SIGABA

Source: <http://www.cs.sjsu.edu/faculty/stamp/students/Sigaba298report.pdf>

* Cipher and control rotors permute one letter of alphabet to another letter
* Index rotors permute one digit to another digit
* 15 rotors
  + 5 cipher (26 contacts on two faces)
  + 5 control (26 contacts on two faces) - Start of encryption mode
  + 5 index (10 contacts)

Encryption:

Note: all Z’s must become X’s and all spaces are changed to Z’s before encryption

1. Plaintext is entered using keyboard
2. Signal is generated that is sent to two banks:
   1. First signal sent to left of cipher rotor bank
      1. Permuted through 5 cipher rotors to produce cipehrtext
   2. Second signal sent to right of control rotor bank
      1. Used to energize the inputs (‘F’, ‘’G’, ‘H, ‘I’)
      2. These inputs are permuted through control rotors right to left
      3. After all get to the left, they go to ORing & determine which inputs for index rotor bank are energized
3. One to three of the control rotors will step forward
   1. From the left to the right: fast control is the third, the medium is the fourth, and the slow is the second (all in the control bank)
   2. Fast steps once every time a letter is keyed, medium steps once every time the fast passes O, and the slow steps every time the medium passes O
   3. First and fifth rotors do not remain fixed in encryption
4. Based on ORing of control rotor bank’s output, 1 to 4 of the index rotor bank’s inputs will be energized
   1. Active signals are permuted by index bank in left to right fashion.
   2. Outputs of index rotor are ORed again (in a different manner) to determine which cipher rotor should step

Source 2: <https://github.com/JoeDunnStable/sigaba/tree/master/sigaba>

Main.cpp:

* Ask the user what they want to do
* Grabs input text
* Call cycle() in header file
* Make the output pretty

Sigaba.cpp:

* Set the default arrays for rotors and other machine settings

Sigaba.h: start at the String cycle() function:

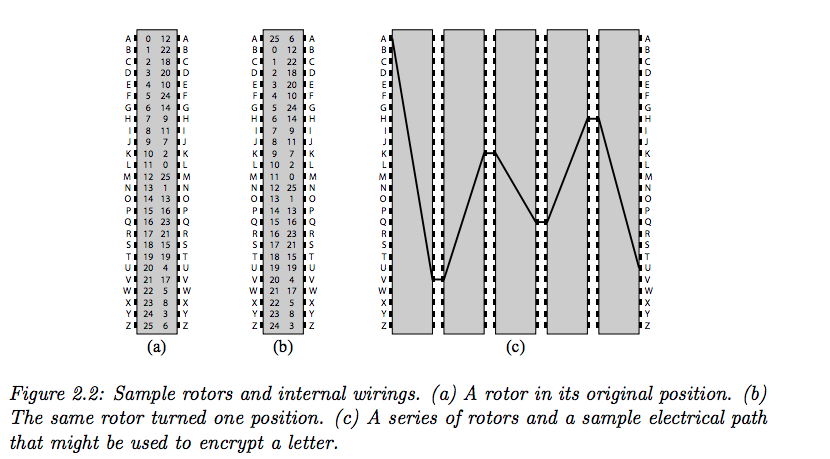
* Filler\_in will modify the plaintext (replace spaces and Z’s)
* Encrypts a single letter
* Steps the cipher rotors
* Steps the control rotors
* Filler\_out will return the ciphertext

To encrypt a single letter: (check to see what right and left is)

|  |  |
| --- | --- |
|  | int encrypt(int in ///< integer to encrypt from 0-25 |
|  | ) { |
|  | if (reversed) |
|  | return mod(pos - right[mod(pos - in,N)], N); |
|  | else |
|  | return mod(left[mod(in + pos,N)] - pos, N); |
|  | } |

Thesis which goes over sigaba : <https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=4626&context=etd_theses>

* Has three banks, each consisiting of 5 rotors
  + https://en.wikipedia.org/wiki/SIGABA
  + Bank 1: Cipher Rotots
    - Each have 26 contacts, when a plaintext letter was entered, a signal would enter one side of the bank and exit the other, denoting the ciphertext letter
  + Bank 2: Control Rotors
    - Each have 26 contact rotors. Each received four signals at each step. After passing through the control rotors the outputs were divided into ten groups of various sizes, ranging from 1-6 wires. Each group corresponded to an input wire for the index rotors
  + Bank 3: Index Rotors
    - Each have 10 contacts. Do not step during encryption. After traveling through index rotors. One to four of five output lines would have power. These turn the cypher rotors.
* Rotor Configuration
  + Cipher and Control rotors can be interchanged with each other. Can also be inserted into the cage in a reversed orientation
  + When inserted in normal position, letters are displayed rightside up in a reverse alphabetic order. When a rotor steps, the rotors will rotate UP the alphabet (O would become N).



Psuedo Code, possible help with decryption

<https://pdfs.semanticscholar.org/b401/54ba54acbffe536d577d1861a892278f2d84.pdf>