Anforderungsspezifikation
Bitcoin Script Debugger

Projektteam x

Rolf Zurbrügg
Samuel Egger

X1.0 - 15.11.17

Berner Fachhochschule
Departement für Technik und Informatik

Inhaltsverzeichnis

1 Zweck des Dokuments

Dieses Dokument beschreibt die Ziele und Anforderungen für den Bitcoin Skript Debugger.

2 Projektvision

Bitcoins können gekauft und verkauft werden. Alle Transaktionen werden mit sogenannten Bitcoin Scripts validiert. Die Skript Sprache ist eine Bitcoin eigene Sprache. Sie ist nicht Turing mächtig und Stack basiert. Jede Transaktion verfügt über ein Locking und Unlocking Skript. Erst wenn beide Skripte zusammen erfolgreich ausgeführt werden konnten, ist eine Transaktion gültig.

Durch dieses Projekt wird eine Open Source Lösung angeboten, welche es dem Nutzer erlaubt offline und sicher seine Bitcoin Skripte zu testen.

Mittels einer visuellen und animierten Darstellung der einzelnen Ausführungsschritte von Bitcoin Skripten, soll deren Verständnis im Unterricht gefördert werden.

Das Projekt dient der Forschung und hat keinen direkten kommerziellen Nutzen

3 Projektziele

Ziel ist es eine Website mit JavaScript zu realisieren, welche es dem Nutzer ermöglicht Bitcoin Skripte zu Debuggen. Dies soll mit Hilfe einer Visualisierung des Bitcoin Skript Stacks passieren.

Die Webseite soll offline voll funktionsfähig sein. Dadurch kann der Benutzer sicherstellen, dass keine heiklen Daten wie private Schlüssel an einen Server übermittelt werden.

Stakeholder:

- Dozenten und Studierende mit entsprechender Fachrichtung
- Bitcoin Enthusiasten, die Ihr Verständnis von Bitcoins festigen und die Bitcoin Scriptsprache verstehen wollen.
- Anwender welche Non Standard Scripts testen, entwickeln oder nutzen wollen.

Non Standard Scripts können verwendet werden um spezielle Anwendungsfälle abzudecken. Zum Beispiel zwei Geschäftspartner zahlen Geld auf eine Bitcoin Adresse. Nun will man aber sichergehen, dass das einbezahlte Geld nur dann ausgegeben werden kann, wenn beide Partner die Transaktion gut heissen, also beide mit ihrem Privat Key signieren. Gleichzeitig will man sich aber absichern, dass wenn einer der beiden Geschäftspartner vom Bus überfahren wird, die Bitcoins nicht verloren sind. Dafür entwirft man nun ein Custom Bitcoin Script, welches zum Beispiel dem Anwalt erlaubt, nach 3 Monaten die Bitcoins zu überweisen. Solche und noch viel komplexere Anwendungsfälle können mit Non Standard Scripts realisiert werden.

4 Systemabgrenzung

4.1 Prozessumfeld

Wikipedia: „Bitcoin (englisch sinngemäß für „digitale Münze“) ist eine digitale Währung, gleichzeitig auch der Name des weltweit verwendbaren dezentralen Buchungssystems sowie die vereinfachende Bezeichnung einer kryptografisch legitimierten Zuordnung von Arbeits- oder Rechenaufwand. Überweisungen werden von einem Zusammenschluss von Rechnern über das Internet mithilfe einer speziellen Peer-to-Peer-Anwendung abgewickelt, sodass anders als im herkömmlichen Bankverkehr keine zentrale Abwicklungsstelle benötigt wird. Eigentumsnachweise an Bitcoin können in einer persönlichen digitalen Brieftasche gespeichert werden. Der Umrechnungskurs von Bitcoin in andere Zahlungsmittel bestimmt sich durch Angebot und Nachfrage.

Das Bitcoin-Zahlungssystem wurde erstmals 2008 in einem unter dem Pseudonym Satoshi Nakamoto veröffentlichten Dokument beschrieben. Im Jahr darauf wurde eine Open-Source-Referenzsoftware dazu veröffentlicht. Das Bitcoin-Netzwerk basiert auf einer von den Teilnehmern gemeinsam mit Hilfe einer Bitcoin-Software verwalteten dezentralen Datenbank (der Blockchain), in der alle Transaktionen verzeichnet sind. Die einzige Bedingung für die Teilnahme ist der Betrieb eines Bitcoin-Clients; alternativ kann auch einer der Online-Dienste genutzt werden (z. B. für mobile Geräte). Dadurch unterliegt das Bitcoin-System keiner geographischen Beschränkung – ein Internetzugang genügt – und kann länderübergreifend eingesetzt werden.

Mit Hilfe kryptographischer Techniken wird sichergestellt, dass Transaktionen mit Bitcoins nur vom jeweiligen Eigentümer vorgenommen und die Geldeinheiten nicht mehrfach ausgegeben werden können. Daher wird Bitcoin auch als Kryptowährung bezeichnet, obwohl der Begriff Währung normalerweise von Staaten emittierte Zahlungsmittel bezeichnet. In deutschsprachigen Medien wird auch die Bezeichnung Kryptogeld benutzt.“

4.2 Systemumfeld

Der Bitcoin Script Debugger baut auf dem etablierten JavaScript Framework Bitcore lib auf (siehe auch https://bitcore.io/).

4.3 Randbedingungen

4.3.1 Technische Rahmenbedingungen

- Offline lauffähige Webapplikation
- Umsetzung mit JavaScript

4.3.2 Organisatorische Vorgaben

- Abgabe des Source Code für Review am 23.12.2017
- Code Reviews erfolgen bis am 15.01.2018
- Projektpräsentation am 18.01.2018
- Projektabschluss am 20.01.2018

4.3.3 Standards

- OP Code Interpretation muss äquivalent zum original Bitcoin Source Code geschehen. Der Bitcoin Source Code gilt als Standard für alle Bitcoin Implementationen.

5 Anforderungen

5.1 Quellen und Vorgehen

Der Projektumfang und die Ziele sind bereits durch den Auftraggeber Kai Brünnler via Projektausschreibung (Bitcoin-Skript-Debugger) vorgegeben. In der Beschreibung befindet sich ein zusätzlicher Verweis auf ein ähnliches Projekt (https://webbtc.com/script), dessen Funktionsumfang als Referenz für die Erhebung der Anforderungen hinzugezogen werden kann.

5.2 Funktionale Anforderungen

ID	Status	Prio	Beschreibung
F1.1	Freigegeben	**M**	Die Applikation erlaubt die Eingabe von Locking und Unlocking Scripts via Copy und Paste. Das einlesen von externen Scripts ist nicht vorgesehen.
F1.2	Freigegeben	**P1**	Die Syntax des Bitcoin Scripts wird farblich hervorgehoben (Syntaxhighlighting).
F1.3	Freigegeben	**W**	Die Scripteingaben können automatisch formatiert werden (jeder OP Code beginnt auf einer neuen Zeile).
F1.3	Freigegeben	**P1**	Alle Syntaxfehler werden in Echtzeit (d.h. während der Eingabe) erkannt. Ungültige OP Codes werden mit einer Hintergrundfarbe hervorgehoben. Eine Fehlerbeschreibung wird als Text in einem Konsolenfenster ausgegeben.
F2.1	Freigegeben	**M**	Die Locking und Unlocking Scripts können ausgeführt werden und das Resultat wird dargestellt.
F2.2	Freigegeben	**M**	Zeilenweises Ausführen der Bitcoin Scripts. Bei jedem Ausführungsschritt werden die aktuelle Operation und der Zustand des Stacks visuell hervorgehoben.
F3.1	Freigegeben	**M**	Kontextvariablen wie die aktuelle länger der Block Chain oder Zeitstempel können durch eine separate Eingabe manipuliert werden.
F4.1	Freigegeben	**M**	Dummy Key Pairs und Hashes können generiert und angezeigt werden damit diese in Test Scripts verwendet werden können.
F5.1	Freigegeben	**P1**	Es können Beispiel Scripts über ein Menu selektiert werden und automatisch in die Eingabefelder abgefüllt werden.
F6.1	Freigegeben	**P2**	Es soll möglich sein zu kontrollieren, ob ein eingegebenes Script als Standard Script betrachtet wird oder nicht.

Tabelle 1: Funktionale Anforderungen

Attribute:
ID: eindeutige Identifikation
Status: Entwurf / geprüft / freigegeben
Priorität: Muss / Optional P1, P2, P3 / Wunsch (Nice to have)

5.3 Technische Anforderungen

ID	Status	Prio	Beschreibung
T1.1	Freigegeben	M	Das Projekt muss als offline Webapplikation ausgeführt werden können.
T2.1	Freigegeben	M	Der Source Code des Projekts ist Open Source.
T3.1	Freigegeben	M	Der vorgegebene Technologie Stack ist HTML, CSS und JavaScript.

Tabelle 2: Technische Anforderungen

5.4 Qualitätsanforderungen

ID	Status	Prio	Beschreibung
Q1.1	Freigegeben	M	Es ist durch manuelle Tests sicherzustellen, dass das Projekt den erwarteten Anforderungen entspricht.
Q2.1	Freigegeben	P1	Jeder OP Code muss innerhalb von 50 Millisekunden auf einem handelsüblichen Computer ausgeführt werden können.

Tabelle 3: Qualitätsanforderungen

6 Glossar

HTML (Hypertext Markup Language)

HTML ist eine Markup Sprache welche von der Organisation w3c standardisiert und gepflegt wird. Mittels HTML können Webseiten und Webapplikationen erstellt werden.

Weitere Informationen zu HTML können unter:
HTML 5 Spezifikation: https://www.w3.org/TR/html5/
Geschichte von HTML: https://en.wikipedia.org/wiki/HTML
nachgelesen werden.

Markup Language

Markup Sprachen erlauben die Annotation eines Dokumentes in einer Weise welche syntaktisch unterscheidbar vom reinen Text ist. In Markup Sprachen werden sogenannte Tags verwendet. Diese dienen dazu den Text, welcher von diesem Tag betroffen ist, eine bestimmte Darstellung / Funktion zu verleihen (Bsp.: Hyperlink oder Unterstrichen).

Weiter Informationen können unter https://en.wikipedia.org/wiki/Markup_language gefunden werden.

CSS (Cascading Style Sheet)

Mittels CSS wird die Präsentation einer Webseite einschliesslich Farbe, Layout und Fonts beschrieben. Es erlaubt einem die Präsentation für verschiedene Endgeräte wie grosse Bildschirme, kleine Bildschirme, Drucker und andere anzupassen. CSS ist unabhängig von HTML. Es kann also auch mit XML-basierten Markup-Sprachen verwendet werden.

Weiter Informationen können unter https://www.w3.org/standards/techs/css respektive https://en.wikipedia.org/wiki/Cascading_Style_Sheets gefunden werden.

JavaScript (JS)

Wikipedia: „Der als ECMAScript (ECMA 262) standardisierte Sprachkern von JavaScript beschreibt eine dynamisch typisierte, objektorientierte, aber klassenlose Skriptsprache."

Weiter Informationen können unter https://de.wikipedia.org/wiki/JavaScript gefunden werden.

Bitcoin

Wikipedia: „Bitcoin (englisch sinngemäß für „digitale Münze") ist eine digitale Währung, gleichzeitig auch der Name des weltweit verwendbaren dezentralen Buchungssystems sowie die vereinfachende Bezeichnung einer kryptografisch legitimierten Zuordnung von Arbeits- oder Rechenaufwand. Überweisungen werden von einem Zusammenschluss von Rechnern über das Internet mithilfe einer speziellen Peer-to-Peer-Anwendung abgewickelt, sodass anders als im herkömmlichen Bankverkehr keine zentrale Abwicklungsstelle benötigt wird."

Weitere Informationen können unter https://de.wikipedia.org/wiki/Bitcoin gefunden werden.

Blockchain

Wikipedia: „Eine Blockchain (auch Block Chain, englisch für Blockkette) ist eine kontinuierlich erweiterbare Liste von Datensätzen, genannt „Blöcke“, welche mittels kryptographischer Verfahren miteinander verkettet sind. Jeder Block enthält dabei typischerweise einen kryptographisch sicheren Hash des vorhergehenden Blocks, einen Zeitstempel und Transaktionsdaten.”

Weiter Informationen können unter https://de.wikipedia.org/wiki/Blockchain gefunden werden.

Bitcoin Script

Bitcoin benutzt ein scriptbasiertes System für Transaktionen. Die Bitcoin Script Sprache ist eine Stack-basierte und nicht Turing mächtige (keine Schleifen möglich) Script Sprache. Die Auswertung geschieht von links nach rechts.

Ein Bitcoin Script ist im Wesentlichen eine Liste von Instruktionen welche auf der Transaktion festgehalten sind. Die Liste von Instruktionen definiert wie jemand Zugriff auf die Bitcoins in der Transaktion erhält.

Weitere Informationen können unter https://en.bitcoin.it/wiki/Script gefunden werden.

OP Code

Wikipedia: „Ein Opcode, auch op code oder operation code, ist eine Zahl, die die Nummer eines Maschinenbefehls für einen bestimmten Prozessortyp angibt.“

In Bezug auf Bitcoins beziehen sich die Opcodes auf die einzelnen Befehle in der Script-Sprache.

Weitere informationen zu Opcodes im allgemeinen und Bitcoin Opcodes im spezfischen können unter https://de.wikipedia.org/wiki/Opcode respektive https://en.bitcoin.it/wiki/Script gefunden werden.

Open Source

Wikipedia: „Als Open Source (aus englisch open source, wörtlich offene Quelle) wird Software bezeichnet, deren Quelltext öffentlich und von Dritten eingesehen, geändert und genutzt werden kann. Open-Source-Software kann meistens kostenlos genutzt werden.“

Weiter Informationen können unter https://de.wikipedia.org/wiki/Open_Source gefunden werden.

Asymmetrische Kryptographie

Wikipedia: „Ein asymmetrisches Kryptosystem oder Public-Key-Kryptosystem ist ein kryptographisches Verfahren, bei dem im Gegensatz zu einem symmetrischen Kryptosystem die kommunizierenden Parteien keinen gemeinsamen geheimen Schlüssel zu kennen brauchen. Jeder Benutzer erzeugt sein eigenes Schlüsselpaar, das aus einem geheimen Teil (privater Schlüssel) und einem nicht geheimen Teil (öffentlicher Schlüssel) besteht. Der öffentliche Schlüssel ermöglicht es jedem, Daten für den Besitzer des privaten Schlüssels zu verschlüsseln, dessen digitale Signaturen zu prüfen oder ihn zu authentifizieren. Der private Schlüssel ermöglicht es seinem Besitzer, mit dem öffentlichen Schlüssel verschlüsselte Daten zu entschlüsseln, digitale Signaturen zu erzeugen oder sich zu authentisieren.“

Weiter Informationen können unter https://de.wikipedia.org/wiki/Asymmetrisches_Kryptosystem gefunden werden.

7 Abbildungsverzeichnis

Es werden keine Abbildungen in diesem Dokument verwendet.

8 Tabellenverzeichnis

9 Literaturverzeichnis

Webseite
https://www.w3.org/TR/html52/introduction.html#introduction

https://en.wikipedia.org/wiki/HTML

Webseite
https://en.wikipedia.org/wiki/Markup_language

Webseite
https://www.w3.org/standards/techs/css

Webseite
https://de.wikipedia.org/wiki/JavaScript

Webseite
https://de.wikipedia.org/wiki/Bitcoin

Webseite
https://de.wikipedia.org/wiki/Blockchain

Webseite
https://en.bitcoin.it/wiki/Script

Webseite
https://de.wikipedia.org/wiki/Opcode

Webseite
https://de.wikipedia.org/wiki/Open_Source

Webseite
https://de.wikipedia.org/wiki/Asymmetrisches_Kryptosystem

10 Anhang

10.1 Abstimmung der Anforderungen

Bei der Abstimmung der Anforderungen sind keine Probleme aufgetreten.

10.2 Definition of Ready – Checklist

Folgende Tests müssen erfolgreich ausführbar sein:

1. Sämtliche Standard Scripts müssen korrekt evaluiert werden und der Stack muss korrekt dargestellt werden. Folgende Scripts werden als Standard Scripts bezeichnet:

   - P2PKH (pay to public key hash)
   - P2SH (pay to script hash)
   - P2PK (pay to public key)
   - Multisignature
   - OP_RETURN metadata

2. Syntaxmarkierung und Syntaxerkennung von Scriptbefehlen. Der Benutzer wird bereits während der Eingabe auf Syntax Fehler hingewiesen.

3. Die Webseite mit dem Bitcoin Script Debugger funktioniert auch offline ohne die zusätzliche Installation von weiteren Tools.

4. Der Benutzer kann mittels Texteingabe Non-Standard Scripts eingeben und diese ausführen.

5. Die schrittweise Ausführung von Scripts ist möglich.

11 Versionskontrolle

Version	Datum	Beschreibung	Autor
X0.1	02.10.2017	Dokument erstellt	Rolf Zurbrügg
X0.2	10.10.2017	Erfassen der Anforderungen	Samuel Egger
X0.3 – X0.8	12.11.2017	Überarbeitung und Korrekturen, Erweiterung des Glossars und Literatur Verzeichnisses, Erstellung der Definition of Ready - Checklist	Rolf Zurbrügg
X0.9	13.11.2017	Gesamtes Dokument überarbeitet. Annotation in Rot bezüglich punkten die ausgebessert, ergänzt werden müssen, gemäss dem Feedback von Herrn G. Schwab	Samuel Egger, Rolf Zurbrügg
X1.0	15.11.2017	Finalisierung des Dokumentes	Samuel Egger, Rolf Zurbrügg