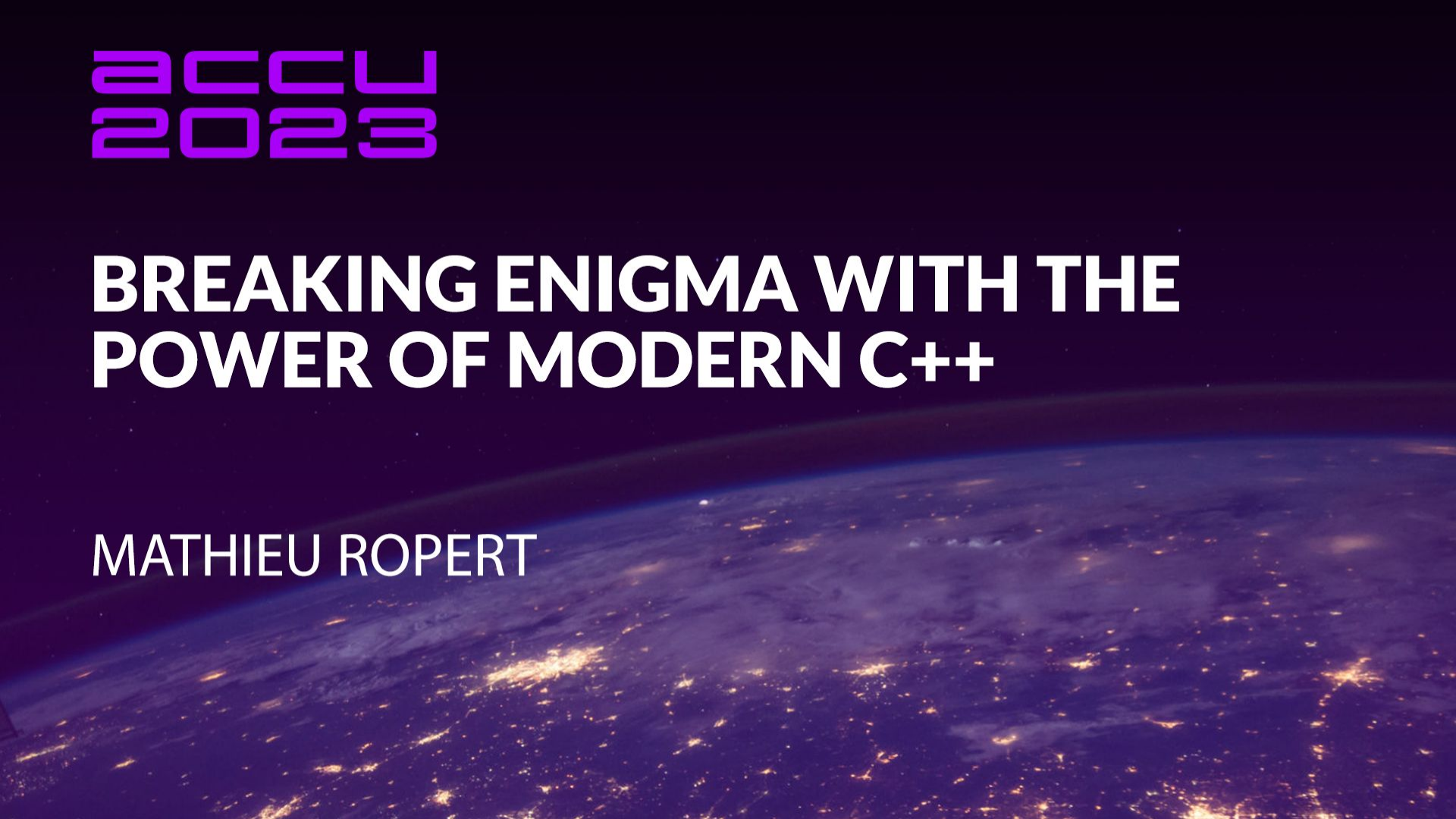
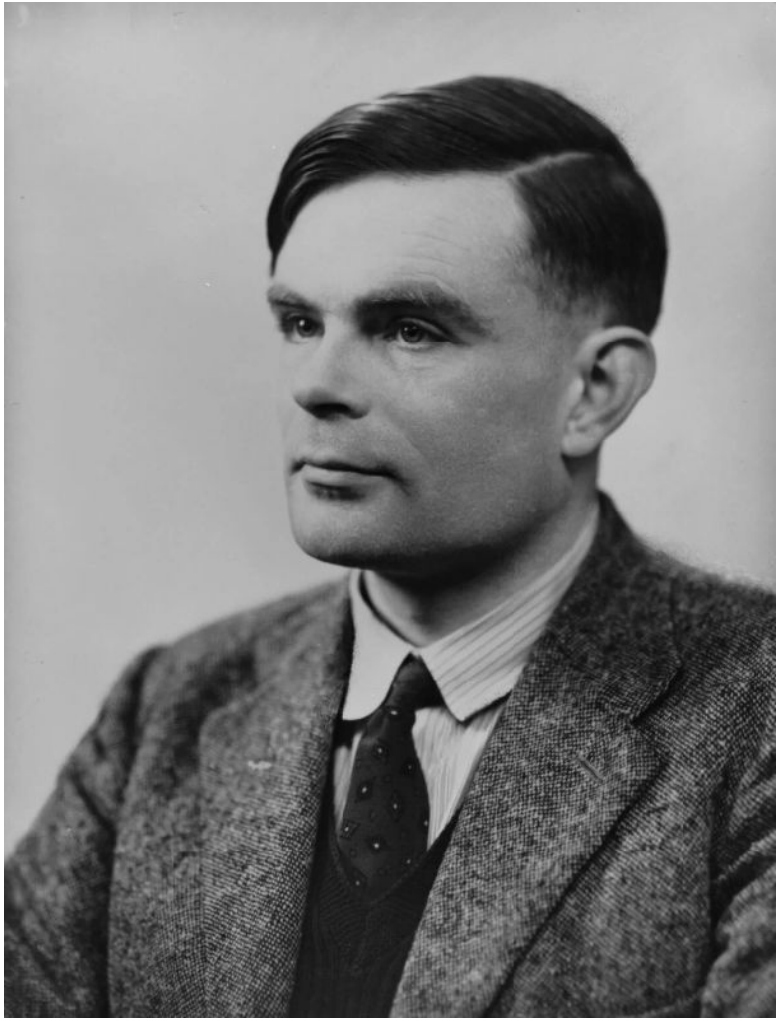
accu 2023

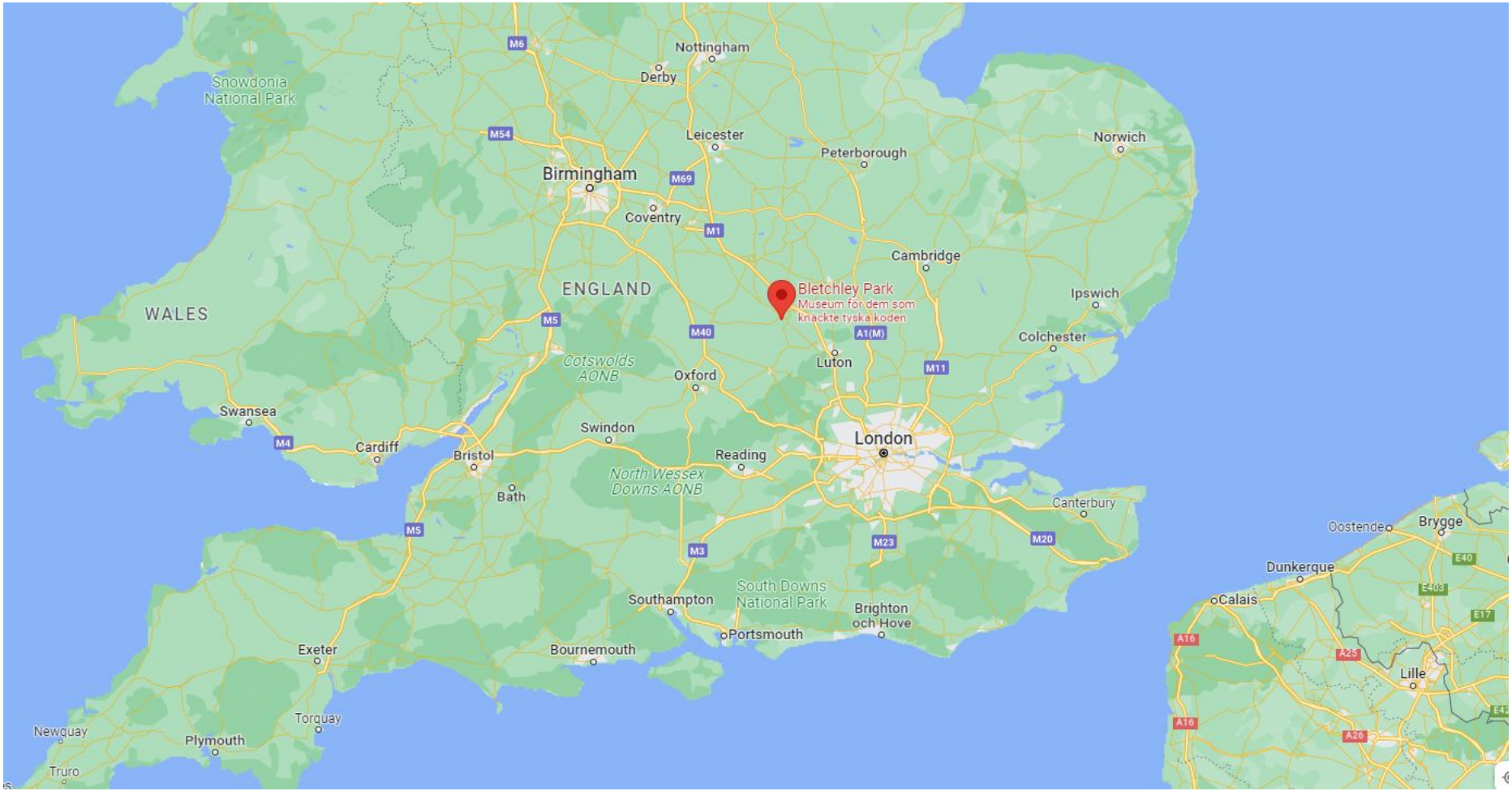
BREAKING ENIGMA WITH THE POWER OF MODERN C++

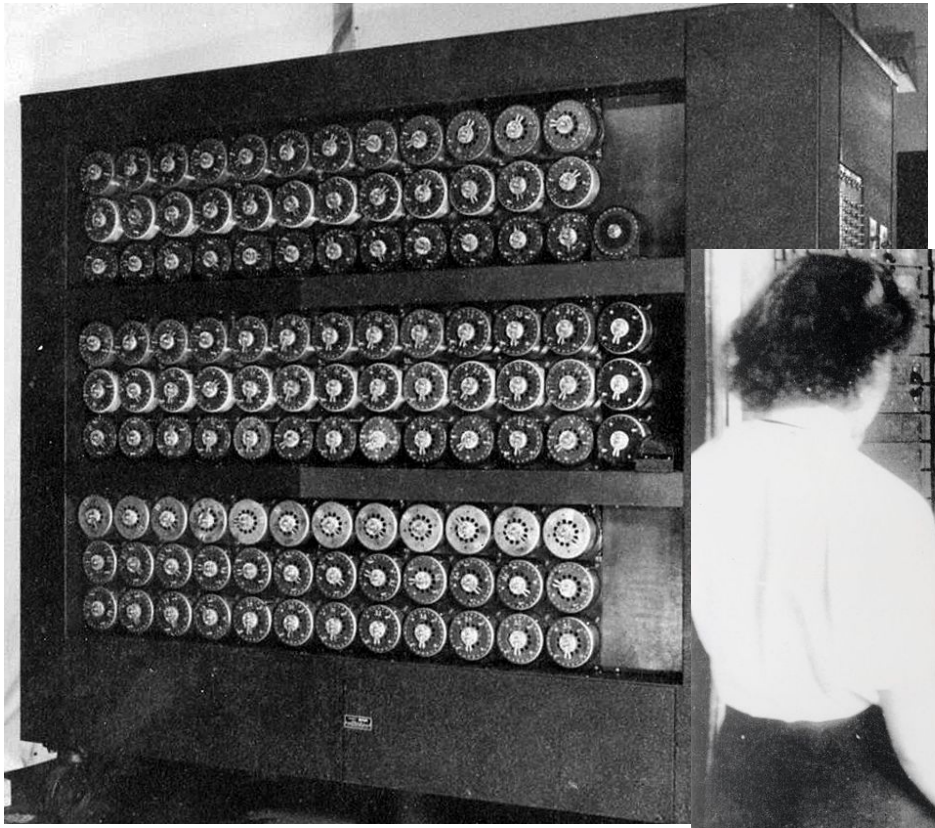
MATHIEU ROPERT

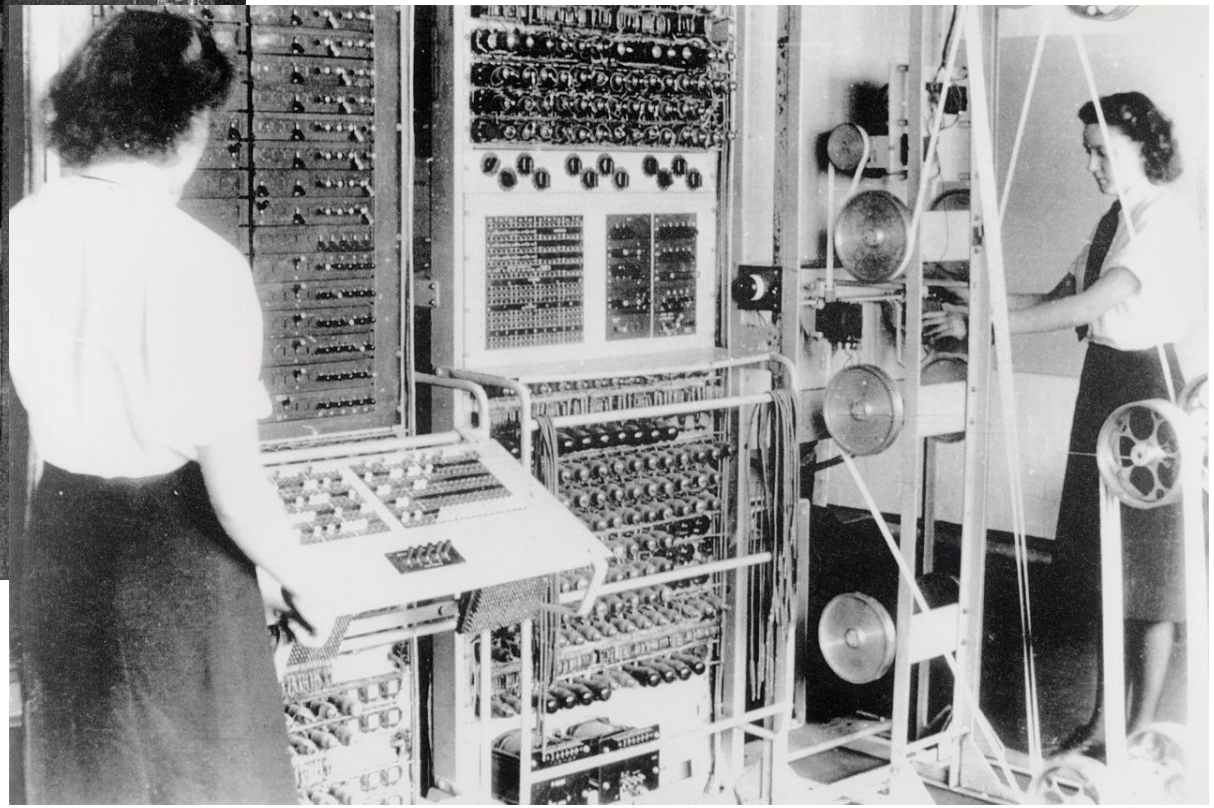


Breaking Enigma With the Power of Modern C++




Bletchley Park
Museum för dem som knackte tyska koden
ENGLAND
WALES
Snowdonia National Park
Nottingham
Derby
Leicester
Peterborough
Norwich
Birmingham
Coventry
Cambridge
Ipswich
Colchester
Luton
Oxford
Cotswolds AONB
Swansea
Cardiff
Bristol
Bath
Swindon
Reading
London
North Wessex Downs AONB
Canterbury
Oostende
Brygge
Dunkerque
Calais
Lille
Southampton
Portsmouth
South Downs National Park
Brighton och Hove
Bournemouth
Exeter
Torquay
Plymouth
Newquay
Truro
M6
M54
M69
M1
M5
M40
A1(M)
M11
M4
M3
M23
M20
A16
A25
A26
E40
E403
E17






Nur Glühlampen mit 12 mm
Durchmesser verwenden.

Hello!

I am Mathieu Ropert

I'm a Tech Lead at Paradox Development Studio where I make Hearts of Iron IV.

You can reach me at:

 mro@puchiko.net

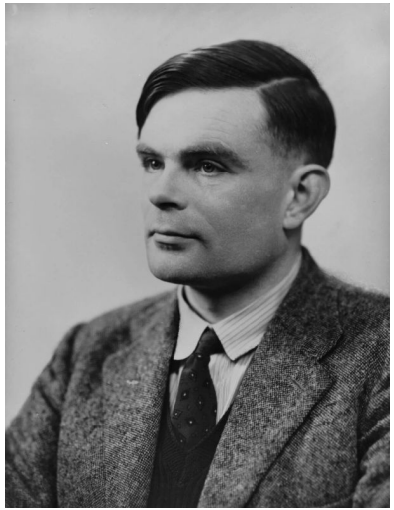
 @MatRopert

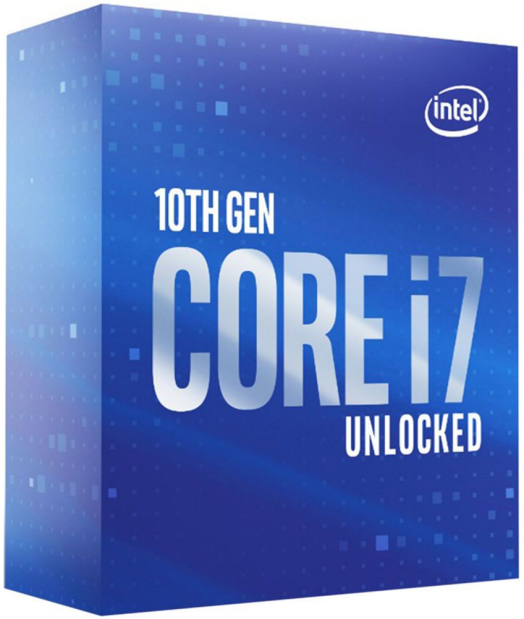
 https://mropert.github.io

About this talk

- Cryptography
- Breaking ciphers
- History
- Some Modern C++

RIP Twitter 2006 - 2022(?)





Who would win?

1 The Enigma Machine

German Space Magic

All warfare is based on deception.
-- Sun Tzu

Warfare 101

- Bigger numbers (usually) wins
- Hit the enemy where they don’t expect you
- Radio coordination helps tremendously…
- … unless the other side can listen to you

Machine Specs

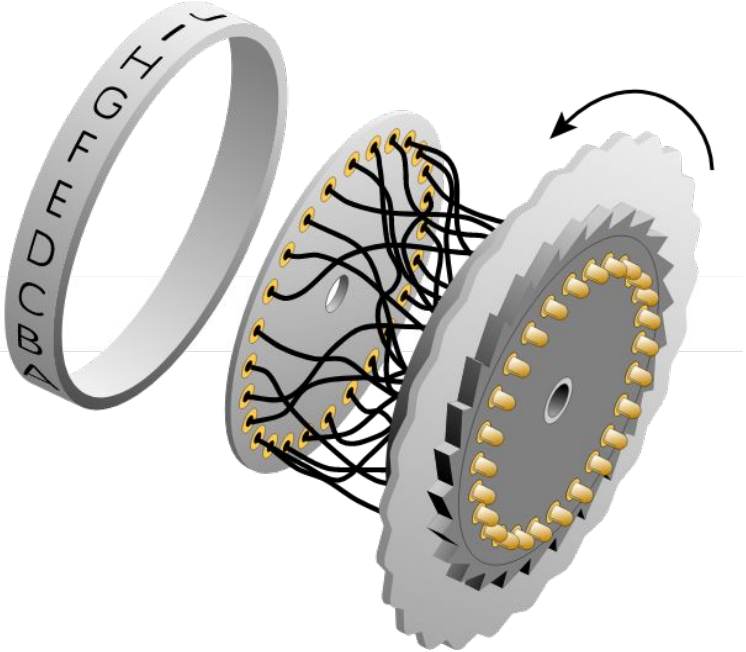
- Invented 1918–1923
- Initially aimed at commercial markets
- Rotors, wires, lamps and a battery

Machine Specs

- Self powered and portable
- Can fit in an armored vehicle or a submarine
- Good fit for enciphering Morse messages

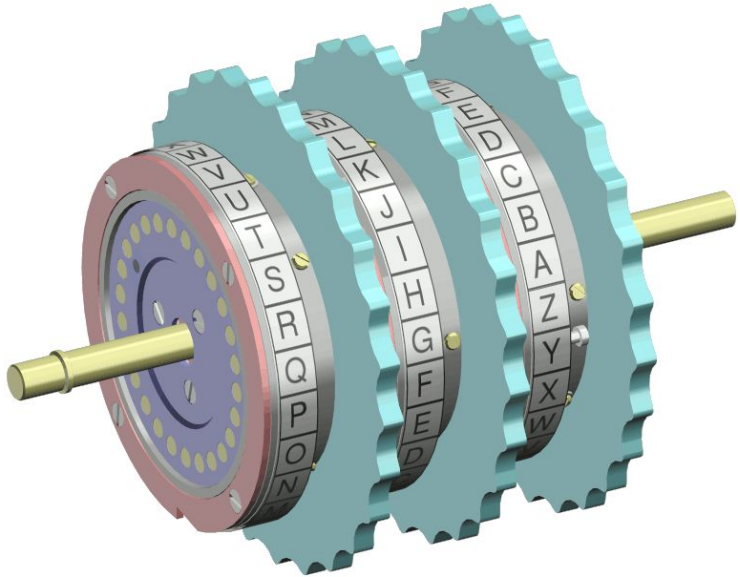
Machine Specs

- Rotors map 26 input pins to 26 output pins 1:1
- Effectively a simple substitution cipher
- Wirings are preset, 8+2 different models (initially 3)



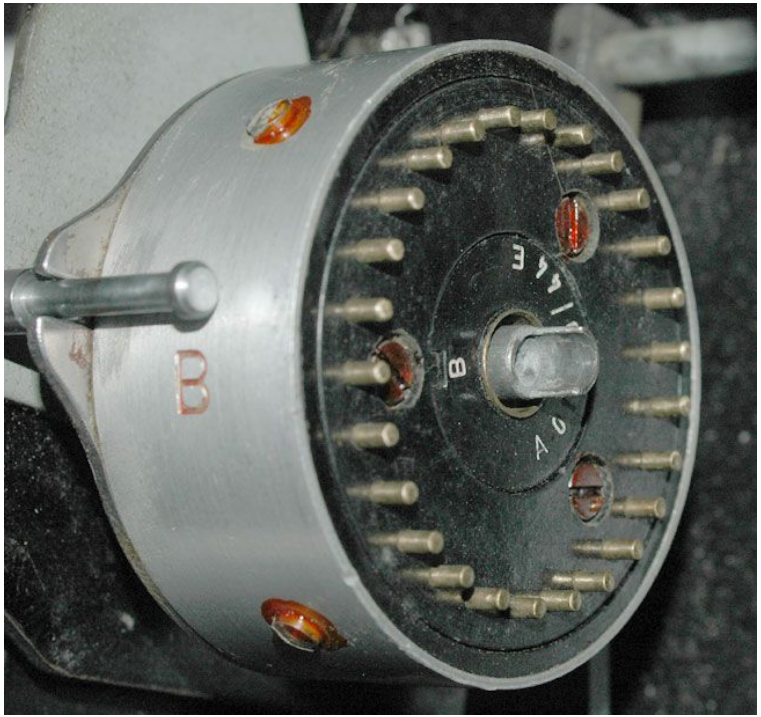
Machine Specs

- Rotors are chained
- Signal sent to one of the 26 input pins at the time
- Keystrokes make rotors turn, changing the input-output map each time



Machine Specs

- No printer (*)
- Leftmost part is a static reflector bouncing signal back through rotors
- Rightmost rotor output signal lights one of 26 lamps

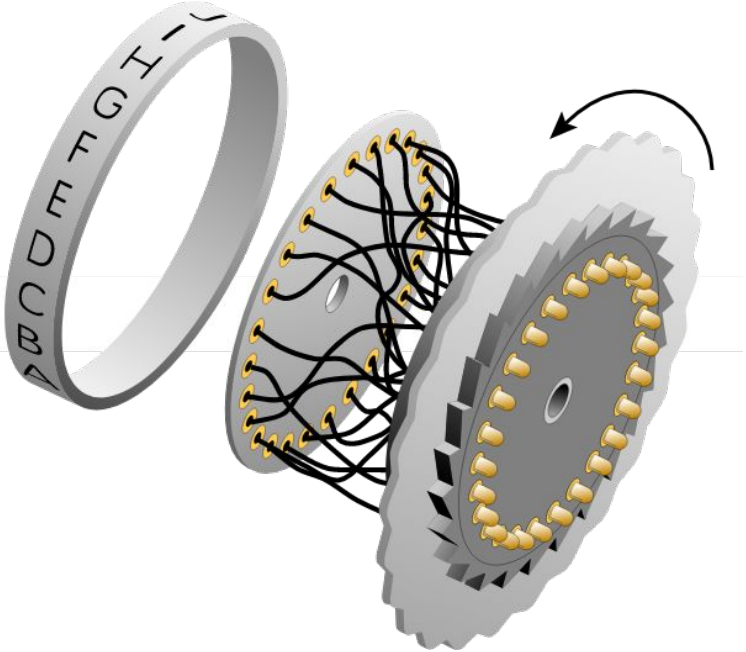


Machine Specs

- 3 rotors in any order => 6
- 26*26*26 initial rotor position (key) => 17 576
- Roughly 105,000 possible combinations
- Not bad for 1920s but...

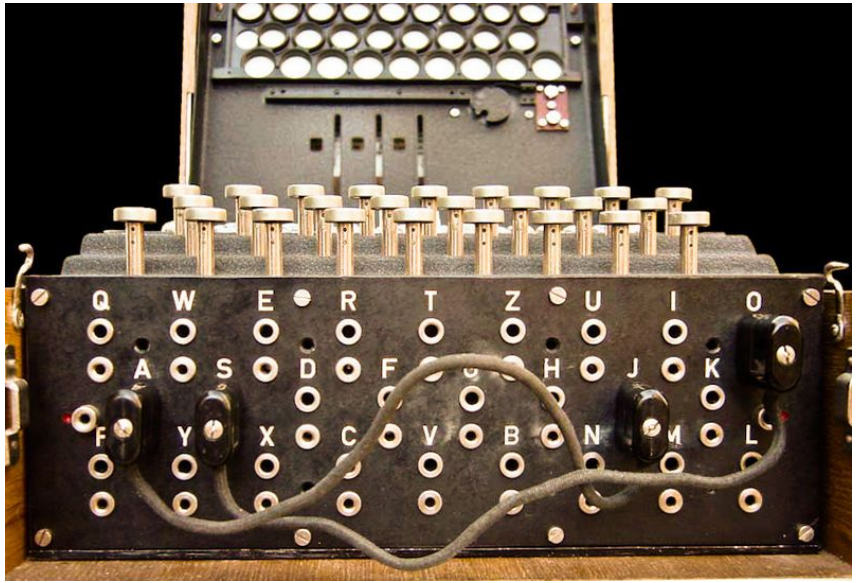
Machine Specs, Revised

- Each rotor has 26 possible ring settings
- Offsets internal wiring by 0–25
- Changes turnover position



Machine Specs, Revised

- Military variant comes with a plugboard
- Remaps input and output letters by pairs
- 6, then 10 pairs are plugged during setup



Machine Specs, Revised

- Add 5 more rotors to the original 3 set
- Shrink the reflector and add a 4th “slim” rotor with 2 sets
- Kriegsmarine M4

Machine Specs, Revised

- 4 rotors out of 10 => 672
- 26^4 initial rotor position (key) => 456 976
- 26^3 rotor ring settings => 17 576
- 10 letter pairs plugged: 150 738 274 937 250

Machine Specs, Revised

- About 8*10^26 combinations
- Roughly 62 bits encryption
- All that with rotating wheels, a few wires, 26 lamps and a mechanical keyboard!

Machine Setup

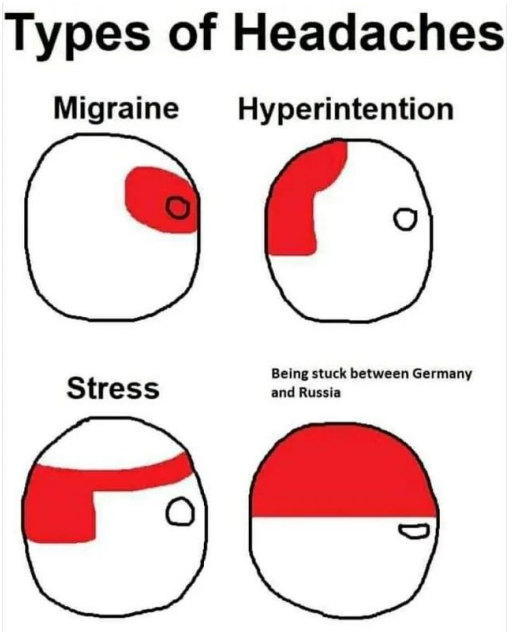
- Install the rotors (changed monthly, later weekly/daily)
- Set ring settings
- Set plugboard

Machine Setup

- Set rotors position to the daily key
- Choose a message key and type it, write the output down
- Reset rotor position to the message key
- Type message and append the enciphered key

2 Cracking Enigma during WW2

How to break 60 bits encryption with math and a few pencils



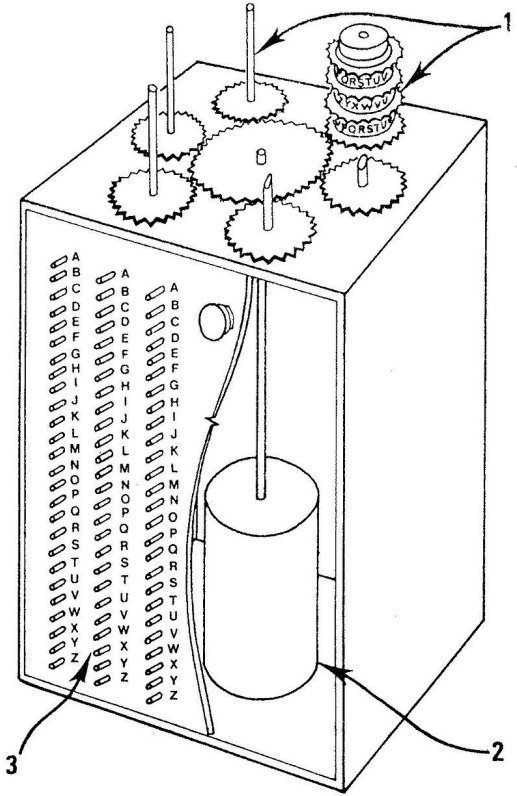
A time before Turing

The Polish Cipher Bureau

- Marian Rejewski (1905–1980)
- Reverse engineered rotors I–V wiring using examples from sales documentation
- Designed the original *bomba*
- Shared with Allies in July 1939

The Polish Cipher Bureau

- German radio operator procedure repeated the message key twice
- ```
  msg[0] == msg[3] &&
  msg[1] == msg[4] &&
  msg[2] == msg[5]
  ```
- Try 17 576 x 6 combinations



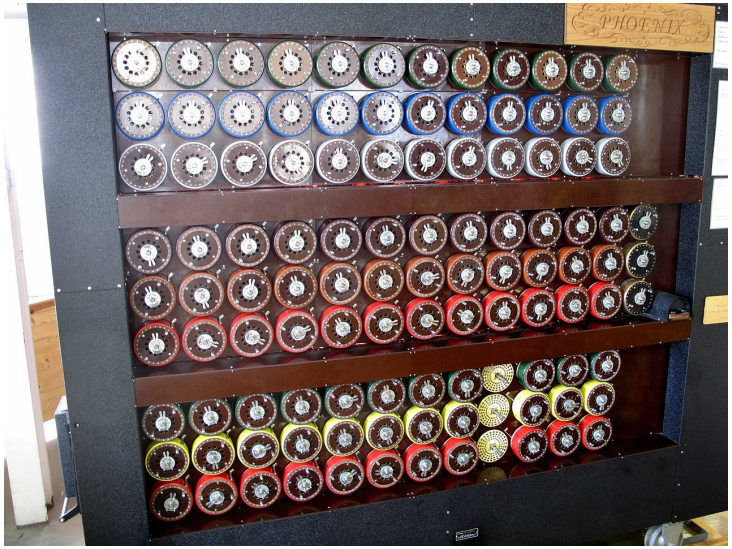
8. Bomba kryptologiczna

(dla przejrzystości ukazano w górnej części bomby tylko jeden zestaw wirników szyfrujących)

1. wirniki,
2. silnik elektryczny,
3. przełączniki
  ```

Bletchley Park

- Turing refined the *Bomba* design into the British *Bombe*
- Message keys stopped repeating in May 1940
- Relied on common formalities and expressions (*cribs*) to detect potential matches



Bletchley Park

- Allies were able to read German ciphers for most of the war
- Decryption took a couple days at worst
- Germany remained convinced Enigma was *theoretically* breakable but too costly to do in practice

3

Cracking Enigma today

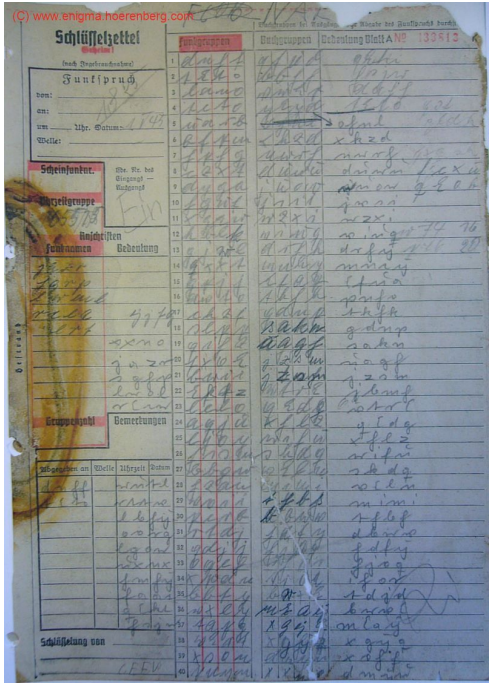
Work harder, not smarter

Data Set

- U534 sunk on May 5th, 1945
- Discovered in 1986
- Raised in 1993
- Contained about 50 encrypted messages, but no keys

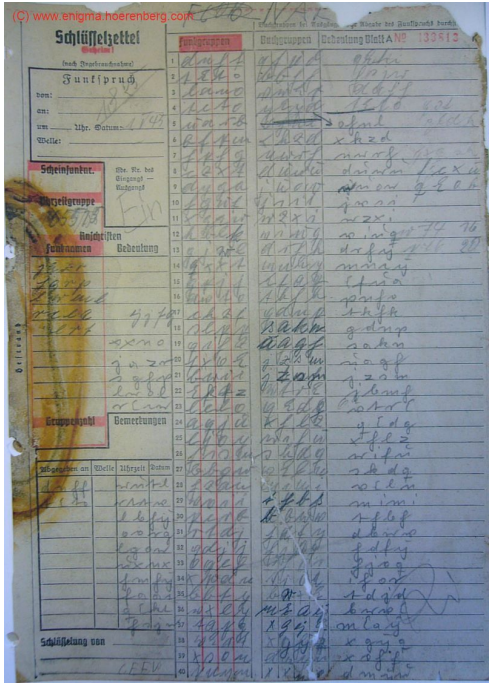
Data Set

- Project setup by Michael Hörenberg in 2012 to crack the messages
- Most have been cracked using Enigma@Home
- Can we do it with a single machine?



Data Set

- Focus on message P1030681
- 372 characters long
- Initially cracked in October 2012



LANOTCTOUARBBFPMHPHGCZXTDYGAHGUFXGEWKBLKGJWLQXX
TGPJJAVTOYJFGSLPPQIHZFXOEBWIIEKFZLCLOAQJULJOYHSSMBBG
WHZANVOIIPYRBRTDJQDJJOQKCXWDNBBTYVXLYTAPGVEATXSON
PNYNQFUDBBHHVWEPYEYDOHNLXKZDNWRHDUWUJUMWWVIIW
ZXIVIUQDRHYMNCYEFUAPNHOTKHKGDNPSAKNUAGHJZSMJBMHV
TREQEDGXHLZWIFUSKDQVELNMIMITHBHDBWVHDFYHJOQIHORT
DJDBWXEMEAYXGYQXOHFDMYUXXNOJAZRSGHPLWMLRECWWUT
LRTTVLBHYOORGLGOWUXNXHMHYFAACQEKTHSJW

KRKRALLEXXFOLGENDESISTSOFORTBEKANNTZUGEBENXXICHHAB
EFOLGENDENBEFEHLERHALTENXXJANSTERLEDESBISHERIGXNREI
CHSMARSCHALLSJGOERINGJSETZTDERFUEHRERSIEYHVRRGRZSSA
DMIRALYALSSEINENNACHFOLGEREINXSCHRIFTLSCHEVOLLMACH
TUNTERWEGSXABSOFORTSOLLENSIESAEMTLICHEMASSNAHMENV
ERFUEGENYDIESICHAUSDERGEGENWAERTIGENLAGEERGEBENXG
EZXREICHSLEITEIKKTULPEKKJBORMANNJXXOBXDXMMMDURNHF
KSTXKOMXADMXUUUBOOIEXKP

KRKRALLEXXFOLGENDESISTSOFORTBEKANNTZUGEBENXXICHHAB
EFOLGENDENBEFEHLERHALTENXXJANSTERLEDESBISHERIGXNREI
CHSMARSCHALLSJGOERINGJSETZTDERFUEHRERSIEYHVRRGRZSSA
DMIRALYALSSEINENNACHFOLGEREINXSCHRIFTLSCHEVOLLMACH
TUNTERWEGSXABSOFORTSOLLENSIESAEMTLICHEMASSNAHMENV
ERFUEGENYDIESICHAUSDERGEGENWAERTIGENLAGEERGEBENXG
EZXREICHSLEITEIKKTULPEKKJBORMANNJXXOBXDXMMMDURNHF
KSTXKOMXADMXUUUBOOIEXKP

P1030681

- Sent on early May 1945
- Announces Grand Admiral Dönitz is now in overall command of Germany
- Germany will surrender only a few days later

Let’s Get Cracking!

Number crunching

- 62 bits brute force would require efforts similar to generating a SHA-1 collision
- Without plugboard, still 5 trillions combinations
- Ignoring leftmost rotor turnover, we get to 208 billions

Algorithm outline

- For every rotor permutation (2*8*7*6)
- For every ring settings (26*26)
- Try every key (26*26*26*26)
- See if we match the plaintext

First attempt

- 2*8*7*6*26*26*26*26*26*26 permutations
- About 208 billions tries on worst case

```
Cracking in progress... 908544/207591401472
(1114872 combinations / second, ETA 3103 minutes)
```

- Roughly 52 hours to exhaust all combinations

Core function

```
input = m_plugboard[ input - 'A' ];

input = m_rotors[ 3 ].m_wiring[ input - 'A' + offsets[ 3 ] + 26 ];
input = m_rotors[ 2 ].m_wiring[ input - 'A' + offsets[ 2 ] - offsets[ 3 ] + 26 ];
input = m_rotors[ 1 ].m_wiring[ input - 'A' + offsets[ 1 ] - offsets[ 2 ] + 26 ];
input = m_rotors[ 0 ].m_wiring[ input - 'A' + offsets[ 0 ] - offsets[ 1 ] + 26 ];

input = m_reflector.m_wiring[ input - 'A' - offsets[ 0 ] + 26 ];
```

Core function

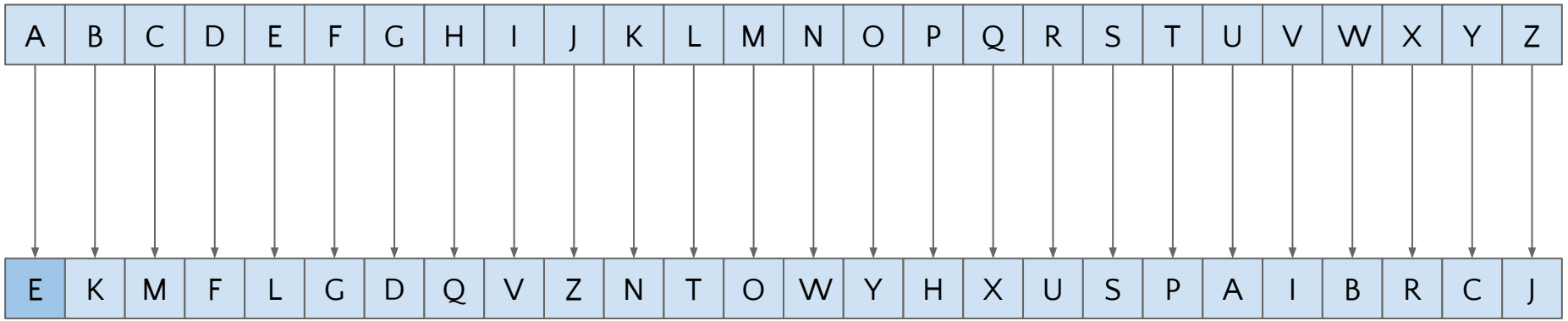
```
input = m_rotors[ 0 ].m_reversed_wiring[ input - 'A' + offsets[ 0 ] + 26 ];
input = m_rotors[ 1 ].m_reversed_wiring[ input - 'A' + offsets[ 1 ] - offsets[ 0 ] + 26 ];
input = m_rotors[ 2 ].m_reversed_wiring[ input - 'A' + offsets[ 2 ] - offsets[ 1 ] + 26 ];
input = m_rotors[ 3 ].m_reversed_wiring[ input - 'A' + offsets[ 3 ] - offsets[ 2 ] + 26 ];

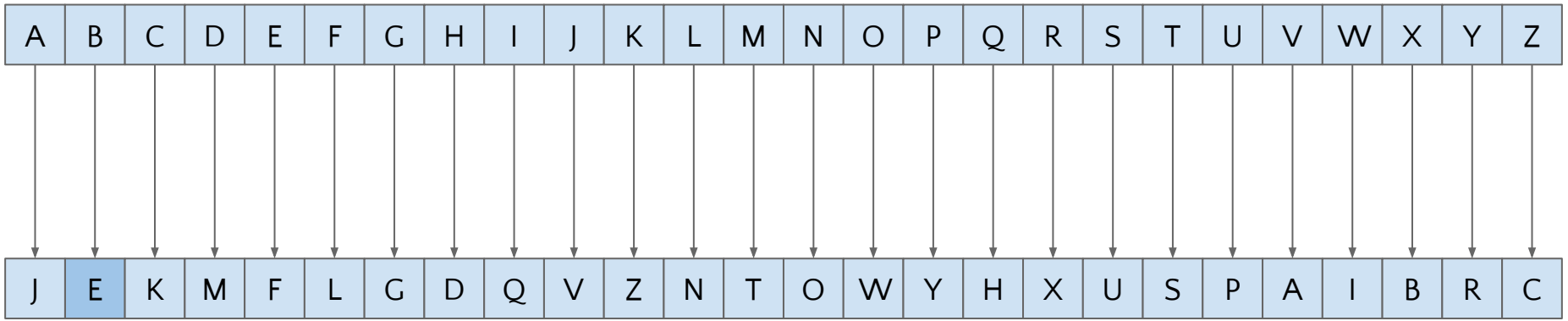
input = enigma::rotors[ static_cast<int>( enigma::rotor_index::ETW ) ]
                            .m_wiring[ input - 'A' - offsets[ 3 ] + 26 ];

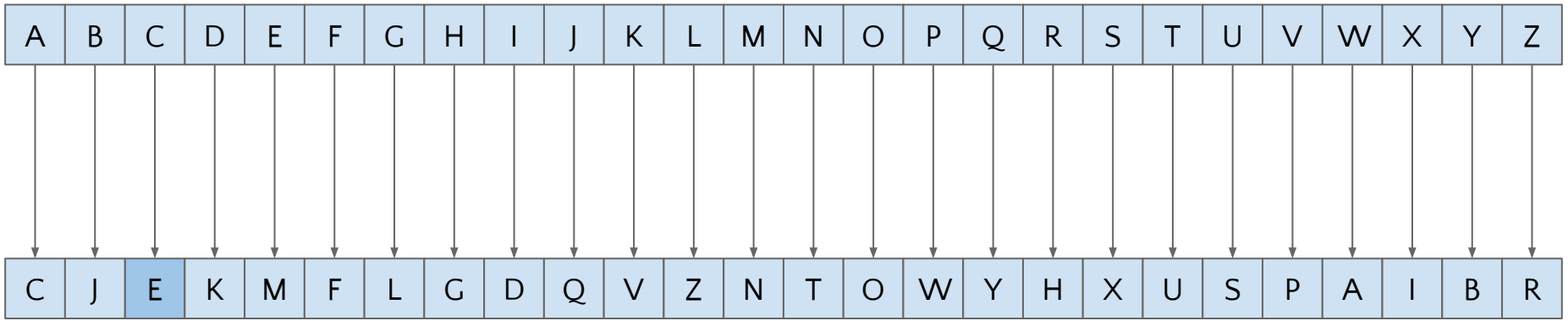
input = m_plugboard[ input - 'A' ];
```

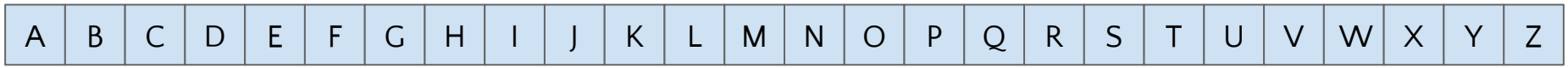
Rotor implementation

- A rotor is a simple 26 character string
- Generate forward and reverse wiring array by repeating the characters 3 times
  - At compile time
- Avoids modulo operation on each rotor

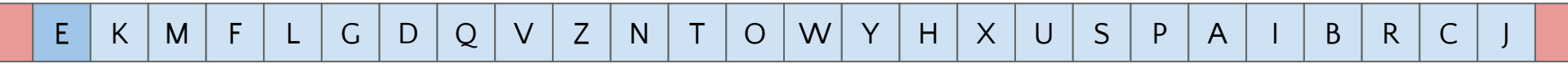

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
E K M F L G D Q V Z N T O W Y H X U S P A I B R C J


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


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




```
output = wiring[ input + offset ];
```



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

```
output = wiring[ ( input + offset ) % 26 ];
```

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

Modulo performance

```
movsx   rax, esi
mov     edx, esi
imul    rax, rax, 1321528399
sar     edx, 31
sar     rax, 35
sub     eax, edx
imul    eax, eax, 26
sub     esi, eax
movsx   rsi, esi
movzx   eax, BYTE PTR [rdi+rsi]
```

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

```
output = wiring[  input + offset + 26 ];
```

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

Modulo performance

```
movsx   rax, esi
mov     edx, esi
imul    rax, rax, 1321528399
sar     edx, 31
sar     rax, 35
sub     eax, edx
imul    eax, eax, 26
sub     esi, eax
movsx   rsi, esi
movzx   eax, BYTE PTR [rdi+rsi]
```

```
movsx   rsi, esi
movzx   eax, BYTE PTR [rsi+26+rdi]
```

Modulo performance

```
movsx   rax, esi
mov     edx, esi
imul    rax, rax, 1321528399
sar     edx, 31
sar     rax, 35
sub     eax, edx
imul    eax, eax, 26
sub     esi, eax
movsx   rsi, esi
movzx   eax, BYTE PTR [rdi+rsi]
```

```
movsx   rsi, esi
movzx   eax, BYTE PTR [rsi+26+rdi]
```

Second attempt

- Multithreading!
- With 16 cores, we should be able to do it in about 3.5 hours
- C++17 makes it easy!

Second attempt

```
std::array<char, 26> values;
std::iota( begin( values ), end( values ), 0 );

std::for_each( std::execution::par_unseq,
                  begin( values ),
                  end( values ),
                  [ & ]( char right_ring_setting )
              { /* ... */ } );
```

Third attempt

- 2nd rotor is unlikely to turn much
  - Every 169 keystrokes at best, 676 at worst
- Assume 3rd rotor ring setting is 0, match if decrypt is at least 1/4 right
- Fine tune the ring settings afterwards

Third attempt

- Try 26*26 ring settings once partial match is found
- Reduces search space to 8 billions combinations

```
Cracking in progress... 14990976/7984284672
(20449310 combinations / second, ETA 6 minutes)
```

But what about the plugboard?

Defeating the plugboard

- There are 150 trillions possible plugboard setups
- Brute force ain’t gonna cut it
- Can we split the problem?

Defeating the plugboard

- With an empty plugboard, 6 letters will be correct anyway
- Brute force with empty plugboard combination
- Count character matches by letter, sum the top 6 matching letters

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
4	1	0	2	6	0	4	0	5	0	1	1	1	1	6	0	0	1	4	2	0	0	0	1	0	0

```
{ "AE", "BF", "CM", "DQ", "HU", "JN", "LX", "PR", "SZ", "VW" };
```

Defeating the plugboard

- After tweaking, the heuristic is able to filter out false positives if `score >= length / 10`
- Rotor order, ring settings and key are broken in the same time with out without knowing plugboard.
- Can we do *better*?

Final touch

- *Ignore* the ring settings, even rightmost rotor
- Turnover will be wrong, but enough fragments will stay intact to distinguish from random
- Fine tuning only requires one 26*26*26 pass

Final touch

```
Cracking in progress... 16451136/307087872 (21410896 combinations / second, ETA 0 minutes)
Cracking in progress... 36558080/307087872 (27888838 combinations / second, ETA 0 minutes)
Cracking in progress... 58492928/307087872 (32043434 combinations / second, ETA 0 minutes)
Cracking in progress... 79056848/307087872 (28903634 combinations / second, ETA 0 minutes)
Cracked message!
- Rotors: 9, 5, 6, 8
- Ring settings: 0, 0, 17, 11
- Message key: YOFZ
```

Last thoughts

- I didn't do the math, the heuristic might not work with all messages
  - Especially shorter ones
- Could index of coincidence work better?
- Purists will argue this isn't *cyphertext-only* cracking

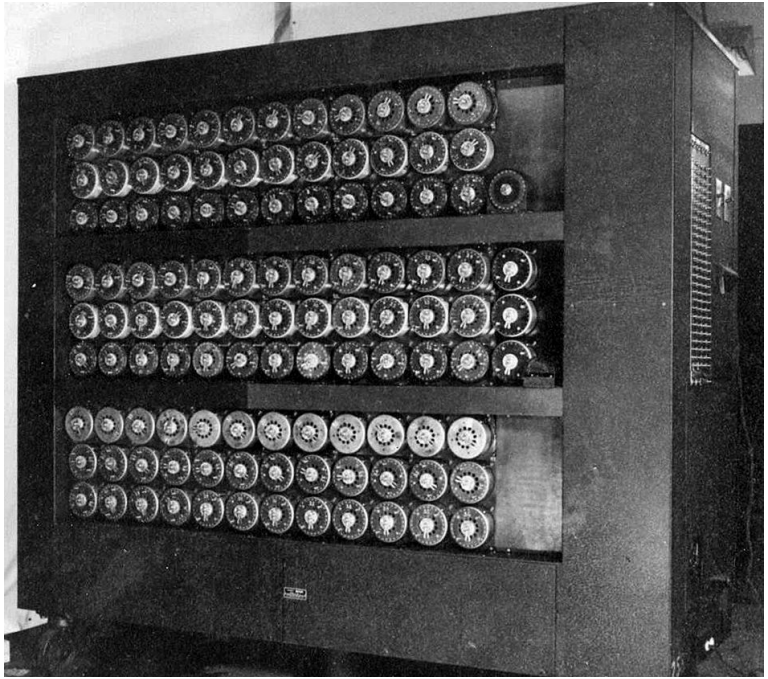
ACCU Bonus Slides

4 A C++ *Bombe* in 2023

C’est de la bombe!

Back in the 40s

- The *Bombes* tried every rotor combination
- Stopped when a potential solution was found
- Required known “crib” plaintext



S	M	B	B	G	W	H	Z	A	N	V	O	I	I	P	Y	R	B	R	T	D	J	Q
R	E	I	C	H	S	M	A	R	S	C	H	A	L	L	*	*	*	*	*	*	*	*
*	R	E	I	C	H	S	M	A	R	S	C	H	A	L	L	*	*	*	*	*	*	*
*	*	R	E	I	C	H	S	M	A	R	S	C	H	A	L	L	*	*	*	*	*	*
*	*	*	R	E	I	C	H	S	M	A	R	S	C	H	A	L	L	*	*	*	*	*
*	*	*	*	R	E	I	C	H	S	M	A	R	S	C	H	A	L	L	*	*	*	*
*	*	*	*	*	R	E	I	C	H	S	M	A	R	S	C	H	A	L	L	*	*	*
*	*	*	*	*	*	R	E	I	C	H	S	M	A	R	S	C	H	A	L	L	*	*
*	*	*	*	*	*	*	R	E	I	C	H	S	M	A	R	S	C	H	A	L	L	*
*	*	*	*	*	*	*	*	R	E	I	C	H	S	M	A	R	S	C	H	A	L	L

Crib Match Strategies

- For each rotor order + key combination, check if any crib location partially decodes
- For potential matches, try tweaking ring settings (26x26)
- Crib needs to be long enough to avoid large number of false positives

Templatize!

```
std::optional<m4_solver::settings> crack_settings(
                                    std::string_view message,
                                    reflector reflector,
                                    std::span<const char* const> plugs )
```

Templatize!

```
template <typename heuristic_type,
          typename score_type,
          typename validate_type>
std::optional<m4_solver::settings> crack_settings(
                                    std::string_view message,
                                    reflector reflector,
                                    std::span<const char* const> plugs,
                                    const heuristic_type& heuristic,
                                    const score_type& score,
                                    const validate_type& validate )
```

Crib Match Strategies

- A random decrypt has 1:26 chance to yield the correct letter
- Two letters: 1:676, three letter: 1:17’576
- Give an exponentially higher score to consecutive letter matches

More Optimizations

- Ring setting exploration can be reduced to linear (rather than quadratic) search
- Score all 26 rightmost ring position, pick best
- Try all 26 middle-right combinations until you get an (almost) perfect match

More Optimizations

- Compute time is a function of the message length
- For cribs with few possible positions, try breaking only that exact fragment, then “rollback” the key to the message start

More Optimizations

- For ex, if we suspect who sent the Dönitz message, we can search for a formal signature in the last quarter of the cyphertext

- `XGEZXREICHSLEITEIKKTULPEKKJBORMANNJXX`
  - “Signed, Reichsleiter (Tulpe) 'Bormann'”

Crib cracking

```
Cracking message of 93 characters using crib
XGEZXREICHSLEITEIKKTULPEKKJBORMANNJXX (12 potential locations) with 20 threads

Cracking in progress... 21934848 / 307087872 (3099017 combinations / second,
830691 false positives) ETA 92 seconds

...

Cracking in progress... 86825440 / 307087872 (4580698 combinations / second,
3282282 false positives) ETA 48 seconds

Cracked message!

- Rotors: 9, 5, 6, 8

- Ring settings: 0, 0, 4, 11

- Message key: YOSZ
```

Last thoughts?

- Still isn't *cyphertext-only* cracking, but closer
- Cyphertext-only attack would use a large dictionary of common 2-4 letters cribs
- What about index of concidence?

Coincidence?

$$\mathbf{IC} = c \times \left(\left(\frac{n_\mathrm{a}}{N} \times \frac{n_\mathrm{a} - 1}{N - 1} \right) + \left(\frac{n_\mathrm{b}}{N} \times \frac{n_\mathrm{b} - 1}{N - 1} \right) + \cdots + \left(\frac{n_\mathrm{z}}{N} \times \frac{n_\mathrm{z} - 1}{N - 1} \right) \right)$$

- Invented in 1922 by William F. Friedman
- Are letters uniformly distributed in the text?
- Natural language doesn't follow uniform distribution

IC in the wild

- Uniform random distribution IC: 1.0
- English: 1.73
- German: 2.05
- Dönitz message: 1.56

IC in practice

```
std::optional<m4_solver::settings> m4_solver::crack_settings( ... )
{
     const auto match_heuristic = []( std::string_view candidate )
     {
          return index_of_coincidence( candidate ) >= 1.2f;
     };

     ...
}
```

IC in practice

```
Cracking message of 372 characters with 20 threads
Cracking in progress... 21477872 / 307087872
(3435360 combinations / second, 0 false positives)
ETA 83 seconds
...
Cracking in progress... 294749520 / 307087872
(4416520 combinations / second, 0 false positives)
ETA 2 seconds
*** FAILED TO CRACK ENIGMA SETTINGS ***
```

IC shortcomings

- Dönitz message: 1.56
- With wrong middle ring setting: 1.21
- With wrong right ring setting: 0.99 😢
- Requires x26 more computations to match

Last thoughts!

- I still didn't do the math, there has to be a better heuristic out there
- In "Modern Breaking of Enigma Ciphertexts" (2017), authors claim pure IC can find rotors, rings and key even with wrong plugboard
- Couldn't reproduce (IC 0.99 with wrong plugs)

C++ Features Usage

In the end, what did we actually use?

Features used

- C++14 constexpr and C++20 constexpr algorithms
- C++17 string_view and C++20 span
- C++17 parallel algorithms
- C++20 format

Features considered

- C++20 Ranges to iterate rotor combinations
- Or maybe Coroutines? 🤔
- C++23 `println` 😠

5 Wrapping Up

Have you been listening closely?

In conclusion

- Enigma is *not* a recommended cipher in 2023
- A true 60 bits cipher will resist pure dumb brute force, even on modern desktop
- Some C+17/20 features will find their way in even the smallest hobby project immediately

Thanks!

Any questions ?

You can reach me at

- mro@puchiko.net
- @MatRopert
- @mropert
- https://mropert.github.io

One Question Remains

What happened to Rejewski and the Polish Cipher Bureau?

The Story Isn't Over

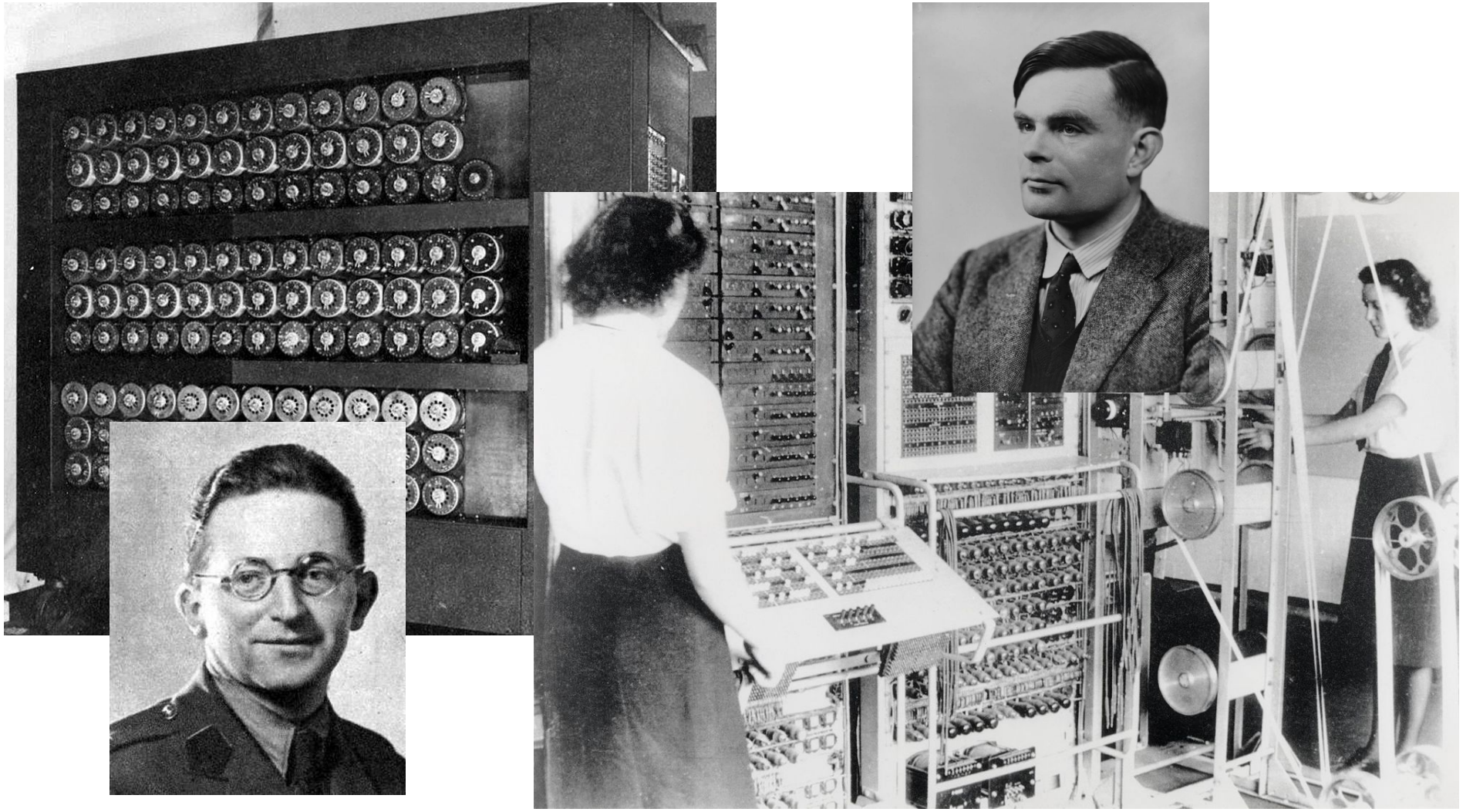
- The Polish Cipher Bureau shared their findings with France and England in July 1939
- Allies spent lots of effort breaking Enigma during the whole of WW2
- Why haven't we heard more from Rejewski?


DENMARK
Copenhagen
LITHUANIA
Kaunas
GERMAN REICH
Berlin
Warsaw
POLAND
London
NETHERLANDS
Amsterdam
BELGIUM
Brussels
Luxemburg
FRANCE
Paris
SLOVAKIA
Bratislava
HUNGARY
Budapest
Bern
SWITZERLAND
KINGDOM OF ROMANIA
Bucharest
YUGOSLAVIA
Belgrade


GERMAN REICH
SWITZERLAND
HUNGARY
SLOVAKIA
KINGDOM
YUGOSLAVIA
KINGDOM OF ITALY
PORTUGAL
NATIONALIST SPAIN
IRELAND


GERMAN REICH
London
Berlin
Bratislava
Bern
Vichy
VICHY FRANCE
KINGDOM OF ITALY
Roma
Madrid
Lisbon

But what if... ?



In Another Universe

- `const` is the default
- No such thing as “Ill-formed, no diagnostic required”
- Coroutines shipped in the 90s, with library support


Franz Ferdinand doesn't linger in Sarajevo in June 1914
Rejewski teams up with Turing at Bletchley Park

We're Hiring

https://career.paradoxplaza.com/

Furthermore

Furthermore, I think your build should be destroyed

Thanks!

Any questions ?

You can reach me at

- mro@puchiko.net
- @MatRopert
- @mropert
- https://mropert.github.io