SSN LAB ASSIGNMENT: ENIGMA

ALI ABDULMADZHIDOV

26 Oct 2016

1. ENIGMA

1.1 Encryption

My plaintext was

> "WhenYouPlayTheGameOfThronesYouWinOrYouDieThereIsNoMiddleGround"

and my birthday is at 03 March.

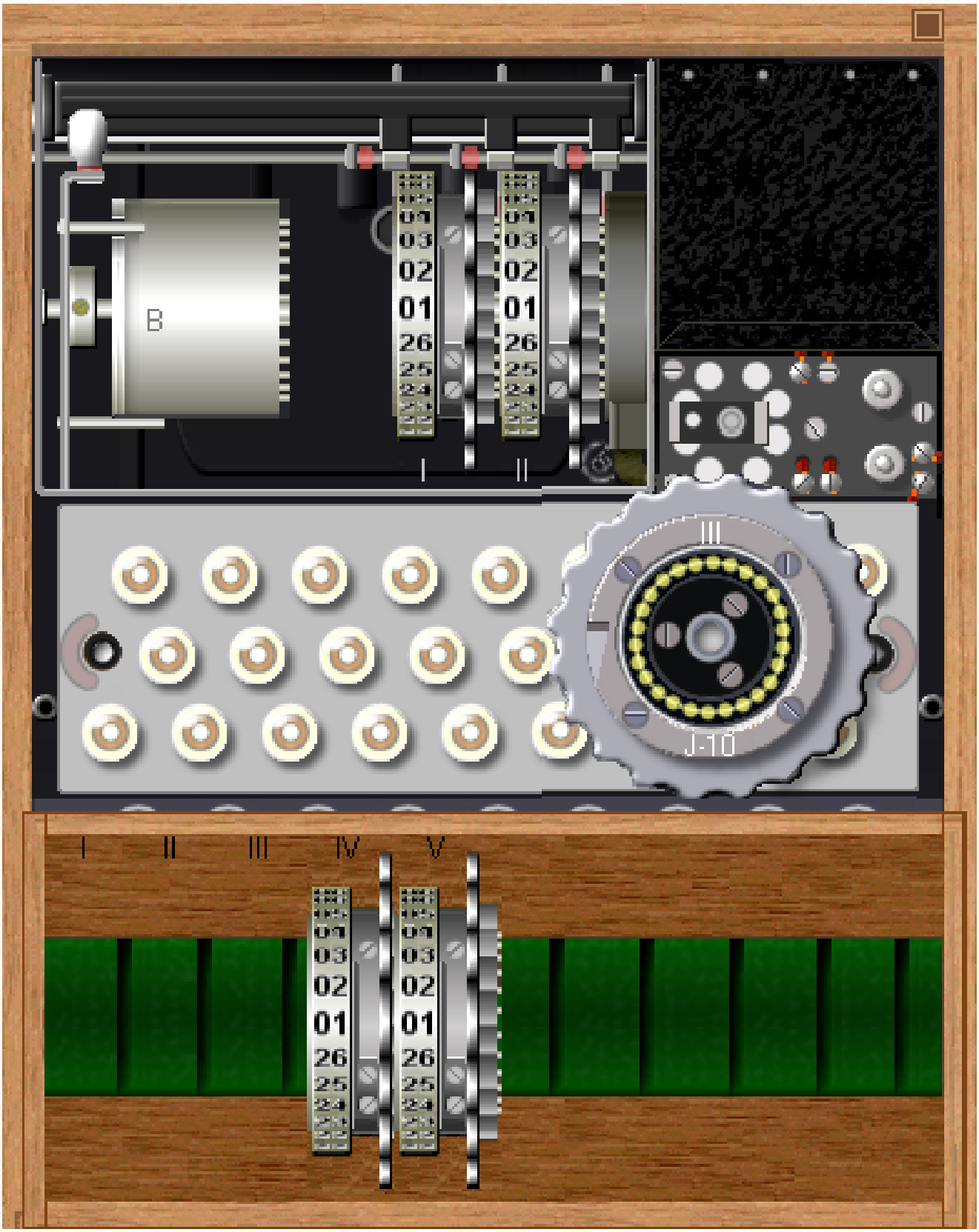
I used **Wermakht UKW B** reflector and rotors **III I II**. Then i setup ring settings on them *10,16,06*. Also i stick all letters in plugboard as it written in codebook. Then i put rotor start position to first key "*NZY*" and encrypted second key "*MID*" result "*DDW*" was putted at the begining of ciphertext for checking all settings when message will be decoded.

After all i got such encrypted message, that i'll send with my birthday date to my neighbor:

> "DDWP EUURD HAZKT XCFLM IREZG PALBL CBXHY FHEJM TNVRV OWSXA OPPVQ YVKZZ KUGBGU"

The final data consist of ciphered text, and trigram at start, that is needed for checking of settings when this would be decrypted.

I don't need to send kengruppen and other values, cause we have key sheets, where are kenngruppens for every date. I only use first triram of kenngruppen as start position of rotors and second for checking purposes (encrypt it, and then recipient after decryption of first 3 symbols can compare them with other kenngruppens from keysheet.)


B
04
03
02
01
26
25
24
I
II
III
J-10
I
II
III
IV
V


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

MIDWH ENYOU PLAYT HEGAM EOFTH RONES
DDWPE UURDH AZKTX CFLMI REZGP ALBLC

1.2 Decryption

I got message:

> BD: 27.01
> JGAJCBAH QDQVD TOEPW GSQJZ RPDFC HUMZB OHCTW KCFSQ VHYBK

I found settings and set up my enigma.

Rotors - *III V IV* with **ring settings** *25 14 22* and **plugs** *AF BQ CK DJ EP GI LY MN RV TU*.

Also i put rotor settings to "**DOG**" and tried to decode first 3 symbols. Result was "**TBF**" second key from book. That means that our machine is correct. Then we reset our rotor setting to "**DOG**" and start decoding message:

> the only truly secure system is one that is powered off

---

2. VIOLA

There is maximum variants of keys for **Viola**:

$$5 * (30^{20}) * 37276043023296000 * 6190283353629375 =$$

$$= 402286475176704820302504069968261460000000000000000000000000000$$

, where 5 - number reflectors, 30^{20} variations (30^2 variations for each rotor (ring settings and rotor settings)), 37276043023296000 variations to put 10 unique rotors from 50 and 6190283353629375 variations of plugboard with 30 elements and 15 pairs.

```
1  def alphabet_variations(n,m):
2  ...  return float(math.factorial(n))/(math.factorial(n-
   2*m)*math.factorial(m)*2**m)
```

Below is maximum variants of keys for **Enigma**:

$$3 * (26^6) * 60 * 150738274937250 = 83817776115255483100800000$$

, where 3 numbers of reflectors, 26^6 variations of each rotor setting and ring setting, 60 variations of 3 rotor from 5, 150738274937250 ways to connect 26 letters with 10 connectors by formula, that i wrote in python above.

Viola, of course, is much more resistant for bruteforce attack, because it has more varitations. That is so mostly, because we have more rotors and also more rotors that take part in encryption.

Formula:

Number of reflectors * Size of alphabet ^ number of rotors * 2 (cause we have rotor and ring settings for each of them) * number of variations to take number of rotors that we need from all rotors * to all variations of connections on plugboard.

Ringsetting is already added to calculation