Login   Forum

Excel	Encrypt and decrypt strings using a Xor algorithm	☆
Ease of Use	Easy	
Version tested with	2003	
Submitted by:	RichardSchollar	
Description:	Encrypts or decrypts a string of text to a string of characters based on a user-specified encryption key.	
Discussion:	The function makes use of bitwise Xor comparisons on the string to be encrypted and a string key provided by the user. This could be useful if you wish to send sensitive information to someone, or want to store sensitive information in a workbook, and wish to use more protection than the standard Excel interface allows (eg worksheet protection). Others can still access the workbook using this method, but if they do not know the key used to encrypt the data, it is very difficult to decipher what the encrypted string is. The function adds one to the individual byte values of the generated encrypted string so that returning Ascii character 0 is avoided (Excel will not display this character and it thus causes encryption to fail). Longer Keys provide better encryption protection (as do longer strings to encrypt). Thanks to Mdmackillop for suggested (and incorporated) improvement to the function.	

Code: instructions for use

```vba
Option Explicit
Sub test()
    'this sub is only present to demonstrate use of the function!
    'it is not required to use the function.
    Dim r As Range, retVal, sKey As String
    sKey = Application.InputBox("Enter your key", "Key entry", "My Key", , , , , 2)
    retVal = MsgBox("This is the key you entered:" & vbNewLine & Chr$(34) & sKey & Chr$(34) & vbNewLine & _
    "Please confirm OK or Cancel to exit", vbOKCancel, "Confirm Key")
    If retVal = vbCancel Then Exit Sub
    For Each r In Sheets("Sheet1").UsedRange
        If r.Interior.ColorIndex = 6 Then
            r.Value = XorC(r.Value, sKey)
        End If
    Next r
End Sub

Function XorC(ByVal sData As String, ByVal sKey As String) As String
    Dim l As Long, i As Long, byIn() As Byte, byOut() As Byte, byKey() As Byte
    Dim bEncOrDec As Boolean
    'confirm valid string and key input:
    If Len(sData) = 0 Or Len(sKey) = 0 Then XorC = "Invalid argument(s) used": Exit Function
    'check whether running encryption or decryption (flagged by presence of "xxx" at start of sData):
    If Left$(sData, 3) = "xxx" Then
        bEncOrDec = False 'decryption
        sData = Mid$(sData, 4)
    Else
        bEncOrDec = True 'encryption
    End If
    'assign strings to byte arrays (unicode)
    byIn = sData
    byOut = sData
    byKey = sKey
    l = LBound(byKey)
    For i = LBound(byIn) To UBound(byIn) - 1 Step 2
        byOut(i) = ((byIn(i) + Not bEncOrDec) Xor byKey(l)) - bEncOrDec 'avoid Chr$(0) by using bEncOrDec flag
        l = l + 2
        If l > UBound(byKey) Then l = LBound(byKey) 'ensure stay within bounds of Key
    Next i
    XorC = byOut
```

```
     If bEncOrDec Then XorC = "xxx" & XorC 'add "xxx" onto encrypted text
End Function
```

How to use:	1. The functions take two arguments - the string to encrypt/decrypt and the key to use for the encyption/decryption.
	2. It can be activated within code or used in the worksheet itself (but note that keeping the key used in the worksheet rather defeats the object of the encryption code - you would probably want to copy the formula results +paste values).
	3. Cell usage eg in B1:
	4. =XorC(A1,"My Key")
Test the code:	1. Open the attached zip file and click on the button on sheet1 and follow the prompts.
Sample File:	EncryptDecryptFunction.zip 10.91KB

Approved by mdmackillop

Login Forum