Kaushik-Encryption : An cryptographic algorithm to encrypt messages

About : A purely ASCII & arithmetic based cryptographic algorithm to encrypt the messages while maintaining a state to decrypt the message. The decryption of the message can be only done by the CPU where the state is in existence which means no other CPU can decrypt it. The architecture of the algorithm might sound useless but the state is shareable with the other CPU over the network which lets them decrypt too.

To be noted : There will be 2 architecture for decryption of the message :

1) Decryption in same CPU
2) Decryption in other CPU

Encryption Algorithm :

m = message, k = key;

Function will take 2 values *E(m,k)* & extract *ASCII* for *m* & *k*

Calculation :
n = ASCII value of character of message, i = ASCII value of character of key;

On each ***n*** there will be arithmetic operation followed by ***i*** to manipulate the numbers

```
if m > k || m == k
        m = m + k;
else
        m = m - k;
```

While doing the upper computation maintain an ***linear structure*** to have logs for ***Arithmetic sequence*** (whether to add or sub). **The sequence will be opposite of what the operation will be if add then in log it will be sub**.Now Append any random Hexa-Alphabet (A-F) at the end of the manipulated number.

Note :

s = Length of structure maintaining sequence, ***l*** = Length of message;
n = Number of arithmetic operations on each message character ASCII,
kl = Length of key

s = l;
n = kl;

Proof for s = l:
In calculation formula was used to operate on each m (each ASCII of message). Therefore, the number of logs in the structure will equal the length of the message.

Proof for n = kl:
In the calculation formula used each k (each ASCII of key) to change the m. Therefore, each log in the structure will have the same length as the length of the key.

Decryption Algorithm :

em = encrypted message, k = key;
logs = structure, x = log index;

Function will take 2 values *E(em,k) & will* segregate Hexa-Alphabets from *em.*

Calculation :
On each ***em*** there will be arithmetic operation followed by ***k*** to manipulate the numbers to get back the actual message

```
if log[ x ] == “plus”
        em = em + k;
else
        em = em - k;
```

At the *em* convert the ASCII to string

ishaankaushik91@gmail.com
https://encrypt.ishaankaushik.com