

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 Private Sub Sifrovani()
9
10 Dim strKlic As String
11 Dim rngOblast As Range
12 Dim rngBunka As Range
13
14 'strKlic = InputBox("Zadejte heslo:", "Šifrování listu")
15
16 strKlic = "Ježibaba"
17
18 'zákaz překreslování
19 Application.ScreenUpdating = False
20
21 'zákaz přepočítávání listu
22 Application.Calculation = xlCalculationManual
23
24 'z využití oblasti listu
25 With ActiveSheet.UsedRange
26
27     'převzetí neprázdných buněk (konstanty, vzorce)
28     Set rngOblast = Union(.SpecialCells(xlCellTypeConstants), _
29         .SpecialCells(xlCellTypeFormulas))
30
31 End With
32
33 'pro každou buňku
34 For Each rngBunka In rngOblast
35
36     'zašifrování obsahu
37     rngBunka.Value = XOREncryption(strKlic, rngBunka.FormulaLocal)
38
39 Next rngBunka
40
41 'povolení přepočítávání listu
42 Application.Calculation = xlCalculationAutomatic
43
44 'povolení překreslování
45 Application.ScreenUpdating = True
46
47 End Sub
48
49 Private Sub Desifrovani()
50
51 Dim strKlic As String
52 Dim rngOblast As Range
53 Dim rngBunka As Range
54
55 'strKlic = InputBox("Zadejte heslo:", "Dešifrování listu")
56
57 strKlic = "Ježibaba"
58
59 'zákaz překreslování
60 Application.ScreenUpdating = False
61
62 'zákaz přepočítávání listu
63 Application.Calculation = xlCalculationManual
64
65 'z využití oblasti listu
66 With ActiveSheet.UsedRange
67
68     'převzetí neprázdných buněk (konstanty)
69     Set rngOblast = .SpecialCells(xlCellTypeConstants)
70
71 End With
72
73 'pro každou buňku
74 For Each rngBunka In rngOblast
75
76     'dešifrování obsahu
77     rngBunka.FormulaLocal = XORDecryption(strKlic, rngBunka.Value)
78
79 Next rngBunka
80
81 'povolení přepočítávání listu
82 Application.Calculation = xlCalculationAutomatic
83
84 'povolení překreslování
85 Application.ScreenUpdating = True
86
87 End Sub
88
89 Public Function XOREncryption(CodeKey As String, DataIn As String) As String
90
91 Dim lonDataPtr As Long
92 Dim strDataOut As String
93 Dim temp As Integer
94 Dim tempstring As String
95 Dim intXorValue1 As Integer
96 Dim intXorValue2 As Integer
97
98
99 For lonDataPtr = 1 To Len(DataIn)
100     intXorValue1 = Asc(Mid$(DataIn, lonDataPtr, 1))
101     intXorValue2 = Asc(Mid$(CodeKey, ((lonDataPtr Mod Len(CodeKey)) + 1), 1))
102     temp = (intXorValue1 Xor intXorValue2)
103     tempstring = Hex(temp)
104     If Len(tempstring) = 1 Then tempstring = "0" & tempstring
105     strDataOut = strDataOut + tempstring
106 Next lonDataPtr
107
108 XOREncryption = strDataOut
109
110 End Function
111
112 Public Function XORDecryption(CodeKey As String, DataIn As String) As String
113
114 Dim lonDataPtr As Long
115 Dim strDataOut As String
116 Dim intXorValue1 As Integer
117 Dim intXorValue2 As Integer
118
119 For lonDataPtr = 1 To (Len(DataIn) / 2)
120     intXorValue1 = Val("&H" & (Mid$(DataIn, (2 * lonDataPtr - 1), 2)))
121     intXorValue2 = Asc(Mid$(CodeKey, ((lonDataPtr Mod Len(CodeKey)) + 1), _
122         1))
123     strDataOut = strDataOut + Chr(intXorValue1 Xor intXorValue2)
124 Next lonDataPtr
125
126 XORDecryption = strDataOut
127
128 End Function
129

```

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](#)