**Network Security**

Assignment 1: Project **0**

**Herschelle Gupta 2019045 | Ritik Vatsal 2019321**

**Character Set (C):**

Character set of the language, L contains 3 symbols - {A, B, C}

**Encoding:**

To convert the plaintext into binary, the following encoding is used:

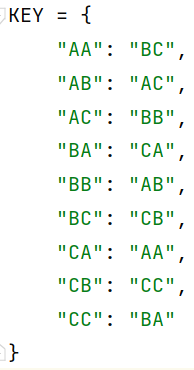
A -> 00

B -> 01

C -> 10

**Key (K):**

Key K, is defined as a permutation of 9 values - {AA, AB, AC, BA, BB, BC, CA, CB, CC} mapping to itself. For example -



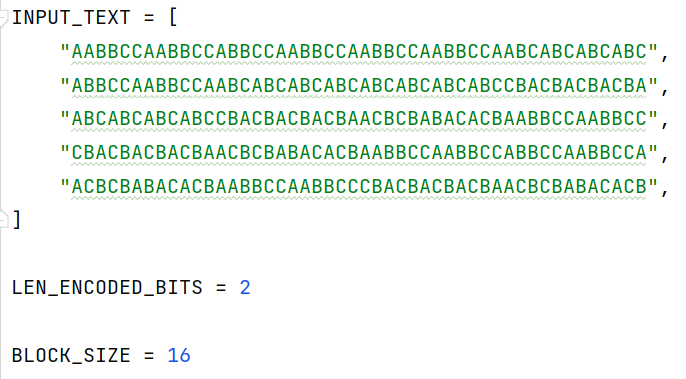
Here, AA is mapped to BC, AB is mapped to AC and so on.

Values on the left (of colon) are fixed, while values on the right are a specific permutation of the values on the left.

Key space consists of 9! = 362880 values

**Plaintext:**

Input text consists of 5 plaintexts. A single character is encoded in 2 bits.

****

**Hash Function:**

Block size of 16 is used and number of rounds depends on length of plaintext divided by block size.

In each round, hash value obtained is equal to bit i in current hash XORed bit i in previous hash

More formally, let H(i) denote hash value in i th round then,

H(i) = Plaintext (in binary) if i = 1

H(i) = H(i) XOR H(i-1) for i= 2 to n (Here XOR denotes bitwise XOR)

**Recognisability:**

To make the plaintext “recognisable” we use the following template for ciphertext -

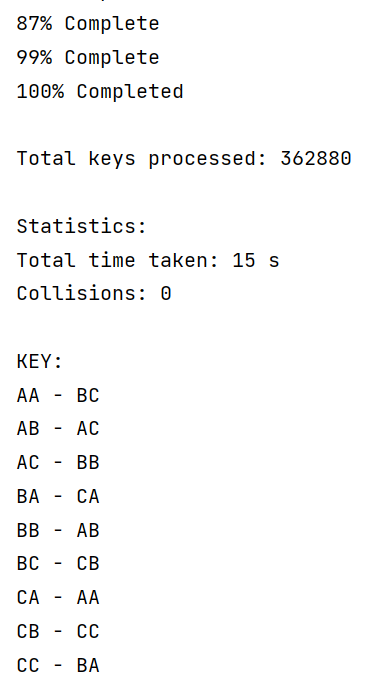
**Plaintext** (*original*) + **Hash** *of Plaintext (original)* = **New Plaintext** (*to be encrypted*)

Example -  
ACCBAAACCBAAACAACBACAACBACAACBACAACBCCBBCACCBBCA ***AABCCABC*  |< HASH >|**

Assumption here is that the length of the plaintext is known.

**Input and Output:**

Input to our system are 5 randomly chosen plaintexts of equal lengths and a key, output shows some statistics and cracked key.



**\*\*\***