Blockchain Engineering

Brieftaschen (Wallets)

Dr. Lars Brünjes

GEFÖRDERT VOM
Bundesministerium
für Bildung
und Forschung

Ostbayerische Technische Hochschule
Amberg-Weiden

OTH
REGENSBURG

MIND
MODULARES INNOVATIVES
NETZWERK FÜR DURCHLÄSSIGKEIT

AUFSTIEG DURCH
BILDUNG
OFFENE HOCHSCHULEN

28. September 2019

- Eine Wallet (elektronische Brieftasche) ist die Benutzer-Schnittstelle zur Kryptowährungs-Blockchain.
- Sie ermöglicht dem Benutzer, seine(n) Schlüssel zu verwalten, Zahlungen zu tätigen und seinen Kontostand einzusehen.
- Es gibt Desktop-, Browser-basierte und mobile Versionen.
- Die Wallet erzeugt Transaktionen und hält in der Blockchain Ausschau nach relevanten Transaktionen (eingehenden Zahlungen).

- Das scheinbar so einfache Konzept des “Kontostands” ist bei näherem Hinsehen komplexer als erwartet.
- Wir müssen zwischen current (aktuellem), available (verfügbarem), total (Gesamt-), und minimal (Mindest-) balance (Kontostand) unterscheiden.
- Diese Konzepte machen sowohl für das Konto-Modell als auch für das UTxO-Modell Sinn, aber wir werden uns im Folgenden auf letzteres beschränken.

Kontostand (Fortsetzung)

- Die Current Balance ist die Gesamtsumme der Beträge aller UTxOs, die dem Besitzer der Wallet gehören.

Kontostand (Fortsetzung)

- Die Current Balance ist die Gesamtsumme der Beträge aller UTxOs, die dem Besitzer der Wallet gehören.
- Die Available Balance ist die Current Balance *abzüglich der Inputs aus pending Transaktionen* (also solchen, die abgeschickt wurden, aber noch nicht Teil der Blockchain sind).

- Die Current Balance ist die Gesamtsumme der Beträge aller UTxOs, die dem Besitzer der Wallet gehören.
- Die Available Balance ist die Current Balance *abzüglich der Inputs aus pending Transaktionen* (also solchen, die abgeschickt wurden, aber noch nicht Teil der Blockchain sind).
- Die Total Balance ist der Kontostand unter der Annahme, dass alle pending Transaktionen Teil der Blockchain werden.

Kontostand (Fortsetzung)

- Die Current Balance ist die Gesamtsumme der Beträge aller UTxOs, die dem Besitzer der Wallet gehören.
- Die Available Balance ist die Current Balance *abzüglich der Inputs aus pending Transaktionen* (also solchen, die abgeschickt wurden, aber noch nicht Teil der Blockchain sind).
- Die Total Balance ist der Kontostand unter der Annahme, dass alle pending Transaktionen Teil der Blockchain werden.
- Die Minimal Balance ist der Mindest-Kontostand, wenn Forks in Betracht gezogen werden.

Kontostand (Beispiel)

- Betrachten wir wieder das einfache Beispiel, in dem Alice ihren einzigen UTxO von 100 ฿ als Input benutzt, um Bob (der am Anfang 50 ฿ hat) 40 ฿ und sich selbst 60 ฿ Wechselgeld zu schicken.
- Wir nehmen an, dass diese Transaktion abgeschickt, aber noch nicht Teil der Blockchain ist.

- Betrachten wir wieder das einfache Beispiel, in dem Alice ihren einzigen UTxO von 100 ฿ als Input benutzt, um Bob (der am Anfang 50 ฿ hat) 40 ฿ und sich selbst 60 ฿ Wechselgeld zu schicken.
- Wir nehmen an, dass diese Transaktion abgeschickt, aber noch nicht Teil der Blockchain ist.
- Die Current Balance ist 100 ฿.

Kontostand (Beispiel)

- Betrachten wir wieder das einfache Beispiel, in dem Alice ihren einzigen UTxO von 100 ฿ als Input benutzt, um Bob (der am Anfang 50 ฿ hat) 40 ฿ und sich selbst 60 ฿ Wechselgeld zu schicken.
- Wir nehmen an, dass diese Transaktion abgeschickt, aber noch nicht Teil der Blockchain ist.
- Die Current Balance ist 100 ฿.
- Die Available Balance ist 0 ฿.

Kontostand (Beispiel)

- Betrachten wir wieder das einfache Beispiel, in dem Alice ihren einzigen UTxO von 100 ฿ als Input benutzt, um Bob (der am Anfang 50 ฿ hat) 40 ฿ und sich selbst 60 ฿ Wechselgeld zu schicken.
- Wir nehmen an, dass diese Transaktion abgeschickt, aber noch nicht Teil der Blockchain ist.
- Die Current Balance ist 100 ฿.
- Die Available Balance ist 0 ฿.
- Die Total Balance ist 60 ฿.

- Betrachten wir wieder das einfache Beispiel, in dem Alice ihren einzigen UTxO von 100 ฿ als Input benutzt, um Bob (der am Anfang 50 ฿ hat) 40 ฿ und sich selbst 60 ฿ Wechselgeld zu schicken.
- Wir nehmen an, dass diese Transaktion abgeschickt, aber noch nicht Teil der Blockchain ist.
- Die Current Balance ist 100 ฿.
- Die Available Balance ist 0 ฿.
- Die Total Balance ist 60 ฿.
-

	Alice	Bob
current	100	50
available	0	50
total	60	90

- Nehmen wir an, dass Alice, Bob und Charlie Amerikaner sind und dass transatlantische Kabel zeitweilig unterbrochen waren.
- Alice hat in einer Transaktion von Bob 50 ฿ bekommen, dann Charlie in einer zweiten Transaktion diese 50 ฿ und weitere 10 ฿ geschickt.

- Nehmen wir an, dass Alice, Bob und Charlie Amerikaner sind und dass transatlantische Kabel zeitweilig unterbrochen waren.
- Alice hat in einer Transaktion von Bob 50 ฿ bekommen, dann Charlie in einer zweiten Transaktion diese 50 ฿ und weitere 10 ฿ geschickt.
- Nach Reparatur des Kabels gibt es drei Fälle:

- Nehmen wir an, dass Alice, Bob und Charlie Amerikaner sind und dass transatlantische Kabel zeitweilig unterbrochen waren.
- Alice hat in einer Transaktion von Bob 50 ฿ bekommen, dann Charlie in einer zweiten Transaktion diese 50 ฿ und weitere 10 ฿ geschickt.
- Nach Reparatur des Kabels gibt es drei Fälle:
  - Keine der beiden Transaktionen ist Teil der wiederhergestellten Blockchain.

Minimal Balance (Beispiel)

- Nehmen wir an, dass Alice, Bob und Charlie Amerikaner sind und dass transatlantische Kabel zeitweilig unterbrochen waren.
- Alice hat in einer Transaktion von Bob 50 ฿ bekommen, dann Charlie in einer zweiten Transaktion diese 50 ฿ und weitere 10 ฿ geschickt.
- Nach Reparatur des Kabels gibt es drei Fälle:
  - Keine der beiden Transaktionen ist Teil der wiederhergestellten Blockchain.
  - Die erste Transaktion wird Teil der Blockchain, aber nicht die zweite.

Minimal Balance (Beispiel)

- Nehmen wir an, dass Alice, Bob und Charlie Amerikaner sind und dass transatlantische Kabel zeitweilig unterbrochen waren.
- Alice hat in einer Transaktion von Bob 50 ฿ bekommen, dann Charlie in einer zweiten Transaktion diese 50 ฿ und weitere 10 ฿ geschickt.
- Nach Reparatur des Kabels gibt es drei Fälle:
  - Keine der beiden Transaktionen ist Teil der wiederhergestellten Blockchain.
  - Die erste Transaktion wird Teil der Blockchain, aber nicht die zweite.
  - Beide Transaktionen werden Teil der Blockchain.

Minimal Balance (Beispiel)

- Nehmen wir an, dass Alice, Bob und Charlie Amerikaner sind und dass transatlantische Kabel zeitweilig unterbrochen waren.
- Alice hat in einer Transaktion von Bob 50 ฿ bekommen, dann Charlie in einer zweiten Transaktion diese 50 ฿ und weitere 10 ฿ geschickt.
- Nach Reparatur des Kabels gibt es drei Fälle:
  - Keine der beiden Transaktionen ist Teil der wiederhergestellten Blockchain.
  - Die erste Transaktion wird Teil der Blockchain, aber nicht die zweite.
  - Beide Transaktionen werden Teil der Blockchain.
  - Es kann nicht sein, dass nur die zweite Transaktion Teil der Blockchain wird, denn sie hat als Input einen Output der ersten Transaktion!

- Nehmen wir an, dass Alice, Bob und Charlie Amerikaner sind und dass transatlantische Kabel zeitweilig unterbrochen waren.
- Alice hat in einer Transaktion von Bob 50 ฿ bekommen, dann Charlie in einer zweiten Transaktion diese 50 ฿ und weitere 10 ฿ geschickt.
- Nach Reparatur des Kabels gibt es drei Fälle:
  - Keine der beiden Transaktionen ist Teil der wiederhergestellten Blockchain.
  - Die erste Transaktion wird Teil der Blockchain, aber nicht die zweite.
  - Beide Transaktionen werden Teil der Blockchain.
  - Es kann nicht sein, dass nur die zweite Transaktion Teil der Blockchain wird, denn sie hat als Input einen Output der ersten Transaktion!
- Alices Minimal Balance ist 90 ฿ (dritter Fall).
- Charlies Minimal Balance ist 0 ฿ (erster oder zweiter Fall).

- UTxOs mit geringem Wert, die sich im Laufe der Zeit “in der Brieftasche” ansammeln können (d.h. solche, die einem bestimmten Benutzer gehören), werden als Staub (dust) bezeichnet.
- Staub ist analog zu einem Haufen Kleingeld, der sich im Portemonnaie ansammelt.
- Staub ist problematisch, weil er Ressourcen (Speicherplatz, Rechenzeit) verschlingt, der seinem Wert nicht gerecht wird.
- Das ist insbesondere ein Problem für “Power User” wie Exchanges, die Tausende von Transaktionen täglich tätigen. Bei Ihnen kann die Zahl der Staub-UTxOs in die Millionen gehen.

- Die Vermeidung von Staub ist Hauptziel der Coin Selection (Münzauswahl) der Wallet.
- Coin Selection ist der Algorithmus, der geeignete UTxOs als Inputs für eine Transaktion auswählt.

- Die Vermeidung von Staub ist Hauptziel der Coin Selection (Münzauswahl) der Wallet.
- Coin Selection ist der Algorithmus, der geeignete UTxOs als Inputs für eine Transaktion auswählt.

Bezahlen im Supermarkt

Stellen Sie sich vor, Sie stehen an der Supermarktkasse und müssen einen bestimmten Betrag bezahlen. Wie vermeiden Sie Kleingeld in Ihrer Brieftasche?

- Bezahlen Sie mit einem großen Schein?
- Versuchen Sie, passend zu bezahlen, wobei Sie riskieren, *zu wenig* Kleingeld für Ihre nächste Bezahlung übrig zu behalten?

Ein naheliegender Algorithmus zur Coin Selection ist Largest-First (größte zuerst): Beginnend mit der wertvollsten Münze nehmen Sie so viele Münzen (sortiert absteigend nach Wert), bis Sie den gewünschten Betrag erreicht haben.

Beispiel

Sagen wir, die Münzen in Ihrer Wallet sind 10 ฿, 6 ฿, 5 ฿ und 2 ฿ und dass Sie 12 ฿ bezahlen müssen.
Sie nehmen die beiden größten Münzen als Input und erzeugen eine Transaktion mit 4 ฿ Wechselgeld.

Ein naheliegender Algorithmus zur Coin Selection ist Largest-First (größte zuerst): Beginnend mit der wertvollsten Münze nehmen Sie so viele Münzen (sortiert absteigend nach Wert), bis Sie den gewünschten Betrag erreicht haben.

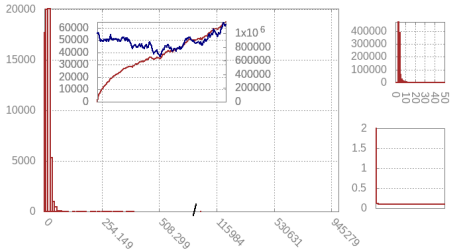

Abb.: Simulation des Largest-First Algorithmus zeigt, dass sich im Laufe der Zeit mehr und mehr Staub in der Wallet ansammelt.

- Trotz der praktischen Bedeutung des Staub-Problems gibt es erschreckend wenige wissenschaftliche Arbeiten zu diesem Thema.
- Eine Ausnahme ist Mark Erhardts Masterarbeit An Evaluation of Coin Selection Strategies.
- Erhardt schlägt vor, zufällige (random) Münzen solange auszuwählen, bis der gewünschte Betrag erreicht ist.
- Erhardts Vorschlag beruht auf seiner Beobachtung, dass, wenn 90% der Münzen Staub sind, zufällige Wahl einer Münze mit 90% Wahrscheinlichkeit eine Münze wählen wird, die Staub ist.

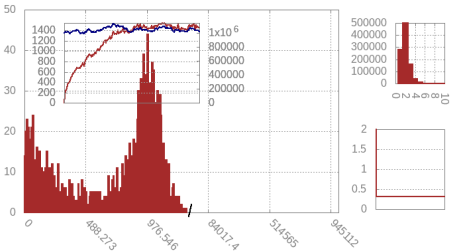

Abb.: Die Simulation von Erhardts Algorithmus zeigt eine dramatische Verbesserung: Wesentlich weniger Staub wird erzeugt. Allerdings ist immer noch mehr Staub vorhanden, als uns lieb ist.

- Edsko de Vries hat Erhardts Algorithmus für Cardano weiterentwickelt und verbessert.
- Die Idee ist, bei der zufälligen Auswahl nicht aufzuhören, wenn der zu zahlende Betrag erreicht ist, sondern zu versuchen, ein gleich-hohes Wechselgeld zu erzielen.
- Dies beruht auf der Idee, dass Wechselgeld in Höhe des Betrages für folgende, ähnliche Transaktionen nützlich sein wird.
- Im Detail muss man aufpassen, rechtzeitig aufzuhören, bevor das Wechselgeld zu hoch wird.

Der Random-Improve Algorithmus:

1. Wähle solange zufällige Münzen aus, bis der zu zahlende Betrag erreicht ist.
2. Wähle eine zufällige Münze aus. Wenn sie eine *Verbesserung* ist, füge sie der Auswahl hinzu und wiederhole diesen Schritt, wenn nicht, höre auf.
3. Eine Münze gilt als Verbesserung, wenn die folgenden Bedingungen erfüllt sind:
   - Die Gesamtsumme aller Münzen ist näher am Doppelten des zu zahlenden Betrages als zuvor.
   - Die Gesamtzahl an Münzen (z,B. zehn) wird nicht überschritten.

Der Random-Improve Algorithmus:

1. Wähle solange zufällige Münzen aus, bis der zu zahlende Betrag erreicht ist.
2. Wähle eine zufällige Münze aus. Wenn sie eine *Verbesserung* ist, füge sie der Auswahl hinzu und wiederhole diesen Schritt, wenn nicht, höre auf.
3. Eine Münze gilt als Verbesserung, wenn die folgenden Bedingungen erfüllt sind:
   - Die Gesamtsumme aller Münzen ist näher am Doppelten des zu zahlenden Betrages als zuvor.
   - Die Gesamtzahl an Münzen (z,B. zehn) wird nicht überschritten.

Bemerkung

Falls die Gesamtzahl der Inputs einer Transaktion beschränkt ist, kann man sowohl den Random- als auch den Random-Improve-Algorithmus dahingehend ändern, dass man bei Überschreiten der Grenze auf Largest-First zurück fällt.

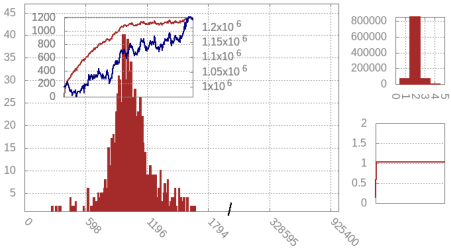

Abb.: Die Simulation des Random-Improve-Algorithmus zeigt, dass jetzt kaum noch Staub erzeugt wird und dass die Verteilung der Münzen in der Wallet sehr gut der Größe der zu erwartenden Zahlungen entspricht.

- Nehmen wir an, der zu zahlende Betrag sei 10 ฿.

- Nehmen wir an, der zu zahlende Betrag sei 10 ฿.
- Wir wählen eine zufällige 7 ฿-Münze. Wir machen weiter, denn 7<10.

- Nehmen wir an, der zu zahlende Betrag sei 10 ฿.
- Wir wählen eine zufällige 7 ฿-Münze. Wir machen weiter, denn 7<10.
- Wir wählen eine zufällige 2 ฿-Münze. Wir machen weiter, denn 7+2=9<10.

- Nehmen wir an, der zu zahlende Betrag sei 10 ฿.
- Wir wählen eine zufällige 7 ฿-Münze. Wir machen weiter, denn 7<10.
- Wir wählen eine zufällige 2 ฿-Münze. Wir machen weiter, denn 7+2=9<10.
- Wir wählen eine zufällige 5 ฿-Münze. Wir beenden den ersten Schritt, denn 7+2+5=14>10.

- Nehmen wir an, der zu zahlende Betrag sei 10 ₿.
- Wir wählen eine zufällige 7 ₿-Münze. Wir machen weiter, denn 7<10.
- Wir wählen eine zufällige 2 ₿-Münze. Wir machen weiter, denn 7+2=9<10.
- Wir wählen eine zufällige 5 ₿-Münze. Wir beenden den ersten Schritt, denn 7+2+5=14>10.
- Wir wählen eine zufällige 3 ₿-Münze. Wir machen weiter, denn 14+3=17 ist näher an 20 als 14.

- Nehmen wir an, der zu zahlende Betrag sei 10 ฿.
- Wir wählen eine zufällige 7 ฿-Münze. Wir machen weiter, denn 7<10.
- Wir wählen eine zufällige 2 ฿-Münze. Wir machen weiter, denn 7+2=9<10.
- Wir wählen eine zufällige 5 ฿-Münze. Wir beenden den ersten Schritt, denn 7+2+5=14>10.
- Wir wählen eine zufällige 3 ฿-Münze. Wir machen weiter, denn 14+3=17 ist näher an 20 als 14.
- Wir wählen eine zufällige 4 ฿-Münze. Wir machen weiter, denn 17+4=21 ist näher an 20 als 17.

Random-Improve Beispiel

- Nehmen wir an, der zu zahlende Betrag sei 10 ฿.
- Wir wählen eine zufällige 7 ฿-Münze. Wir machen weiter, denn 7<10.
- Wir wählen eine zufällige 2 ฿-Münze. Wir machen weiter, denn 7+2=9<10.
- Wir wählen eine zufällige 5 ฿-Münze. Wir beenden den ersten Schritt, denn 7+2+5=14>10.
- Wir wählen eine zufällige 3 ฿-Münze. Wir machen weiter, denn 14+3=17 ist näher an 20 als 14.
- Wir wählen eine zufällige 4 ฿-Münze. Wir machen weiter, denn 17+4=21 ist näher an 20 als 17.
- Wir wählen eine zufällige 1 ฿-Münze. Wir hören auf, denn 21+1=22 ist weiter von 20 als 21.

- Nehmen wir an, der zu zahlende Betrag sei 10 ฿.
- Wir wählen eine zufällige 7 ฿-Münze. Wir machen weiter, denn 7<10.
- Wir wählen eine zufällige 2 ฿-Münze. Wir machen weiter, denn 7+2=9<10.
- Wir wählen eine zufällige 5 ฿-Münze. Wir beenden den ersten Schritt, denn 7+2+5=14>10.
- Wir wählen eine zufällige 3 ฿-Münze. Wir machen weiter, denn 14+3=17 ist näher an 20 als 14.
- Wir wählen eine zufällige 4 ฿-Münze. Wir machen weiter, denn 17+4=21 ist näher an 20 als 17.
- Wir wählen eine zufällige 1 ฿-Münze. Wir hören auf, denn 21+1=22 ist weiter von 20 als 21.
- Ergebnis des Algorithmus ist also, dass wir Inputs mit 7 ฿, 2 ฿, 5 ฿, 3 ฿ und 4 ฿ verwenden und 11 ฿ Wechselgeld bekommen.

- Bisher haben wir immer von *dem* öffentlichen Schlüssel des Benutzers (bzw. dessen Hash) gesprochen.
- Zur Wahrung der Privatsphäre ist es besser, möglichst für jeden Zahlungseingang *eine neue Adresse* zu benutzen.
- Bitcoin ermöglicht dies (und andere, fortgeschrittene Szenarien) mittels BIP-0032 (HD Wallets).
- Der Standard wird auch in vielen anderen Kryptowährungen verwendet.

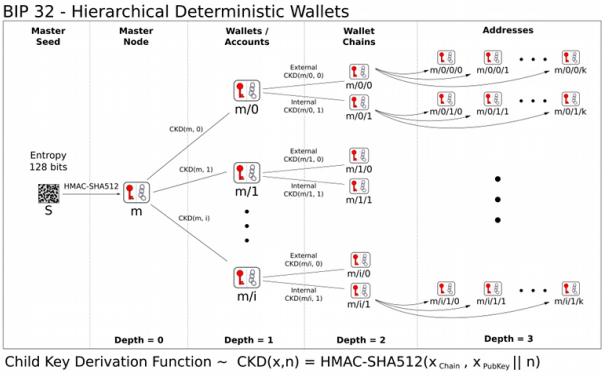

Abb.: Hierarchische Schlüssel

Eine Hierarchical Deterministic Wallet (oder HD Wallet) erlaubt die Generierung (fast) beliebig vieler Schlüssel ausgehend von einem einzigen Master Seed.

Hinweis

Diese Publikation wurde im Rahmen des vom Bundesministerium für Bildung und Forschung (BMBF) geförderten Bund- Länder- Wettbewerbs “Aufstieg durch Bildung: offene Hochschulen” erstellt. Die in dieser Publikation dargelegten Ergebnisse und Interpretationen liegen in der alleinigen Verantwortung der Autor/innen.