**The Lu-Shuffle Algorithm**

1. Given a sequence of N items, e.g. S={a, b, c}
2. Create a blank sequence of similar length, S\*={\_,\_,\_}
3. Given a seed number K such that 1 <= K <= |S|, e.g. K = 2
4. Select the K-th term in S e.g. S[K] -> S[2] = b
5. If K is even (K%2 == 0): shift the selected term K+1 positions in reverse direction to the first blank position in S\* starting from position K and jumping over positions in S\* that are already occupied, wrapping around if necessary. e.g. we move ‘b’ as such: 2 -> 1 -> 3, so that S\* = {\_,\_,b}
   1. Set K to that new position. E.g. K= 3
   2. Jump to Instruction #7
6. Else: If K is odd (K%2==1): shift the selected term K+1 positions in forward direction to the first blank position in S\* starting from position K and jumping over positions in S\* that are already occupied, wrapping around if necessary.
   1. Set K to that new position.
   2. Jump to Instruction #7
7. If |S\*| = |S| meaning all items in S have been successfully processed, then return S\* as the shuffled list. Otherwise Loop from Instruction #4.

Thus, applying this algorithm to the set S={a,b,c} and initial seed K = 2, we obtain the following states:

S = {a,b,c} | S\* = {\_,\_,\_}

K = 2 | S[K] = S[2] = b 🡪 S\* = {\_,\_,b} | K\* = 3

K = 3 | S[K] = S[3] = c -> S\* = {c,\_,b} | K\*=1

K = 1 | S[K] = S[1] = a -> S\*={c,a,b} | END

If S = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, and K = 4

K = 4 | S[K] = S[4] = 3 -> S\* = {\_,\_,\_,\_,\_,\_,\_,\_,\_,3} | K\*= 10

K = 10| S[K] = S[10] = 9 -> S\* = {\_,\_,\_,\_,\_,\_,\_,\_,9,3} | K\*=9

K = 9| S[K] = S[9] = 8 -> S\* = {8,\_,\_,\_,\_,\_,\_,\_,9,3} | K\*=1

K = 1| S[K] = S[1] = 0 -> S\* = {8,0,\_,\_,\_,\_,\_,\_,9,3} | K\*=2

K = 2| S[K] = S[2] = 1 -> S\* = {8,0,\_,\_,\_,\_,1,\_,9,3} | K\*=7

K = 7| S[K] = S[7] = 6 -> S\* = {8,0,6,\_,\_,\_,1,\_,9,3} | K\*=3

K = 3| S[K] = S[3] = 2 -> S\* = {8,0,6,\_,\_,2,1,\_,9,3} | K\*=6

K = 6| S[K] = S[6] = 5 -> S\* = {8,0,6,\_,\_,2,1,5,9,3} | K\*=8

K = 8| S[K] = S[8] = 7 -> S\* = {8,0,6,7,\_,2,1,5,9,3} | K\*=4

K = 4| S[K] = S[4] = 3 -> S\* = {8,0,6,7,3,2,1,5,9,3} | END

If S = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, and K = 3

K = 3 | S[K] = S[3] = 2 -> S\* = {\_,\_,\_,\_,\_,2,\_,\_,\_,\_} | K\*= 6

K = 6 | S[K] = S[6] = 5 -> S\* = {\_,\_,\_,\_,\_,2,\_,\_,\_,5} | K\*= 10

K = 10| S[K] = S[10] = 9 -> S\* = {\_,\_,\_,\_,\_,2,\_,9,\_,5} |K\*= 8

K = 8 | S[K] = S[8] = 7 -> S\* = {\_,\_,\_,\_,\_,2,7,9,\_,5} |K\*= 7

K = 7 | S[K] = S[7] = 6 -> S\* = {\_,\_,\_,\_,\_,2,7,9,6,5} |K\*= 9

K = 9 | S[K] = S[9] = 8 -> S\* = {\_,\_,\_,8,\_,2,7,9,6,5} |K\*= 4

K = 4 | S[K] = S[4] = 3 -> S\* = {\_,\_,\_,8,3,2,7,9,6,5} |K\*= 5

K = 5 | S[K] = S[5] = 4 -> S\* = {4,\_,\_,8,3,2,7,9,6,5} |K\*= 1

K = 1 | S[K] = S[1] = 0 -> S\* = {4,0,\_,8,3,2,7,9,6,5} |K\*= 2

K = 2 | S[K] = S[2] = 1 -> S\* = {4,0,1,8,3,2,7,9,6,5} |END

If S = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, and K = 2

K = 2 | S[K] = S[2] = 1 -> S\* = {\_,\_,\_,\_,\_,\_,\_,\_,\_,1} | K\*= 10

K = 10| S[K] = S[10] = 9 -> S\* = {\_,\_,\_,\_,\_,\_,\_,\_,9,1} |K\*= 9

K = 9| S[K] = S[9] = 8 -> S\* = {8,\_,\_,\_,\_,\_,\_,\_,9,1} |K\*= 1

K = 1| S[K] = S[1] = 0 -> S\* = {8,0,\_,\_,\_,\_,\_,\_,9,1} |K\*= 2

K = 2| S[K] = S[2] = 1 -> S\* = {8,0,\_,\_,\_,\_,\_,\_,9,1} |K\*= 2

UPDATED LGES Algorithm: Plato-Cipher Step:

So, when a sequence such as S = {A, C, A, T, T, G} needs to be expressed in the Lu Genome Expression System, we proceed as such:

1. Compute the FGE prefix: IFA by encoding S in base-OZIN to produce OZ(S)
2. Compute the IFA MSS from S: MSS(OZ(S))
3. Compute the Cardinality of the IFA MSS: k = |MSS(OZ(S))|
4. Use that K to generate the corresponding Platonic Form Cipher (PFC\*) by shuffling the order of the forms in the standard PF-cipher using the Lu-Shuffle Algorithm and the provided K.
5. Then, encode IFA MSS into platonic form to generate the PFA by mapping each element w[i] in MSS(OZ(S)) to PFC\*[i] such that: PFA[i] = PFC\*[i] + j x w[i] where j = I(PFC[i],PFC) --🡪 so that, if the PFC\*[1] maps to j = 3, then PFA[1] shall be a 3 sided polygon with 3 attachments of w[1]
6. Finally, Encode IFA MSS in OZ to generate GES: GES = OZ(MSS(IFA))
7. Return FGE = OZ(S).PFA.GES