RIM Crypto

Offizielles Technical Data Sheet

V1.0

05. Juni 2018

www.rimcrypto.eu

info@rimcrypto.eu

Inhaltsverzeichnis

1. Blockchains

Blockchains als Zahlungssysteme sind prinzipiell verteilte Datenbanken, die das double spending problem lösen. Das double spending problem ist im Grunde ein Sonderfall des Problems der byzantinischen Generäle, welches bei Bitcoin mit Proof-of-Work gelöst wird. Das bedeutet, dass solche Probleme bei dem Bitcoin sehr unwahrscheinlich sind, da die jeweiligen Blöcke schwer zu berechnen sind und somit Gleichzeitigkeit in Transaktionen verhindert wird. Die Implementierung einer Blockchain erfordert asymmetrische kryptographische Algorithmen (Public-Key Kryptosystem) und kryptographische Einweg-Hash-Funktionen. Erstere werden als Schlüssel verwendet, um auf Ihr Konto zuzugreifen und Sie als Eigentümer zu verifizieren, letztere werden verwendet, um Zahlungen im Zusammenhang mit dem vorherigen Kontostand zu verifizieren. Zahlungen werden durch die Blockchain mit digitalen Signaturen und Hash-Summen verifiziert.

2. Blockchain - Die Umsetzung

Wir werden eine Blockchain verwenden, welche die distributed Ledger Technologie anstelle von UTXOs und verteilte Vereinbarungen anstelle von Proof-of-Work verwendet, um das double spending Problem zu lösen. Wir richten einen Master-Server ein, der die Berechnungen des Netzwerks delegiert. Verschiedene andere Server (bzw. Miner) können mit dem Master-Server interagieren, um dessen Berechnungen für die Blockchain zu unterstützen. Das bedeutet, dass der Master-Server nach einer Anfangsphase nicht nur die für die Transaktionen notwendigen Berechnungen durchführt, sondern auch die Berechnungen anderer Server verifiziert und so zu den anderen Servern im Netzwerk homogen wird. Jeder Server im Blockchain-Netzwerk hat eine eigene Signatur und einen über das Netzwerk propagierten Vertrauenswert, d.h. wenn Berechnungen eines Servers korrekt sind, wird sein Vertrauenswert erhöht, wenn die Berechnungen falsch sind, wird sein Vertrauenswert reduziert. Das Vertrauen in Blockchain-Transaktionen wird durch das Vertrauen der signierenden Server erhöht. Für die Überprüfung einer Transaktion ist ein bestimmter Vertrauenswert erforderlich. Mehr als die Hälfte des summierten Vertrauenswerts aller reaktionsschnellen Server wird für die Überprüfung einer Transaktion benötigt. Dies bedeutet, dass der Vertrauenswert, der für eine zu verifizierende Transaktion benötigt wird, mit dem

Wachstum des Netzwerks steigt. In der Anfangsphase einer Transaktion ist es also ausreichend, dass der Master-Server sie signiert hat, da andere Server am Anfang nicht vertrauenswürdig sind. Später wird der Vertrauenswert, der zum Signieren einer Transaktion benötigt wird, viel größer sein als der maximale Vertrauenswert einzelner Server, was bedeutet, dass mehrere hochgradig vertrauenswürdige Server eine Transaktion signieren müssen, um sie verifizieren zu lassen. Eine Transaktion kann eine Geldüberweisung mit einer solchen Struktur sein:

```
typedef char key;
typedef char sig;

struct _transfer{
 key sender[KEY_LENGTH];     //senders public key
 key receiver[KEY_LENGTH];   //receivers public key
 unsigned long long amount; //amount to be tranfered
 sig signature[SIG_LENGTH]; //signature to verify the transaction
};
```

Um eine Transaktion zu signieren, muss der Server prüfen, ob die Transaktion gültig ist, hier ist ein Pseudo-C-Code für eine solche Gültigkeitsprüfung:

```
int verify_transfer(struct _transfer trans){
 sig* sig_hash=hash(trans);
 sig* sig_decrypt=decrypt(trans.signature,trans.sender);
 if(memcmp(sig_hash,sig_decrypt,SIG_LENGTH)!=0){
  printf("Invalid transfer received: Wrong signature!\n");
  return 1;
 }
 P(change_blockchain); //semaphore for preventing double spending of coins
 if(get_amount(trans.sender)<trans.amount){
  printf("Invalid transfer received: Not enough coins to transfer!\n");
  return 1;
 }
 apply(trans); //modify balance of the accounts on the local blockchain
 sign(trans); //sign the transaction
 V(change_blockchain); //free the semaphore
 return 0;
}
```

Außerdem gibt es ein Zeitlimit für den Rückruf von Transaktionen, so dass neben der Überprüfung der Transaktion durch das signierte Vertrauen auch eine bestimmte Zeit gewartet werden sollte, um sicherzustellen, dass die Transaktion wirklich gültig ist. Ripple verwendet eine Verifikationszeit von 10 Sekunden, Stellar eine Verifikationszeit von 5 Sekunden. Wir streben ähnliche Überprüfungsfristen an

Verhindern des double spending Problems

Angenommen, ein Konto hat 3 Coins und versucht, zwei Transaktionen durchzuführen, welche jeweils 2 Münzen ausgeben, also insgesamt 4 Coins. Wenn ein einzelner Server diese Transaktionen erhält, wird nur die erste Transaktion signiert, da nach der ersten Transaktion nicht mehr genügend Coins übrig sind, um den Betrag der zweiten Transaktion zu decken. Wenn nun beide Transaktionen an verschiedene Server gesendet würden, würden beide Transaktionen von zwei verschiedenen Servern signiert und dann über das Netzwerk verbreitet. Hierbei würden sie um den benötigten Vertrauenswert konkurrieren. Mehr als die Hälfte des summierten Vertrauenswerts aller reaktionsschnellen Server wird für die Überprüfung einer Transaktion benötigt. Das bedeutet, dass immer noch nur eine Transaktion verifiziert werden kann und die andere lokal auf den Servern, die sie signiert haben, rückgängig gemacht werden muss. Wir sehen also, dass es nicht möglich ist, Coins mit diesem Ansatz der Überprüfung von Transaktionen zu verdoppeln. Angenommen, wir haben bösartige Server, dann wird mehr als die Hälfte des Vertrauenswerts der verfügbaren reaktionsschnellen Server benötigt, um eine bösartige Transaktion zu verifizieren, was bedeutet, dass ein bösartiger Angreifer viele Server über eine lange Zeit laufen lassen müsste, um genügend Vertrauen zu gewinnen damit ein Angriff durchgeführt werden kann. Aber wenn das passiert, können andere vertrauenswürdige Server, wie zum Beispiel der Master-Server einen Widerruf der Transaktion auslösen, was bedeutet, dass das Vertrauen der bösartigen Transaktionen verringert wird und die für ihre Überprüfung erforderliche Zeitspanne verlängert wird. Der Vertrauenswert, der für eine zu widerrufende Transaktion benötigt wird, ist daher viel kleiner als der Vertrauenswert, der für die Überprüfung von Transaktionen benötigt wird. Aber wenn ein vertrauenswürdiger Server eine Transaktion ohne triftigen Grund widerruft, wird sein eigener Vertrauenswert unter den Wert gesenkt, der für zukünftige Widerrufe erforderlich ist. Eine Transaktion, die ohne triftigen Grund widerrufen wird (wie z.B. eine gleichzeitige Transaktion oder eine widerrufende Unterschrift des Absenders der Transaktion), wird trotzdem nach einer erhöhten Zeitspanne verifiziert.

Schlüssellänge

Aktuelle Bitcoin Miner haben eine maximale Rate von einigen GH/s, was $k \cdot 10^9$ Hashes pro Sekunde bedeutet. Nehmen wir nun einen Worst Case von $k \text{ of } 10^3$ an und nehmen wir auch an, dass wir eine Schlüssellänge von 512 Bit haben, dann hat das Finden eines bestimmten Schlüssels mit der gegebenen Hashrate innerhalb von tausend Jahren eine Wahrscheinlichkeit von $\frac{10^{15} \cdot 60^2 \cdot 24 \cdot 365.25}{2^{512}} \approx 2.353673334589236 \cdot 10^{-132}$.

Dies bedeutet, dass es ziemlich unmöglich ist, einen bestimmten Schlüssel zu finden und, dass die Schlüssellänge von 512 mehr als genug ist, um ein Wallet zu sichern. Gehen wir weiter und untersuchen die Möglichkeit eines Geburtstagsangriffs. Das heißt, dass man das Geburtstagsparadoxon als möglichen Angriffsvektor ausnutzt. Dies bedeutet nichts anderes als, dass wir nicht versuchen, einen bestimmten Schlüssel zu erraten, sondern nach einem zufällig vorhandenen Schlüssel suchen. Daraus folgt, dass bei Verwendung aller 2^{512} möglichen Schlüssel die Wahrscheinlichkeit, einen zufälligen Schlüssel zu finden, 100% beträgt, obwohl die Wahrscheinlichkeit, einen bestimmten Schlüssel zu finden, immer noch so gering ist wie oben berechnet. Also steigt die tatsächliche Wahrscheinlichkeit, einen zufälligen Schlüssel zu finden, um die Anzahl der verwendeten Schlüssel. Angenommen, wir haben 10 Milliarden (10^{10}) Menschen und jeder verwendet 10^{20} Schlüssel, d.h. wir haben insgesamt 10^{30} Schlüssel, dann ist die Wahrscheinlichkeit für eine Übereinstimmung $1 - \frac{2^{512}!}{(2^{512}-10^{30})! \cdot (2^{512})(10^{30})}$. Dies ist ziemlich unmöglich zu berechnen, aber eine Annäherung gibt uns eine Wahrscheinlichkeit 10^{-94} von . Selbst wenn wir $1.6 \cdot 10^{68}$ Hashes hätten, ist die Wahrscheinlichkeit einer Übereinstimmung nur ungefähr 10^{-18}. .

Das bedeutet, dass es selbst mit dem Geburtstagsparadoxon unmöglich ist, einen Schlüssel zu finden. Um die Sicherheit weiter zu erhöhen, werden wir mindestens 1024 Bit Schlüssellänge verwenden.

Wechsel auf die RIM-Blockchain

Der Übergang von einem WAVES-Tokens zu einer unabhängigen RIM-Blockchain kann mit einem einfachen Smart-Contract erfolgen, bei dem WAVES-Tokens, die an eine von uns definierte Adresse übertragen werden, in der WAVES-Blockchain geburned und durch einen festen Wechselkurs auf eine neue Public-Key-Adresse in der neuen RIM-Blockchain gutgeschrieben werden.

3. Die RIM-Karte

Die RIM-Karte funktioniert ähnlich wie eine normale Debitkarte. Der einzige Unterschied besteht darin, wie die Zahlungen im Bank-Backend erfolgen. RIM-Karten verwenden eine individuelle Blockchain-Adresse. Bei Zahlungen berechnet die Bank, dass genügend **RIM Token** auf dieser Blockchainadresse vorhanden sind, wobei die untere Korridorwand als Wechselkurs verwendet wird. Die Zahlung erfolgt dann über eine bestimmte RIM-Blockchain-Adresse, die mit einem echten Bankkonto von **RIM Crypto** verbunden ist, um die gewünschte Währung zu überweisen. Das bedeutet, dass wir im Falle einer Zahlung mit einer RIM-Karte einfach ihre Token zurückkaufen und das Geld überweisen. Wir werden dann versuchen, die erhaltenen Token zu einem höheren Preis zu verkaufen, um ihnen zusätzliche Coins auf ihre RIM-Karte zurück zu transferieren. Dies geschieht natürlich mit Abzug einer kleinen Überweisungsgebühr. Bei abnormalem Marktverhalten (aktueller Token-Preis, der den unteren Korridor verlässt) werden Zahlungen mit RIM-Paycard deaktiviert. Kunden können auch eine zusätzliche obere Korridorgrenze festlegen, die Zahlungen mit der RIM-Karte bei Überschreitung deaktiviert. Sobald der Wechselkurs wieder innerhalb der Korridore liegt, werden die Zahlungen mit der RIM-Karte wieder aktiviert.

4. Der Stabilisierungsmechanismus

Einführung

Das Ziel dieses Kapitels ist es, einige Ansätze für das machine learing Konzept im Rahmen des Stabilisierungsmechanismus aufzuzeigen, welche die **RIM Token** Wechselkurse in einem kontrollierten Preiskorridor halten. Einer dieser Ansätze ist der Einsatz künstlicher neuronaler Netze. Deep Learning ist eine revolutionäre maschinelle Lerntechnik, die auf einer Vielzahl an Schichten von künstlichen neuronalen Netzen (ANN) basiert. Deep learning verbesserte die Mustererkennung im Vergleich zu früheren Ansätzen. ANNs bestehen aus künstlichen Neuronen, wobei jedes Neuron im Wesentlichen ein lineares Punktprodukt über einem Eingangsvektor und einem Gewichtsvektor ist, gefolgt von einer nichtlinearen Aktivierungsfunktion. Ein ANN wird durch Änderung der Gewichtsvektoren trainiert. Ein einfacher Ansatz für das Training eines Feed Forward ANN ist die Verwendung des Backpropagation-Algorithmus, der im Grunde ein stochastischer Gradientenabstieg über die Backpropagated Error-Funktion ist, wobei E der Fehler, η eine w_{ij} Lernrate und der Gewichtswert ist, der i Neuronen mit j Neuronen verbindet. Dies bedeutet, dass die w_{ij} Änderung durch berechnet $\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$. wird.

Unvollständige Daten

Eine wichtige Angelegenheit, um die wir uns kümmern müssen, ist das Fehlen von einigen Datensätzen. Insbesondere fehlen einige Werte, da nicht alle Krypto-Währungen über die gesamt aufgezeichnete Zeit existierten. Ein Beispiel ist der **RIM Token** selbst, daher werden die Daten für unseren eigenen Token kontinuierlich verbessert. Aber auch ältere Daten aus anderen Krypto-Währungen sind nützlich, um ein Modell zu erstellen. Es gibt verschiedene Möglichkeiten wie z.B. Imputation um mit fehlenden Daten umzugehen. Ein vielversprechender Ansatz ist in[Tre94] skizziert, wo mangelhafte Daten zur Verbesserung der Trainingsergebnisse in einem neuronalen Netzwerk verwendet werden. Ein naiver Umgang

mit fehlenden Werten ist es, diese mit Mittelwerten zu füllen. Eine genauere Möglichkeit, mit fehlenden Werten umzugehen, ist die Vorhersage, z.B. die Regression der fehlenden Werte selbst. Für den Fall, dass es keine Währung gab, ist es am besten, die Werte so zu wählen, dass sie keinen Einfluss auf konkurrierende Werte haben, d.h. für diese Trainingsschritte sollten die Ergebnisse dem Training ähnlich sein. Dies liegt daran, dass Daten, die vor einem ICO fehlen, systematisch anders sind als Daten, die zufällig fehlen, oder Daten, die aufgrund eines zugrunde liegenden Ereignisses fehlen, das unvorhersehbar ist, aber einen wirklichen Einfluss auf die Daten hat.

Der Kurskorridor

Wir wollen auf verschiedenen Korridoren trainieren. Dies kann entweder durch eine Vorhersage der tatsächlichen Wechselkurse und -volumina und der darauf basierenden Berechnung der Korridore erreicht werden oder auf eine diskrete Art und Weise. Ein Training für die Korridore ist in diesem Fall nur möglich, wenn diskrete Korridorstufen verwendet werden. Dann ist es sinnvoll, eine ordinale Klassifizierung für die Korridore an zu trainieren. Wenn kontinuierliche Korridore benötigt werden, ist es besser, eine Regression des Wechselkurses durchzuführen, was auch bei einer konformen Vorhersage passieren könnte. Dies bedeutet, dass wir Vertrauensintervalle bei einem spezifischen Wechselkurs erreichen würden. Konforme Vorhersagen hätte auch die Eigenschaft, dass sie in einer Online-Lernumgebung durchgeführt werden könnte. Daraus folgt, dass das Modell die tatsächlichen Daten ohne explizite Umschulung kontinuierlich verbessern könnte.

Worst case Simulation unter Verwendung ähnlicher Tokendaten

In der ersten Phase vor dem ICO führen wir einige Worst-Case-Simulationen durch, um die Machbarkeit unseres Ansatzes zu überprüfen. Für diese Simulation verwenden wir reale Daten von ähnlichen Token, um zu berechnen, wie lange unser System im schlimmsten Fall stützen

kann. Dies ist der schlimmste Fall, da in einem realen Szenario unsere oberen und unteren Walls den Wechselkurs direkt beeinflussen, was bei dieser ersten Simulation nicht der Fall ist. Diese Simulation ist in Python mit Hilfe von Pandas implementiert.

Interferenz des trainierten Modells bezüglich ähnlicher Tokendaten

In der zweiten Phase trainieren wir ein adaptives Modell des intrinsischen Wechselkursverhaltens, um ein realistischeres Verhalten zu modellieren. Bei diesem Modell handelt es sich bereits um ein mit TensorFlow trainiertes künstliches feed forward neuronales Netzwerk. Für die Phase vor dem ICO werden die gecrawlten Daten in Trainingsdaten und Testdaten aufgeteilt. Zusätzlich werden einige Monate vom Training ausgeschlossen. Das Modell wird auf die Trainingsdaten angelernt und die Simulation ist dann ein Rückschluss des trainierten Modells auf die Testdaten. Dies wird mehrere Verbesserungen gegenüber dem ursprünglichen neuronalen Netz ermöglichen, indem seine Struktur für bessere Vorhersagen geändert wird.

Interferenz bezüglich Hot Data

Nach dem ICO ist die Trennung in Trainingsdaten und Testdaten nicht mehr zwingend erforderlich, da die Struktur des neuronalen Netzes bereits für den jeweiligen Datentyp verifiziert ist. Ziel dieser Phase ist es, ein über den gesamten Datensatz trainiertes Modell zur Vorhersage des aktuellen **RIM Token** Wechselkurses zu verwenden. Natürlich werden die maschinellen Lernalgorithmen parallel zu dieser Anwendung noch verbessert.

Verbesserte Trainingsdaten

Für weitere Verbesserungen bezüglich der Genauigkeit unseres Modells werden wir später nicht nur die Wechselkurse, sondern auch die für RIM wichtigen Orderboks von Krypto-Währungen sammeln, sondern ebenso die Wechselkurse und Volumina, die die Anfragen und Gebote auch im JSON-Format aus den Web-APIs lesen und dann an Pandas weiterleiten. Später werden wir auch gängige Blogs und News-Einträge zu Krypto-Währungen überprüfen, um Textdaten zu sammeln, die in unser Trainingsmodell aufgenommen werden können, indem wir Worteinbettungen wie word2vec verwenden, bevor wir sie in ein neuronales Netzwerk einspeisen. Die Textdaten können entweder mit dem numerischen Modell durch Umschulung über den gesamten Datensatz oder durch Training einzelner neuronaler Netze und deren Stapelung kombiniert werden.

5. Referenzen und Ausblick

Referenzen

[Tre94] Neuneier Tresp Ahmad. Training Neural Networks with Deficient Data. 1994.

Ausblick

In den folgenden Versionen dieses Datenblattes werden wir zusätzliche Informationen über die Anwendung und einige technische Informationen zu unserer Full Node zur Verfügung stellen.
Zusätzlich werden wir weitere Informationen über den aktuellen Stand unseres Projektes veröffentlichen.

Impressum und Kontakt

RIM Crypto

Technologiepark 8

33100 Paderborn

www.rimcrypto.eu

info@rimcrypto.eu

Made in Germany

garage33

Freiraum für Gründer und Unternehmer

TEC UP

UNIVERSITÄT PADERBORN

IDnow

mtp

PADERBORN