

Leitfaden: Verschlüsseltes Google Drive als natives Laufwerk unter macOS

Die Integration von Cloud-Speichern wie Google Drive direkt in das macOS-Dateisystem bietet einen erheblichen Komfortgewinn. Werden Dateien jedoch unverschlüsselt übertragen und gespeichert, gibt man die Souveränität über die eigenen Daten vollständig an den Provider ab. Die Kombination aus **Rclone** und einer **FUSE-Schicht** löst dieses Problem, indem sie eine lokale Verschlüsselungsebene einzieht.

Bei der Wahl der technischen Schnittstelle stehen Nutzer heute jedoch vor einer Richtungsentscheidung: dem bewährten, aber alternden **macFUSE** oder dem modernen **FUSE-T**.

Warum FUSE-T die notwendige Wahl für moderne Systeme ist

Lange Zeit war **macFUSE** der unangefochtene Standard. Es operiert jedoch als **Kernel-Extension (KEXT)**. Da Apple die Architektur von macOS konsequent in Richtung eines abgesicherten Kernels entwickelt, bringt dieser Ansatz auf aktuellen Systemen (insbesondere mit Apple Silicon M1/M2/M3-Chips) gravierende Nachteile mit sich:

1. **Eingriff in die Systemintegrität (SIP):** Um macFUSE zu nutzen, muss die Sicherheit des Macs im Wiederherstellungsmodus auf „**Reduzierte Sicherheit**“ herabgestuft werden. Dies öffnet Tür und Tor für potenzielle Angriffe auf Systemebene, da die strikte Trennung zwischen Drittanbieter-Software und dem Betriebssystemkern aufgeweicht wird.
2. **Systemstabilität:** Da macFUSE im Kernel-Space läuft, führt ein Programmierfehler oder ein Konflikt nicht nur zum Absturz der App, sondern oft zu einem **Kernel Panic** – der gesamte Mac stürzt ab.
3. **Zukunftssicherheit:** Apple hat den Support für Kernel-Extensions faktisch abgekündigt. In kommenden macOS-Versionen wird die Nutzung dieser Treiber immer hürdenreicher oder gänzlich unmöglich.

FUSE-T hingegen geht einen fundamental anderen Weg. Es arbeitet vollständig im **User-Space**. Statt sich tief in das System einzugraben, nutzt es moderne Protokolle wie NFS oder das neue Apple FSKit, um dem Finder vorzugaukeln, es handele sich um ein lokales Netzwerk-Laufwerk.

Die Vorteile von FUSE-T auf einen Blick:

- **Volle Sicherheit:** Dein Mac bleibt im Modus „Volle Sicherheit“. SIP (System Integrity Protection) muss nicht angetastet werden.
- **Stabilität:** Ein Fehler im Dateisystem-Prozess beendet lediglich den Mount, reißt aber niemals das gesamte Betriebssystem mit in den Abgrund.
- **Einfachheit:** Keine komplizierten Neustarts im Recovery-Modus nötig. Die Installation erfolgt sauber und ohne tiefgreifende Systemänderungen.

Mit diesem Setup schaffst du eine Umgebung, die deine Daten schützt (durch Rclone-Verschlüsselung) und gleichzeitig die Integrität deines Macs bewahrt (durch FUSE-T).

Leitfaden: Verschlüsseltes Google Drive als natives Laufwerk unter macOS

Schritte der Installation:

[1. Die Installation der Software unter \(macOS\)](#)

[2. Konfiguration](#)

[Schritt A: Basis-Remote \(gdrive_secure\)](#)

[Schritt B: Crypt-Remote \(gdrive_crypt\)](#)

[3. Die dauerhafte Einbindung \(macOS\)](#)

[A. Den Ordner erstellen](#)

[B. Die LaunchAgent-Datei erstellen](#)

[C. Den Dienst aktivieren](#)

[4. Wiederherstellung \(Time Machine\)](#)

1. Die Installation der Software unter (macOS)

Wichtig: Installiere Rclone **nicht** über Homebrew, da die Brew-Version unter macOS den mount-Befehl nicht unterstützt.

4. FUSE-T installieren (via Brew okay):

```
Bash
brew tap macos-fuse-t/homebrew-cask
brew install --cask fuse-t
```

5. Rclone (Offizielle Binary) installieren:

```
Bash
sudo -v ; curl https://rclone.org/install.sh | sudo bash
```

Prüfung:

```
Bash
rclone version
```

Dies sollte eine aktuelle Version wie Darwin anzeigen.

Leitfaden: Verschlüsseltes Google Drive als natives Laufwerk unter macOS

2. Konfiguration

Führe

```
Bash  
rclone config
```

aus und folge diesen Schritten:

Schritt A: Basis-Remote (gdrive_secure)

- n (New Remote) -> Name: gdrive_secure -> Typ: drive.
- Alle Abfragen (Client-ID, Service Account, Advanced Config) mit **Enter** (leer lassen) überspringen.
- Use web browser to automatically authenticate? -> **y**.
- Im Browser einloggen und Zugriff erlauben.
- Configure this as a Shared Drive? -> **n**.
- Mit y bestätigen und speichern.

Schritt B: Crypt-Remote (gdrive_crypt)

- n -> Name: gdrive_crypt -> Typ: crypt.
- Remote to encrypt: gdrive_secure:geheim
- filename_encryption: **1** (standard).
- directory_name_encryption: **1** (true).
- **Passwörter:** Wähle y, um dein eigenes **Secret** und danach dein eigenes **Salt** einzugeben. **Sichere beide Passwörter extern!**

3. Die dauerhafte Einbindung (macOS)

A. Den Ordner erstellen

```
Bash  
mkdir -p ~/Cloud/GoogleDrive
```

Leitfaden: Verschlüsseltes Google Drive als natives Laufwerk unter macOS

B. Die LaunchAgent-Datei erstellen

Erstelle die Datei plist-Datei mit nano

```
Bash
nano ~/Library/LaunchAgents/com.user.rclone.gdrive.plist
```

und füge diesen Inhalt ein (ersetze DEIN_USER durch deinen echten Namen):

```
XML
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>Label</key>
  <string>com.user.rclone.gdrive</string>
  <key>ProgramArguments</key>
  <array>
    <string>/usr/local/bin/rclone</string>
    <string>mount</string>
    <string>gdrive_crypt:</string>
    <string>/Users/DEIN_USER/Cloud/GoogleDrive</string>
    <string>--vfs-cache-mode</string>
    <string>full</string>
    <string>--vfs-cache-max-size</string>
    <string>3G</string>
    <string>--vfs-cache-max-age</string>
    <string>24h</string>
    <string>--volname</string>
    <string>Google Drive Crypt</string>
  </array>
  <key>RunAtLoad</key>
  <true/>
  <key>KeepAlive</key>
  <true/>
</dict>
</plist>
```

Leitfaden: Verschlüsseltes Google Drive als natives Laufwerk unter macOS

C. Den Dienst aktivieren

Damit macOS die Datei registriert und den Dienst startet:

```
Bash  
launchctl load ~/Library/LaunchAgents/com.user.rclone.gdrive.plist
```

4. Wiederherstellung (Time Machine)

Falls du deinen Mac neu aufsetzt:

1. **Installiere FUSE-T und Rclone** (Schritt 1 dieser Anleitung).
2. **Hole die Config zurück:** Kopiere rclone.conf aus dem Backup (versteckter Pfad ~/.config/rclone/) zurück an den Platz, falls sie nicht mehr da sein sollte
3. **OAuth-Token erneuern:**

```
Bash  
rclone config reconnect gdrive_secure:
```

4. **Dienst laden:** Führe Schritt 3 noch einmal aus.