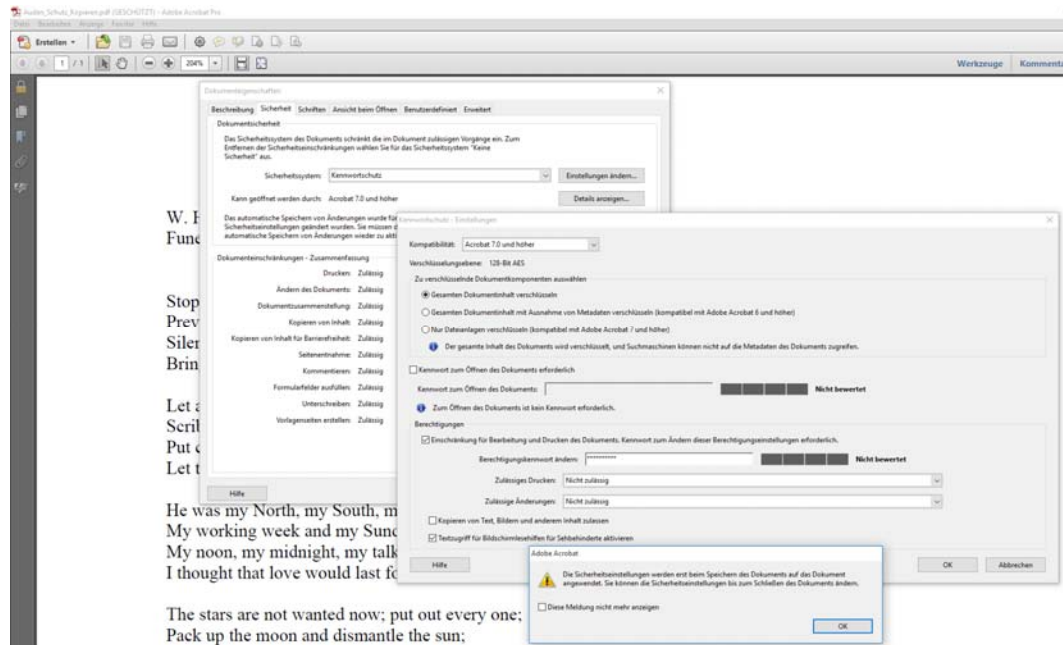


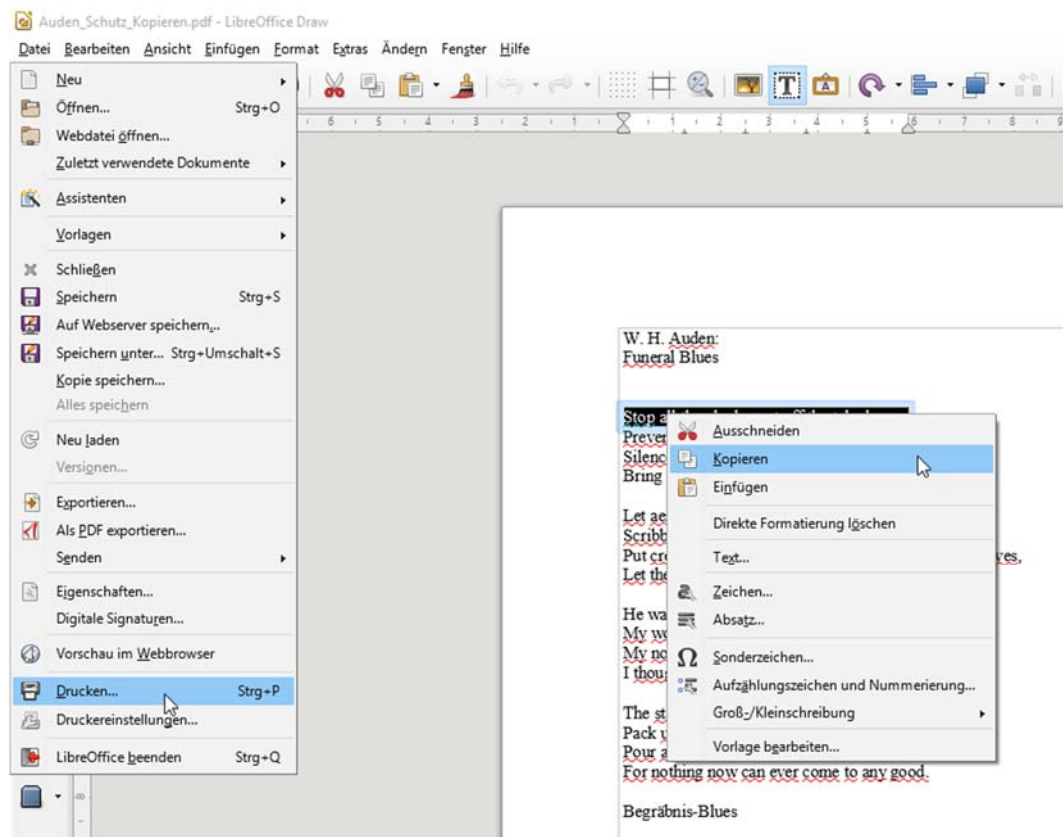
Excel hacken

1. PDF

In Adobe Acrobat wird ein Dokument mit einem Kopierschutz versehen.



Dieses kann in LibreOffice (und einigen anderen Programmen) problemlos geöffnet werden – Kopier- und Druckschutz ist entfernt:

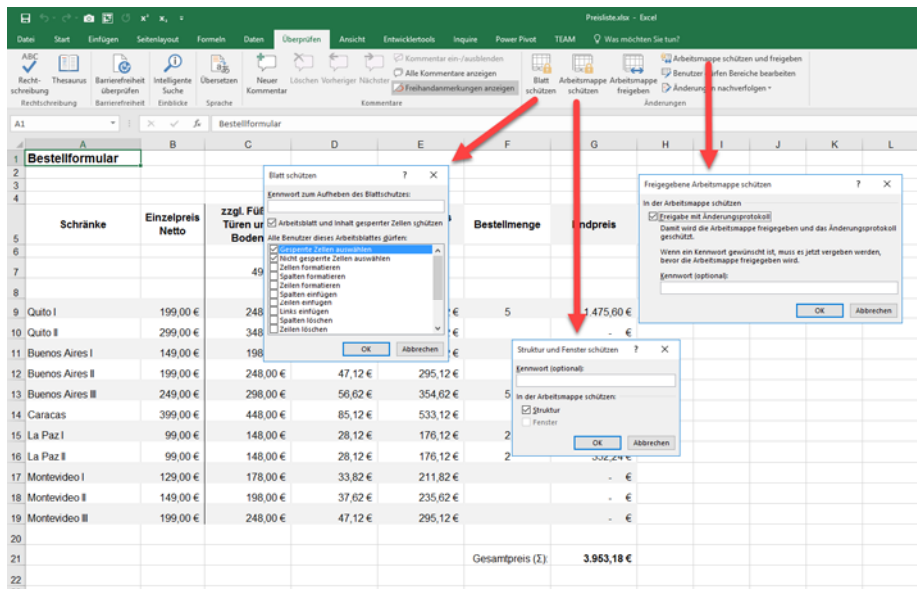


Der Dateischutz kann leider nicht über LibreOffice geknackt werden.

2. Blattschutz

In der Registerkarte „Überprüfen“ stehen drei Schutzmechanismen mit Kennwort zur Verfügung:

- Blattschutz
- Arbeitsmappe schützen
- Freigegebene Arbeitsmappe schützen

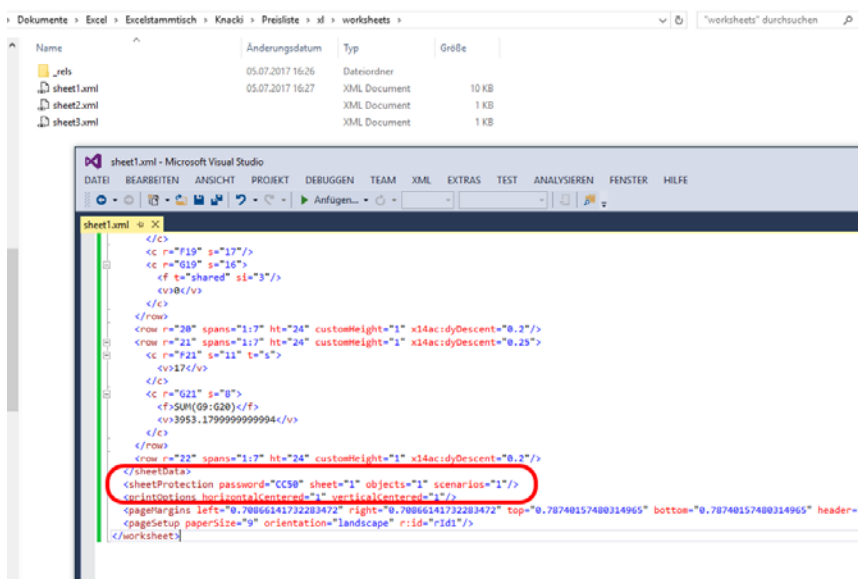


2a) Blatt schützen wird aufgehoben wenn die Zellen auf ein anderes Tabellenblatt kopiert werden.

Blatt schützen wird aufgehoben, wenn man die Datei in libreOffice (oder Numbers) öffnet.

Blatt schützen verbirgt sich im ZIP-Archiv: In dem Worksheet steht am Ende von sheetxx.xml folgendes Element:

```
<sheetProtection password="CC50" sheet="1" objects="1" scenarios="1"/>
```



Knoten löschen; zippen -> Schutz ist entfernt.

2b) und 2c) In der Datei workbook.xml befindet das Element **workbookProtection**:

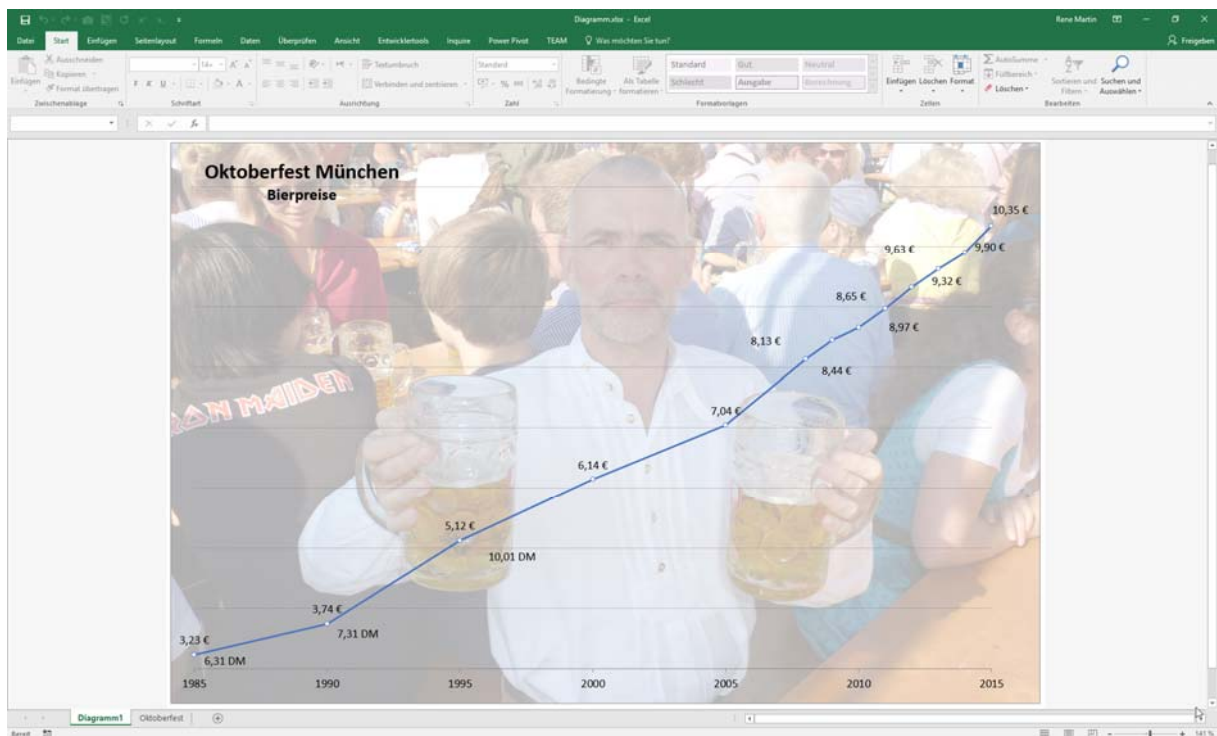
```
workbook.xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<workbook xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main" xmlns:r="http://schemas.m...
<fileVersion appName="xl" lastEdited="7" lowestEdited="7" rupBuild="18201"/>
<workbookPr/>
<mc:AlternateContent xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006">
  <mc:Choice Requires="x15">
    <x15:absPath url="D:\Eigene Dateien\Excel\Excelstammtisch\Knacki\" xmlns:x15ac="http://schema...
  </mc:Choice>
</mc:AlternateContent>
<workbookProtection workbookPassword="CC50" lockStructure="1"/>
<bookViews>
  <workbookView xWindow="0" yWindow="0" windowWidth="25200" windowHeight="12000"/>
</bookViews>
<sheets>
  <sheet name="Tabelle1" sheetId="1" r:id="rId1"/>
  <sheet name="Tabelle2" sheetId="2" r:id="rId2"/>
  <sheet name="Tabelle3" sheetId="3" r:id="rId3"/>
</sheets>
<calcPr calcId="171027"/>
<extLst>
  <ext uri="{140A7094-0E35-4892-8432-C4D2E57EDB5}" xmlns:x15="http://schemas.microsoft.com/office/...
  <x15:workbookPr chartTrackingRefBase="1"/>
  </ext>
</extLst>
</workbook>
```

Element löschen -> Schutz ist aufgehoben!

3. Diagrammschutz

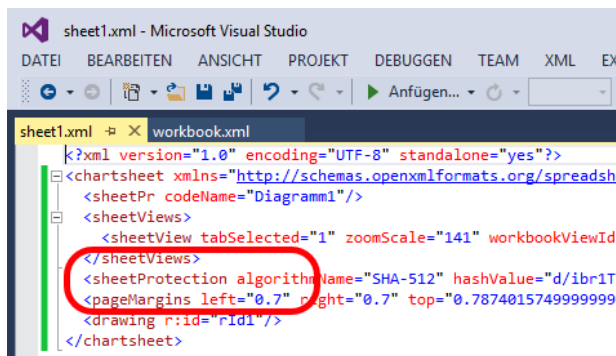
Mit ein paar Zeilen Code lässt sich ein Diagramm schützen:

Charts(1).Protect Password="Bier"



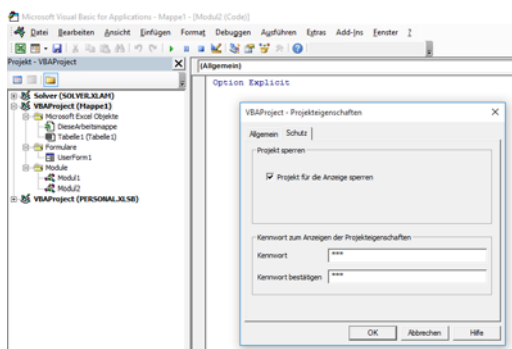
Auch dieser Schutz befindet sich im XML-Archiv:

Im Verzeichnis chartsheets in der Datei sheetxx.xml findet sich die sheetProtection:



4. VBA-Schutz

In VBA kann man auf das Projekt (die Datei einen Schutz mit Kennwort legen):



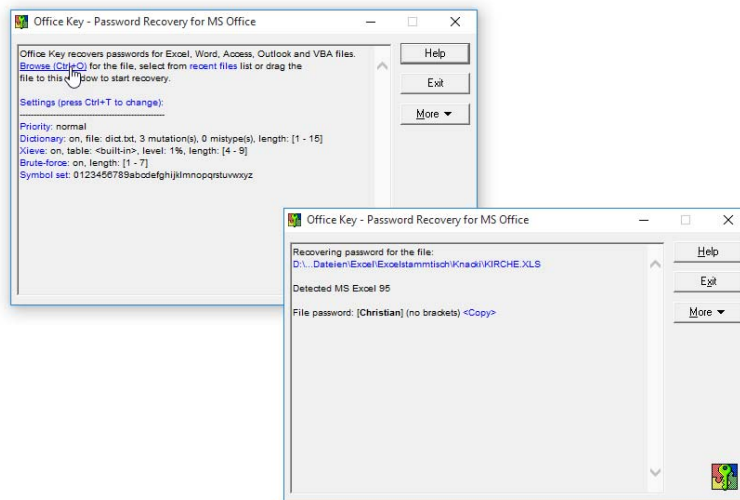
Da VBA nicht als XML-Information abgelegt wird, sondern im binären Code, muss man einen Hex-Editor besorgen (beispielsweise MX von www.nextsoft.de).

Man öffnet die Datei im Editor, sucht die Zeichenfolge DPB und ersetzt sie beispielsweise durch DBx. Datei speichern; Datei in Excel öffnen – eine Warnmeldung wegklicken und schon ist der Schutz aufgehoben.

```
ID="{00000000-00
00-0000-0000-000
00000000}"..Doc
ument=DieseArbei
tsmappe/&H000000
00..Document=Tab
elle1/&H00000000
..HelpFile=""..N
ame="VBATEST"..H
elpContextID="0"
..VersionCompati
ble32="393222000
"..CMG="C7C56B74
757C7E807E807A84
7A84"..DPB="CCCE
607F60997D997D66
839A7D1A4D989F1E
FC16C8B389DA48D2
5F64E2B3EA837F42
8407C838"..GC="D
1D37D7E7E7E7E7E"
....[Host Extend
er Info]..&H0000
0001={3832D640-C
F90-11CF-8E43-00
A0C911005A};VBE;
&H00000000....[W
orkspace]..Diese
Arbeitsmappe=0,
```

5. Werkzeuge zum Knacken

Bis Office 2003 war der Schutz sehr schwach – es gab eine Menge (kostenloser) Werkzeuge zum Knacken:



6. Dateischutz

Leider kann eine geschützte Datei weder entzippt noch mit einem Fremdprogramm geöffnet werden. Es gibt (kostenpflichtige) Programme, die mit brute force den Code knacken.

Wikipedia schreibt zu Brute Force:

Oft sind Passwörter mit Hilfe von kryptographischen Hashfunktionen verschlüsselt. Eine direkte Berechnung des Passworts aus dem Hashwert ist praktisch nicht möglich. Ein Cracker kann jedoch die Hashwerte vieler Passwörter berechnen. Stimmt ein Wert mit dem Wert des hinterlegten Passwortes überein, hat er das (oder ein anderes, zufällig passendes) Passwort gefunden. Brute Force bedeutet hier also simples Ausprobieren von möglichen Passwörtern.

Michael schreibt: „Die Bücher von Simon Singh kann ich sehr empfehlen.“

Und Stefan kommentiert: „Wenn ein Programmierer etwas schützen möchte, dann hat dies einen guten Grund. Überlegt bitte, ob ihr diesen Schutz wirklich knacken wollt und was ihr mit den Informationen anstellt.“