

The formula for the encryption algorithm

Bartosz Burgiel

Variables

Let str be the string of the length n_{str} be the string that will be encrypted.

Let key be the key of the length n_{key} that the string str will be encrypted with.

Let s be the sum of all characters' values as in $s = \sum_i^n str[i]$

Let the byte b be equal to the value of a byte of the string str with the index i

Let the byte k be equal to the value of a byte of the string key with the index i

Let the integer i be the index of the byte that is currently being encoded.

Let the integer val be the base for the encryption of each byte of the string str being equal to $val = b * n_{key}$

Formula

$$b_{encrypted} = \left(\sum_{i=0}^{n_{key}-1} \frac{2^k}{sum + i_{key} + j} + i_{key} \right) + sum * n_{key}$$

Code

```
func encodeByte(b byte, key string, i int) string {
    var temp uint64 = uint64(b)

    // sum of all key letters
    sum := 0
    for _, s := range key {
        sum += int(s)
    }

    // iterate over the key
    for j, k := range key {
        temp += uint64(math.Pow(2.0, float64(k))) / uint64(sum+i+j)
    }

    temp += uint64(sum * len(key))
    temp += uint64(i * len(key))

    // return temp in hexadecimal
    return fmt.Sprintf("%x-", temp)
}
```