

Předpokládejme, že tentýž sešit sdílí několik uživatelů, každý s jinými pravomocemi a funkcí. Údaje na jednom listu přitom chceme zobrazovat jen určeným osobám. Tohle za nás bezpečně nevyřeší zámeček listu ani snadno zobrazitelné (super)skryté listy. Nabízí se ovšem jedna převážně matematická disciplína – šifrování.

Jestliže matkou algoritmů je funkce MOD, pak pro šifrování je jí funkce XOR, známá z oblasti booleovské algebry a logiky. Pokud ji zkombinujeme s možností šifrovacího klíče (zjednodušeně hesla), dostaneme účinnou zbraň (ale také sobě nebezpečnou v případě zapomenutí klíče). Tím, že zveřejním kód pro šifrování v jinak děravém VBA, se také příliš nezmění. Bez znalosti klíče je rozluštění „ve hvězdách“.



Odkryté hodnoty a vzorce na listu



Zašifrované hodnoty a vzorce na listu

Pozn. Nejsem si jistý, proč se při šifrování mění barva v grafu, nicméně podoba grafu závislého na tabulce se samozřejmě změní také. Texty bez provázanosti na hodnoty v buňkách zohledněny nejsou.

Důrazně upozorňuji, že aplikování níže uvedeného kódu provádíte „na vlastní triko“. Teoreticky je limitní množství znaků v buňce, nicméně běžný sešit by neměl v buňkách obsahovat více jak 255 znaků (novodobé verze Excelu mají tento limit už řádově jinde). Procedury byly vyzkoušeny na formátovaných hodnotách (číslo, datum, text) i na vzorcích. O případných chybách mě informujte.

```

1 'http://www.freevbcode.com/ShowCode.asp?ID=5676
2
3
4 'autor níže uvedeného kódu nenese žádnou odpovědnost
5 'za škody způsobené jeho užíváním
6 'doporučena je nesířovaná záloha sešitu
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129

```

```

'http://www.freevbcode.com/ShowCode.asp?ID=5676

'autor níže uvedeného kódu nenese žádnou odpovědnost
'za škody způsobené jeho užíváním
'doporučena je nesířovaná záloha sešitu

Private Sub Sifrovani()
    Dim strKlic As String
    Dim rngOblast As Range
    Dim rngBunka As Range

    'strKlic = InputBox("Zadejte heslo:", "Sifrování listu")

    strKlic = "Ježibaba"

    'zákaz překreslování
    Application.ScreenUpdating = False

    'zákaz přepočítávání listu
    Application.Calculation = xlCalculationManual

    'z využití oblasti listu
    With ActiveSheet.UsedRange
        'převzetí neprázdných buněk (konstanty, vzorce)
        Set rngOblast = Union(.SpecialCells(xlCellTypeConstants), _
            .SpecialCells(xlCellTypeFormulas))

    End With

    'pro každou buňku
    For Each rngBunka In rngOblast

        'zašifrování obsahu
        rngBunka.Value = XOREncryption(strKlic, rngBunka.FormulaLocal)

    Next rngBunka

    'povolení přepočítávání listu
    Application.Calculation = xlCalculationAutomatic

    'povolení překreslování
    Application.ScreenUpdating = True

End Sub

Private Sub Desifrovani()
    Dim strKlic As String
    Dim rngOblast As Range
    Dim rngBunka As Range

    'strKlic = InputBox("Zadejte heslo:", "Dešifrování listu")

    strKlic = "Ježibaba"

    'zákaz překreslování
    Application.ScreenUpdating = False

    'zákaz přepočítávání listu
    Application.Calculation = xlCalculationManual

    'z využití oblasti listu
    With ActiveSheet.UsedRange
        'převzetí neprázdných buněk (konstanty)
        Set rngOblast = .SpecialCells(xlCellTypeConstants)

    End With

    'pro každou buňku
    For Each rngBunka In rngOblast

        'dešifrování obsahu
        rngBunka.FormulaLocal = XORDecryption(strKlic, rngBunka.Value)

    Next rngBunka

    'povolení přepočítávání listu
    Application.Calculation = xlCalculationAutomatic

    'povolení překreslování
    Application.ScreenUpdating = True

End Sub

Public Function XOREncryption(CodeKey As String, DataIn As String) As String
    Dim IonDataPtr As Long
    Dim strDataOut As String
    Dim temp As Integer
    Dim tempstring As String
    Dim intXorValue1 As Integer
    Dim intXorValue2 As Integer

    For IonDataPtr = 1 To Len(DataIn)
        intXorValue1 = Asc(Mid$(DataIn, IonDataPtr, 1))
        intXorValue2 = Asc(Mid$(CodeKey, ((IonDataPtr Mod Len(CodeKey)) + 1), 1))
        temp = (intXorValue1 Xor intXorValue2)
        tempstring = Hex(temp)
        If Len(tempstring) = 1 Then tempstring = "0" & tempstring
        strDataOut = strDataOut + tempstring
    Next IonDataPtr

    XOREncryption = strDataOut

End Function

Public Function XORDecryption(CodeKey As String, DataIn As String) As String
    Dim IonDataPtr As Long
    Dim strDataOut As String
    Dim intXorValue1 As Integer
    Dim intXorValue2 As Integer

    For IonDataPtr = 1 To (Len(DataIn) / 2)
        intXorValue1 = Val("&H" & (Mid$(DataIn, (2 * IonDataPtr - 1), 2)))
        intXorValue2 = Asc(Mid$(CodeKey, ((IonDataPtr Mod Len(CodeKey)) + 1), _
            1))
        strDataOut = strDataOut + Chr(intXorValue1 Xor intXorValue2)
    Next IonDataPtr

    XORDecryption = strDataOut

End Function

```

V procedurách je klíč obsažen natvrdo a nad jím jen lehce v komentáři naznačen způsob, jak kód zadávat (InputBox). V praxi si heslo před šifrováním ověřujte (dvojití užití InputBoxu a kontrola), ať se vyvarujete chyby překlepu. Z pohledu fungování i bezpečnosti neexistuje žádná kontrola před pokusem o dešifrování špatným klíčem...

Hádanka s trochou nadsázky: Víte, kdo má dnes nejlepší přehled o šifrách a hodně z nich si osvojí? (Geokačeři, tj. hráči geocachingu, kteří luští mystery cache.)

Příloha:

[sifrovani_listu_xor.zip](#)