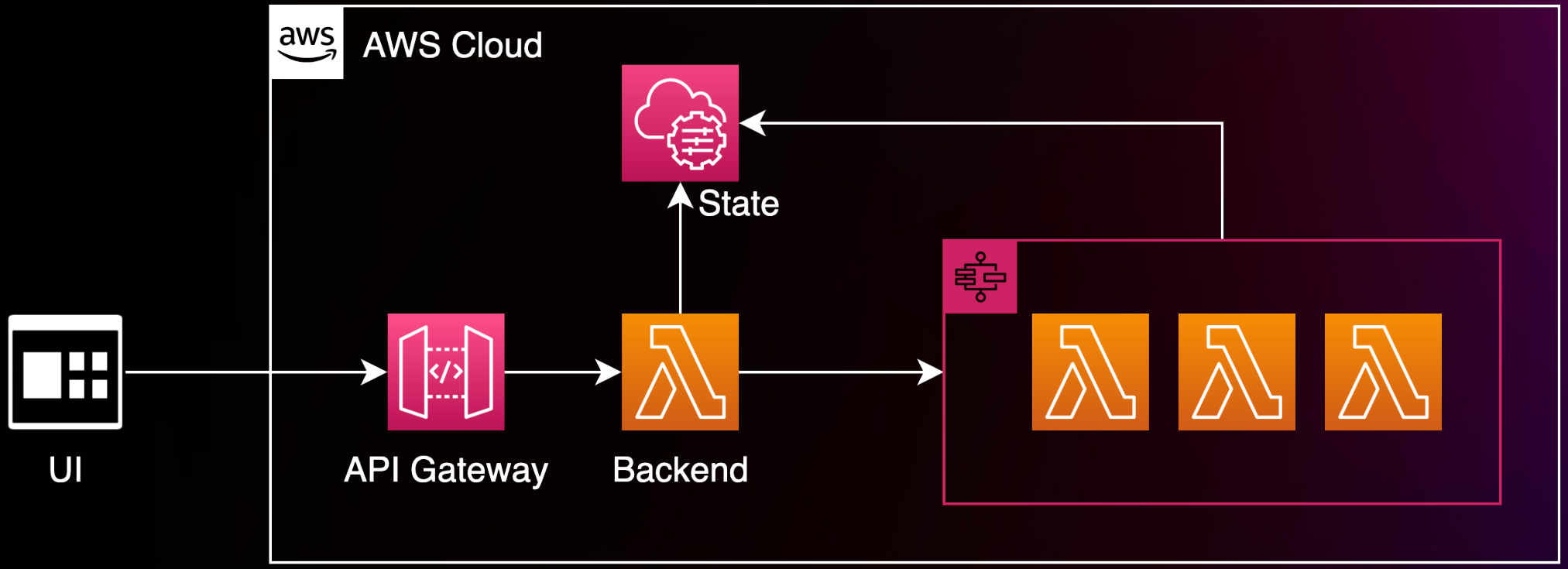
- A set of loosely coupled components
- Has initial state, configurable
- Receives input
- Processes input through a workflow
- Modifies input at each step
- Produces output
- Essentially a **state machine**!

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

Reimplementing the Enigma Machine

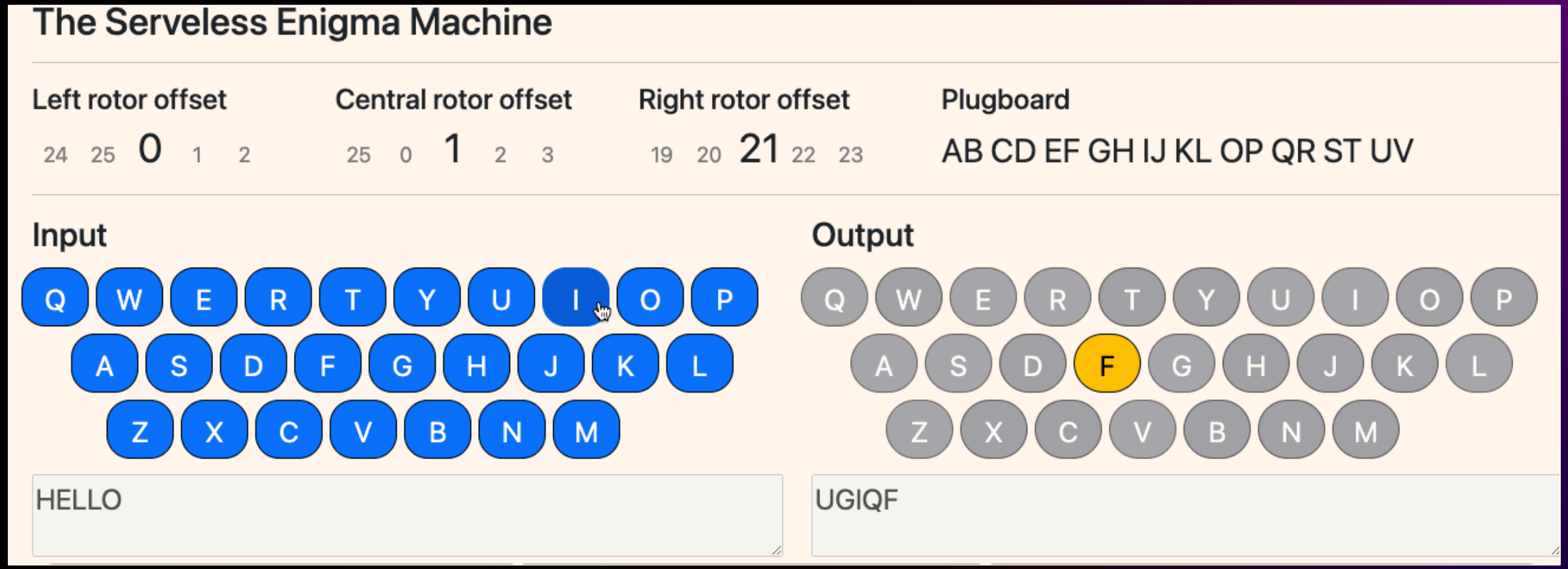


Reimplementing the Enigma Machine

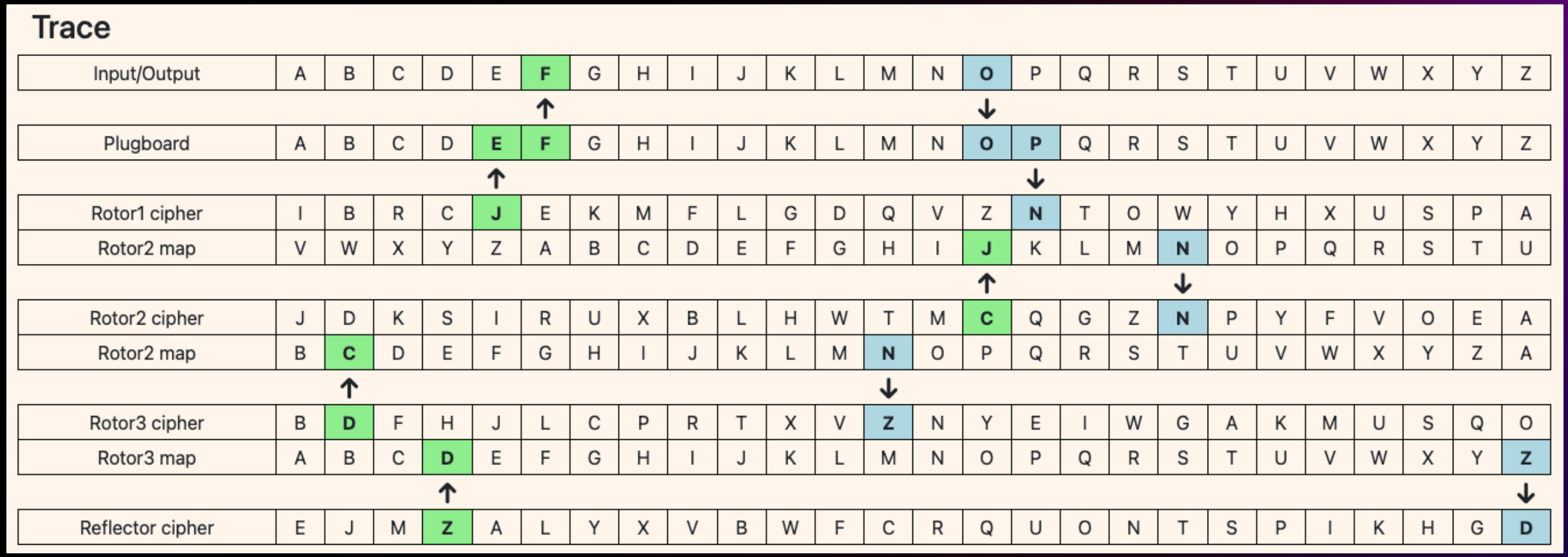


Demo

Reimplementing the Enigma



Reimplementing the Enigma



Reimplementing the Enigma

Reflector

```
const CIPHER = 'EJMZALYXVBWFCRQUONTSPIKHGD';

export const handler = async(event) => {
    const inputPin = event.inputPin;
    const cipherLetter = CIPHER[inputPin];
    const outputPin = cipherLetter.charCodeAt(0) - 65;
    console.log(`[handler] inputPin=${inputPin} cipherLe
    return { inputPin, cipherLetter, outputPin, cipher:
};
```

inputPin=4 outputPin=0

Reimplementing the Enigma

Rotor

```
function encode(offset, inputPin){
    const inputIdx = (inputPin + offset) % 26;
    const cipherLetter = CIPHER[inputIdx];
    const cipherLetterIdx = cipherLetter.charCodeAt(0) - 65;
    let outputPin = cipherLetterIdx - offset;
    if (outputPin < 0) outputPin = outputPin + 26;
    console.log(`[encode] inputPin=${inputPin} offset=${offse
    return {inputPin, offset, inputIdx, cipherLetter, cipherL
}
```

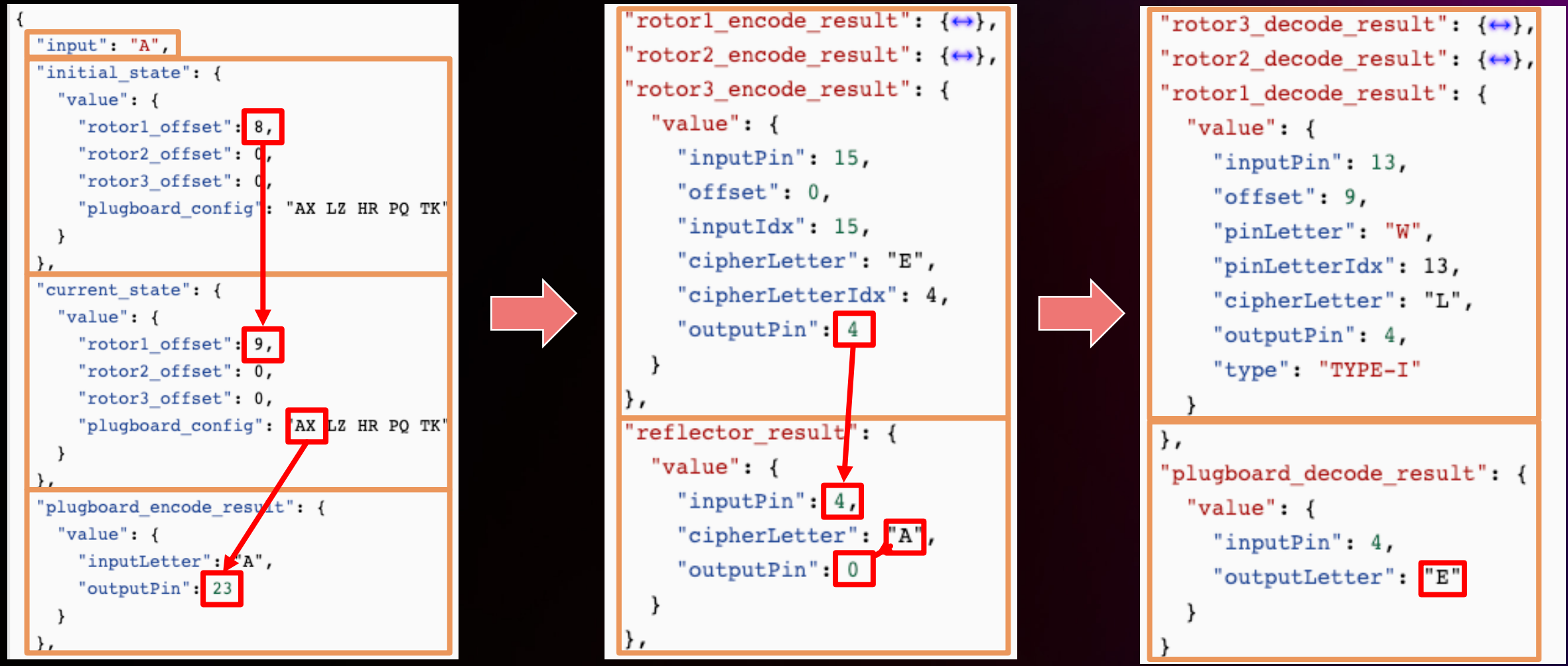
Reimplementing the Enigma

Plugboard

```
function encode(plugs, inputLetter){
    const outputLetter = plugs[inputLetter] || inputLetter;
    const outputPin = outputLetter.charCodeAt(0) - 65;
    console.log(`[encode] inputLetter=${inputLetter} outputLe
    return { inputLetter, outputLetter, outputPin };
}
```

```
plugs = {
    "A":"B",
    "D":"X",
    "P":"K"
}
```

Reimplementing the Enigma Machine




```
const CIPHER = 'EJMZALYXVBWFCRQUONTSPIKHGD';
```

Thank you!

Anton Aleksandrov
Principal Solutions Architect
AWS Serverless