Praxissession 1 - Grundlagen Blockchain

Linux

Installation der Wallet Software Electrum

1. Laden Sie die folgende Datei herunter: https://download.electrum.org/3.3.8/electrum-3.3.8-x86_64.AppImage

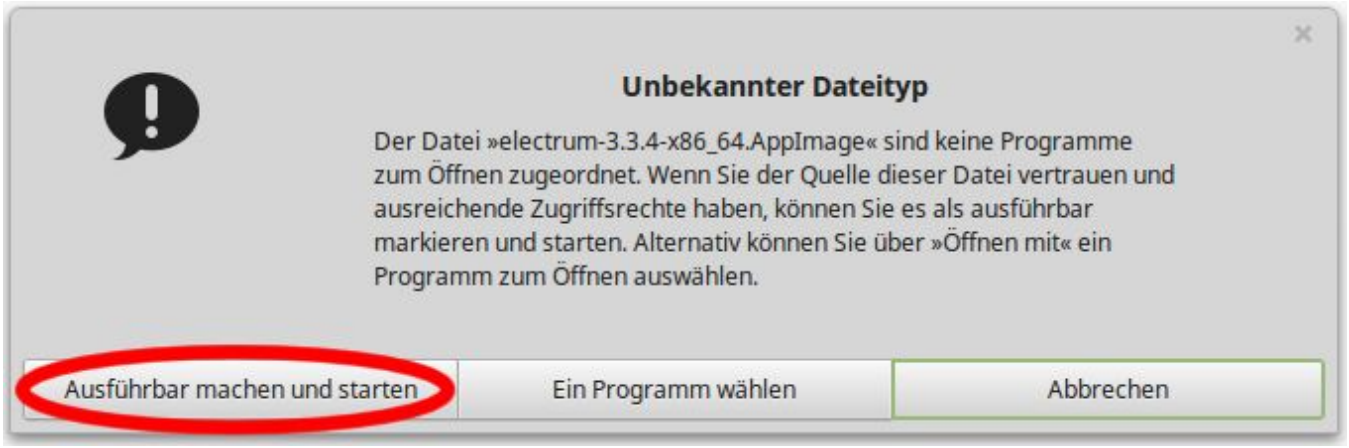
   Sie wird in folgendem Ordner gespeichert:

   ```
   /vol/vol_home_bcgst/bcgst<ihre persönliche Nummer>/Downloads
   ```

   Alternativ können Sie Electrum auch aus dieser vertrauenswürdigen Quelle beziehen

   ```
   /usr/local/Archiv/lehre/Meisel
   ```

2. Öffnen Sie den Dateiexplorer z.B. über das Startmenü (Klick auf das grüner Ordnersymbol) und Doppelklicken Sie auf die Datei.
   - Es erscheint eine Fehlermeldung. Klicken Sie auf "**Ausführbar machen und starten**"



- Sollte dies nicht funktionieren, klicken Sie mit der rechten Maustaste auf die Datei, wählen Sie Eigenschaften und Wählen Sie dann "Ausführbar machen" aus.

Konfiguration von Electrum und Import Ihres privaten Schlüssels

1. Im ersten Dialogfenster **Automatisch Verbinden** wählen und **Weiter** klicken
2. Im zweiten Dialogfenster einen Namen für Ihre Wallet wählen (kann man auf default_wallet lassen) und **Weiter** klicken
3. **Bitcoin-Adressen oder private Schlüssel importieren** auswählen

Öffnen Sie nun die folgende Website, die der Farbe ihrer Gruppe entspricht und geben Sie das Passwort ein.

Gruppe Rot: https://blockchain.hs-mittweida.de/2958-2/

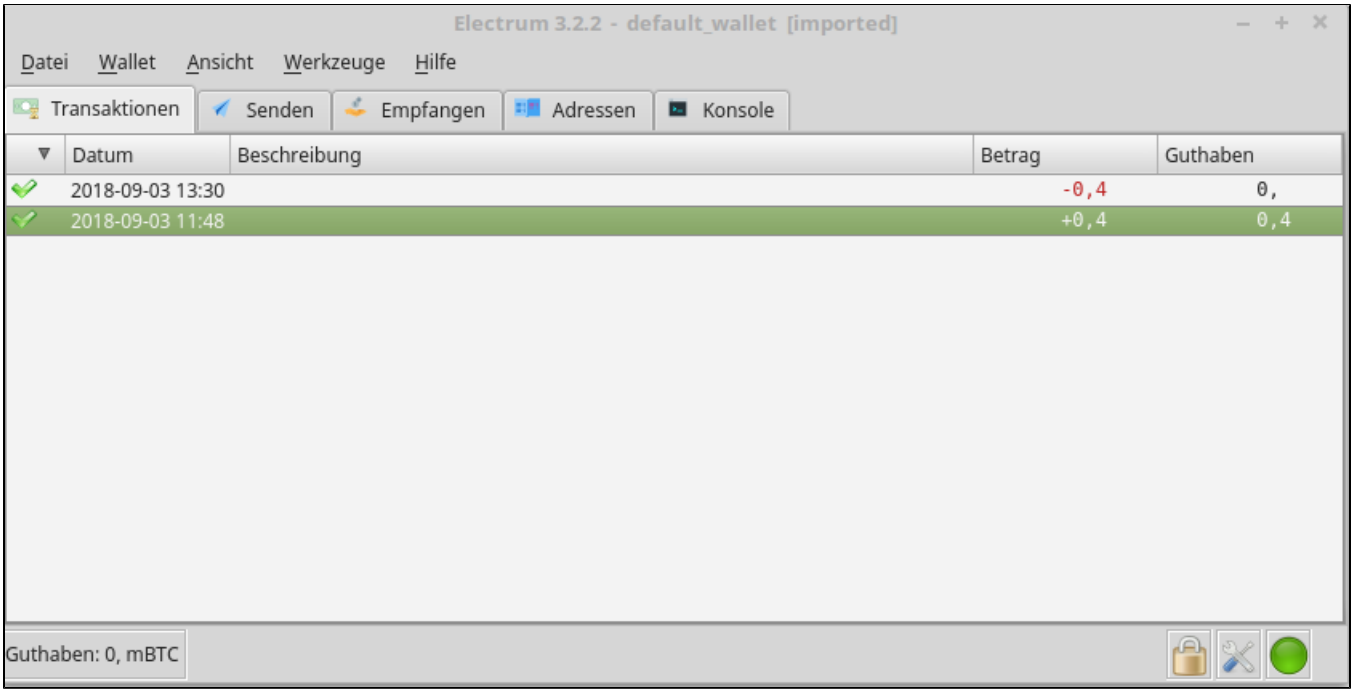
Gruppe Gelb: https://blockchain.hs-mittweida.de/private-keys-bas2019-gruppe-gelb/

Gruppe Blau: https://blockchain.hs-mittweida.de/2964-2/

Dort sehen Sie eine Liste mit Bitcoin-Adressen und den dazugehörigen privaten Schlüsseln.

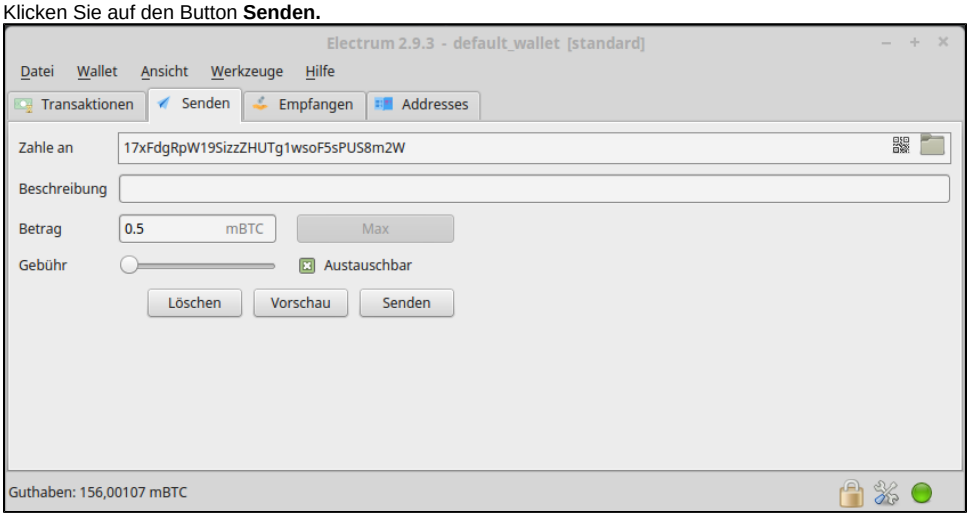
Nutzen Sie bitte nur die Adressen und Schlüssel, die Ihrer Nummer entsprechen.

1. Kopieren Sie sich nun den **Privaten Key aus der linken Spalte (Key A) der Liste (siehe oben)** aus der **Zeile entsprechend der Nummer, die Ihnen zugeteilt wurde**.
2. Passwort wählen
3. Folgendes Fenster sollte erscheinen, wenn das Wallet erfolgreich installiert wurde. Es sollte jetzt ein kleiner Betrag in mBTC angezeigt werden.



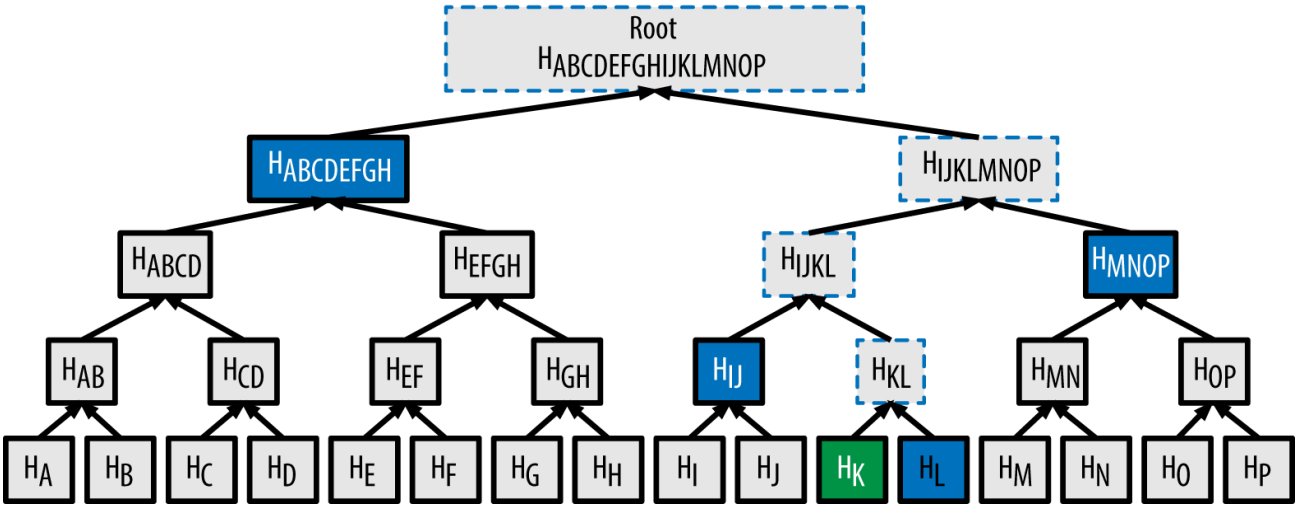
Anleitung Senden von Bitcoins

1. Wechseln Sie in den **Reiter: "Senden"**
2. Geben Sie in das Feld "Zahle an" (Pay to) die **zweite Adresse (Spalte ganz rechts Adresse B)** Ihrer "Paperwallet" (Tabelle der Website) ein. Achten Sie wieder darauf, die Adresse aus der Ihnen zugeteilten Zeilennummer zu verwenden!
3. Wählen Sie den zu sendenden Betrag (z. B. **0,13mBTC**).
4. Sie können dann die **Gebühren** verändern. Niedrigere Gebühren führen zu einer längeren Wartezeit. Lassen Sie die Gebühren (Fee) unverändert.
5. Klicken Sie auf den Button **Senden.**



6. Nachdem Sie die Transaktion erstellt haben, wechseln Sie bitte in den **Reiter: "Transaktionen"** und klicken Sie mit der rechten Maustaste auf die soeben erstellte Transaktion und dann auf **Details**.
7. Schauen Sie sich die Angaben an.
8. Kopieren Sie die Transaktions-ID und öffnen Sie die Webseite: https://www.blockchain.com/explorer.
9. Fügen Sie in das Suchfeld Ihre Transaktions-ID ein.
10. **Bis die TX im Netzwerk propagiert und in einen Block aufgenommen wurde, kann es eine Weile dauern.**
11. **In der Zwischenzeit schauen wir uns Merkle Trees und Bloom Filter an.**

Merkle Trees



Öffnen Sie nun ein neues Terminalfenster.

1. Installieren der JavaScript Bibliothek **merkletree:**

Terminalfenster / Konsole

```
$ npm install merkletree
```

2. Starten der **Node.js** Konsole:

```
$ node
```

Die folgenden Befehle werden innerhalb der soeben gestarteten Node.js Konsole (zu erkennen am **>** Symbol) ausgeführt.

1. Laden der Bibliothek **merkletree:**

Node.js Konsole

```
> merkle = require('merkletree')
```

2. Erzeugen von Testdaten:

```
> trxs = ['von a nach b','von x nach y','von t nach e','von m nach n']
```

3. Erzeugen eines neuen Merkle Trees aus den Testdaten:

```
> tree = merkle.default(trxs)
```

4. Anzeigen des Merkle Root:

```
> mroot = tree.root()

'288c3fa3bd8fc181b5…'
```

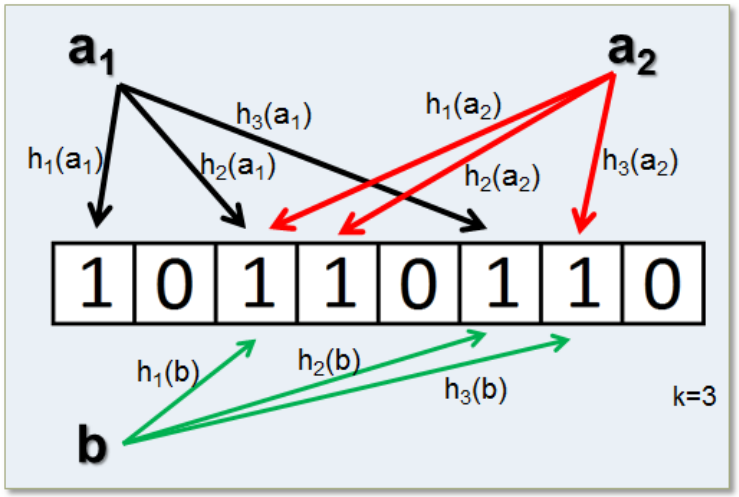
5. Merkle Pfad der Transaktion **'von x nach y'** berechnen:

```
> mpath = tree.proof('von x nach y')
```

6. Verschiedene Merkle Proofs durchführen:

```
> merkle.verifyProof('von x nach y', mroot, mpath)

true

> merkle.verifyProof('von a nach b', mroot, mpath)

true

> merkle.verifyProof('von m nach n', mroot, mpath)

false
```

Bloomfilter



Aufgabe

Es sollen die Worte "Baum", "Welt" und "DAS" in einem Bloomfilter der Länge 10 Bit eingetragen werden. Die Zahl der Hashfunktionen (k) beträgt 2.

Verwenden Sie folgende Hashfunktionen:

- H1: Einstellige Quersumme (Addieren aller ASCII-Werte der Buchstaben)
- H2: Einstellige Quersumme (Ermitteln aller ASCII-Werte + 1, dann addieren)

Eine ASCII-Tabelle finden Sie hier: https://de.wikipedia.org/wiki/American_Standard_Code_for_Information_Interchange.

H1(BAUM) = 66 + 65 + 85 + 77 = 293; Einstellige Quersumme: 2 + 9 + 3 = 14, 1+4 = 5

H2(BAUM) = 67 + 66 + 86 +78 = 297, Einstellige Quersumme: 2 + 9 + 7 = 18, 1 + 8 = 9

H1(WELT) = 87 + 69 + 76 + 84 = 316, Einstellige Quersumme: 3 + 1 + 6 = 10, 1 + 0 = 1

H2(WELT) = 88 + 70 + 77 + 85 = 320, Einstellige Quersumme: 3 + 2 + 0 = 5

H1(DAS) = 68 + 65 + 83 = 216, Einstellige Quersumme: 2 + 1 + 6 = 9

H2(DAS) = 69 + 66 + 84 = 219, Einstellige Quersumme: 2 + 1 + 9 = 12, 1 + 2 = 3

Der belegte Bloomfilter sieht dann so aus:

Bit	0	1	2	3	4	5	6	7	8	9
	0	1	0	1	0	1	0	0	0	1

Fragen

1. Kann man Werte aus einem einfachen Bloomfilter löschen?
2. Ist das Wort "DER" in dem oben erstellten Bloomfilter enthalten?
3. Ist das Wort "ABC" in dem oben erstellten Bloomfilter enthalten?

1. Nein
2. Nein, da H1(DER) = 3 und H2(DER) = 6 und das Bit 6 im Filter nicht gesetzt ist.
3. Wahrscheinlich, da H1(ABC) = 9 und H2(ABC) = 3, beide Bits sind gesetzt. Man kann aber nicht davon ausgehen, dass der Filter tatsächlich mit ABC belegt wurde.

Umsetzung in JavaScript

1. Öffnen Sie den Datei Explorer.
2. Speichern Sie die folgende Datei in Ihrem persönlichen Ordner. (Rechtsklick auf Link **Link speichern unter**):

   formatter.js

3. Öffnen Sie ein neues Terminalfenster.
4. Installieren der JavaScript Bibliothek **bloomfilter**:

   Terminalfenster / Konsole

   ```
   $ npm install bloomfilter
   ```

5. Wechseln Sie in das Terminalfenster mit der **Node.js** Konsole oder starten Sie eine neue Node.js Konsole:

   ```
   $ node
   ```

Die folgenden Befehle werden innerhalb der Node.js Konsole (zu erkennen am **>** Symbol) ausgeführt.

1. Laden der JavaScript Bibliothek **bloomfilter**:

   Node.js Konsole

   ```
   > bf = require('bloomfilter')
   ```

2. Laden der JavaScript Bibliothek **formatter**:

   ```
   > format = require('./formatter')
   ```

3. Neuen leeren Bloomfilter mit 32 Bit Länge und 3 Hashfunktionen erstellen:

   ```
   > filter = new bf.BloomFilter(32,3)
   ```

4. Leeren Bloomfilter anzeigen lassen:

   ```
   > format.toBin(filter)

   '00000000000000000000000000000000'
   ```

5. Neu Werte dem Filter hinzufügen und Filter anzeigen:

```
> filter.add('a')
> format.toBin(filter)

'00100000000000001000100000000000'

> filter.add('b')
> format.toBin(filter)

'00100000001000001000110000000001'

> filter.add('c')
> format.toBin(filter)

'10100000001100001000111000000001'
```

6. Testen, ob der Filter die enthaltenen Werte erkennt und nicht enthaltene Werte verwirft:

```
> filter.test('a')

true

> filter.test('x')

false
```

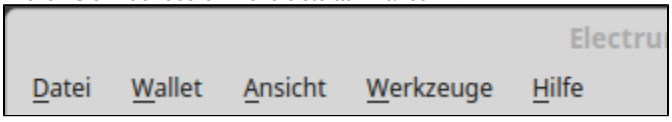
7. Neuen Filter mit 32 Bit Länge und **16 Hashfunktionen** anlegen und testen:

```
> filter = new bf.BloomFilter(32,16)
> filter.add('a')
> format.toBin(filter)

'10101010101010101010101010101010'

> filter.add('b')
> filter.add('c')
> filter.test('y')

true

> format.toBin(filter)

'11111110101111111010111111101011'
```

Sweepen/Entleeren von privaten Schlüsseln

> Electrum ermöglicht aus Sicherheitsgründen, Guthaben von (externen) privaten Schlüsseln auf die Wallet-Schlüssel zu überweisen, anstatt den externen Schlüssel zu importieren.
>
> Dies sollte anstelle eines Imports eines privaten Schlüssels genutzt werden, da der externe private Schlüssel aus dem Seed nicht wiederhergestellt werden kann.
>
> Dieser Vorgang wird bei Electrum "Entleeren" (Sweepen / Reinigen) genannt, da das Guthaben auf dem zu integrierenden privaten Schlüssels vollständig abgehoben wird.

1. Klicken Sie in der oberen Menüleiste auf **Wallet**.



2. Wählen Sie **Private Schlüssel Entleeren** (Sweep)
3. Geben Sie den **privaten Schlüssel der zweiten Adresse (vorletzte Spalte Key B)** aus der Liste an.

4. Klicken Sie auf **Entleeren** (Sweep).

Da dies eine Transaktion im Bitcoinnetzwerk ist, kann dies eine Weile dauern.

Bearbeiten Sie in Zwischenzeit die folgenden Aufgaben für den Blockexplorer.

https://www.blockchain.com/explorer

Aufgaben für den Blockexplorer

1. Wann wurde der erste Block gemined ?
2. Wie hoch war der Blockreward (Blockbelohnung) des ersten Blocks ?
3. Wie lautet die Nonce des ersten Blocks ?
4. Wie viele Transaktionen sind im 1. Block enthalten ?
5. Wie lautet der Hash der Coinbasetransaktion des 1. Blocks ?
6. Wie lautet das MerkleRoot des 1. Blocks ?
7. Wie lautet der Coinbaseparameter (Coinbase) der Coinbasetransaktion des 1. Blocks ?
8. Wie hoch waren die Gebühren der folgenden Transaktion ?

```
ee29b55cd3b1bbb26e9a298d96859a84eec6b9f189a687604f2f3c44acd85df5
```

9. Wie viele Transaktionen enthielt der Block 401661 ?
10. Wie hoch war die Summe der Ausgänge in diesem Block insgesamt ?
11. Wie hoch war das **Wechselgeld** der Adresse **1JWnmQW9k9TnGakPR2oGeu1mjdFGQDjrJt** in der folgenden Transaktion ?

```
05dcac053f396dd7d30992c465608aad3de79a03c8ee77ac9752dabfe0b94904
```

12. Was fällt ihnen an Block 409008 auf ?

```
cc455ae816e6cdafdb58d54e35d4f46d860047458eacf1c7405dc634631c570d
```

Exportieren eines privaten Schlüssels

> Private Schlüssel zu exportieren stellt ein Sicherheitsrisiko dar. Stellen Sie sicher, dass Ihre privaten Schlüssel geheim bleiben.

Trotzdem ist es manchmal notwendig bzw. gewollt, seine privaten Schlüssel an anderer Stelle zu speichern.

1. Klicken Sie in der oberen Menüleiste auf **Wallet**.
2. Wählen Sie **Private Schlüssel Exportieren**.
3. Geben Sie Ihr **Wallet-Passwort** ein.
4. Es werden Ihre privaten Schlüssel mit Adresse angezeigt.
5. Wählen Sie das Dateiformat **csv** (ist bereits voreingestellt).
6. Klicken Sie auf **Exportieren** und wählen Sie den Speicherpfad.

Jetzt wollen wir uns die exportierten Schlüssel anschauen.

1. Navigieren Sie zu dem Pfad, unter dem die privaten Schlüssel gespeichert wurden.
2. Öffnen Sie die csv-Datei mit Libre Office Calc oder einem anderen Editor.

Schließen Sie die Anwendung wieder und löschen Sie die Datei.

Rückgabe der BTC an das BCCM

Wenn Sie möchten, können Sie die BTC wieder an das BCCM zurücksenden.

Nutzen Sie dafür folgende Adresse: 15zYvVAEv6eHpieTcrQFF5erBXjF8PXJBD