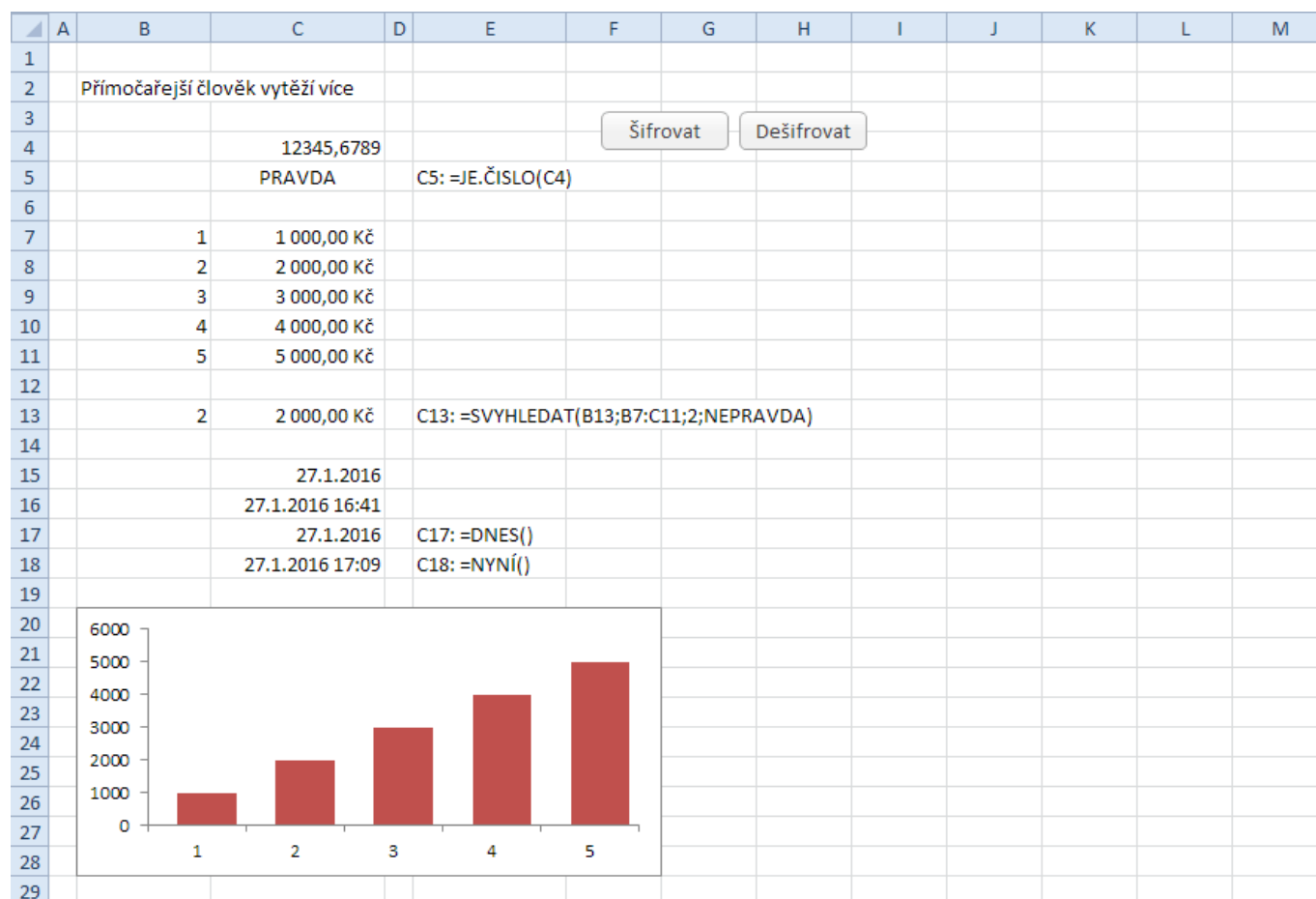
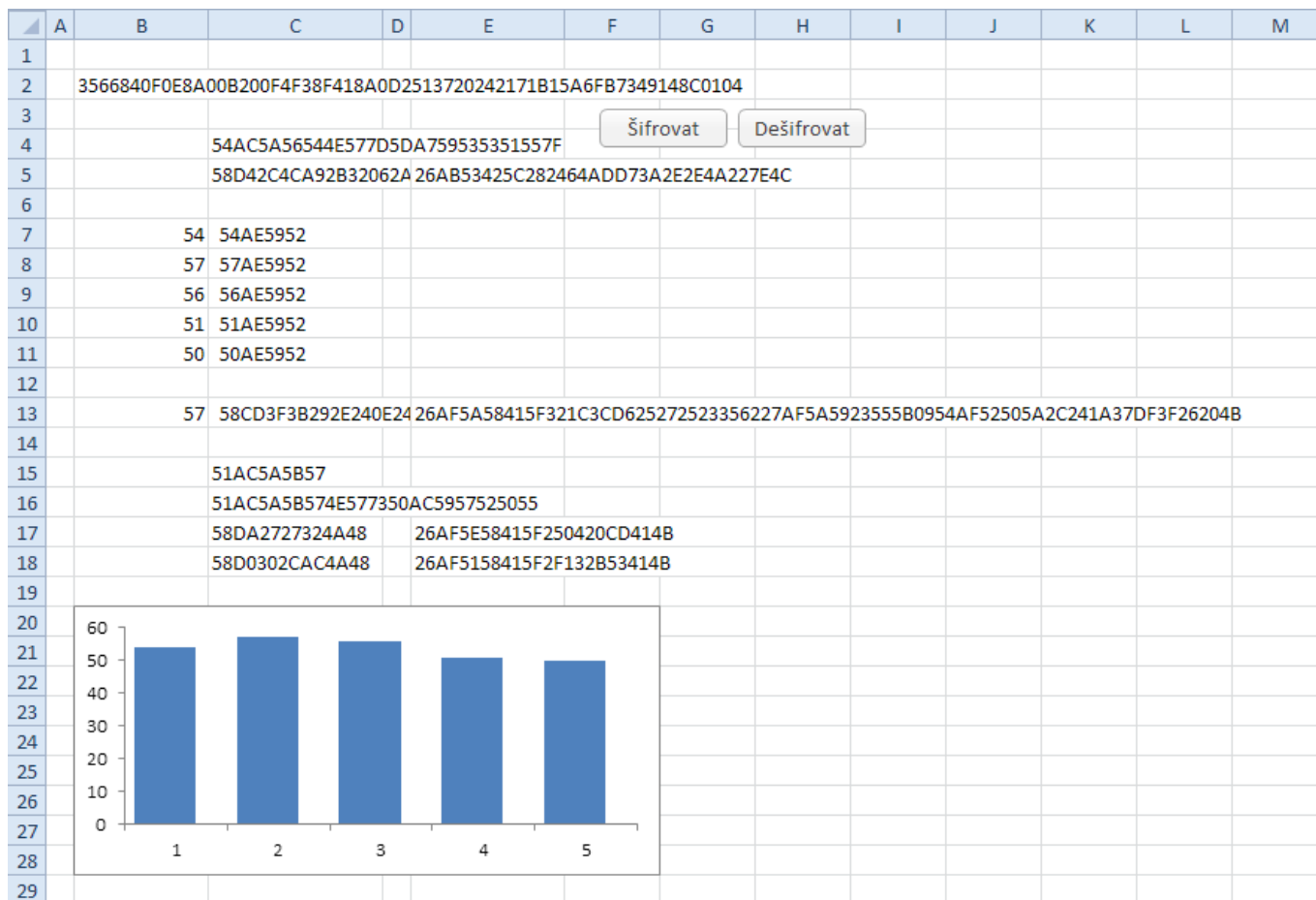


Předpokládejme, že tentýž sešit sdílí několik uživatelů, každý s jinými pravomocemi a funkcí. Údaje na jednom listu přitom chceme zobrazovat jen určeným osobám. Tohle za nás bezpečně nevyřeší zámeček listu ani snadno zobrazitelné (super)skryté listy. Nabízí se ovšem jedna převážně matematická disciplína – 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 'autor níže uvedeného kódu nenese žádnou odpovědnost
4 'za škody způsobené jeho užíváním
5 'doporučena je nešifrování záloha sešitu
6
7 Private Sub Šifrovani()
8
9     Dim strKlic As String
10    Dim rngOblast As Range
11    Dim rngBunka As Range
12
13    'strKlic = InputBox("Zadejte heslo:", "Šifrování listu")
14
15    strKlic = "Ježibaba"
16
17    'zákaz překreslování
18    Application.ScreenUpdating = False
19
20    'zákaz přepočítávání listu
21    Application.Calculation = xlCalculationManual
22
23    'z využití oblasti listu
24    With ActiveSheet.UsedRange
25
26        'převzetí neprázdných buněk (konstanty, vzorce)
27        Set rngOblast = Union(SpecialCells(xlCellTypeConstants), _
28            .SpecialCells(xlCellTypeFormulas))
29
30    End With
31
32    'pro každou buňku
33    For Each rngBunka In rngOblast
34
35        'zašifrování obsahu
36        rngBunka.Value = XOREncryption(strKlic, rngBunka.FormulaLocal)
37
38    Next rngBunka
39
40    'povolení přepočítávání listu
41    Application.Calculation = xlCalculationAutomatic
42
43    'povolení překreslování
44    Application.ScreenUpdating = True
45
46 End Sub
47
48 Private Sub Desifrovani()
49
50     Dim strKlic As String
51     Dim rngOblast As Range
52     Dim rngBunka As Range
53
54     'strKlic = InputBox("Zadejte heslo:", "Dešifrování listu")
55
56     strKlic = "Ježibaba"
57
58     'zákaz překreslování
59     Application.ScreenUpdating = False
60
61     'zákaz přepočítávání listu
62     Application.Calculation = xlCalculationManual
63
64     'z využití oblasti listu
65     With ActiveSheet.UsedRange
66
67         'převzetí neprázdných buněk (konstanty)
68         Set rngOblast = .SpecialCells(xlCellTypeConstants)
69
70     End With
71
72     'pro každou buňku
73     For Each rngBunka In rngOblast
74
75         'dešifrování obsahu
76         rngBunka.FormulaLocal = XORDecryption(strKlic, rngBunka.Value)
77
78     Next rngBunka
79
80     'povolení přepočítávání listu
81     Application.Calculation = xlCalculationAutomatic
82
83     'povolení překreslování
84     Application.ScreenUpdating = True
85
86 End Sub
87
88
89 Public Function XOREncryption(CodeKey As String, DataIn As String) As String
90
91     Dim IonDataPtr 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 IonDataPtr = 1 To Len(DataIn)
100        intXorValue1 = Asc(Mid$(DataIn, IonDataPtr, 1))
101        intXorValue2 = Asc(Mid$(CodeKey, ((IonDataPtr 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 IonDataPtr
107
108    XOREncryption = strDataOut
109
110 End Function
111
112
113 Public Function XORDecryption(CodeKey As String, DataIn As String) As String
114
115     Dim IonDataPtr As Long
116     Dim strDataOut As String
117     Dim intXorValue1 As Integer
118     Dim intXorValue2 As Integer
119
120     For IonDataPtr = 1 To (Len(DataIn) / 2)
121        intXorValue1 = Val("&H" & (Mid$(DataIn, (2 * IonDataPtr - 1), 2)))
122        intXorValue2 = Asc(Mid$(CodeKey, ((IonDataPtr Mod Len(CodeKey)) + 1), _
123            1))
124        strDataOut = strDataOut + Chr(intXorValue1 Xor intXorValue2)
125    Next IonDataPtr
126
127    XORDecryption = strDataOut
128
129 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](#)