SSC Methodology

The Spanish Strip Cipher (SSC) is ***a homophonic substitution cipher***, in which a plaintext letter not only maps to one cipher text character (as in monoalphabetic substitution ciphers), but it can map to different ones. In this kind of ciphers, the cipher text characters are called **homophones**, which are arranged in a table, where each column is mapped by one letter of the plaintext alphabet. During the Spanish civil war (1936-1939) this method was widely adopted by both sides, Republicans and Nationalists.

Normally, the number of homophones in a column is related with the **frequency** of a plaintext letter. For example, in an English text, the letter **E** occurs with a frequency of **12.7%** approximately. On the other hand, the letter **X** approximately occurs with a frequency of **0.074%.** Thus, the column (*strip*) assigned to the letter **E** should contain more homophones than the column assigned to the letter **X**. In this way, frequency analysis attacks become more difficult.

In addition to the homophones table, the SSC encompasses three more elements: A **random alphabet**, a **keyword**, which is used to generate the random alphabet, and an **initial position** that is used to shift the random alphabet.

Encryption:

In order to encrypt a plaintext, sender and receiver agree on a key which consists of three elements: a keyword, a homophones table, and an initial position. After generating and shifting the random alphabet, the encryption can begin. For each plaintext letter:

1. We look for the same letter in the random alphabet.
2. We substitute the plaintext letter by one the homophones of the same column of the random-alphabet letter.

Decryption:

The decryption is a straightforward process, in which each cipher text homophone is replaced by its corresponding letter of the random alphabet

Example:

Language: English.

No. of Ordered Alphabet: 26 letters.

Keyword: Yahya.

Initial Pos.: 1.

Plain Text: Hello.

Encryption:

1. **Keyword:** First remove all duplicates from the keyword: so Yahya → Yah
2. **Generate Random Alphabet:**
   The ordered alphabet is sequentially written under the remaining letters of the keyword ignoring the letters which already occur in the keyword.

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

   Then, each column of the above table is placed horizontally. The resulting random alphabet is:

   YBEILORUXACFJMPSVZHDGKNQTW

3. **Initial Position:** used to shift the random alphabet, in this example we shift it 1 position because Initial pos. = 1.
   The Final Random Alphabet:

   WYBEILORUXACFJMPSVZHDGKNQT

4. **The Homophones Table:** numbers in two-digits format start from 00 to 99. The columns in the Homophones Table are the strips in which each strip is assigned to one letter in the random alphabet.

The more frequent the letter is in the language; the bigger strip should be assigned to it.

Numbers in each strip are randomly generated to fill out each strip.

WYBEILORUXACFJMPSVZHDGKNQT

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
W	Y	B	E	I	L	O	R	U	X	A	C	F	J	M	P	S	V	Z	H	D	G	K	N	Q	T
51	38	06	39	81	02	88	79	28	87	92	98	93	45	63	32	59	48	20	77	36	47	24	60	40	41
80	29	67	53	95	61	76	56	21	10	70	96	33	55	30	90	44	34	69	27	14	17	11	91	97	64
50	85	05	78	54	57	13	75	86		22	46	42		49	72	94	83		65	62	19		99		03
			73	84	37	18	31			23						89			35	74			07		15
			82	58	26	00	04			08						66			16	68			01		52

Encryption:

In order to encrypt the word for example “Hello” we look at the Random Alphabet which is in the red color and we randomly map every letter in our plain text with one number from the strip assigned to that letter.

T	D	F	G
H	E	L	O
77	39	02	88
27	53	61	76
65	78	57	13
35	73	37	18
16	82	26	00

Hello → 16 78 61 37 88

An improvement I added here and which should strengthen the cipher and confuse the hacker is: we pick up randomly from 3 symbols, a symbol to replace each digit, even the spaces are allocated symbols for more security.

Symbols:

```
Space:{"#","$",";"}, 0:{"!","@","€"}, 1:{"~","|", "'"},
 2:{"%","^","["}, 3:{"&","*","]"}, 4:{"(",")","{"}, 5:
  {"-","+","}"}, 6:{"=","/","£"}, 7:{"\\","\"",";"},
          8:{"<",">","¿"}, 9:{".",":","±"}.
```

So in our example: the digit “1” can be replaced randomly with one of the following symbols: ~ or | or ' …. And so on for the rest of the digits.

Example of the *Cipher Text* for the plain text “Hello” is:

|=\¿=~*"¿<

Decryption:

1. First we unhide the homophones(digits) by using the same Symbols Table above
2. Via the unique key we generate the Random Alphabet.
3. We map each homophone in the cipher text with the corresponding letter in the Random Alphabet.

|=\¿=~*"¿ → 16 78 61 37 88 → HELLO.