UNCATEGORIZED

ALLES VERSCHLÜSSELT!

3. MÄRZ 2019 | HERMANN APFELBÖCK

Mobile Notebooks, handliche USB-Sticks, öffentliche Cloud: Alles, was das Haus und das heimische Netz verlässt, kann in fremde Hände gelangen oder ist in fremden Händen. Verschlüsselung sorgt dafür, dass die Daten nichts Persönliches preisgeben.

Hinsichtlich Datenschutz und Verschlüsselung spaltet sich die Gesellschaft so schizophren wie sonst auch: Die einen werfen ihre Privatsphäre bedenkenlos ins World Wide Web, die anderen sorgen sich bei jeder Dropbox-Datei, dass die NSA mitlesen könnte. Es ist aber nicht Ziel dieses Beitrags, die Naiven zu bekehren oder die Paranoiden zu beruhigen. Hier geht es allein um die technischen Möglichkeiten, die Linux in großer Vielfalt und Abstufung bereithält, um Daten zu verschlüsseln. Der Artikel stellt alle Methoden vor, bewertet sie und bringt eine vollständige Praxis-Anleitung für die Einrichtung und Nutzung.

1. LUKS-verschlüsseltes Linux-System

Die kompromisslose Methode, die lokalen Daten vor Fremdzugriff zu schützen, ist die Verschlüsselung der kompletten Festplatte. Mit dem auf dem Kernelmodul dm-crypt basierenden Linux Unified Key Setup (LUKS) lassen sich sowohl externe USB-Datenträger (siehe Punkt 2) als auch die Systemfestplatte selbst sicher verschlüsseln. Wir beginnen mit dem technisch anspruchsvollsten Szenario der verschlüsselten Systemfestplatte, da es durch moderne Installer zur einfachen Übung gerät und bei der Alltagsbenutzung nicht mehr Aufwand bedeutet als die Schlüsseleingabe beim Systemstart. Trotzdem sollte sich jeder Anwender, der seine Systemfestplatte verschlüsselt im Klaren sein, dass die Partitionierung komplexer wird und bei Bootproblemen höheren Reparaturaufwand verursacht.

Empfehlung: Eine LUKS-verschlüsselte Systemfestplatte ist die richtige Maßnahme für Notebooks, die viel unterwegs sind und auch jenseits des Home-Verzeichnisses vertrauliche und private Daten enthalten. Der verschlüsselte Datenträger lässt beim Booten durch ein Fremdsystem keinerlei Einblick in die Verzeichnisstruktur und in die Daten zu. Das Einzige, was ein Fremdzugriff anhand der Partitionierungsfakten in Erfahrung bringen kann, ist die Tatsache, dass die Festplatte LUKS-verschlüsselt ist.

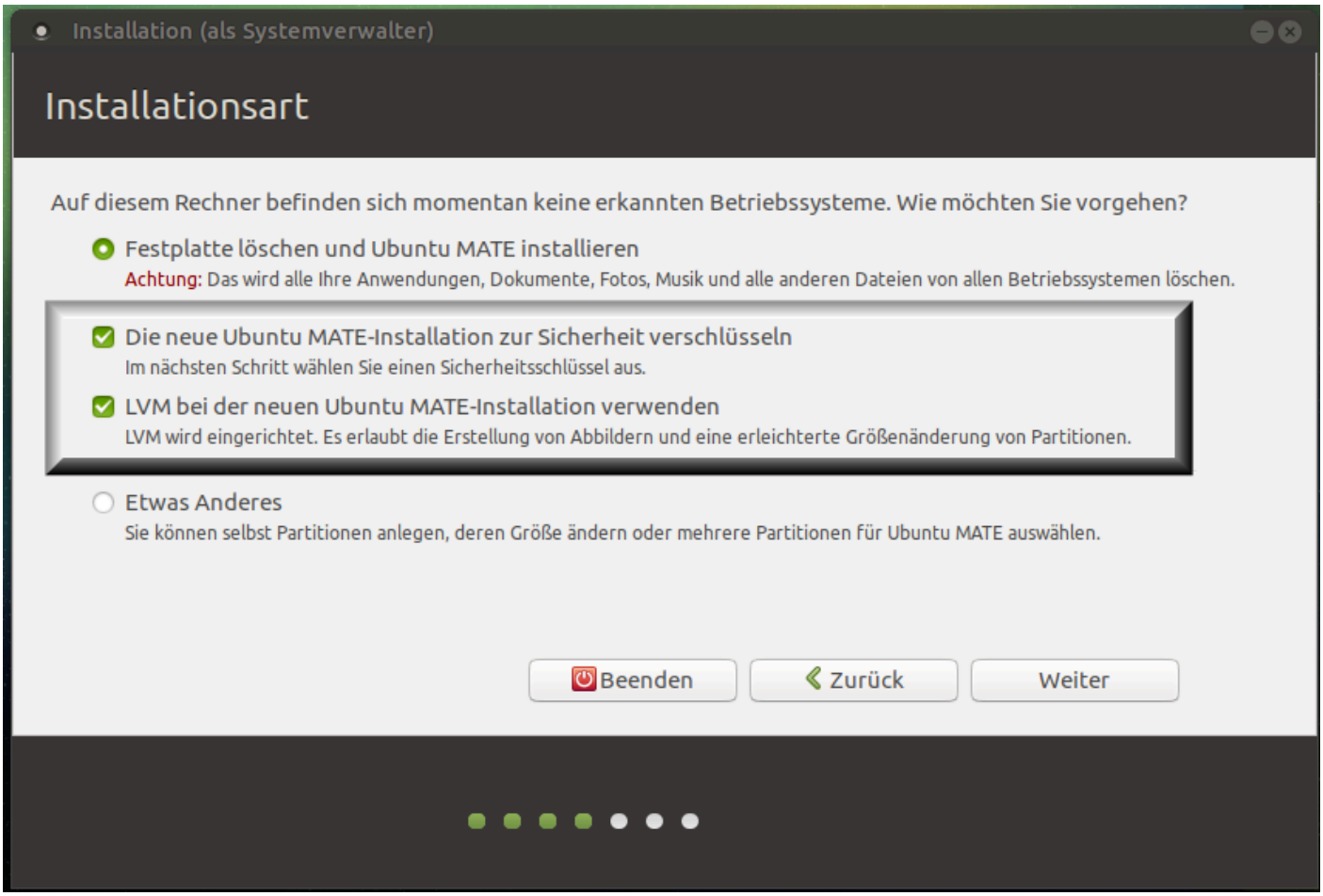
Installation mit LUKS und LVM: Es gibt diverse grafische Linux-Installer, die beim Setup eine LUKS-verschlüsselte Systempartition einrichten können. Neben Yast unter Open Suse, dem Debian-Instal-

ler und dem Fedora-Installer bieten alle Ubuntu-basierten Distributionen inklusive Linux Mint den Installer Ubiquity, der dies beherrscht. Die folgende Anleitung orientiert sich an Ubiquity. Beachten Sie aber, dass der Ubuntu-Installer den bequemen Weg zur LUKS-verschlüsselten Systemfestplatte nur anbietet, wenn Sie ihm dafür die gesamte primäre Festplatte überlassen. Eine kompliziertere Situation mit Multiboot oder anderweitigen Partitionsaufteilungen ist nicht vorgesehen. Die Festplatte wird bei diesem Vorgang komplett gelöscht.

Starten Sie die Installation im Livesystem eines Ubuntu-Systems, und folgen Sie dem Setup-Assistenten bis zum Punkt „Installationsart“. Hier wählen Sie die erste Option „Festplatte löschen und […] installieren“. Darunter aktivieren Sie das Kästchen „Die neue Ubuntu-Installation zur Sicherheit verschlüsseln“. Sobald Sie dies tun, wird zugleich der weitere Punkt „LVM […] verwenden“ aktiv. Der Logical Volume Manager ist eine Abstraktionsschicht, um Festplatten und Partitionen flexibler zu verwalten, zusammenzufassen und dynamisch zu erweitern. In diesem Fall ist LVM notwendig, um neben der kleinen unverschlüsselten Bootpartition die LUKS-formatierte Partition und die virtuelle LVM-Partition unterzubringen, die bei korrekter Kennworteingabe unverschlüsselt ins Dateisystem geladen wird.

Wenn Sie im Assistenten mit den genannten Optionen auf „Weiter“ klicken, folgt noch die Abfrage des Sicherheitsschlüssels. Dieses Kennwort sollte komplex genug sein, um vom Assistenten als „Starkes Passwort“ gelobt zu werden. Andererseits muss die Eingabe zumutbar bleiben, denn sie ist künftig bei jedem Systemstart erforderlich. Die weitere Installation unterscheidet sich nicht mehr von einem üblichen Ubuntu-Setup.

Wenn Sie ein LUKS-verschlüsseltes System booten, erscheint künftig das Eingabefeld „Please unlock disk […]“. Dort geben Sie das Passwort ein, und erst danach kann der Systemstart fortsetzen, wobei das LUKS-Volume entsperrt und unverschlüsselt nach /dev/mapper/[sd…] gemountet wird.



Systemverschlüsselung per Setup: Dies ist die entscheidende Einstellung beim Ubuntu-Installer. Mit LVM und LUKS-Verschlüsselung bootet das System erst nach korrekter Kennworteingabe.

2. LUKS-verschlüsselte (USB-) Datenträger

Interne Laufwerke, die als reine Datenpartition dienen, sowie externe USB-Datenträger lassen sich ebenfalls mit LUKS verschlüsseln. Technisch ist dies weniger anspruchsvoll und kommt ohne LVM-Unterstützung aus. Die Einrichtung und Benutzung erfolgt auf modernen Linux-Distributionen komplett mit grafischen Werkzeugen, die den komplizierteren Weg über Terminalbefehle in der Regel überflüssig machen.

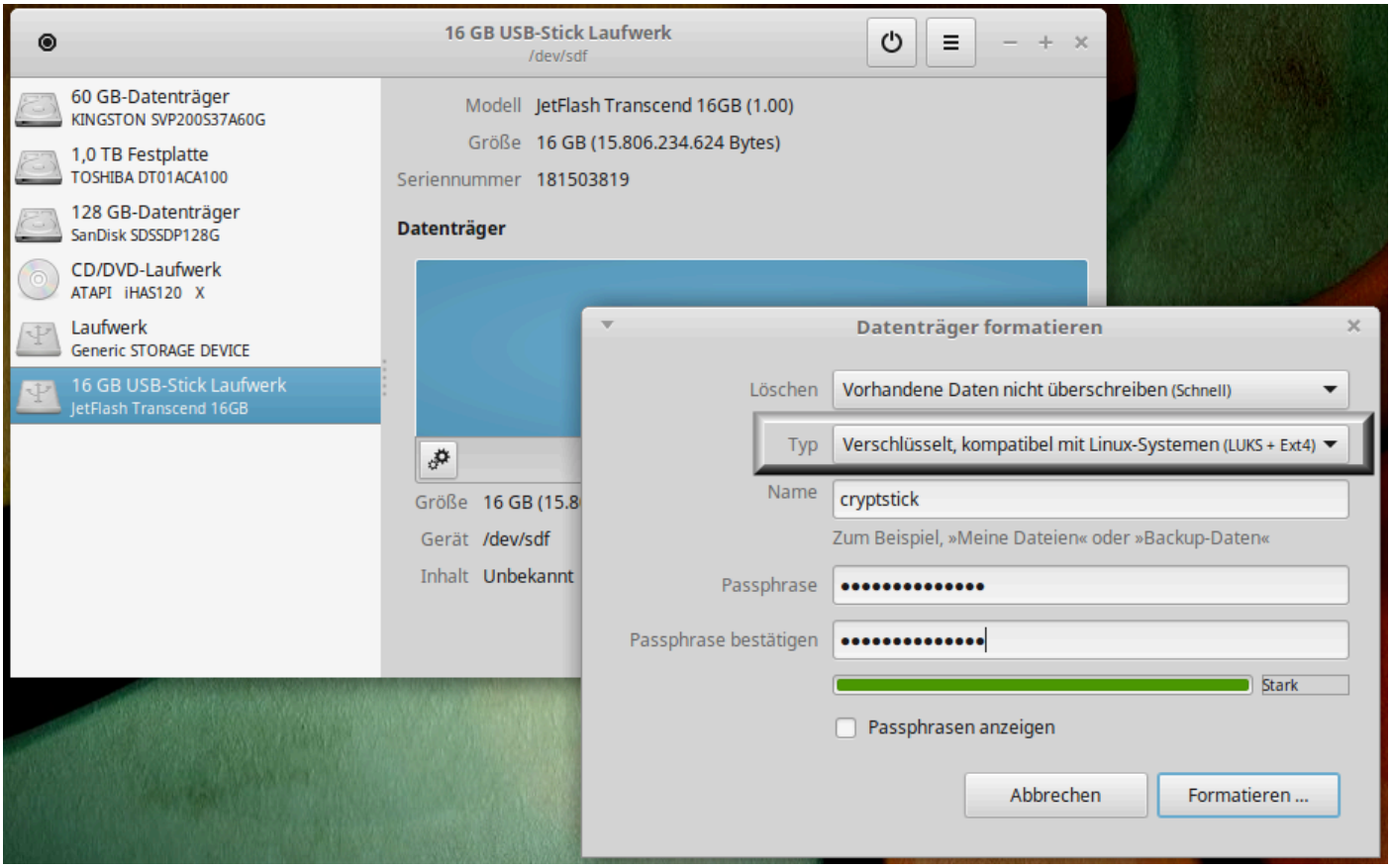
Empfehlung: Besonders USB-Sticks und handliche USB-Festplatten gehen oft verloren oder werden vergessen. LUKS ist für mobile Speicher erste Wahl, sofern die Datenträger überwiegend mit Linux gelesen und beschrieben werden. Dort, wo auch ein Windows oder ein Mac OS zugreifen soll, ist Veracrypt (siehe Punkt 6) die bessere Option.

Einrichten mit grafischen Werkzeugen: Erfreulicherweise hat LUKS-Verschlüsselung in die Systemwerkzeuge längst Einzug gehalten. Die KDE-Umgebung bietet den „KDE Partition Manager“ (partitionmanager), und die Gnome-affinen Desktops (Gnome, Mate, Unity, Cinnamon, XFCE) haben das Tool „Laufwerke“ (gnome-disks) an Bord. Wir beschreiben die wenigen Klicks zur LUKS-Verschlüsselung eines USB-Laufwerks am Beispiel von gnome-disks:

Nach Anschließen des USB-Datenträgers hängen Sie das Laufwerk zunächst mit dem kleinen schwarzen Symbol links unterhalb der Partitionsanzeige aus. Danach klicken Sie auf das Zahnrad-

symbol und verwenden die Option „Partition formatieren“. Im Folgedialog wählen Sie als „Typ“ den Eintrag „Verschlüsselt, kompatibel mit Linux-Systemen (LUKS + Ext4)“. Der Eintrag „Name“ ist nicht unbedingt erforderlich, macht aber den späteren Mountpunkt lesbarer. Entscheidend ist darunter die „Passphrase“ – also das Kennwort. Hier gilt wie unter Punkt 1: Das Kennwort sollte komplex sein, die Eingabe aber zumutbar bleiben, denn sie ist künftig bei jeder Nutzung des Datenträgers erforderlich. Mit Klick auf „Formatieren“ schließen Sie den Vorgang ab. Sie können nach der Formatierung den Datenträger sofort mit gnome-disks einhängen und nutzen, indem Sie auf den unteren Balken der symbolischen Anzeige klicken und die Partition mit dem Pfeilsymbol links einhängen.

Für die künftige Alltagsbedienung genügen die typischen Dateimanager Nautilus, Nemo, Caja, Dolphin. Wenn Sie das USB-Gerät anschließen, erscheint nach kurzer Frist automatisch der Dialog „Geben Sie eine Passphrase zum Entsperren […] ein“. Nach Eingabe des korrekten Kennworts ist das Medium entsperrt und im Dateimanager unter „Geräte“ normal benutzbar. An gleicher Stelle im Dateimanager können Sie den Datenträger wieder trennen („Laufwerk sicher entfernen“).



LUKS-Verschlüsselung für USB-Datenträger: Diese Aufgabe beherrschen die typischen Partitionsmanager von Desktop-Distributionen – hier gnome-disks unter Linux Mint.

Exkurs: Manuelle LUKS-Verschlüsselung

LUKS-Verschlüsselung für ein externes USB-Laufwerk kann auch ohne grafische Werkzeuge etwa auf einem Headless-Server eingerichtet werden. Die folgende Ergänzung zu den Punkten 1 und 2 dient nicht nur der Vollständigkeit, sondern soll auch die zugrundeliegenden Werkzeuge vorstellen, die unter der Haube auch von grafischen Tools wie gnome-disks genutzt werden.

Zunächst ermitteln Sie mit

```
lsblk
```

die Gerätekennung des USB-Datenträgers. Alle folgenden Kommandos gehen von der Beispielkennung /dev/sde aus, die in Ihrem Fall natürlich anders lauten kann und unbedingt entsprechend angepasst werden muss. Zunächst wird die Partitionstabelle des Sticks mit fdisk neu geschrieben:

```
sudo fdisk /dev/sde
```

Geben Sie am fdisk-Prompt „o“ ein. Dieser Befehl legt eine neue DOS-Partitionstabelle an. Sie müssen die Aktion anschließend mit dem Schreibbefehl „w“ realisieren, was zugleich fdisk beendet. Starten Sie dann fdisk erneut:

```
sudo fdisk /dev/sde
```

Jetzt legen Sie mit dem Befehl „n“ eine neue Partition an und verwenden dabei „p“ für „primary“, „1“ für Partition 1. Die zwei Abfragen der Start- und End-Sektoren quittieren Sie einfach mit der Eingabetaste. Auch hier muss abschließend der Write-Befehl „w“ erfolgen, um die Aktion tatsächlich auf den Datenträger zu schreiben.

Nun hängen Sie das Laufwerk mit

```
sudo umount /dev/sde?
```

aus und formatieren es mit LUKS. Das dazu notwendige Tool cryptsetup steht auf allen verbreiteten Distributionen zur Verfügung:

```
sudo cryptsetup luksFormat /dev/sde1
```

Der Parameter „luksFormat“ muss genau so eingegeben werden. Die nachfolgende Bestätigung mit „YES“ ist ebenfalls case-sensitiv und erfordert Großbuchstaben. Dann werden Sie nach dem „Passsatz“ gefragt, also dem Zugangskennwort. Die Eingabe erfolgt ohne Textanzeige und ohne Stellvertreterzeichen.

Nun können Sie das Laufwerk mit „luksOpen“

```
sudo cryptsetup luksOpen /dev/sde1 Stick
```

in das System laden. Der Name, hier „Stick“, ist frei wählbar. Das Laufwerk wird nun unter /dev/mapper/Stick gemountet. Zu guter Letzt braucht das Laufwerk neben LUKS noch ein normales, unverschlüsseltes Dateiformat, was Sie mit

```
sudo mkfs.vfat /dev/mapper/Stick -n Stick
```

erledigen. Das war’s. Entfernen Sie nun den Stick einfach vom Rechner. Die Prozedur ist für jeden USB-Datenträger nur einmal erforderlich.

3. Verschlüsseltes Home-Verzeichnis (Ecryptfs)

Ubuntu-Systeme einschließlich Linux Mint machen bei der Installation das Angebot, das Home-Verzeichnis zu verschlüsseln. Diese Option ist bei einer Neuinstallation immer gut zu überlegen, zumal sie nachträglich für den ersteingerichteten Benutzer nicht mehr vorgesehen ist und dann doch einige Klimmzüge erfordert. Technisch zuständig ist in diesem Fall das Modul Ecryptfs, das Ubuntu & Co. standardmäßig im Kernel mitbringen. Ecryptfs ist nicht zu verwechseln mit Encfs (siehe Punkt 5), wenngleich sich beide Techniken ähneln.

Empfehlung: Ein Ecryptfs-verschlüsseltes Home-Verzeichnis ist für typische Desktop-Systeme, auch für mobile Notebooks, oft der angemessene und der komfortabelste Schutz. Bei Fremdzugriff ist zwar der Großteil des Dateisystems lesbar, nicht aber der Inhalt von /home/[user]. Dieser liegt verschlüsselt unter /home/.ecryptfs/[user]/.Private und wird automatisch unverschlüsselt nach /home/[user] geladen, sobald sich der Benutzer am System anmeldet. Wenn sich der Gerätebesitzer daran hält, seine Daten stets unter /home abzulegen, ist für Datendiebe nichts zu holen. Lediglich die Anzahl der Verzeichnisse und Dateien sowie deren ungefähre Größen sind unter /home/.ecryptfs/[user]/.Private ersichtlich – die Inhalte nicht. Die Dateinamen sind ebenfalls verschlüsselt.

Einrichten der Home-Verschlüsselung: Bei der Installation von Ubuntu-Systemen erscheint zu einem späteren Zeitpunkt das Fenster „Wer sind Sie?“. Hier legen Sie den Erstbenutzer des System an. An unterster Stelle gibt es die Option „Meine persönlichen Daten verschlüsseln“. Ein Häkchen genügt, um Ecryptfs für das Home-Verzeichnis des Erstbenutzers zu aktivieren. Nach der ersten Anmeldung am neu installierten System erscheint dann ein Fenster mit dem Hinweis „Ihre Verschlüsselungspassphrase notieren“. Klicken Sie auf „Diese Aktion ausführen“. Danach geben Sie das Systempasswort ein und bestätigen mit der Eingabetaste. Sie sehen dann das von Ubuntu & Co. zufällig generierte Passwort für die Home-Verschlüsselung. Notieren Sie sich dieses, denn Sie benötigen es für den (unwahrscheinlichen) Fall, dass einmal die Wiederherstellung eines defekten Dateisystems nötig werden sollte.

Für den alltäglichen Zugriff auf das Home-Verzeichnis genügt die Anmeldung am System. Vor anderen Benutzern am selben PC ist das verschlüsselte Home-Verzeichnis ebenfalls sicher. Diese erhalten beim Zugriff eine Fehlermeldung, die auf fehlende Rechte hinweist. Besitzen andere Systembenutzer root-Recht, können Sie zwar den Ordner /home/.ecryptfs/[user]/.Private betreten, sehen dort aber nicht mehr als verschlüsselte Ordner- und Dateinamen.

Weitere verschlüsselte Home-Verzeichnisse: Die Home-Verschlüsselung bei der Installation gilt nur für den dort eingerichteten Erstbenutzer. Wenn ein weiterer Benutzer ein verschlüsseltes Home-Verzeichnis erhalten soll, gibt es zwei Wege:

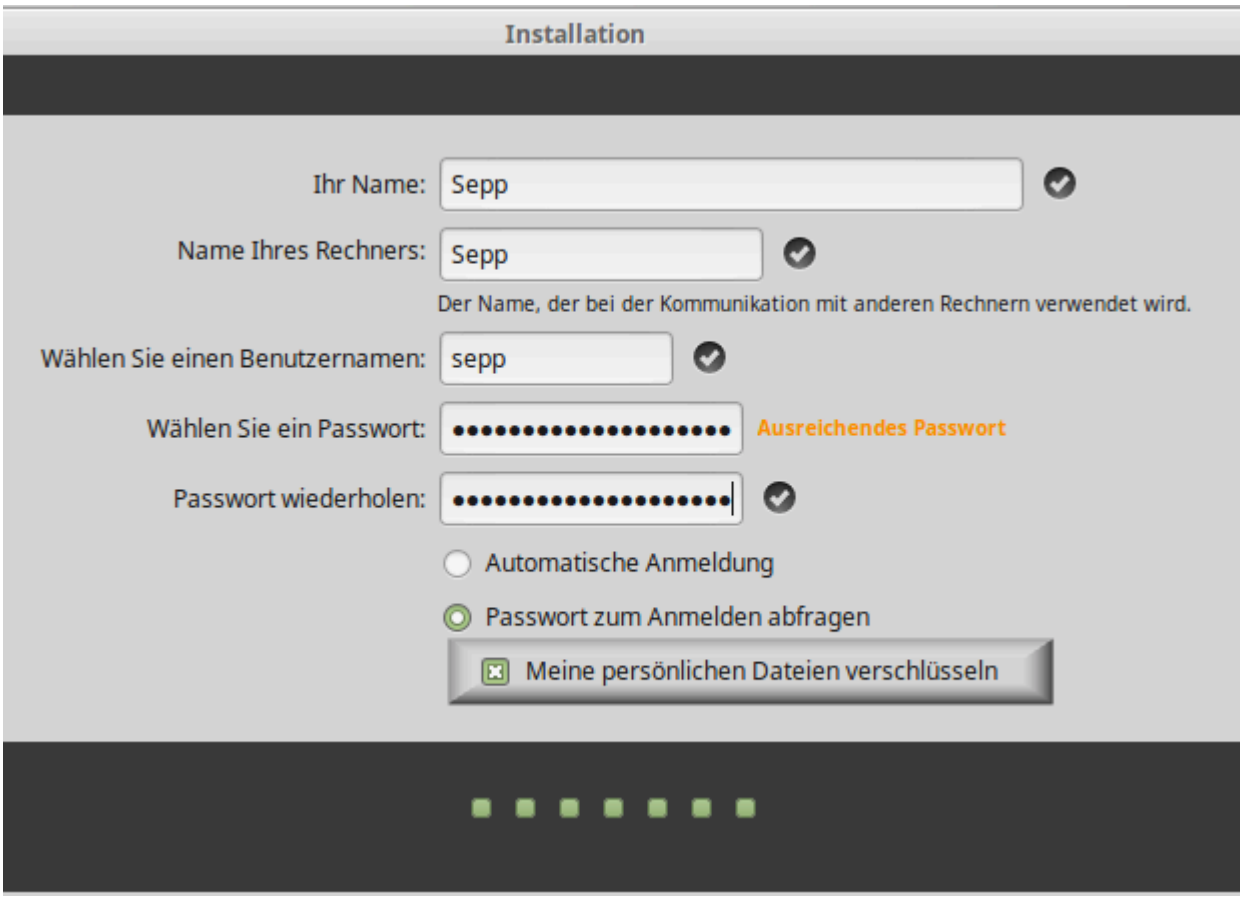
1. Die grafische Methode über „Einstellungen -> Benutzer/Users“ ist aktuell noch die Ausnahme. Linux Mint 18.2 bietet in der Benutzerverwaltung beim Anlegen eines neuen Kontos ganz unten die Option „Persönlichen Ordner verschlüsseln […]“.

2. Bei den meisten Distributionen ist die Home-Verschlüsselung nur über die Kommandozeile zu erreichen. Es genügt dieser Befehl:

```
sudo adduser --encrypt-home [username]
```

Falls der Befehl scheitert, installieren Sie das Paket ecryptfs-utils (*sudo apt install ecryptfs*) und wiederholen den Befehl. Legen Sie das Passwort für den neuen Benutzer hinter „Geben Sie ein neues UNIX-Passwort ein:“ fest. Danach geben Sie die Benutzerinformationen ein oder bestätigen einfach alles mit Eingabetaste. Ab sofort kann sich der neue Benutzer anmelden und das verschlüsselte Home-Verzeichnis nutzen. Auch er erhält einen Hinweis, sich die Verschlüsselungspassphrase zu notieren.

Home-Verzeichnis nachträglich verschlüsseln: Es ist nicht vorgesehen, die Verschlüsselung nachträglich zu aktivieren. Die einfachste Lösung ist es daher, alle Dateien im Home-Verzeichnis vorübergehend an einen anderen Ort zu verschieben, dann ein neues Benutzerkonto mit verschlüsseltem Home-Verzeichnis anzulegen und die gesicherten Dateien danach in das neue Home-Verzeichnis zu kopieren. Das ursprüngliche Konto kann danach im Prinzip gelöscht werden. Tun Sie dies aber erst, wenn der Systemalltag reibungslos funktioniert: So sollte zum Beispiel der sudo-berechtigte Erstbenutzer nicht gelöscht werden, solange kein neues Konto mit sudo-Recht eingerichtet ist (*visudo*).



Linux Mint bietet bei der Installation nach wie vor an, das Home-Verzeichnis des Erstbenutzers zu verschlüsseln. Das gilt auch für weitere, später angelegte Benutzer .

4. Verschlüsselung mit Encfs

Encfs ist ein flexibles Ordner- und Datei-orientiertes Verschlüsselungswerkzeug. Der Ruf des Tools hat etwas gelitten, nachdem vor Jahren eine theoretische Lücke bekannt wurde, welche die Robustheit von Encfs infrage stellte. Diese Lücke besteht immer noch und wird bei der Installation des Pakets gemeldet. Unterm Strich handelt es sich aber um ein akademisches Problem, das normale Anwender nicht betrifft. Technisch ähnlich wie bei Ecryptfs wird ein verschlüsselter Ordner durch Eingabe des richtigen Passwort entsperrt und der Inhalt unverschlüsselt in einen zweiten Ordner gemountet, wo die Dateien dann normal zu verwenden sind. Verschlüsselte Encfs-Ordner sind überall flexibel einzurichten – auf internen und externen Datenträgern mit FAT, FAT32, NTFS oder einem Linux-Dateisystem.

Empfehlung: Encfs ist das richtige Werkzeug für alle Situationen, die im Alltag plötzlich Verschlüsselung ratsam erscheinen lassen: Dieses oder jenes Verzeichnis auf der USB-Festplatte hat Verschlüsselung verdient, ein Unterordner von Home muss geschützt werden oder ein Cloud-Dienst soll über den verschlüsselten Sync-Ordner nur noch geschützte Dateien bevorraten. Encfs kann an jeder Stelle und auf jedem Datenträger einspringen und eignet sich auch für größere Datenmengen. Neben Linux können Android-Geräte mit der App Cryptonite Encfs-Daten lesen. Für Mac-Anwender gibt es nach einem gewissen Installationsaufwand ein praktisch identisches Encfs. Für den Austausch mit Windows gibt es Encfs4win, das allerdings experimentell bleibt (https://encfs.win).

Encfs mit grafischem Cryptkeeper: Encfs ist meist nicht installiert, doch finden Sie das relativ kleine Paket in den Repositories aller wichtigen Linux-Distributionen. Unter Ubuntu/Mint installieren Sie es mit

```
sudo apt install encfs
```

und können dann sofort loslegen. Encfs ist an sich ein reines Kommandozeilen-Tool, jedoch werden die meisten Anwender Encfs über das grafische Frontend Cryptkeeper bedienen. Auch dieses kleine Tool muss mit

```
sudo apt install cryptkeeper
```

erst nachinstalliert werden. Nach dem Aufruf *cryptkeeper* präsentiert sich dieser als Schlüsselsymbol in der Hauptleiste. Die Option „Erstelle verschlüsselten Ordner“ richtet ein neues verschlüsseltes Verzeichnis ein, wobei Sie in der oberen Zeile den Ordnernamen vergeben und unten zum gewünschten Ort navigieren, etwa zu einem USB-Stick unter /media im Dateisystem. Mit der Schaltfläche „Vor“ geht es weiter zur Passwortvergabe. Der neue und noch leere Mount-Ordner wird zum Abschluss automatisch im Dateimanager geöffnet und kann dann befüllt werden. Sie arbeiten in diesem Ordner wie mit unverschlüsselten Dateien. Die eigentlichen Dateien liegen auf gleicher Ebene in einem versteckten Ordner .[name]_enfcs. Um einen Encfs-Ordner auszuhängen und damit zu schützen, klicken Sie auf das Cryptkeeper-Symbol und dann auf den betreffenden Ordnereintrag.

Über die „Einstellungen“ können Sie vorgeben, dass Mount-Ordner nach dem Entladen („Aushängen“) gelöscht und dass nicht genutzte Encfs-Ordner nach bestimmter Frist automatisch entladen werden. Diese zweite Sicherheitsmaßnahme ist ein Alleinstellungsmerkmal von Encfs.

Encfs auf der Kommandozeile: Im Terminal ist Encfs-Verschlüsselung etwas mühsamer, andererseits flexibler. Die Kernsyntax lautet:

```
encfs /Pfad1/verschlüsselte_Daten/ /Pfad2/unverschlüsselte_Daten/
```

Pfad1 ist der zu verschlüsselnde Ordner, Pfad2 der Mountpunkt, wo die Daten unverschlüsselt genutzt werden. Das Beispiel

```
mkdir /media/sepp/data/privat
```

```
mkdir ~/privat
```

```
encfs /media/sepp/data/privat ~/privat
```

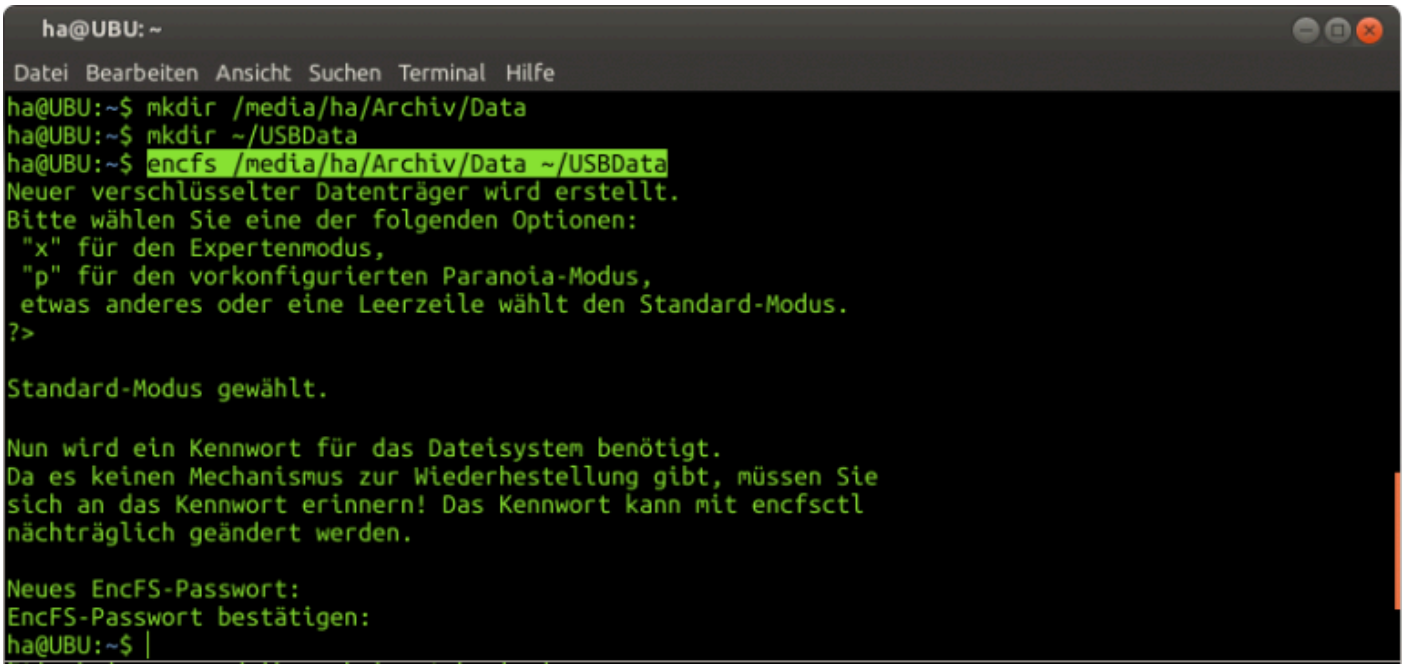
erstellt unter „/media/sepp/data“ das neue Verzeichnis „privat“, ferner den gleichnamigen Mountpunkt im Home-Verzeichnis. Die dritte Zeile lädt das noch leere Verzeichnis in den Mountpunkt. Danach ist noch die Vergabe eines neuen Kennworts notwendig. Unter „~/privat“ arbeiten Sie mit den Daten.

Während bei Cryptkeeper das verschlüsselte Verzeichnis und das Mount-Verzeichnis stets auf gleicher Ebene liegen, kann Encfs auf Kommandozeile von beliebiger Stelle in einen beliebigen Mountpunkt laden. Wenn Sie einen verschlüsselten Ordner nicht mehr benötigen, entladen Sie seinen Mountpunkt:

```
fusermount -u ~/privat
```

Neuerliches Laden geschieht mit genau demselben Encfs-Befehl wie bei der Ersteinrichtung.

Tipp: Auf der Kommandozeile (nicht im Cryptkeeper) können Sie Encfs auch auf ein bereits bestehendes Verzeichnis ansetzen. Alle Dateien, die sich dort bereits befinden, bleiben dort allerdings weiterhin unverschlüsselt. Sollen diese nachträglich verschlüsselt werden, verschieben Sie die Dateien einfach in das Mountverzeichnis des Encfs-Ordner-Paares.



Ersteinrichtung eines Encfs-Ordners: Sie benötigen lediglich einen Ordner als Mountpunkt (hier USBData im Home-Verzeichnis) und ein Kennwort.

Grafisches Frontend für EncFS: Das kleine Tool Cryptkeeper erscheint als Schlüsselsymbol in der Hauptleiste und bietet die wesentlichen EncFS-Funktionen.

5. Container mit Veracrypt

Veracrypt ist der Nachfolger des Verschlüsselungsklassikers Truecrypt, der 2014 eingestellt wurde. Veracrypt arbeitet mit verschlüsselten Containerdateien und einem eigenen Format. Der Inhalt solcher Container wird durch die „Mount"- (oder „Einbinden"-) Schaltfläche und nach korrekter Passworteingabe unverschlüsselt ins Dateisystem gemountet, wobei auch gleich der zuständige Dateimanager zur Anzeige und Dateibearbeitung gestartet wird. Die Größe der Containerdateien muss bei der Einrichtung definiert werden und ist später nicht mehr dynamisch erweiterbar.

Empfehlung: Veracrypt eignet sich für große und sehr große Datenmengen, allerdings nur auf lokalen Rechnern oder im lokalen Netzwerk. Für den Transfer in die Cloud ist es ungeeignet, da auch bei geringen Datenänderungen immer der Transport des gesamten Containers notwendig wäre. Ein entscheidendes Plus von Veracrypt sind Versionen für Linux, Windows, Mac OS, Free BSD und Raspbian. Damit sind Veracrypt-Container zwischen allen PC-Systemen austauschbar.

Installation und Container-Betrieb: Anlaufstelle für die meisten Betriebssysteme ist die Projektseite https://www.veracrypt.fr/en/Downloads.html. Für Ubuntu, Mint & Co. ist die Installation über ein PPA allerdings deutlich komfortabler:

```
sudo add-apt-repository ppa:unit193/encryption
```

```
sudo apt-get update
```

```
sudo apt-get install veracrypt
```

Im Unterschied zur Windows-Version bietet Veracrypt unter Linux keine deutsche Übersetzung, weswegen die nachfolgenden Menübezeichnungen englischsprachig ausfallen. Um eine neue Containerdatei anzulegen, klicken Sie im Hauptdialog auf „Create Volume -> Create an encrypted file container“ und anschließend auf “Standard VeraCrypt volume“. Hier geben Sie Pfad und Namen einer bisher nicht existierenden Datei an. „Encryption Option“ belassen Sie einfach auf den Standardvorgaben. Danach geben Sie die Größe der Containerdatei an. Diese sollte großzügig ausfallen, weil die Kapazität nicht mehr zu ändern ist und viele kleine Container unübersichtlicher sind als wenige große.

Danach kommt die Passwortvergabe. Zur Schlüsselerstellung auf Basis des Passworts erwartet Veracrypt Mausbewegungen im eigenen Fenster. Nach beendeter Fortschrittsanzeige schließen Sie mit „Format“. Damit ist der Container einsatzbereit. Mit „Select File“ im Hauptdialog navigieren Sie zur Containerdatei. Mit Klick auf „Mount“ wird diese geladen und im Dateimanager geöffnet (falls nicht, lässt sich das unter „Preferences -> System Integration“ einstellen). Linux mountet Container nach „/media/veracrypt[nummer]“, Windows auf freie Laufwerkbuchstaben. Auf diesen Datenträgern lesen, arbeiten, kopieren Sie wie auf einem normalen Laufwerk. Mit „Dismount“ im Hauptdialog entladen Sie den Container, der somit wieder geschützt ist.

Zur besseren Systemintegration nistet sich das Veracrypt-Symbol zusätzlich in der Systemleiste des Linux-Desktops ein. Hier sind im Kontextmenü einige fundamentale Aktionen wie das Mounten aller „Favorites“ oder das Abschalten aller aktuell geladenen Container. Eingerichtete „Favorites“ ersparen die Sucherei nach verstreuten Containerdateien, sodass es sich durchaus lohnt, das Menü „Favorites“ zu organisieren.

Beachten Sie, dass Sie beim Mounten von Veracrypt-Containern zusätzlich zum Container-Passwort auch noch nach dem sudo-Kennwort gefragt werden, das mit dem Veracrypt-Passwort nichts zu tun hat und vermutlich anders lautet.

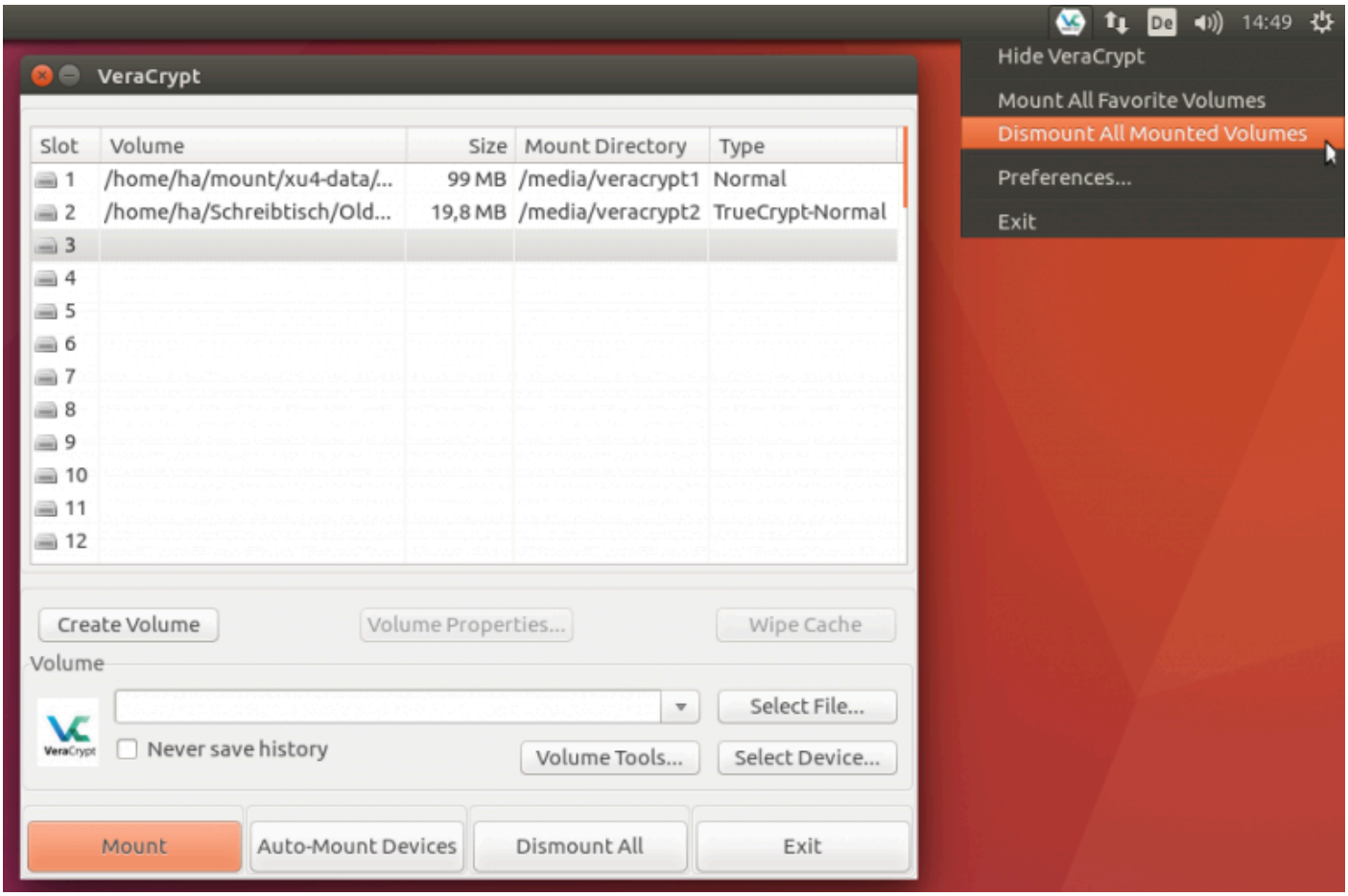
Tipp 1: Veracrypt ist auch komplett über Terminalbefehle zu steuern (siehe *veracrypt –help*). So entlädt etwa

```
veracrypt --dismount
```

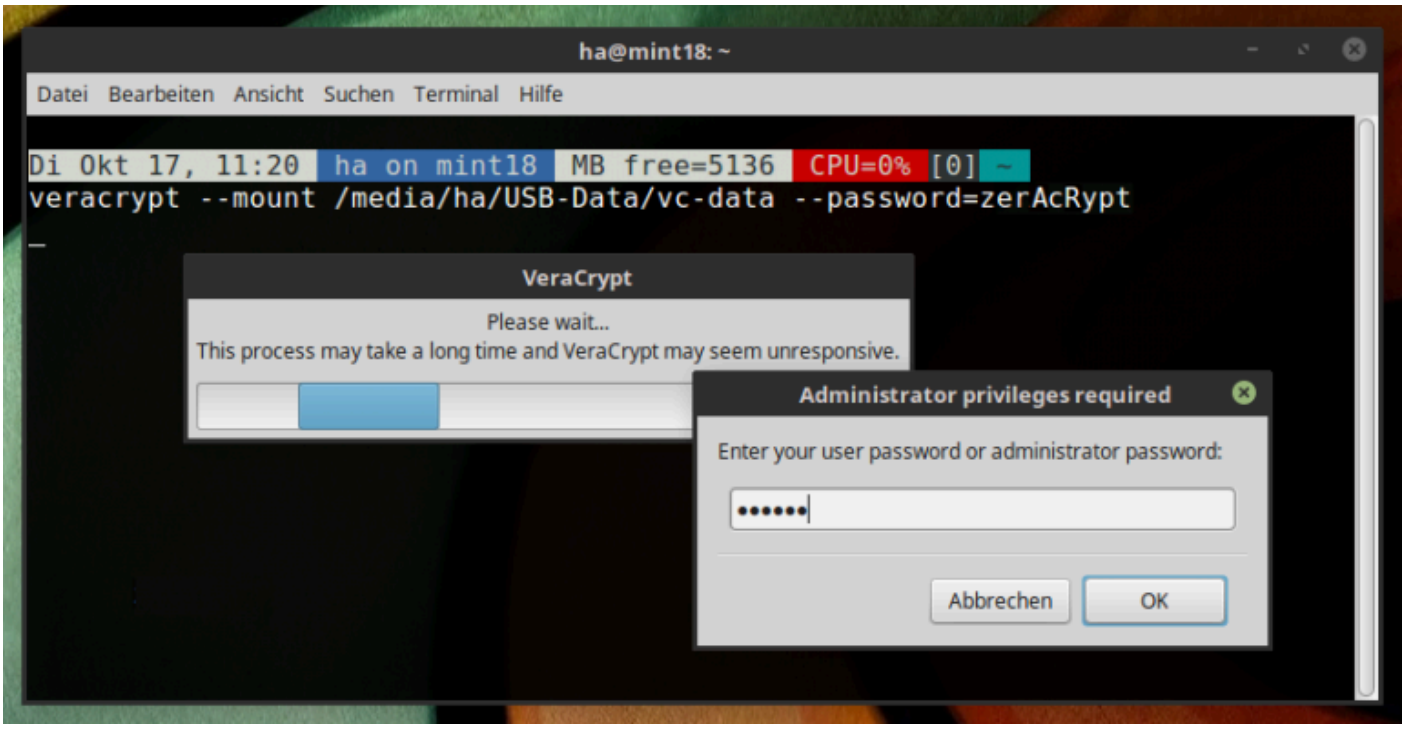
alle geladenen Containerdateien. Und auf stationären, privaten PCs kann es vertretbar sein, einen Container durch ein Terminal-Alias zu laden und dabei das Passwort im Klartext mitzugeben:

```
veracrypt --mount /home/ha/vc-data
--password=Sehr-GeH3im
```

Tipp 2 für ältere Truecrypt-Container: Unter „Select File“ wählen Sie die Container-Datei aus oder mit „Select Device“ das verschlüsselte Laufwerk. Nach einem Klick auf „Mount“ aktivieren Sie im angezeigten Dialog vor der Angabe des Passworts die Option „TrueCrypt Mode“. In einigen Fällen funktioniert das nicht, und Veracrypt öffnet den Container nicht. Hier hilft es, die Einstellung „Options -> Mount volume as read-only“ zu aktivieren. Damit sind die Daten erreichbar und können bei Bedarf an eine andere Stelle kopiert werden.



Hauptdialog und Systemintegration von Veracrypt: Veracrypt entspricht bei der Bedienung seinem Truecrypt-Vorgänger. Laden und Entladen von Containern gelingt auch über das Panelsymbol.



Veracrypt im Terminal: Alle Funktionen der Verschlüsselungs-Software sind auf Wunsch auch über die Kommandozeile zu steuern.

6. Einfaches Verschlüsseln mit Packer

Einfachster Schutz bei geringeren Datenmengen ist die Ad-hoc-Verschlüsselung von Einzeldateien oder eines Ordners. Ohne Einschränkung für alle Dateien und Ordner anwendbar ist ein Packer mit eingebauter Verschlüsselung wie 7-Zip. Diese ist sicher, wenn Sie das Passwort komplex und lang wählen. Packer-Verschlüsselung erfordert diszipliniertes Verhalten und ist nicht so komfortabel wie andere Methoden.

Empfehlung: Geschützte Packer-Archive eignen sich für Dateien in der Cloud, können aber auch für mobile Datenträger ausreichen, wenn die Dateimengen überschaubar sind. Da es 7-Zip für Linux, Windows und Mac OS (7zX) gibt, ist der Austausch solcher Archive problemlos. Unter Android können nicht alle Apps mit einem „zip“ im Namen auch mit passwortgeschützten Archiven umgehen, aber der kostenlose, allerdings werbefinanzierte 7Zipper (von Polar Bear) beherrscht dies.

Manuelles Verpacken: Installieren Sie zunächst, sofern noch nicht geschehen, den 7-Zip-Packer:

```
sudo apt install p7zip-full
```

7-Zip erscheint unter Desktop-Linux nicht als selbständiges, grafisches Programm, sondern integriert sich in die „Archivverwaltung“. In Zusammenarbeit mit dieser Archivverwaltung oder dem 7z-Filemanager unter Windows ist Verschlüsseln und Entschlüsseln recht komfortabel: Sie ziehen Datei oder Ordner einfach mit der Maus in das Fenster („Archivverwaltung“ oder „7-Zip“), bestätigen unter Linux, dass damit ein neues Archiv angelegt werden soll und geben dann das Format „7z“ an. Unter „Erweiterte Einstellungen“ vergeben Sie das Passwort. Die Option „Dateiliste ebenfalls verschlüsseln“ sorgt dafür, dass die Archivverwaltung später auch keine Dateinamen verrät. Beim späteren

Doppelklick des Archivs wird automatisch das Kennwort abgefragt und nur bei richtiger Eingabe entpackt.

Komfortfunktionen: Der Hauptaufwand für sichere passwortgeschützte Archive entsteht durch die Eingabe des komplexen Kennworts. Dieser Komfortverlust lässt sich etwa durch Terminal-Aliases oder -Funktionen minimieren. Für hier soll ein Beispiel für das Terminal genügen, das am besten als Function in der Datei ~/.bashrc zu realisieren ist:

```
function cc (
```

```
{
```

```
name=${1%/}
```

```
echo $name | grep ".7zEnc"
```

```
if [ $? -eq 0 ]; then
```

```
  7z x
-p'linuX*Welt' "$name"
```

```
else
```

```
  7z a
-p'linuX*Welt' -t7z -mhe=on "$name.7zEnc" "$name"
```

```
fi
```

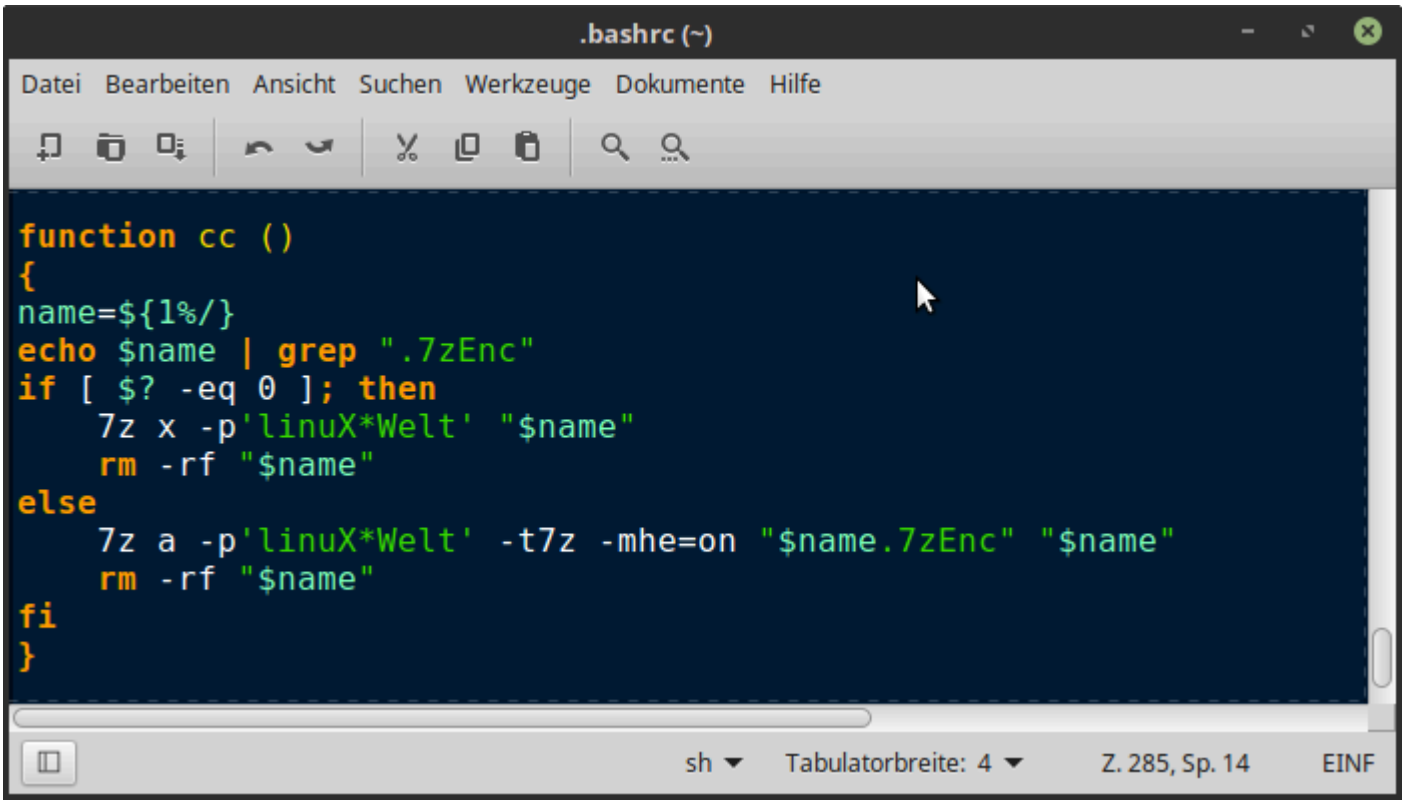
```
}
```

Danach erledigt im Terminal die Eingabe

```
cc [datei]
```

das Ein- oder Auspacken des Archivs im aktuellen Verzeichnis. Ob Einpacken oder Auspacken als Aufgabe ansteht, erkennt die Funktion anhand der Dateiextension. Das Kennwort „linuX*Welt“ ist natürlich anzupassen.

Ähnliche und zum Teil noch komfortablere grafische Lösungen hat die LinuxWelt in früheren Ausgaben vorgestellt, so den Ausbau des jeweiligen Dateimanagers mit speziellen Kontextmenüs. Die letzte Ausgabe der LinuxWelt zeigte eine Lösung mit einem Incron-überwachten Ordner, der die Verschlüsselung per Drag & Drop erledigt. Alle nötigen Infos dazu finden Sie im PDF-Booklet auf Heft-DVD. Im Prinzip basieren aber alle diese Komfortlösungen auf einem Shellscript ähnlicher Machart wie oben.



Vereinfachte Packer-Verschlüsselung: Ein Script kann die lästige Aufgabe übernehmen, das Kennwort an 7-Zip zu übergeben.

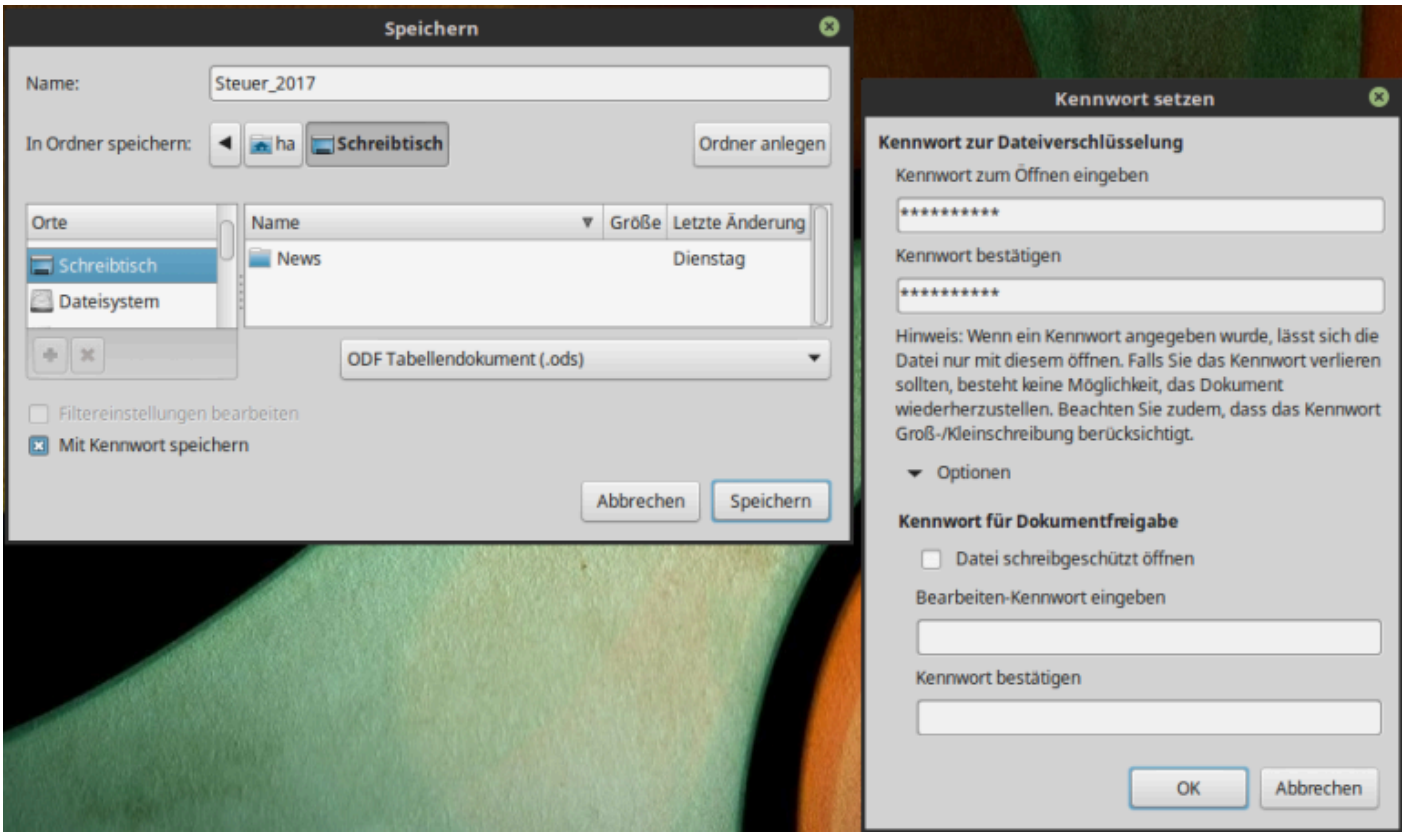
7. Kennwortschutz in Office-Software

Libre Office und Microsoft Office bieten eine eigene integrierte Verschlüsselung. Das ist bequem, bleibt aber eine Insellösung, die auf die wenigen Office-Formate beschränkt ist. Außerdem hat diese Software-interne Kryptographie den großen Nachteil, dass Sie auf Office-Suiten angewiesen sind, um ein Dokument lesen zu können. Immerhin kann Libre Office auch Passwort-geschützte Microsoft-Dateien öffnen, umgekehrt ist das nicht der Fall.

Empfehlung: Diese Methode, Dateien einzeln zu verschlüsseln, eignet sich nur für wenige sensible Texte oder Tabellen. Für größere Datenmengen ist sie zu unbequem. Der häufigste Einsatz ist der

Austausch vertraulicher Tabellen innerhalb eines Arbeitsteams.

Praktische Nutzung: Libre Office bietet die Option „Datei -> Speichern unter -> Mit Kennwort speichern“. Das Kennwort muss jeweils beim Öffnen eingegeben werden. Dass das Dokument geschützt ist, ist Libre Office bei der Weiterbearbeitung klar: Es genügt daher künftig, normal zu speichern. In Microsoft Office findet sich die Verschlüsselung unter „Datei -> Speichern unter -> Tools -> Allgemeine Optionen“.



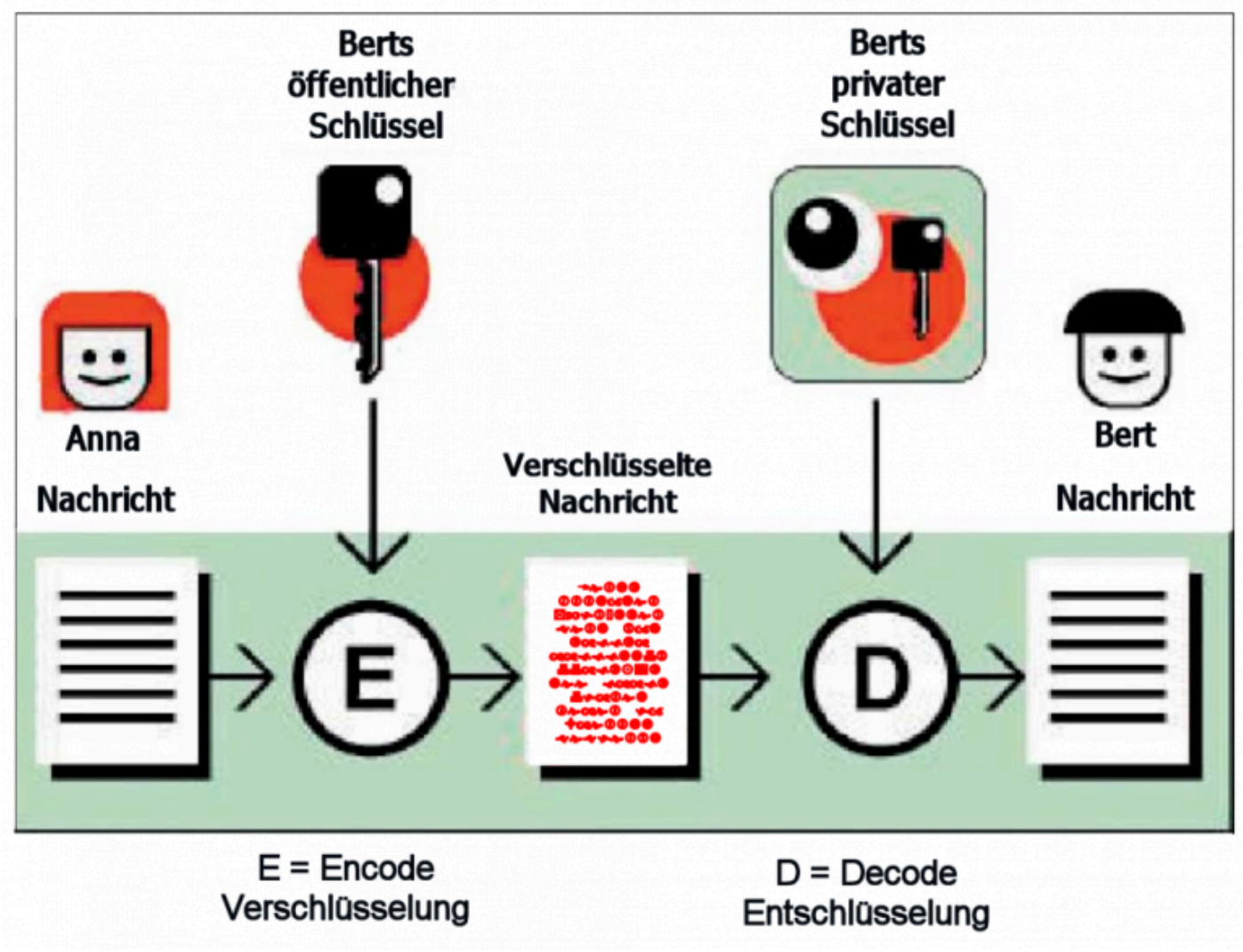
Einzeldateien unter Libre Office verschlüsseln: Diese Ad-hoc-Maßnahme ist ein Notbehelf für geringe Datenmengen.

Exkurs: Asymmetrische Verschlüsselung

Alle bisher genannten Verschlüsselungsvarianten (Punkt 1 bis 8) gehören zur Kategorie symmetrischer Verschlüsselung: Ein Kennwortschlüssel verändert die Ausgangsdaten unlesbar, genau derselbe Schlüssel stellt den lesbaren Zustand wieder her. Dieses Verfahren ist optimal, wenn Sender und Empfänger dieselbe Person sind: Sie verschlüsseln Dateien oder Datenträger, die Sie später wieder entschlüsseln. Die Verschlüsselung hat nur den Zweck, dass keine andere Person die Datei lesen kann. Sobald Sender und Empfänger verschiedene Personen sind, wird symmetrische Verschlüsselung problematisch: Erstens muss der Schlüssel auf einem sicheren Weg von Person A zu Person B kommen. Zweitens brauchen Sie strenggenommen für Person C einen anderen Schlüssel, für Person D einen weiteren und so fort. Bei einem Austausch vieler Personen wie bei der Mail-Korrespondenz ist dies nicht praktikabel.

Wesentliches Merkmal der asymmetrischen Verschlüsselung (siehe Punkt 9) sind zwei unabhängige Schlüssel: ein öffentlicher zum Verschlüsseln, ein privater zum Entschlüsseln. Die komplementären Schlüssel generiert die Software – etwa GnuPG – auf Ihrem Rechner. Beide Schlüssel stehen zwar in

eindeutigem Verhältnis, jedoch ist die Berechnung des privaten Schlüssels aus dem öffentlichen durch den Einsatz mathematischer Einwegfunktionen extrem aufwendig bis unmöglich. Der öffentliche Schlüssel zum Verschlüsseln kann daher ohne Geheimniskrämerei an alle Kommunikationspartner direkt oder zu einem öffentlichen Key-Server im Web geschickt werden. Nun chiffrieren alle Partner Nachrichten an Sie mit Ihrem öffentlichen Schlüssel – und Sie sind die einzige Person, die diese Nachrichten mit dem passenden privatem Schlüssel lesbar machen kann. Umgekehrt codieren Sie Ihre Nachrichten mit den öffentlichen Schlüsseln Ihrer Partner und haben die Sicherheit, dass nur der Empfänger mit dem komplementären privaten Schlüssel die Nachricht lesen kann.



Asymmetrische Verschlüsselung: Der Absender einer verschlüsselten Nachricht benötigt zum Chiffrieren („E" – Encrypt) nur den öffentlichen Schlüssel des Empfängers. Empfangen und entschlüsselt („D" – Decrypt) wird mit dem privaten Schlüssel.

8. Mailverschlüsselung unter Thunderbird

Ob der persönliche Mail-Austausch Verschlüsselung benötigt, muss jeder selbst entscheiden. Tatsache ist, dass US-Anbieter wie Google oder Yahoo der Neugier von Geheimdiensten wenig Datenschutz-Anstrengungen entgegensetzen. Auch wenn Sie deutsche Provider oder sogar einen eigenen Mail-Server benutzen, ist die Mail doch an den Knotenpunkten theoretisch abzufangen – am einfachsten in öffentlichen WLANs.

Empfehlung: Mail-Verschlüsselung ist wie jede Datenschutzmaßnahme mit Mehraufwand verbunden. Die Kombination von GnuPG (GNU Privacy Guard) plus Thunderbird mit Erweiterung Enigmail ist die wohl komfortabelste Lösung, aber auch sie erfordert Gewöhnung und zumindest einen Anteil von Mail-Partnern, die ebenfalls GnuPG nutzen. Unter Linux sind Thunderbird und GnuPG oft vorinstalliert, und falls nicht, über die Paketnamen „thunderbird“ und „gnupg“ schnell nachgerüstet. Für Windows gibt es Downloads unter www.mozilla.org und www.gnupg.org). Enigmail finden und installieren Sie dann direkt in Thunderbird über „Add-ons“.

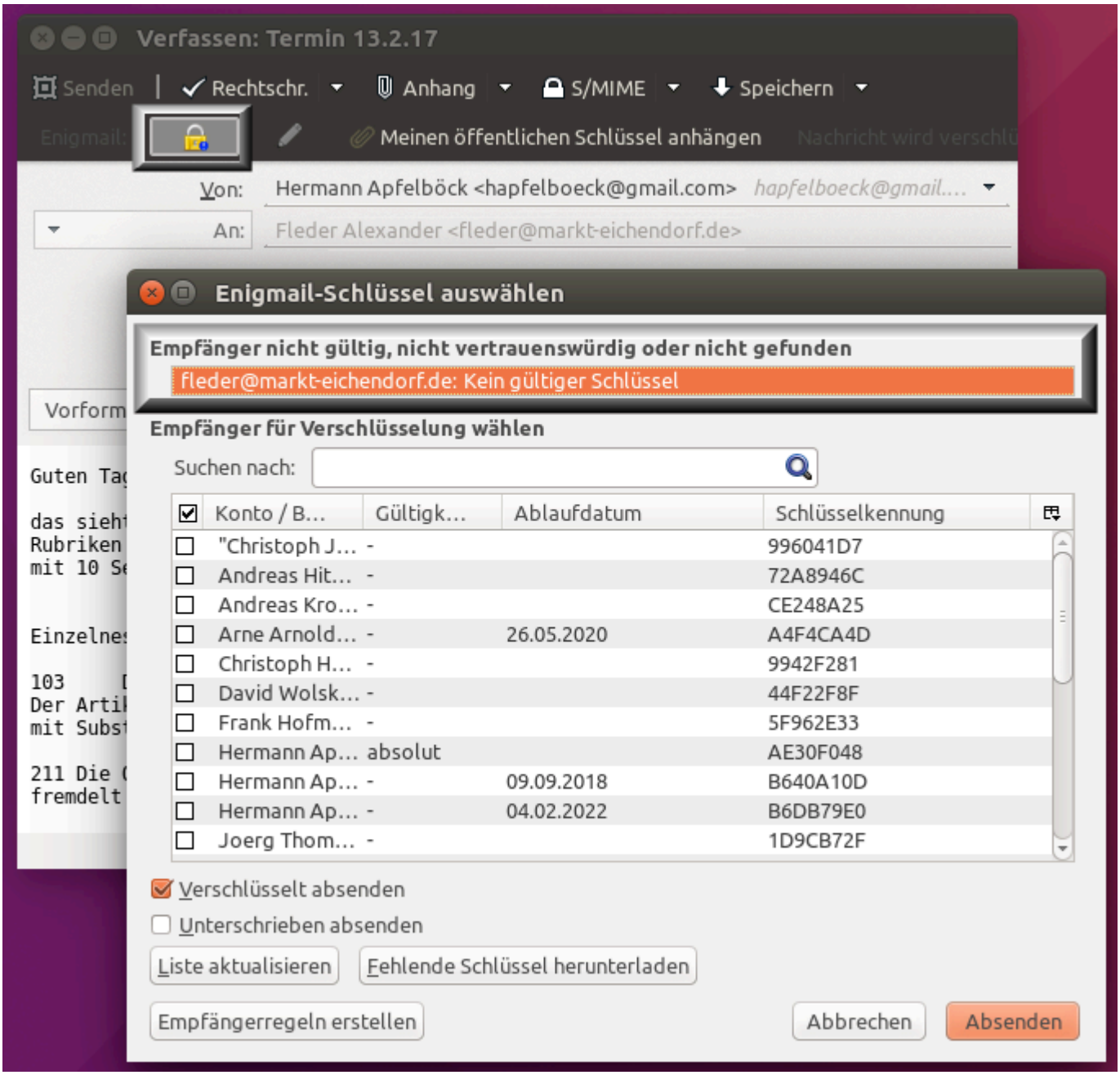
Einrichtung und Mail-Alltag: Nach der Installation der Enigmail-Erweiterung und einem Thunderbird-Neustart verwenden Sie im automatisch startenden Einrichtungsassistenten die „ausführliche Konfiguration“. Im ersten Schritt geben Sie die „Passphrase“ ein. Das Passwort benötigen Sie später stets, um auf Ihre Schlüssel zuzugreifen. Es bildet auch die Grundlage für die beiden Schlüssel. Nach der doppelten Eingabe legt Enigmail das neue Schlüsselpaar an (öffentlich/privat). Falls Sie auf einem anderen Rechner bereits ein eingerichtetes Enigmail und ein Schlüsselpaar besitzen, wählen Sie im Assistenten die Option, bestehende Schlüssel zu importieren. Schlüssel lassen sich über „Enigmail -> Schlüssel verwalten“ als Ascii-Dateien exportieren und auf anderen Rechnern importieren.

Öffnen Sie wie gewohnt den Editor zum Verfassen von Nachrichten. Dort hat Enigmail jetzt eine weitere Symbolleiste platziert. Möchten Sie eine ausgehende Nachricht verschlüsseln, benötigen Sie den öffentlichen Schlüssel des Empfängers. Wenn dieser als Textdatei vorliegt, können Sie den Schlüssel über „Enigmail -> Schlüssel verwalten -> Datei importieren“ einlesen. Alternativ gibt es Schlüsselserver, die öffentliche Schlüssel aufbewahren. Über „Schlüsselserver -> Schlüssel suchen“ sehen Sie nach, ob die Empfängeradresse dort eingetragen ist; falls ja, importieren Sie den Schlüssel mit einem Klick. Umgekehrt ist es sinnvoll, den eigenen öffentlichen Schlüssel über „Schlüsselserver -> Schlüssel hochladen“ im Web zugänglich zu machen.

Nach einem Schlüsselimport ist der neue Mail-Empfänger Enigmail/GnuPG bekannt. Künftig klicken Sie beim Verfassen einer Nachricht an diesen Empfänger auf das Symbol mit dem Schloss. Um Mails verschlüsselt zu versenden, müssen Sie Ihr Passwort eingeben. Wenn Sie mit dem Schloss-Symbol verschlüsselt senden wollen, jedoch für den Empfänger kein Schlüssel vorliegt, erscheint automatisch der Hinweis, dass dieser Empfänger „nicht gültig“ ist. Dann besorgen Sie sich entweder den öffentlichen Schlüssel oder Sie senden unverschlüsselt.

Erhalten Sie umgekehrt eine Mail, die verschlüsselt wurde, erkennt Enigmail das automatisch. Wenn Sie im Vorschaubereich von Thunderbird auf das Element klicken, werden Sie dazu aufgefordert, das Passwort einzugeben. Wenige Augenblicke später erscheint die Nachricht.

Beachten Sie, dass Sie bei der Nutzung mehrerer Rechner die Schlüsselverwaltung manuell synchron halten müssen. Eine wichtige Hilfe ist wieder „Enigmail -> Schlüssel verwalten -> Datei exportieren“, wobei Sie einfach sämtliche Schlüssel markieren. Die resultierende Ascii-Datei lässt sich auf dem nächsten Rechner importieren.



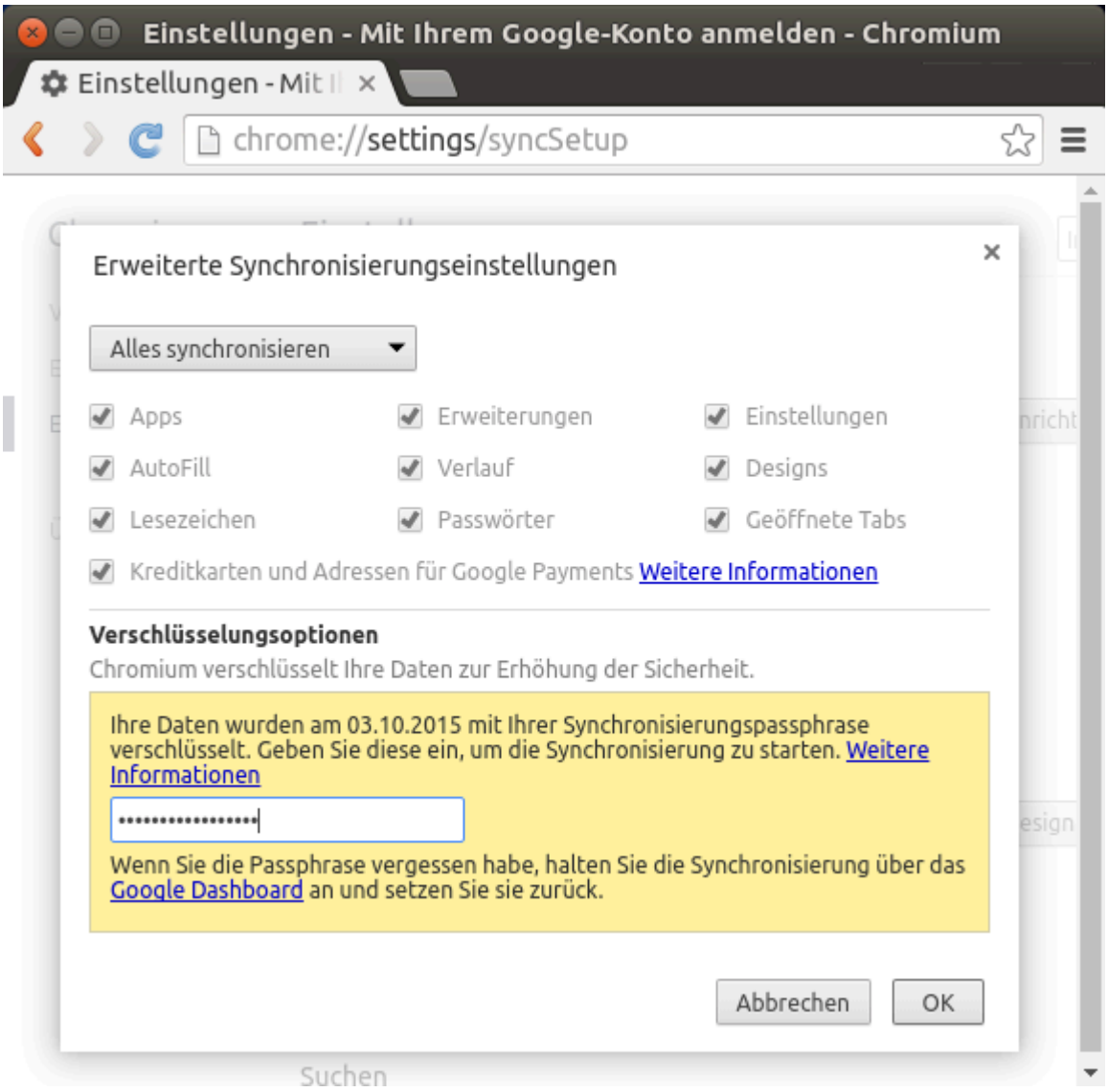
Verschlüsselte Mail mit GnuPG und Enigmail: Wenn Sie versuchen, verschlüsselt zu senden, aber für den Adressaten kein Schlüssel vorliegt, öffnet sich automatisch die Schlüsselverwaltung.

9. Verschlüsselte Browser-Synchronisierung

Die Browser-Synchronisierung von Lesezeichen, Einstellungen, Erweiterungen und Skins bedeutet für Nutzer mehrerer Geräte unschätzbaren Komfort. Bedenklich scheint allerdings der Nebenaspekt, dass dabei Mengen von persönlichen Daten auf Google- oder Mozilla-Servern hinterlegt werden müssen.

Empfehlung: Firefox verschlüsselt standardmäßig alle Daten, wobei der Schlüssel auf dem Gerät des Benutzers verbleibt. Damit ist der Mozilla-Browser in puncto Datenschutz erste Wahl. Jedoch lässt sich auch der Google-Browser so einstellen, dass alle Synchronisierungsdaten sicher verschlüsselt sind.

Abhörsichere Synchronisierung für Chrome/Chromium: Standardmäßig werden hier nur die Kennwörter verschlüsselt. Aber unter „Einstellungen -> Erweiterte Synchronisierungseinstellungen“ (vorherige Google-Anmeldung vorausgesetzt) gibt es die Option „Alle synchronisierten Daten […] verschlüsseln“, bei der Sie ein individuelles Kennwort zur Sync-Verschlüsselung vergeben, das unabhängig vom Google-Kennwort ist. Der Komfortverlust ist nicht gravierend, da Sie dieses Kennwort auf jedem weiteren Gerät nur ein einziges Mal eingeben müssen. Damit landen sämtliche Daten verschlüsselt auf dem Google-Server, der Schlüssel dazu (Kennwort) verbleibt auf dem lokalen Gerät.



Sync-Daten verschlüsseln: Diese Maßnahme hält Googles Big-Data-Sammler von Lesezeichen und Verlaufsdaten fern.

10. Verschlüsselung im Internet

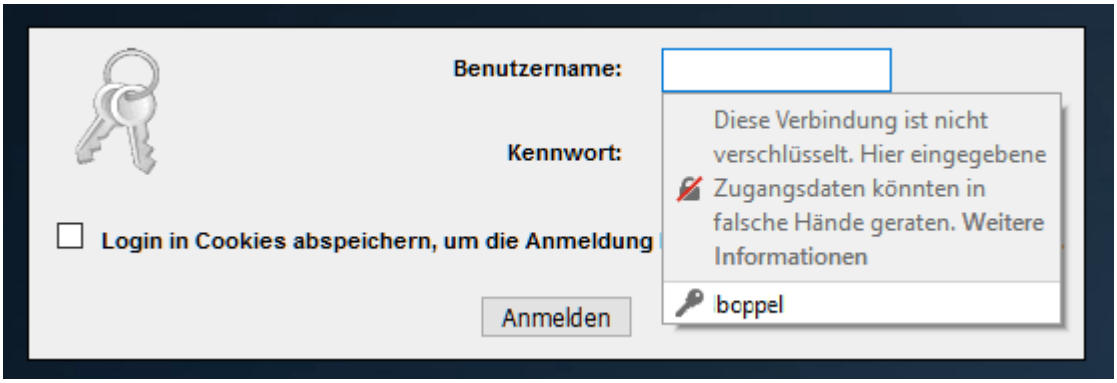
Der Datenaustausch über unverschlüsselte Verbindungen ins Internet kann im Klartext mitgelesen werden. Das gilt verschärft in öffentlichen Funknetzen, innerhalb von lokalen Firmennetzen und theoretisch auch außerhalb des lokalen Netzwerks an Verteilerknoten, die von Providern, Geheimdiensten oder Hackern abgehört werden. Im Fokus stehen die meistgenutzten Protokolle HTTP (Webseiten) und FTP (Datentransfer). Die folgenden Infos beziehen sich ausschließlich auf die Client-Seite des Web-Nutzers, nicht auf die Server-Seite des Betreibers.

HTTP und HTTPS: Im Sinne des Datenschutzes ist, wo immer möglich, auf verschlüsselte Verbindung zu achten. Zwingend erforderlich ist dies überall, wo zur Anmeldung persönliche Zugangsdaten verschickt werden (Bank, Online-Shop). Alle Browser zeigen sichere (HTTPS-) Web-Adressen in der Adresszeile mit einem grünen Schloss-Symbol an. HTTPS garantiert, dass es für Kriminelle und Geheimdienste selbst dann nichts Lesbares zu lesen gibt, wenn der Angreifer im Netz sitzt und den Netzverkehr abhört.

Firefox und Chrome signalisieren unverschlüsselte HTTP-Sites explizit als „nicht sicher". Dies ist kein Urteil über die Seriosität der Website, sondern ausschließlich die Aussage, dass eine Site keine TLS- (Transport Layer Security) oder SSL-Verschlüsselung bietet (Secure Socket Layer). Datenschutztechnisch noch einen Schritt weiter geht Firefox, der Anmeldungen auf unverschlüsselten Seiten automatisch bremst: „Diese Verbindung ist nicht verschlüsselt…". Das ist im Prinzip verdienstvoll, kann aber – insbesondere bei lokalen Servern (Router, NAS) – auch nerven und über „about:config" deaktiviert werden („security.insecure_field_warning…").

FTP, FTPS und SFTP: Das File Transfer Protocol (FTP) bietet keine Verschlüsselung. Daher sollten Sie sich die Anmeldung auf unverschlüsselten FTP-Servern zumindest in öffentlichen WLANs verkneifen. Man mag sich als Client-Nutzer auf den Standpunkt stellen, das Sicherheitsproblem sei Sache des Server-Betreibers. Jedoch fällt der Erstverdacht zunächst auf den Client-Nutzer, wenn dessen unverschlüsselte Anmeldedaten abgegriffen und destruktiv missbraucht werden. Sicherheitsbewusste FTP-Betreiber werden FTPS (FTP mit SSL- oder TLS-Verschlüsselung) anbieten. FTP-Clients wie Filezilla zeigen im Servermanager unter „Verschlüsselung" an, ob die Verbindung abhörsicher ist.

Eine sichere Alternative zu FTP ist der Datenaustausch über SSH, das über sein Protokoll SFTP auch die direkte verschlüsselte Dateiübertragung vorsieht. Mit den ähnlich klingenden Protokollen FTP und FTPS hat das nichts zu tun, sondern mit SSH-Servern, die auf Linux-Systemen SSH-Verbindungen entgegennehmen. Mit Rücksicht auf Windows-Systeme, die standardmäßig keinen SSH-Client enthalten, bleibt FTP und FTPS das verbreitete Austauschprotokoll. Wirklich triftig ist diese Rücksicht auf Windows allerdings nicht, da der auch unter Windows vielgenutzte FTP-Client Filezilla auch das Protokoll „SFTP – SSH File Transfer Protocol" beherrscht.



Firefox ultravorsichtig: Der Mozilla-Browser zeigt bei Anmeldungen auf unverschlüsselten Web-Seiten diese Warnung.

Exkurs: Unentbehrlicher Browser-Tipp

Mit dem Thema „Verschlüsselung“ hat dieser kleine Exkurs nichts zu tun, aber viel mit dem verwandten Thema „Datenschutz“: Wer gerade vorhat, sich eine Jacke, Gitarre oder Kettensäge zu kaufen, sollte die Angebote tunlichst nicht über normales Google & Co. recherchieren. Dann sieht man nämlich die nächsten Wochen im Web überall nur noch Jacken, Gitarren und Kettensägen. Einfache Abhilfe schafft der „Private Modus“ im Firefox oder „Inkognito“ bei Chrome. Damit können Sie in Google & Co. suchen, ohne die Werbeindustrie über Ihre Interessen zu informieren. Wer solche Belästigung generell hasst, kann auch die Suchmaschine duckduckgo.com verwenden – dort ist der Datenschutz inklusive.