Table of Contents

Chinese Gold Bar (CGB) Ciphers

The most notorious cipher devices in the WWII era would be German Army's Enigma machines. It was at the center of cipher warfare and deciphering encrypted messages of German army in time was detrimental to winning the war. When the CGB cryptograms were glanced and examined, it gives a strong resemblance of messages or cipher texts that would have been generated by those Enigma machines. Even though German's wartime Enigma machines were not responsible of generating the CGB cryptograms, there had been several variations of Enigma machines available for commercial applications based on the principles of German's version in 1930s.

Swiss-K Enigma machine was one of commercially available Enigma machines at that time. Switzerland has been accepted as a neutralized nation and banks in the nation were open and accessible without conflictions to any political parties in the world and the secrecy of depositors from all over the world has been safely kept under. The existence of the Swiss version of Enigma machine tells that the records of banking transactions conducted by and within banks might have been encrypted safely and hidden from any accidental exposure to outsiders. It is a reasonable assumption to make that if the CGB ciphers were generated by Enigma machine, Swiss-K Enigma machine would be a sounding choice taken by the National City Bank in China for the transaction.

Cryptograms of CGB Ciphers

Cryptograms inscribed on the gold bars can be found at IACR and are replicated below. Among them, the cipher "HLMTAHGBGFNIV" in the original transcriptions was changed to "MLMTAHGBGFNIV" based on its decryption result. The first cipher letter should be not "H", but "M" and it is interpreted as "A".

Gold Bar (with text only)	KO 08124	KO 00803 (or 00808)
JKGFIJPMCWSAEK	(LEFT)	(MIDDLE)
SKCDKJCDJCYQSZKTZJPXPWIRN		
MQOLCSJTLGAJOKBSSBOMUPCE	MQOLCSJTLGAJOKBSSBOMUPCE	UGMNCBXCRLDEY
FEWGDRHDDEEUMFFTEEMJXZR	FEWGDRHDDEEUMFFTEEMJXZR	
RHZVIYQIYSXVNQXQWIOVWPJO	YQHUDTABGALLOWLS	RHZVIYQIYSXVNQXQWIOVWPJO
MQOLCSJTLGAJOKBSSBOMUPCE	VIOHIKNNGUAB	SKCDKJCDJCYQSZKTZJPXPWIRN
FEWGDRHDDEEUMFFTEEMJXZR	MVERZRLQDBHQ	MQOLCSJTLGAJOKBSSBOMUPCE
SKCDKJCDJCYQSZKTZJPXPWIRN	ZUQUPNZN	FEWGDRHDDEEUMFFTEEMJXZR
RHZVIYQIYSXVNQXQWIOVWPJO	VIOHIKNNGUAB	
MQOLCSJTLGAJOKBSSBOMUPCE	JKGFIJPMCWSAEK	(RIGHT)
SKCDKJCDJCYQSZKTZJPXPWIRN	SKCDKJCDJCYQSZKTZJPXPWIRN	
		VIOHIKNNGUAB
(UPSIDE DOWN)	(RIGHT)	HFXPCQYZVATXAWIZPVE
		YQHUDTABGALLOWLS
MLMTAHGBGFNIV	MLMTAHGBGFNIV	XLYPISNANIRUSFTFWMIY
ZUQUPNZN ABRYCTUGVZXUPB	FEWGDRHDDEEUMFFTEEMJXZR	KOWVRSRWTMLDH
MVERZRLQDBHQ	RHZVIYQIYSXVNQXQWIOVWPJO	JKGFIJPMCWSAEK
GKJFHYXODIE UGMNCBXCRLDEY	HFXPCQYZVATXAWIZPVE	ABRYCTUGVZXUPB
HFXPCQYZVATXAWIZPVE	KOWVRSRWTMLDH	GKJFHYXODIE
		ZUQUPNZN
		GKJFHYXODIE

The longest cipher is "SKCDKJCDJCYQSZKTZJPXPWIRN" and it has 25 letters. The shortest cipher is "ZUQUPNZN" and it is 8 letters long.

Swiss-K Enigma Machine Simulator in Python on Jupyter Notebook

It is very difficult to get a hand on and use a real life Swiss-K Enigma machine in solving the CGB ciphers. Instead, a virtual Swiss-K Enigma machine written in Python will be used. Jupyter notebook is a powerful tool to run the simulator on.

The codes in this project have been cloned from pyEnigma on GitHub and they can be found here: https://github.com/cedricbonhomme/pyEnigma.

How Enigma machine works can be found here: https://en.wikipedia.org/wiki/Enigma_machine.

The wiring details used in Enigma machines can be found here: https://www.cryptomuseum.com/crypto/enigma/wiring.htm.

Basically, an Enigma machine substitutes the letter entered on the key board with another letter as an output result. After the substitution was completed, a light bulb associated with the output letter is lit up on the look up panel and rotors are rotated into new positions waiting for next inputs and in that way the machine substitutes next input letter differently than before. Even if a same letter was entered repeatedly, output letter changes every time. The initial start positions for rotors can be set up differently at the start and even in the fly, if wanted. A rotor associates an input letter with an output letter through internal wiring. Each rotor has its own 26 internal wirings and they are reconfigurable before use. Sometimes more than one rotor can advance their positions depending on the settings of notches. In the software simulator, several parameters have been defined to emulate the behavior of real Enigma machine.

- Rotor Wiring
- KEY
- NOTCH
- RING
- Plug Board

Swiss-K Enigma machine uses three rotors, one reflector, and one entry disk.
Unless altered by end user, Swiss-K Enigma machine is shipped out with the following default settings.

Rotor	**ABCDEFGHIJKLMNOPQRSTUVWXYZ**	**Notch**
ETW	QWERTZUIOASDFGHJKPYXCVBNML	
I	LPGSZMHAEOQKVXRFYBUTNICJDW	G
II	SLVGBTFXJQOHEWIRZYAMKPCNDU	M
III	CJGDPSHKTURAWZXFMYNQOBVLIE	V
UKW	IMETCGFRAYSQBZXWLHKDVUPOJN	

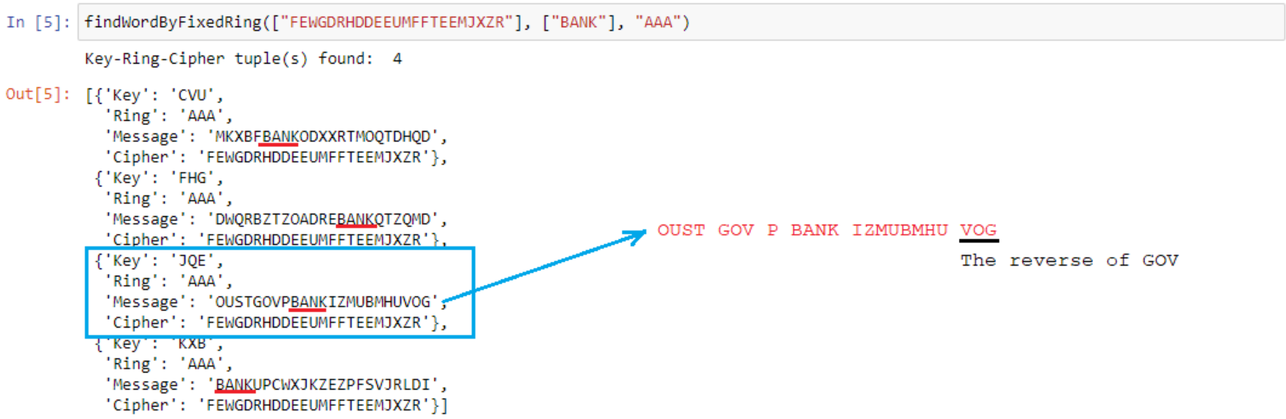
The simulator written in Python uses the factory settings without any customization. In other words, it is assumed that the City Bank did not alter any default settings including plug board settings when the CGB ciphers were generated. With those factory settings intact, there are numerous ways to set the runtime configuration of KEY, RING, and NOTCH to hide information in ciphers. Without knowing exact settings of KEY, RING, and NOTCH, it is extremely difficult to decode the ciphers and restore original texts. It is true that altering the factory configuration can provide with a higher degree of complexity to the encrypted messages, but it also would be a burden to keep the record of configuration changes and runtime settings straight for later decryption because the counterpart, i.e. the depositor in case of CGB ciphers, also should have the same information on the ciphers they are about to decipher.

Furthermore, it is assumed that the NOTCHs were unchanged as well. It leaves us that KEY and RING settings are only changeable parameters involved in the CGB ciphers. Each cryptogram in the CGB ciphers would have had different combination of KEY and RING settings when generated.

The First Crack: A Pure Luck

The longest CGB cipher consists of 25 alphabetic letters. The shortest cipher is 8 letters long. There are all 26 alphabet letters on the rotor of Swiss-K Enigma machine. If the starting KEY value on the first rotor was “G”, the only moving rotor would be the first one during the encryption and the remaining rotors won't move at all. When the encryption crosses "F" of the KEY of the first rotor, the first and second rotors will move one notch. As the second rotor has to move, it will happen only once at most since none of CGB ciphers exceeds 26 letters. Since the third rotor never moves for the CGB ciphers, both third rotor and the reflector can be combined into one new reflector for its wiring. In case that the second rotor doesn't move, both second and third rotors and the reflector altogether can be replaced with a single reflector.

There is no objective reason on why I decided to start with FEW23 cipher: "FEWGDRHDDEEUMFFTEEMJXZR". With the RING set to "AAA", FEW23 cipher was decrypted by varying the KEY from "AAA" to "ZZZ" and it generated 17,576 texts. I ran a string match operation on each output to see if the output contains the word "BANK" in it. The following is an excerpt of the output.



```
In [5]:  findWordByFixedRing(["FEWGDRHDDEEUMFFTEEMJXZR"], ["BANK"], "AAA")

         Key-Ring-Cipher tuple(s) found:  4

Out[5]:  [{'Key': 'CVU',
           'Ring': 'AAA',
           'Message': 'MKXBFBANKODXXRTMOQTDHQD',
           'Cipher': 'FEWGDRHDDEEUMFFTEEMJXZR'},
          {'Key': 'FHG',
           'Ring': 'AAA',
           'Message': 'DWQRBZTZOADREBANKQTZQMD',
           'Cipher': 'FEWGDRHDDEEUMFFTEEMJXZR'},
          {'Key': 'JQE',
           'Ring': 'AAA',
           'Message': 'OUSTGOVPBANKIZMUBMHUVOG',
           'Cipher': 'FEWGDRHDDEEUMFFTEEMJXZR'},
          {'Key': 'KXB',
           'Ring': 'AAA',
           'Message': 'BANKUPCWXJKZEZPFSVJRLDI',
           'Cipher': 'FEWGDRHDDEEUMFFTEEMJXZR'}]
```

OUST GOV P BANK IZMUBMHU VOG

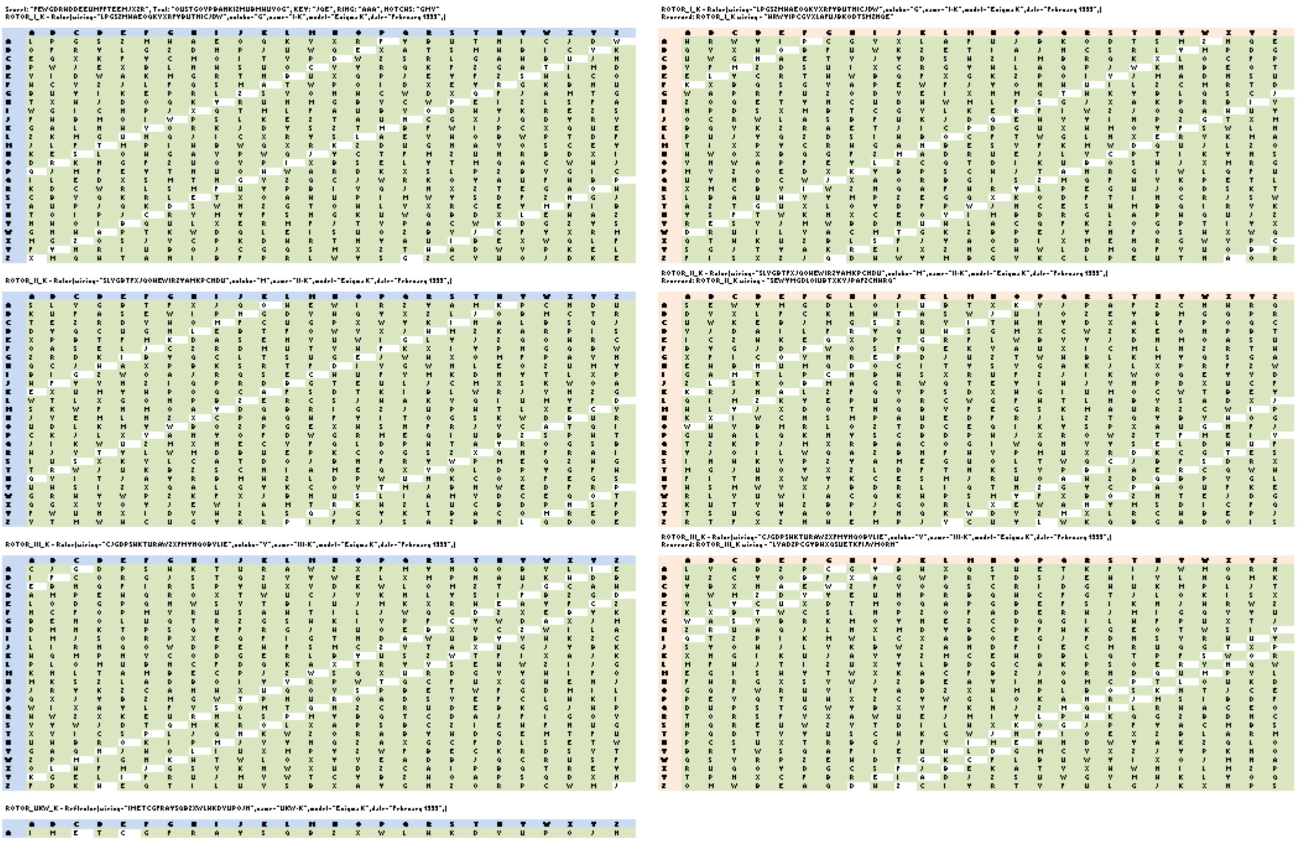
The reverse of GOV

Four messages in total have the word "BANK". When the KEY reached "JQE" with the RING set to "AAA", the Swiss-K simulator decoded the FEW23 cipher into "OUSTGOVPBANKIZMUBMHUVOG", or "OUST GOV P BANK IZMUBMHU VOG" in more readable form with spaces inserted. It was a pure luck to get extra "OUST GOV" and "VOG" at the same time in addition to "BANK". The movement of rotors is sequential, which means that having "VOG" decoded at the end of FEW23 cipher makes the decipherment result complete till the end and "IZMUBMHU" or IZU08 cipher in short is a valid cipher embedded in another cipher. In other words, IZU08 is not random, but intended. This is the first evidence that the CGB cipher is a multiply enciphered kind.

This is the first crack to the decipherment of CGB ciphers. The "P" could stand for Purchase, or something else like Private. "OUST GOV" indicates the buyer as an anonymous party from the ousted government who purchased the shares of the National City Bank. A decrypted text of FEW23 cipher finally has been produced on Swiss-K Enigma machine simulator with a KEY-RING pair set to "JQE" and "AAA".

Swiss-K Enigma Machine: The One

Swiss-K Enigma machine uses 3 rotors and each rotor has 26 internal wirings to pass the incoming electric signals through to the next wheel. When a letter was pressed down, the electric current passes down through ROTOR-I, ROTOR-II, and ROTOR-III. The output signal of ROTOR-III is bounced by the non-rotating fixed reflector and is passed back up to ROTOR-I. Finally, the output current of ROTOR-I turns on the light bulb wired to the corresponding letter. For each letter entered, two forward and backward wires of the rotor are involved in the signal activation. As every letter of FEW23 cipher was being deciphered from start to end, 24 wires of ROTOR-I, 24 wires of ROTOR-II, 24 wires of ROTOR-III, and 12 wires of the reflector were involved in the conduction of electric signals. It is 84 out of 91 wires in total, which covers 92.3% usage for the decipherment of FEW23 cipher.



Secret: "FEWGDRHDDEEUHFFTEEHJX2R", Text: "OUSTGOVPBAHKIZHUBHHUVOG", KEY: "JQE", RING: "AAA", NOTCHS: "GMV"
ROTOR_I_K - Rotor(wiring="LPGSZMHAEOQKVXRFYBUTHICJDW",notch="G",name="I-K",model="Enigma K",date="February 1939",)

ROTOR_I_K - Rotor(wiring="LPGSZMHAEOQKVXRFYBUTHICJDW",notch="G",name="I-K",model="Enigma K",date="February 1939",)
Reversed: ROTOR_I_K wiring - "HRWYIPCGVXLAFUJBKODTSMZHQE"

ROTOR_II_K - Rotor(wiring="SLVGBTFXJQOHEWIRZYAMKPCHDU",notch="M",name="II-K",model="Enigma K",date="February 1939",)

ROTOR_II_K - Rotor(wiring="SLVGBTFXJQOHEWIRZYAMKPCHDU",notch="M",name="II-K",model="Enigma K",date="February 1939",)
Reversed: ROTOR_II_K wiring - "SEWYMGDLOIUBTXKVJPAFZCHHRQ"

ROTOR_III_K - Rotor(wiring="CJGDPSHKTURAWZXFMYHQOBVLIE",notch="V",name="III-K",model="Enigma K",date="February 1939",)

ROTOR_III_K - Rotor(wiring="CJGDPSHKTURAWZXFMYHQOBVLIE",notch="V",name="III-K",model="Enigma K",date="February 1939",)
Reversed: ROTOR_III_K wiring - "LVADZPCGYBHXQSUETKFIJWMORH"

ROTOR_UKW_K - Reflector(wiring="IMETCGFRAYSQBZXWLHKDVUPOJH",name="UKW-K",model="Enigma K",date="February 1939",)

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

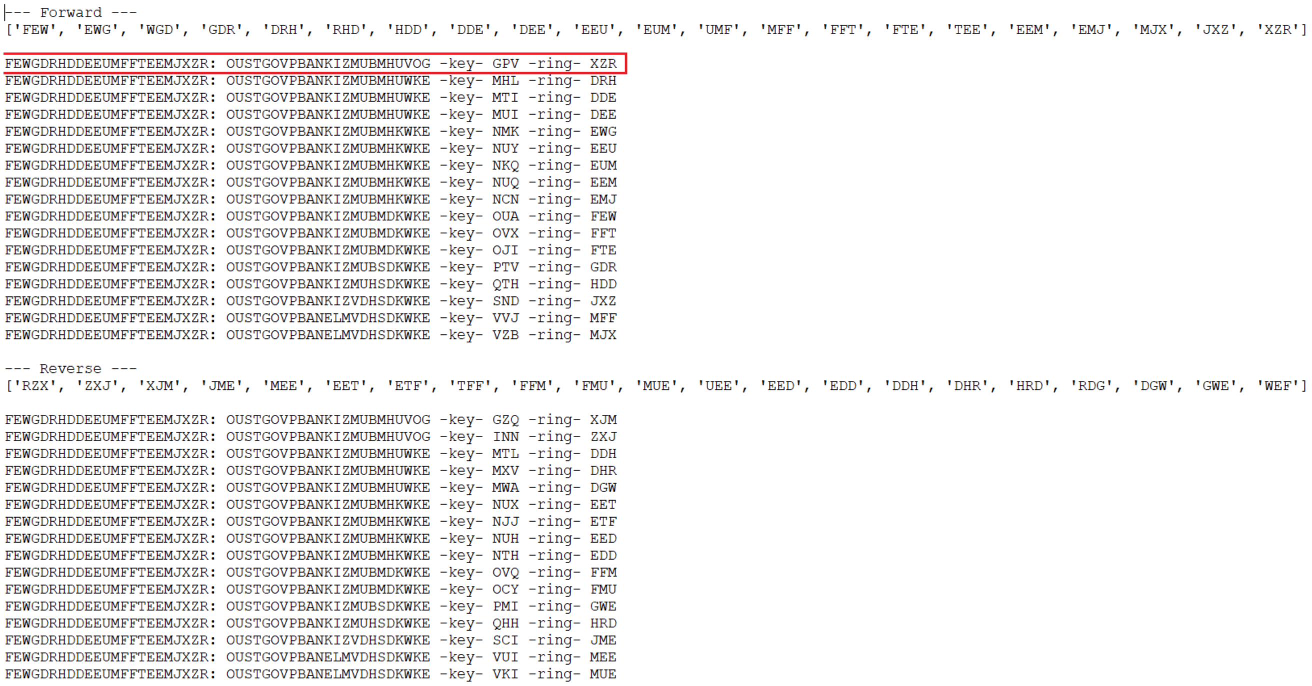
The dark colors in the tables indicate the wires of each rotor that were involved in passing the signals. Each row of a table shows the wiring depending on the KEY position. Since the rotor advances one position per key press, the rotor wiring shifts left by one.

Three tables on the left indicate the forward signal transmission. The table at the bottom is for the reflector. Three tables on the right shows the wirings used for the backward signal mappings. Based on the decoded text of FEW23 cipher, it would be safe to say that Swiss-K Enigma machine was the one involved in the generation of the CGB ciphers. We now can focus on the search for KEY-RING pairs and leave everything else out of the way.

Self Sufficient Cipher: Last Three Letters (L3L)

When deciphering CGB cipher, two parameter values need to be known to both parties, the sender and receiver, and they are the initial KEY and the RING values. The KEY value changes after every key stroke, but the RING values don’t change once configured. In the FEW23 decipherment, the RING was fixed to the value of “AAA”. It would be interesting to see whether the same RING setting equally applies to other CGB ciphers or not. It seems very unlikely though.

It is required for the sending party of ciphers to memorize KEY and RING values they used for each cipher and the receiving party needs to know those values in advance to decode ciphers. At this point, I bluntly assumed further that what if each CGB cipher conveys the RING value in it. I made up a routine to extract RING values from the given cipher and use them to see if they produce the same results. For FEW23 cipher, all possible combinations of 3 consecutive letters were extracted from the string in both ways, forward and backward. Since we already know that FEW23 cipher contains “OUSTGOV”, it is used as a target string for the search. For each 3 letter RING value, FEW23 cipher was decoded by varying KEY values from “AAA” to “ZZZ”. The followings are the result.

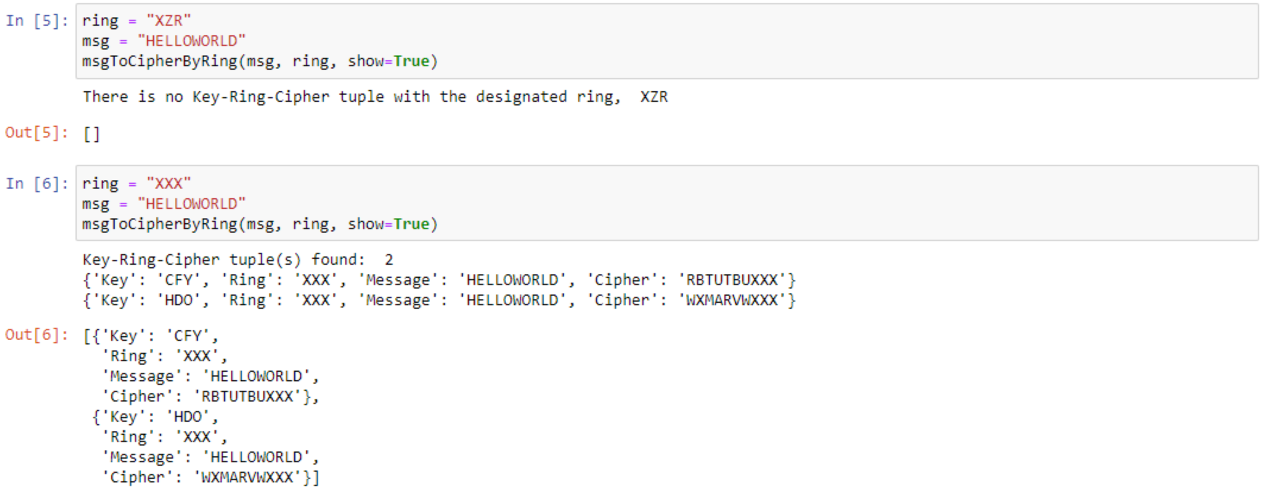


```
--- Forward ---
['FEW', 'EWG', 'WGD', 'GDR', 'DRH', 'RHD', 'HDD', 'DDE', 'DEE', 'EEU', 'EUM', 'UMF', 'MFF', 'FFT', 'FTE', 'TEE', 'EEM', 'EMJ', 'MJX', 'JXZ', 'XZR']

FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMHUVOG -key- GPV -ring- XZR
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMHUWKE -key- MHL -ring- DRH
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMHUWKE -key- MTI -ring- DDE
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMHUWKE -key- MUI -ring- DEE
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMHKWKE -key- NMK -ring- EWG
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMHKWKE -key- NUY -ring- EEU
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMHKWKE -key- NKQ -ring- EUM
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMHKWKE -key- NUQ -ring- EEM
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMHKWKE -key- NCN -ring- EMJ
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMDKWKE -key- OUA -ring- FEW
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMDKWKE -key- OVX -ring- FFT
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMDKWKE -key- OJI -ring- FTE
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBSDKWKE -key- PTV -ring- GDR
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUHSDKWKE -key- QTH -ring- HDD
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZVDHSDKWKE -key- SND -ring- JXZ
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANELMVDHSDKWKE -key- VVJ -ring- MFF
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANELMVDHSDKWKE -key- VZB -ring- MJX

--- Reverse ---
['RZX', 'ZXJ', 'XJM', 'JME', 'MEE', 'EET', 'ETF', 'TFF', 'FFM', 'FMU', 'MUE', 'UEE', 'EED', 'EDD', 'DDH', 'DHR', 'HRD', 'RDG', 'DGW', 'GWE', 'WEF']

FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMHUVOG -key- GZQ -ring- XJM
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMHUVOG -key- INN -ring- ZXJ
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMHUWKE -key- MTL -ring- DDH
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMHUWKE -key- MXV -ring- DHR
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMHUWKE -key- MWA -ring- DGW
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMHKWKE -key- NUX -ring- EET
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMHKWKE -key- NJJ -ring- ETF
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMHKWKE -key- NUH -ring- EED
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMHKWKE -key- NTH -ring- EDD
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMDKWKE -key- OVQ -ring- FFM
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBMDKWKE -key- OCY -ring- FMU
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUBSDKWKE -key- PMI -ring- GWE
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZMUHSDKWKE -key- QHH -ring- HRD
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANKIZVDHSDKWKE -key- SCI -ring- JME
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANELMVDHSDKWKE -key- VUI -ring- MEE
FEWGDRHDDEEUMFFTEEMJXZR: OUSTGOVPBANELMVDHSDKWKE -key- VKI -ring- MUE
```

17 cases out of all three consecutive letters extracted from FEW23 cipher produced the same deciphered texts. After having FEW23 string reversed, three consecutive letters have been extracted and 16 cases also produced the same text. It is surprising to see that there are so many KEY-RING pairs that produce the same result. The most interesting one is the RING value of “XZR”, which is the L3Ls of FEW23 cipher.

The idea of using the last three letters (L3L) of cipher as the RING is absolutely brilliant, if it works every time, because of the realization that how one possibly could know what the L3L of encrypted text would be before having encrypted it. It is not certain how it is possible to get the L3L of encrypted text from its plain text in advance. I don't have any answer to this requirement, but I think it might be viable if a RING value was assigned arbitrarily in advance and then try to find out if there is any KEY value that could produce the L3L of cipher coincidentally identical to the RING used in the encryption. It led me to the next simulation.

In the simulation, the plain text to encrypt is "HELLOWORLD". I arbitrarily picked up two different RING values to get the L3L at the end of cipher text. They are "XZR" and "XXX", respectively.



```
In [5]: ring = "XZR"
        msg = "HELLOWORLD"
        msgToCipherByRing(msg, ring, show=True)

        There is no Key-Ring-Cipher tuple with the designated ring,  XZR

Out[5]: []

In [6]: ring = "XXX"
        msg = "HELLOWORLD"
        msgToCipherByRing(msg, ring, show=True)

        Key-Ring-Cipher tuple(s) found:  2
        {'Key': 'CFY', 'Ring': 'XXX', 'Message': 'HELLOWORLD', 'Cipher': 'RBTUTBUXXX'}
        {'Key': 'HDO', 'Ring': 'XXX', 'Message': 'HELLOWORLD', 'Cipher': 'WXMARVWXXX'}

Out[6]: [{'Key': 'CFY',
          'Ring': 'XXX',
          'Message': 'HELLOWORLD',
          'Cipher': 'RBTUTBUXXX'},
         {'Key': 'HDO',
          'Ring': 'XXX',
          'Message': 'HELLOWORLD',
          'Cipher': 'WXMARVWXXX'}]
```

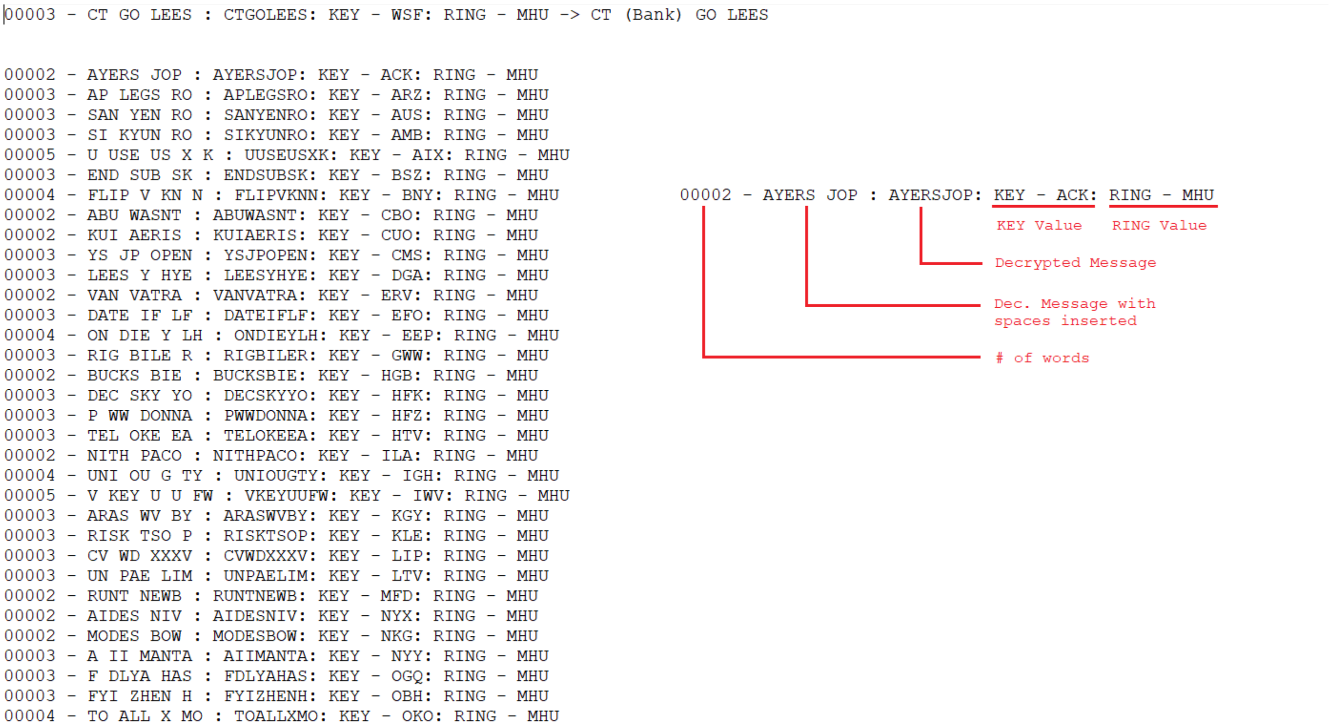
The RING of "XZR" didn't produce any cipher ending with "XZR". The RING of "XXX" produced two encrypted texts ending with "XXX" and the KEY values obviously were different. Either one can be chosen for the encryption of "HELLOWORLD". Now the cipher carries the RING value with which the original text can be produced in the decryption. The simulation result tells that knowing the RING value doesn't specify the KEY for that cipher.

Since the KEY value can't be assigned arbitrarily, encapsulating the KEY value in the encrypted message would be impossible. This implies that both the sender and receiver in the exchange of secret texts need to know what KEY value was used in the given ciphers. It also implies that there could be predetermined KEY-RING pairs shared by each party and each party uses their own KEY-RING pair to exchange information including the KEY value beforehand. By the way, it is surprising how they have calculated the L3L without using the computing power of modern computer systems.

Decoding IZU08 Cipher: The Completion of FEW23 Cipher

IZU08 cipher ("IZMUBMHU") is a part of the decoded FEW23 cipher. It looks random, but it could be another encrypted text itself. So, let's find out. It is assumed that the L3L "MHU" was used as the RING value for the encryption of IZU08 cipher. Since it is unknown which KEY value was used to get the cipher, we have to collect deciphered texts with all the KEY values from "AAA" to "ZZZ". It produces a list of 17,576 texts and each and every decrypted text needs to be examined to narrow down into a manageable size and finalize the plausible text. It is noted that this process is inevitable and prone to human errors.

The following picture shows a list of the decrypted texts selected from 17,576 texts. In order to facilitate with the selection process, a string operation is adopted to insert spaces to increase readability. I used a library called "ninja" from GitHub. After space insertions, the words are counted and labeled. All decrypted texts are sorted in ascending order.



```
00003 - CT GO LEES : CTGOLEES: KEY - WSF: RING - MHU -> CT (Bank) GO LEES


00002 - AYERS JOP : AYERSJOP: KEY - ACK: RING - MHU
00003 - AP LEGS RO : APLEGSRO: KEY - ARZ: RING - MHU
00003 - SAN YEN RO : SANYENRO: KEY - AUS: RING - MHU
00003 - SI KYUN RO : SIKYUNRO: KEY - AMB: RING - MHU
00005 - U USE US X K : UUSEUSXK: KEY - AIX: RING - MHU
00003 - END SUB SK : ENDSUBSK: KEY - BSZ: RING - MHU
00004 - FLIP V KN N : FLIPVKNN: KEY - BNY: RING - MHU
00002 - ABU WASNT : ABUWASNT: KEY - CBO: RING - MHU
00002 - KUI AERIS : KUIAERIS: KEY - CUO: RING - MHU
00003 - YS JP OPEN : YSJPOPEN: KEY - CMS: RING - MHU
00003 - LEES Y HYE : LEESYHYE: KEY - DGA: RING - MHU
00002 - VAN VATRA : VANVATRA: KEY - ERV: RING - MHU
00003 - DATE IF LF : DATEIFLF: KEY - EFO: RING - MHU
00004 - ON DIE Y LH : ONDIEYLH: KEY - EEP: RING - MHU
00003 - RIG BILE R : RIGBILER: KEY - GWW: RING - MHU
00002 - BUCKS BIE : BUCKSBIE: KEY - HGB: RING - MHU
00003 - DEC SKY YO : DECSKYYO: KEY - HFK: RING - MHU
00003 - P WW DONNA : PWWDONNA: KEY - HFZ: RING - MHU
00003 - TEL OKE EA : TELOKEEA: KEY - HTV: RING - MHU
00002 - NITH PACO : NITHPACO: KEY - ILA: RING - MHU
00004 - UNI OU G TY : UNIOUGTY: KEY - IGH: RING - MHU
00005 - V KEY U U FW : VKEYUUFW: KEY - IWV: RING - MHU
00003 - ARAS WV BY : ARASWVBY: KEY - KGY: RING - MHU
00003 - RISK TSO P : RISKTSOP: KEY - KLE: RING - MHU
00003 - CV WD XXXV : CVWDXXXV: KEY - LIP: RING - MHU
00003 - UN PAE LIM : UNPAELIM: KEY - LTV: RING - MHU
00002 - RUNT NEWB : RUNTNEWB: KEY - MFD: RING - MHU
00002 - AIDES NIV : AIDESNIV: KEY - NYX: RING - MHU
00002 - MODES BOW : MODESBOW: KEY - NKG: RING - MHU
00003 - A II MANTA : AIIMANTA: KEY - NYY: RING - MHU
00003 - F DLYA HAS : FDLYAHAS: KEY - OGQ: RING - MHU
00003 - FYI ZHEN H : FYIZHENH: KEY - OBH: RING - MHU
00004 - TO ALL X MO : TOALLXMO: KEY - OKO: RING - MHU
```

```
00002 - AYERS JOP : AYERSJOP: KEY - ACK: RING - MHU
```

For IZU08 cipher, the decrypted text with the KEY of "WSF" was chosen as the result, which goes like "CT GO LEES". The decrypted text of FEW23 cipher is given as "OUST GOV P BANK CT GO LEES VOG". When the last 9 letters were read in reverse, it becomes "GOV SEEL OG", or "GOV SEE LOG". It is obvious that "BANK CT" indicates the National City Bank". The final deciphered text of FEW23 cipher is "OUST GOV P BANK CT GOL EES VOG", which could be interpreted as "OUST GOV P (purchased) BANK CT (City Bank), GOV SEE LOG (of transaction)".

In summary, the followings are verified.

- Swiss-K Enigma machine with the factory configuration is the one that was used in the generation of CGB ciphers.
- Original NOTCH values were untouched.
- Original ROTOR and Reflector wirings were not altered.
- L3L of a CGB cipher are used as the RING value for its decipherment unless instructed otherwise.
- CGB Cipher had plain text encrypted and another encrypted text can be included as a part of original text. In other words, the CGB ciphers allow multiple encryptions.

- It would be highly unlikely that two different CGB ciphers use the same set of KEY and RING values since it is not guaranteed that the same RING value will be available to both ciphers at the same time.

Deciphering the CGB ciphers is straight forward from this point on.

HTG12 (HUD TAB GALLOW) Cipher: A Place Holder for another Cipher

On the left side of Gold Bar labeled with KO 08124, the following transcription is inscribed.

MQO24: MQOLCSJTLGAJOKBSSBOMUPCE
FEW23: FEWGDRHDDEEUMFFTEEMJXZR
YQH16: YQ**HUDTABGALLOW**LS
VIO12: VIOHIKNNGUAB
MVE12: MVERZRLQDBHQ
ZUQ08: ZUQUPNZN
VIO12: VIOHIKNNGUAB
JKG14: JKGFIJPMCWSAEK
SKC25: SKCDKJCDJCYQSZKTZJPXPWIRN

YQH16 cipher contains a plain text in the middle, which is "HUDTABGALLOW" or "HUD TAB GALLOW" (HTG12). It starts with "YQ" and ends with "LS". It is highly unlikely to get the YQH16 cipher from a plain text. If both "YQ" and "LS" were plug board settings, the decryption of FEW23 cipher would be changed to "OU**L**T GOV P BANK CT GO**S** EE**L** VOG". It breaks the plausibility. They couldn't be any plug board settings. The length of HTG cipher is 12 letters wide. It is the same as the length of VIO12 cipher. The YQH16 cipher is not a cipher itself and it is regarded as a place holder. The puzzle can be solved when MVE12 cipher is decrypted first. The MVE12 cipher is decrypted as follows.

00005 - TWO MSG XOR S RH : TWOMSGXORSRH: KEY - GIV: RING – BHQ

With the KEY of "GIV" and the RING of "BHQ", the MVE12 cipher is translated into "TWO MSG XOR SRH". Two messages referred by the MVE12 cipher are YQH16 and VIO12 ciphers. The operation of XOR with those two messages produces a new YQV16 cipher.

YQV16: YQ**VIOHIKNNGUAB**LS

It is assumed that "SRH" at the end is used as the RING value for the YQV16 cipher. (Actually, I tried to get the deciphered text of YQV16 cipher with the RING of "BLS", but it didn't produce any plausible results. It also is possible that I missed other plausible text though.) From the decipherment of YQV16 cipher with "SRH" as the RING value produced the following result.

00005 - CONTACT FW T DEW VRY : CONTACTFWTDEWVRY: KEY - ZDE: RING - SRH -> CONTACT [FWT] DEWVRY

The deciphered text is "CONTACT FWT DEW VRY". The name of DEW can be found on the inscriptions of gold bars and it looks like "Dewly" or something else. “FWT” could be a RING value for another cipher. When the length of CGB cipher was intended to be shorter than 26 letters, the space requirement would have forced to make original text compacted. It is expected to use abbreviations. ”DEWVRY” could be the name of the person who was the contact point.

Since the VIO12 cipher doesn’t decipher itself, the repeated occurrence of VIO12 cipher below ZUQ08 cipher is considered to be a contextual marker to indicate the separation of MVE12 from others. The YQH16 cipher is a place holder and it is not decrypted as well.

Deciphering MQO24 Cipher

The decipherment of MQO24 cipher is a little bit more complicated because it contains double coded ciphers in it. At the first decryption, the following message was decoded and chosen. (When anyone disagrees with me on this, they can do their own decryption different from the one presented here.)

```
00011 - USE AL TY K D BJ Z GOV DVD ULI DUD :
        USEALTYKDBJZGOVDVDULIDUD: KEY - FWQ: RING - PCE -> USE
        ALTYKDBJZ GOV DVDULIDUD
```

With the KEY of “FWQ” and the RING of “PCE”, we have a message divided into two parts. One starts with “USE” and the other with “GOV”. The clause of “USE” has a cipher labeled ALT09 (“ALTYKDBJZ”) and the “GOV” clause has a cipher labeled DVD09 (“DVDULIDUD”). Both ciphers are 9 letters long each. They are decrypted as follows.

```
00003 - D JUD BORIS : DJUDBORIS: KEY - YCI: RING – BJZ
ALT09: ALTYKDBJZ ◊ D JUD BORIS

00003 - IDP KEY SEA : IDPKEYSEA: KEY - CPU: RING – DUD
DVD09: DVDULIDUD ◊ ID P KEY SEA
```

Altogether, the MQO24 cipher can be deciphered.

```
MQU24: MQOLCSJTLGAJOKBSSBOMUPCE ◊ USED JUD BORIS GOV ID P KEY SEA
```

The “P” could indicate “Purchase” as in the “P BANK” of the FEW23 cipher. Here we have another name BORIS the JUDGE. “SEA” is the P-KEY and it is uncertain how this key is being used. Nonetheless, based on the deciphered text, it might be used to identify the OUST GOV or someone on that side.

```
MQO24: MQOLCSJTLGAJOKBSSBOMUPCE ◊ USED JUD BORIS GOV ID P KEY SEA
FEW23: FEWGDRHDDEEUMFFTEEMJXZR ◊ OUST GOV P BANK CT GOL EES VOG
YQH16: YQHUDTABGALLOWLS (place holder not decrypted)
VIO12: VIOHIKNNGUAB (not decrypted alone)
MVE12: MVERZRLQDBHQ ◊ TWO MSG XOR SRH
YQV16(YQH16 XOR VIO12) : YQVIOHIKNNGUABLS ◊ CONTACT FWT DEWVRY
ZUQ08: ZUQUPNZN
VIO12: VIOHIKNNGUAB
JKG14: JKGFIJPMCWSAEK
```

```
      SKC25: SKCDKJCDJCYQSZKTZJPXPWIRN
```

Deciphering ZUQ08 Cipher

When ZUQ08 was deciphered with the RING of "NZN", several decoded texts could be picked up based on its plausibility. For example, I narrowed them down into 3 texts as the candidates for the decoded text.

```
     00003 - GOD BY ILL : GODBYILL: KEY - GVC: RING - NZN
     00003 - ALL OR RUE : ALLORRUE: KEY - JFR: RING - NZN
     00003 - XOR ID IWA : XORIDIWA: KEY - PTB: RING - NZN
```

Any one of them could provide with certain context to the deciphered texts and can be considered as the deciphered text. The problem is that as long as the right KEY value is unknown, none of them would be the original text.

If ZUQ08 cipher was deciphered with the RING of "FWT" from the deciphered YQV16 cipher, the following texts are picked up.

```
     00003 - XOR ID IWA : XORIDIWA: KEY - HQH: RING - FWT
     00003 - GOD BY ILL : GODBYILL: KEY - YSI: RING - FWT
     00003 - ALL OR MEP : ALLORMEP: KEY - BCX: RING - FWT
```

It seems that "ALL OR RUE" was replaced with "ALL OR MEP". The phrase "GOD BY ILL" doesn't seem to go well with the other ciphers. The context of "GOD BY ILL" would disagree with other deciphered texts. It leaves "XOR ID IWA" as the final text to choose on.

Deciphering JKG14 Cipher

With the RING of "AEK", the JKG14 cipher is decoded.

```
          00006 - ACT DMZ EF G SIG CT : ACTDMZEFGSIGCT: KEY - CAS: RING –
               AEK ◊ ACT DMZ [EFG] SIG CT
```

"EFG" is regarded as the RING for the next cipher SKC25. With the deciphered YQV16 cipher, the deciphered texts together indicate that the contact person DEW (or DEWVRY) acts like DMZ and signals CT Bank ("CT" from the IZU08 cipher contained in the FEW23 cipher).

Deciphering SKC25 Cipher

With the RING of "EFG" from the deciphered text of JKG14 cipher, the SKC25 cipher can be decrypted as follows.

```
     00012 - BAY W X DY FX J F EX CBN S TACKY LOT :
          BAYWXDYFXJFEXCBNSTACKYLOT: KEY - HNN: RING – EFG ◊ BAY
          WXDYFXJFEXCBN STACKY LOT
     WXD13: WXDYFXJFEXCBN
```

With the RING of "CBN", the WXD13 cipher is decrypted.

00004 - ADJ GATA MAST WY : ADJGATAMASTWY: KEY - RPY: RING – CBN ◊ ADJ GATA MAST WY

It looks like a street name. The final text will look like below.

00012 - BAY W X DY FX J F EX CBN S TACKY LOT : BAYWXDYFXJFEXCBNSTACKYLOT: KEY - HNN: RING – EFG ◊ BAY [ADJ GATA MAST WY] STACKY LOT

“ADJ” could stand for “adjacent”. Once again, deciphering a CGB cipher is a task to single out one text out of all possible 17,576 outcomes and it doesn’t guarantee that the finalized one will be the original message all the time. The result presented in this report is merely an example.

Deciphered Texts on the Left Column of Gold Bar (KO 08124)

All ciphers on the left column of the gold bar labeled KO 08124 are listed below with their deciphered texts.

Label	**Cipher Message**	**KEY:RING**	**Deciphered Message**
MQO24	MQOLCSJTLGAJOKBSSBOMUPCE	FWQ:PCE	USE ALTYKDBJZ GOV DVDULIDUD ◊ USED JUD BORIS GOV ID P KEY SEA
◊ALT09	ALTYKDBJZ	YCI:BJZ	D JUD BORIS
◊DVD09	DVDULIDUD	CPU:DUD	ID P KEY SEA
FEW23	FEWGDRHDDEEUMFFTEEMJXZR	GPV:XZR	OUST GOV P BANK IZMUBMHU VOG ◊ OUST GOV P BANK CT GOL EES VOG
◊IZM08	IZMUBMHU	WSF:MHU	CT GOL EES
YQH16	YQ**HUDTABGALLOW**LS		(A place holder)
VIO12	**VIOHIKNNGUAB**		(Not deciphered as it is)
MVE12	MVERZRLQDBHQ	GIV:BHQ	TWO MSG XOR [SRH]
YQV16	YQ**VIOHIKNNGUAB**LS	ZDE:SRH	CONTACT [FWT] DEWVRY
ZUQ08	ZUQUPNZN	HQH:FWT	XOR ID IWA
VIO12	VIOHIKNNGUAB		(Not deciphered as it is)
JKG14	JKGFIJPMCWSAEK	CAS:AEK	ACT DMZ [EFG] SIG CT
SKC25	SKCDKJCDJCYQSZKTZJPXPWIRN	HNN:EFG	BAY WXDYFXJFEXCBN STACKY LOT ◊ BAY ADJ GATA MAST WY STACKY LOT
◊WXD13	WXDYFXJFEXCBN	RPY:CBN	ADJ GATA MAST WY

Three letters enclosed in brackets are used as the RING value for the decipherment of the cipher next in line. “SRH” from MVE12 cipher was contributed to the decipherment of the YQV16 cipher. Note that “TWO MSG” indicates both YQH16 and VIO12 ciphers. By performing “XOR” operation on them, the VIO12 cipher takes the place of “HUDTABGALLOW” in the YQH16 cipher so that the YQV16 cipher is formed.

Three letters “IWA” from ZUQ08 does not apply as the RING value for the JKG14 cipher. Actually, when “IWA” was applied to the decipherment of JKG14 cipher, it didn’t produce any plausible results. The presence of VIO12 cipher just below ZUQ08 cipher might imply that the application of “IWA” as the RING value doesn’t apply beyond that point. This interpretation also might imply that if there is no VIO12 cipher after ZUQ08 cipher, the RING for the cipher next to ZUQ08 cipher would be “IWA”.

Synchronizing the RING value to the L3Ls of encrypted cipher doesn't work with any randomly chosen three letters. It is easier to determine in a matter of seconds if the cipher will end with the RING value with the aid of modern computer systems. Finding the RING value for any ciphers every time wouldn't be an easy task back in 1930s. Sometimes they couldn't find out the RING value for the encryption of text in time. Embedding the RING value within a cipher for the next cipher might be an alternative solution to this kind of problem. It doesn't require the cipher to end with the embedded RING value from other cipher.

Limitations on the Decryption of CGB Ciphers

If both KEY and RING values for the given CGB cipher are known, the original message can always be determined decisively. Mostly, the last 3 letters of each CGB ciphers are being used as the RING value for the decryption unless implied otherwise. Unfortunately, the KEY values for each CGB cipher are unknown and the lack of information makes the recovery process very hard since a single message should be picked and chosen out of a lot of texts for the presumed original text.

In addition to that, CGB cipher might not be consisted of one plain text. They could contain other encrypted texts in the deciphered text. When the decrypted text contains only partial plain text, there could be other embedded ciphers to be verified. Nonetheless the generation process of the CGB ciphers has been reverse engineered and detailed well in this report.

The Bottom Part of Gold Bar with Dual Text (Upside Down)

The gold bar with Chinese texts and Cryptograms side by side doesn't have a label on it. The bottom part of the text consists of five lines of ciphers.

```
Line 1: (MLM13) MLMTAHGBGFNIV
Line 2: (ZUQ08) ZUQUPNZN (ABR14) ABRYCTUGVZXUPB
Line 3: (MVE12) MVERZRLQDBHQ (GKJ11) GKJFHYXODIE
Line 4: (UGM13) UGMNCBXCRLDEY
Line 5: (HFX19) HFXPCQYZVATXAWIZPVE
```

Deciphering MLM13 Cipher

With the RING of "NIV", the MLM13 cipher is decoded at the KEY of "ZNZ" as follows.

```
00005 - A GOVT CIV BOU CO : AGOVTCIVBOUCO: KEY - ZNZ: RING - NIV
```

GOVT stands for "GOVERNMENT". CIV stands for "CIVILIAN". BOU stands for "BOUGHT". "CO" stands for "COMPANY". The message implies that the oust government as a civilian bought the shares of the National City Bank.

Deciphering ABR14 Cipher on Line 2

We already decoded the ZUQ08 cipher and the recovered text is “XOR ID IWA”. It is assumed that the presence of ZUQ08 cipher in front of the ABR14 cipher implies that the L3Ls for the ABR14 cipher is replaced (“XORed”) with “IWA” before being deciphered. In other words, the ABR14 cipher will be modified from “ABRYCTUGVZXUPB” to “ABRYCTUGVZXIWA” (ABR14a). The new ABR14a cipher is decoded as follows.

00005 - REPAYS PIB L DZ CQ : REPAYSPIBLDZCQ: KEY - ADD: RING - IWA ◊ REPAY <u>SPIBLDZCQ (SPI09)</u>

The SPI09 then is deciphered.

00003 - USO GD WHEN : USOGDWHEN: KEY - VMA: RING - ZCQ

Altogether, the deciphered message of ABR14a cipher is given as follows.

ABR14: ABRYCTUGVZXUPB
ABR14a: ABRYCTUGVZXIWA ◊ REPAY US OGD WHEN

“OGD” can be a RING value for other cipher or it could stand for “OF GOLD”. It is noted that if the RING for the ABR14 cipher wasn’t replaced with “IWA”, the SPI09 cipher would have had different three letters at the end and the deciphered result wouldn’t be the same as presented here.

Deciphering GKJ11 Cipher on Line 3

The MVE12 cipher in front of GKJ11 is deciphered as “TWO MSG XOR SRH”. Nonetheless there are no two messages to perform XOR operation on. Instead, “SRH” is used as the RING for the decipherment of GKJ11 cipher. With the RING of “SRH”, the GKJ11 is deciphered as follows.

00004 - A ZH LATE TOLD : AZHLATETOLD: KEY - ZRQ: RING - SRH ◊ AZH LATE TOLD

When the L3Ls “DIE” was used as the RING instead of “SRH”, the GKJ11 cipher is decoded as follows.

GKJFHYXODIE: AZHLATETXXS -key- KIN -ring- DIE ◊ AZH LATE TXXS

Different RING value produces different result. It is interesting to notice that only three letters at the end of text are mismatch, not the entire message. Errors on the deciphered text are isolated and not propagated through the remaining part of text. It can be stated that one letter error on the deciphered text can easily be spotted as a typo and would be fixed after the decryption has ended.

Deciphering UGM13 Cipher

With the L3Ls of UGM13 as the RING, the following message was chosen out of 17,576 possible outcomes.

```
00005 - LU QX GUMS SIG CO : LUQXGUMSSIGCO: KEY - JEB: RING - DEY ◊
      LUQXGUMS SIG CO

LUQ08: LUQXGUMS
```

It has a double encrypted cipher LUQ08 with the RING of “UMS”. Two outputs were chosen as the plausible deciphered texts.

```
00003 - ASIAN MX F : ASIANMXF: KEY - MJV: RING - UMS ◊ ASIAN MXF
00003 - US VETS WW : USVETSWW: KEY - ULJ: RING - UMS ◊ US VET SWW
```

We have two deciphered messages for the UGM13 cipher.

```
A. UGM13: UGMNCBXCRLDEY ◊ ASIAN MXF SIG CO
B. UGM13: UGMNCBXCRLDEY ◊ US VET SWW SIG CO
```

“ASIAN” implies the people on the oust government side since they were Chinese. If “US VET” was picked out, it would imply that the person named “DEWVRY” from the YQV16 cipher. “US” could indicate the bank, or the United States. “VET” could indicate either a war veteran of the States or a veteran of the banking. If “DEWVRY” was a US war veteran, the final deciphered text for the UGM13 cipher would be “US VET SWW SIG CO”. Otherwise, “ASIAN MXF SIG CO” can be chosen as the recovered text.

Deciphering HFX19 Cipher

It took me lots of time decoding HFX19 cipher because using the L3Ls “PVE” as the RING value won’t produce any plausible outcomes. It is partly because the length of HFX19 cipher is relatively long and the readability of deciphered texts is very low. Spotting plausible texts out of 17,576 texts per a RING value isn’t an easy task to do with. Since the L3Ls of its own cipher was no good, it is time to try with the RING value originated from other deciphered texts. Using three letters from deciphered texts as the RING value makes the decryption process more complicated because we don’t know which one to pick. The working three letters might have been put inside a cipher that wasn’t decoded yet.

On the right column of the gold bar labeled KO 00803, the inscription line starts as follows.

```
VIO12: VIOHIKNNGUAB
HFX19: HFXPCQYZVATXAWIZPVE
YQH16: YQHUDTABGALLOWLS
XLY20: XLYPISNANIRUSFTFWMIY
```

Both VIO12 and YQH16 ciphers on the gold bar labeled KO 08124 aren’t deciphered because there is no MVE12 cipher next to it.

```
MVE12: MVERZRLQDBHQ ◊ TWO MSG XOR SRH
```

The HFX19 cipher was placed between VIO12 and YQH16 ciphers. It might imply that the RING value of “SRH” applies to the decipherment of HFX19 cipher. By using “SRH” as the RING, the HFX19 cipher produces the following output.

```
      00009 - US BERGE U IF EK S FG VW TZ : USBERGEUIFEKSFGVWTZ: KEY - HEV:
            RING - SRH ◊ US BERGE UIFEKSFGVWTZ
```

We get another UIF12 cipher in the decrypted text.

```
       UIF12: UIFEKSFGVWTZ
```

When the UIF12 cipher was decoded with the RING of “WTZ”, we have the following outcome.

```
      00005 - GOV YD ZX UFC PC : GOVYDZXUFCPC: KEY - ZQN: RING - WTZ ◊ GOV
            YDZXUFCPC
```

The decrypted text has another cipher named YDZ09.

```
       YDZ09: YDZXUFCPC
```

The YDZ09 cipher produces the final text as follows.

```
      00003 - KEY JD SIAA : KEYJDSIAA: KEY - SHZ: RING - CPC ◊ KEY JDS IAA
```

Altogether, the HFX19 cipher has its final text as follows.

```
       HFX19: HFXPCQYZVATXAWIZPVE ◊ US BERGE GOV KEY JDS IAA
```

It seems too much to encrypt a message three times in a row by adding extra text. The result taken here might be wrong and there could be other plausible outcomes. Nonetheless if the decryption is right, those two KEYs for the government kept by the bank should have very important role. They might be a KEY-RING pair used for the encryption and/or decryption in message exchanges between the government and the bank.

Deciphered Texts on the Bottom Part of Gold Bar with Text Only (Upside Down)

All ciphers on the bottom of the gold bar with dual texts have been listed here with their deciphered texts.

Label	Cipher Message	KEY:RING	Deciphered Message
MLM13	MLMTAHGBGFNIV	ZNZ:NIV	A GOVT CIV BOU CO
ZUQ08	ZUQUPNZN	HQH:FWT	XOR ID IWA
ABR14	ABRYCTUGVZXUPB		(Last 3 letters replaced with IWA)
ABR14a	ABRYCTUGVZXIWA	ADD:IWA	REPAY SPIBLDZCQ ◊ REPAY US OGD WHEN
◊SPI09	SPIBLDZCQ	VMA:ZCQ	US OGD WHEN
MVE12	MVERZRLQDBHQ	GIV:BHQ	TWO MSG XOR SRH
GKJ11	GKJFHYXODIE	ZRQ:SRH	AZH LATE TOLD
UGM13	UGMNCBXCRLDEY	JEB:DEY	LUQXGUMS SIG CO ◊ (a) US VET SWW SIG GO or (b) ASIAN MXF SIG CO
◊LUQ08	LUQXGUMS	ULJ:UMS	(a) US VET SWW
		MJV:UMS	(b) ASIAN MXF
HFX19	HFXPCQYZVATXAWIZPVE	HEV:SRH	US BERGE UIFEKSFGVWTZ ◊ US BERGE GOV KEY JDS IAA
◊UIF12	UIFEKSFGVWTZ	ZQN:WTZ	GOV YDZXUFCPC
◊YDZ09	YDZXUFCPC	SHZ:CPC	KEY JDS IAA

The Right Column of Gold Bar (KO 08124)

The ciphers on the left column have been decrypted earlier. On the right column, there are five ciphers.

```
MLM13: MLMTAHGBGFNIV ◊ A GOVT CIV BOU CO
FEW23: FEWGDRHDDEEUMFFTEEMJXZR ◊ OUST GOV P BANK CT GOL EES VOG
RHZ24: RHZVIYQIYSXVNQXQWIOVWPJO
HFX19: HFXPCQYZVATXAWIZPVE ◊ US BERGE GOV KEY JDS IAA
KOW13: KOWVRSRWTMLDH
```

Three of five ciphers have already been decrypted above.

Deciphering RHZ24 Cipher

With the RING of “PJO”, the RHZ24 cipher is decrypted. The deciphered text contains two ciphers and they are NOC13 and PAS06 ciphers.

```
00009 - PAY NO CNN TTC LAS ES IF PASO LI : PAYNOCNNTTCLASESIFPASOLI:
        KEY - FHK: RING - PJO ◊ PAY NOCNNTTCLASES IF PASOLI

NOC13: NOCNNTTCLASES
PAS06: PASOLI
```

The decrypted NOC13 text contains one more cipher.

00006 - KEY QI AHU NO Q UP : KEYQIAHUNOQUP: KEY - DKU: RING - SES ◊ KEY QIAHUNOQ UP

QIA08: QIAHUNOQ

The QIA08 cipher is decrypted.

00003 - IN USS KAN : INUSSKAN: KEY - GGK: RING - NOQ

The decrypted text of NOC13 cipher is "KEY IN USS KAN UP". The PAS06 cipher is decrypted.

00002 - DUE ASK : DUEASK: KEY - YQX: RING - OLI

The final decrypted text of RHZ24 cipher is "PAY KEY IN USS KAN UP IF DUE ASK". The RHZ24 cipher contains a text encrypted three times like HFX19 does. Each RHZ24 and HFX19 ciphers contains a pair of three letter keys.

Deciphering KOW13 Cipher

With the RING of "LDH", the KOW13 cipher is decrypted. It contains another cipher in the decrypted text.

00006 - ACC NON LB RH Q EM : ACCNONLBRHQEM: KEY - NQS: RING - LDH ◊ ACC NO NLBRHQEM

NLB08: NLBRHQEM

The NLB08 cipher is decrypted.

00003 - ASH KEEP I : ASHKEEPI: KEY - OMN: RING - QEM ◊ I KEEP HSA (in reverse)

The final decrypted text is "ACC NO [ASH KEEP I]" or it could be read like "ACC NO [I PEEK HSA]". Remember that the decrypted text of FEW24 cipher also has reversed words like "GOL EES VOG" at the end and it is read like "GOV SEE LOG".

Deciphered Texts of the Right Column of Gold Bar (KO 08124)

All ciphers on the right column have been listed here with their deciphered texts.

Label	**Cipher Message**	**KEY:RING**	**Deciphered Message**
MLM13	MLMTAHGBGFNIV	ZNZ:NIV	A GOVT CIV BOU CO
FEW23	FEWGDRHDDEEUMFFTEEMJXZR	GPV:XZR	OUST GOV P BANK IZMUBMHU VOG ◊ OUST GOV P BANK CT GOL EES VOG
RHZ24	RHZVIYQIYSXVNQXQWIOVWPJO	FHK:PJO	PAY NOCNNTTCLASES IF PASOLI ◊ PAY KEY IN USS KAN UP IF DUE ASK
◊NOC13	NOCNNTTCLASES	DKU:SES	KEY QIAHUNOQ UP
◊QIA08	QIAHUNOQ	GGK:NOQ	IN USS KAN
◊PAS06	PASOLI	YQX:OLI	DUE ASK
KOW13	KOWVRSRWTMLDH	NQS:LDH	ACC NO NLBRHQEM ◊ ACC NO ASH KEEP I/ACC NO [I PEEK HSA]
◊NLB08	NLBRHQEM	YQX:OLI	ASH KEEP I/I PEEK HSA

The Middle Column of Gold Bar (K00803)

There are five ciphers on the middle column and they all were decrypted.

UGM13: UGMNCBXCRLDEY ◊ A GOVT CIV BOU CO
RHZ24: RHZVIYQIYSXVNQXQWIOVWPJO ◊ PAY [KEY IN USS KAN] UP IF DUE ASK
SKC25: SKCDKJCDJCYQSZKTZJPXPWIRN ◊ BAY [ADJ GATA MAST WY] STACKY LOT
MQO24: MQOLCSJTLGAJOKBSSBOMUPCE ◊ USED JUD BORIS GOV ID P KEY SEA
FEW23: FEWGDRHDDEEUMFFTEEMJXZR ◊ OUST GOV P BANK CT [GOL EES VOG]

The Right Column of Gold Bar (K00803)

There is one cipher left to decrypt on the right column of gold bar labeled KO 00803.

VIO12: VIOHIKNNGUAB ◊ (not deciphered)
HFX19: HFXPCQYZVATXAWIZPVE ◊ US BERGE GOV KEY JDS IAA
YQH16: YQHUDTABGALLOWLS ◊ (not deciphered)
XLY20: XLYPISNANIRUSFTFWMIY
KOW13: KOWVRSRWTMLDH ◊ ACC NO ASH KEEP I/ACC NO [I PEEK HSA]
JKG14: JKGFIJPMCWSAEK ◊ ACT DMZ [EFG] SIG CT
ABR14: ABRYCTUGVZXUPB (ABRYCTUGVZXIWA) ◊ REPAY US OGD WHEN
GKJ11: GKJFHYXODIE ◊ AZH LATE TOLD
ZUQ08: ZUQUPNZN ◊ XOR ID IWA
GKJ11: GKJFHYXODIE ◊ AZH LATE TOLD

Deciphering XLY20 Cipher

Since the XLY20 cipher is located below the YQH16 cipher, the RING of “SRH” will be chosen for its decipherment. The following text was picked and chosen as the deciphered outcome. It contains two other ciphers.

```
00010 - QUIZ TW GY X TO A EW MY SX FH : QUIZTWGYXTOAEWMYSXFH: KEY -
     DLX: RING - SRH ◊ QUIZ TWGYX TO AEWMYSXFH

TWG05: TWGYX
AEW09: AEWMYSXFH
```

The TWG05 cipher is decrypted.

```
00001 - BUYER : BUYER: KEY - BDZ: RING - GYX
```

The AEW09 cipher is decrypted.

```
00003 - BYE ANY OPP : BYEANYOPP: KEY - IKU: RING - XFH
```

The deciphered text of XLY20 is given as follows.

```
XLY20: XLYPISNANIRUSFTFWMIY ◊ QUIZ BUYER TO BYE ANY OPP
```

For the deciphered text of TWG05 cipher, there are a few more alternatives.

```
00001 - ASHOK : ASHOK: KEY - CIX: RING - GYX ◊ ASH OK (ASH from "ASH
     KEEP I" of KOW13 cipher)
00002 - A CLUE : ACLUE: KEY - OLB: RING - GYX
00001 - BYNUM : BYNUM: KEY - TUV: RING - GYX ◊ BY NUM
```

With all these alternatives, the deciphered text of XLY20 is finalized.

```
XLY20: XLYPISNANIRUSFTFWMIY ◊ QUIZ BUYER[/ASH OK/A CLUE/BY NUM] TO BYE
     ANY OPP
```

The deciphered text implies that the buyer will be challenged with quiz to reject any illegitimate opponent.

Full List of the Decrypted CGB Ciphers

Gold Bar (with text only)	**KO 08124**	**KO 00803 (or 00808)**
JKGFIJPMCWSAEK (ACT DMZ [EFG] SIG CT) SKCDKJCDJCYQSZKTZJPXPWIRN (BAY ADJ GATA MAST WY STACKY LOT) MQOLCSJTLGAJOKBSSBOMUPCE (USED JUD BORIS GOV ID P KEY SEA) FEWGDRHDDEEUMFFTEEMJXZR (OUST GOV P BANK CT GOL EES VOG) RHZVIYQIYSXVNQXQWIOVWPJO (PAY KEY IN USS KAN UP IF DUE ASK) MQOLCSJTLGAJOKBSSBOMUPCE (USED JUD BORIS GOV ID P KEY SEA) FEWGDRHDDEEUMFFTEEMJXZR (OUST GOV P BANK CT GOL EES VOG) SKCDKJCDJCYQSZKTZJPXPWIRN (BAY ADJ GATA MAST WY STACKY LOT) RHZVIYQIYSXVNQXQWIOVWPJO (PAY KEY IN USS KAN UP IF DUE ASK) MQOLCSJTLGAJOKBSSBOMUPCE (USED JUD BORIS GOV ID P KEY SEA) SKCDKJCDJCYQSZKTZJPXPWIRN (BAY ADJ GATA MAST WY STACKY LOT) (UPSIDE DOWN) MLMTAHGBGFNIV (A GOVT CIV BOU CO) ZUQUPNZN (XOR ID IWA) ABRYCTUGVZXUPB (XOR ID IWA) ◊ ABRYCTUGVZXIWA (REPAY US OGD WHEN) MVERZRLQDBHQ (TWO MSG XOR SRH) GKJFHYXODIE (AZH LATE TOLD) UGMNCBXCRLDEY (US VET SWW SIG GO) HFXPCQYZVATXAWIZPVE (US BERGE GOV KEY JDS IAA)	(LEFT) MQOLCSJTLGAJOKBSSBOMUPCE (USED JUD BORIS GOV ID P KEY SEA) FEWGDRHDDEEUMFFTEEMJXZR (OUST GOV P BANK CT GOL EES VOG) YQHUDTABGALLOWLS (Not deciphered, see YQV16.) VIOHIKNNGUAB (Not deciphered, see YQV16.) MVERZRLQDBHQ (TWO MSG XOR SRH) ◊ YQVIOHIKNNGUABLS (YQV16) (CONTACT FWT DEWVRY) ZUQUPNZN (XOR ID IWA) VIOHIKNNGUAB (Not deciphered) JKGFIJPMCWSAEK (ACT DMZ EFG SIG CT) SKCDKJCDJCYQSZKTZJPXPWIRN (BAY ADJ GATA MAST WY STACKY LOT) (RIGHT) MLMTAHGBGFNIV (A GOVT CIV BOU CO) FEWGDRHDDEEUMFFTEEMJXZR (OUST GOV P BANK CT GOL EES VOG) RHZVIYQIYSXVNQXQWIOVWPJO (PAY KEY IN USS KAN UP IF DUE ASK) HFXPCQYZVATXAWIZPVE (US BERGE GOV KEY JDS IAA) KOWVRSRWTMLDH (ACC NO ASH KEEP I)	(MIDDLE) UGMNCBXCRLDEY (US VET SWW SIG GO) RHZVIYQIYSXVNQXQWIOVWPJO (PAY KEY IN USS KAN UP IF DUE ASK) SKCDKJCDJCYQSZKTZJPXPWIRN (BAY ADJ GATA MAST WY STACKY LOT) MQOLCSJTLGAJOKBSSBOMUPCE (USED JUD BORIS GOV ID P KEY SEA) FEWGDRHDDEEUMFFTEEMJXZR (OUST GOV P BANK CT GOL EES VOG) (RIGHT) VIOHIKNNGUAB (Not deciphered) HFXPCQYZVATXAWIZPVE (US BERGE GOV KEY JDS IAA) YQHUDTABGALLOWLS (Not deciphered) XLYPISNANIRUSFTFWMIY (QUIZ BUYER/ASH OK/A CLUE/BY NUM TO BYE ANY OPP) KOWVRSRWTMLDH (ACC NO ASH KEEP I) JKGFIJPMCWSAEK (ACT DMZ EFG SIG CT) ABRYCTUGVZXUPB (XOR ID IWA) ◊ ABRYCTUGVZXIWA (REPAY US OGD WHEN) GKJFHYXODIE (AZH LATE TOLD) ZUQUPNZN (XOR ID IWA) GKJFHYXODIE (AZH LATE TOLD)

* Unless the original text was known to the public, I believe that the recovered text presented here would be a very close version of it. I have to say that the methods I discovered also might not be the methods actually used in the generation of the CGB ciphers, but I am confident with them.