

Eine Excel-Mappe mit Office 2019 für Verschlüsselungsmakros anlegen

Bitte beachten: ich gebe hier keine Anleitung, wie man Excel bedient oder VBA programmiert bzw. wie man mit dem Editor umgeht. Alles, was Ihr ab hier mit Eurem Programm macht, geschieht auf eigenes Risiko!

1. EXCEL mit einer neuen Mappe starten. Ein Arbeitsblatt umbenennen in „CryptoRW“. Diese Arbeitsmappe speichern, wichtig: als Endung „.xlsm“ verwenden (Excel Arbeitsmappe mit Makros).
2. In Spalte C1 in diese Formel einfügen: =LÄNGE(A1)
Die Formel dreimal nach unten kopieren, also in C2 bis C4
3. Den VBA-Editor starten. Ein Modul einfügen.
4. Folgende drei Makros einfügen:

Public Sub Generate ()

```
Dim nLow As Integer, nHigh As Integer, cChar As String, cLine As String, k As Integer, y As Integer, nZeichen As Integer
```

```
Dim oWS As Worksheet, oWB As Workbook
```

```
Set oWB = ThisWorkbook
```

```
Set oWS = oWB.Worksheets("CryptoRW")
```

```
nLow = 32 ' Starte mit Leerzeichen
```

```
nHigh = 255 '
```

```
nZeichen = oWS.Cells(1, 3) ' Wieviel Zeichen brauche ich für das OTP
```

```
cLine = ""
```

```
For y = 1 To nZeichen
```

```
    Randomize
```

```
    k = Int((nHigh - nLow + 1) * Rnd + nLow)
```

```
    cLine = cLine + Chr(k)
```

```
    oWS.Cells(2, 1) = cLine
```

```
Next y
```

```
' Debug.Print cLine
```

```
oWS.Cells(2, 1) = cLine
```

```
Call enkodieren
```

End Sub

Public Sub enkodieren()

```
Dim oWS As Worksheet, oWB As Workbook, i As Integer, nTemp As Integer, cQuelltext  
As String, cSchluesseltext As String, cChiffprat As String
```

```
Dim cTempzeichen As String, cTempKey As String, nTempkey As Integer, nChiffre As  
Integer
```

```
Set oWB = ThisWorkbook
```

```
Set oWS = oWB.Worksheets("CryptoRW")
```

```
cQuelltext = oWS.Cells(1, 1)
```

```
cSchluesseltext = oWS.Cells(2, 1)
```

```
cChiffprat = ""
```

```
For i = 1 To Len(cQuelltext)
```

```
    cTempzeichen = Mid(cQuelltext, i, 1)
```

```
    nTemp = Asc(cTempzeichen)
```

```
    If nTemp < 32 Then
```

```
        MsgBox ("Zeichen unterhalb von ASCII-Code 32 (Leerzeichen) entdeckt,  
Funktion wird abgebrochen.")
```

```
        Exit Sub
```

```
    End If
```

```
    cTempKey = Mid(cSchluesseltext, i, 1)
```

```
    nTempkey = Asc(cTempKey)
```

```
    nChiffre = nTemp + nTempkey
```

```
    If nChiffre > 255 Then
```

```
        nChiffre = nChiffre - 255
```

```
    End If
```

```
    cChiffprat = cChiffprat & Chr(nChiffre)
```

```
    oWS.Cells(3, 1) = cChiffprat
```

```
    ' Debug.Print nTemp & " wird zu " & nTempkey
```

```
Next i
```

```
Call dekodieren
```

End Sub

Private Sub dekodieren()

```
Dim oWS As Worksheet, oWB As Workbook, i As Integer, nTemp As Integer, cQuelltext  
As String, cSchluesseltext As String, cChiffprat As String
```

```
Dim cTempzeichen As String, cTempKey As String, nTempkey As Integer, nKlartext As  
Integer
```

```
Set oWB = ThisWorkbook
```

```
Set oWS = oWB.Worksheets("CryptoRW")
```

```
cSchluesseltext = oWS.Cells(2, 1)
```

```
cChiffprat = oWS.Cells(3, 1)
```

```
cQuelltext = ""
```

```
For i = 1 To Len(cChiffprat)
```

```
    cTempzeichen = Mid(cChiffprat, i, 1)
```

```
    nTemp = Asc(cTempzeichen)
```

```
    cTempKey = Mid(cSchluesseltext, i, 1)
```

```
    nTempkey = Asc(cTempKey)
```

```
    ' Abfrage wenn negativ später einbauen
```

```
    nKlartext = nTemp - nTempkey
```

```
    If nKlartext < 32 Then
```

```
        nKlartext = nKlartext + 255
```

```
    End If
```

```
    ' Debug.Print nTemp & " - " & nTempkey & " = " & (nTemp - nTempkey)
```

```
    cQuelltext = cQuelltext & Chr(nKlartext)
```

```
    oWS.Cells(4, 1) = cQuelltext
```

```
Next i
```

End Sub

5. Eine Schaltfläche auf das Arbeitsblatt legen, die mit dem Makro Generate() verknüpft wird. Dann laufen alle Schritte bequem mit einem Klick.
6. In der Zelle A1 kann man den zu verschlüsselnden Text eingeben.
7. Die Funktion ist wie folgt:
 - a. Das Makro „Generate()“ schaut in C1, wie lang der Quelltext ist. Genauso lang wird der Schlüsseltext. Der Schlüsseltext wird aus den Zeichen 32 bis 255 generiert. Vor jedem Durchlauf wird der Zufallszahlen-Generator mit „randomize“ neu initialisiert.

- b. Das Makro „enkodieren()“ läuft zeichenweise durch den Quelltext und addiert den Schlüssel auf. Überschreitet der Wert 255, ist das kein zulässiges Zeichen. Dann wird 255 abgezogen. Zeichen unterhalb von 32 sind Steuerzeichen für den PC. Da befinden sich auch Absatzmarken etc., weswegen diese Zeichen nicht – jedenfalls nicht mit diesem unprofessionellen Makro – zu verschlüsseln sind.
- c. Danach startet das Makro „dekodieren()“ und entschlüsselt das Chiffre zur Kontrolle wieder.
- d. Das Verschlüsseln /Entschlüsseln ginge natürlich wesentlich schneller, wenn man die Aktualisierung der Zellen unterläßt und das erst zum Schluß erledigt. So ist es wenigstens noch was für's Auge 😊

Viel Spaß,

Andreas Janke

P.S.: Laßt die Tante Frieda ja in Ruhe!