FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Správa blockchainu pro FITCOIN
Student:	Andrea Zábojníková
Vedoucí:	Mgr. Jan Starý, Ph.D.
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2019/20

Pokyny pro vypracování

1. Navrhněte a implementujte co nejjednodušší strukturu blockchainu pro triviální měnu FITCOIN.
2. Implementaci proveďte v jazyce C. Výsledkem budou moduly, které může používat zastřešující full node.
3. Dbejte na čistotu, korektnost a přenositelnost kódu mezi operačními systémy typu UNIX (Linux, *BSD, MacOS, ...).
4. Implementaci řádně zdokumentujte.
5. Implementaci řádně otestujte.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 6. prosince 2018

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Bakalářská práce

Správa blockchainu pro Fitcoin

Andrea Zábojníková

Katedra softwarového inženýrství
Vedoucí práce: Mgr. Jan Starý, Ph.D.

3. června 2020

Poděkování

Ráda bych poděkovala Mgr. Janu Starému, Ph.D., za rady a konzultace při vývoji Fitcoinu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo"), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 3. června 2020 . . . . . .

Odkaz na tuto práci

Zábojníková, Andrea. *Správa blockchainu pro Fitcoin*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2020.

Abstrakt

Bakalářská práce popisuje návrh a implementaci blockchainu pro kryptoměnu. Kryptoměny využívají blockchain pro uložení transakcí, tedy blockchain plní roli distribuované účetní knihy. V teoretické části popisuji potřebné koncepty používané v kryptoměnách a z těchto poznatků je pak odvozena implementace pro kryptoměnu Fitcoin. Praktickou částí je pak implementace v jazyce C.

Klíčová slova blockchain, kryptoměna, transakce, virtuální měna, decentralizace, Bitcoin, platební systém

Abstract

The bachelor thesis describes the design and implementation of a blockchain for the cryptocurrency. Cryptocurrencies use a blockchain technology to record transactions, so the blockchain acts as a distributed ledger. In the theoretical part I describe the necessary concepts used in cryptocurrencies and from this knowledge the implementation for the cryptocurrency Fitcoin is derived. The practical part is the implementation in the C language.

Keywords blockchain, cryptocurrency, transakce, virtual currency, decentralization, Bitcoin, payment system

Obsah

Seznam obrázků

Seznam tabulek

Cíl práce

Bakalářská práce popisuje jak navrhnout a implementovat blockchain pro kryptoměnu Fitcoin. Většina kryptoměn využívá blockchain jako databázi proběhlých transakcí, tedy blockchain plní úlohu distribuované účetní knihy.

Fitcoin je společným cílem bakalářských prací Filipa Volfa a Lukáše Danga. Naše práce vede Mgr. Jan Starý, Ph. D. a pro tento účel implementuje ve Fitcoinu peer-to-peer síťovou komunikaci a udržuje projekt pohromadě. Fitcoin je školní kryptoměna běžící v operačních systémech typu Unix, je psaná v jazyku C. Motivací je pochopit jak fungují kryptoměny a z jakých jednotlivých částí se skládají. Mým úkolem je navrhnout správu blockchainu, zajistit správné vkládání bloků do blockchainu a implementovat tuto funkčnost do Fitcoinu. Fitcoin se vyvíjí od nuly a měl by mít hlavně minimalistickou strukturu. V poslední fázi by měl být kód řádně zdokumentovaný a otestovaný.

Úvod

V září roku 2008 burzy po celém světě zaznamenaly hluboký propad a svět se začal potácet ve finanční krizi. Satoshi Nakamoto[1] 31. října prostřednictvím kryptografického mailing listu publikoval článek s názvem „Bitcoin: A Peer-to-Peer Electronic Cash System“ [2] ve kterém představil koncept kryptoměny Bitcoin, 3. ledna 2009 vytvořil nultý blok, tzv. *genesis blok* a pár dní na to publikoval zdrojový kód Bitcoinového klienta. Bitcoin se dodnes aktivně vyvíjí komunitou jako open source.

„Bitcoin je kolekce konceptů a technologií, která vytváří základy ekosystému digitálních peněz“[1]. Kryptoměny se postupně staly fenoménem a komoditou o kterou nemají zájem jenom technologičtí nadšenci, ale i široká veřejnost. Dnes se tak můžeme účastnit společenského a ekonomického experimentu. Často média a banky publikují zprávy o konci Bitcoinu, ale navzdory všem černým scénářům adopce uživateli stále roste. Bitcoinu se často vytýkají problémy se škálováním, které se projevují tím, že se v síti nahromadí nezpracované transakce, poplatky za transakce se navýší a potvrzení transakcí trvá déle. Bitcoin neohrožuje malý zájem uživatelů, ale spíše velký – což také není ideální stav.

K masivnějšímu rozšíření Bitcoinu je také potřeba dobrá infrastruktura, a chce-li člověk přispět k dalšímu rozvoji v této oblasti, potřebuje znát základní principy, na kterých kryptoměny fungují. Kryptoměny pravděpodobně budou stále častěji běžná součást platebních metod. Dnes je velké množství informací a textů o kryptoměnách, avšak samotná práce s kódem a technologií vede k lepšímu pochopení technologie. Některé části Bitcoinu ani dosud nejsou dostatečně zdokumentované a pro jeho pochopení je stejně nakonec potřeba se do zdrojového kódu podívat.

O blockchainu se nejčastěji mluví v souvislosti s kryptoměnami, díky kterým

[1] Pseudonym autora nebo autorů, kteří za vznikem Bitcoinu stojí. Z podstaty této technologie to však není důležité, jelikož ani její autor není schopen síť ovládat nebo jí jakkoliv manipulovat. Maximálně utratí své vytěžené mince.

se tento pojem dostal do povědomí široké veřejnost. Satoshi Nakamoto při uvedení Bitcoinu ani slovo blockchain nepoužil, nazýval jej slovy chain of block, Proof of Work chain, distributed timestamp server.

Decentralizace

Satoshi považoval za největší slabinu plateb *model důvěry*, konkrétně nutnost důvěřovat nějaké třetí straně (kterou je vláda, banka, instituce nebo jiná entita). Spojením několika technologií, vznikl systém důvěry založený na kryptografickém důkazu. V Bitcoinu proběhlé transakce kontrolují všichni účastníci sítě na základě dohodnutých podmínek formou konsenzu. Většina účastníků je finančně motivovaná pravidla dodržovat a propagují se pouze ty transakce, na kterých panuje shoda. Systém je transparentní, otevřený všem zúčastněným a spolehlivý bez lidského zásahu nebo kontrolního orgánu. Také je rezistentní proti cenzuře.

Chytré kontrakty

Bitcoin neuchovává pouze účetní transakce, základní úlohou Bitcoinové sítě je provádění a ověřování tzv. chytrých kontraktů. Myšlenku chytrých kontraktů popsal Nick Szabo v roce 1997 a definoval je jako *„automatizovaný transakční protokol, který vykonává podmínky kontraktu“*. Můžeme si to představit tak, jako by se každá platba uzavírala proti samovymahatelné smlouvě. Vzhledem k tomu, že se stále jedná o software, vše se vykonává automaticky, rychle a bezpečně a tím se značně snižují transakční náklady. Tyto kontrakty jsou schopny ve velké míře změnit naše interakce v podnikání, právu a společnosti.

Tato představa se nyní realizuje především v kryptoměně Ethereum, kde je možné programovat kontrakty v turingovsky úplném programovacím jazyce postaveném nad virtuálním strojem (Ethereum Virtual Machine). V tomto duchu se Ethereum jeví spíše jako decentralizovaná výpočetní síť, než jako platební systém. Skriptovací jazyk Bitcoinu Script nepodporuje cykly ani skoky a je odvozený od jazyka Forth.

Aplikace chytrých kontraktů v kryptoměně jsou široké a spousta z nich nachází uplatnění ve finanční službách. Pro zjednodušení implementace Fitcoinu jsme se rozhodli s transakcemi pracovat jen jako s účetním záznamem, než jako s programovatelným kontraktem a v dalších kapitolách chytré kontrakty v práci zmiňovány nebudou. Na to, aby Fitcoin plnil svou funkci toto zjednodušení mít vliv nebude, chytré kontrakty fungující nad účetní knihou slouží k rutinnímu podepsání a ověření podpisu transakce před změnou vlastnictví tokenu.

Struktura práce

V práci je často zmíněn Bitcoin, to má svůj důvod, Bitcoin je hlavní předlohou Fitcoinu a také mnoho kryptoměn vzniklo jako jeho fork. Některé měny se snažily konkurovat Bitcoinu v některé konkrétní oblasti (v anonymitě, jiným protokolem konsensu apod.). Mezi úspěšnými altcoiny (alternativní kryptoměny k Bitcoinu), které nejsou přímým derivátem Bitcoinu jsou například Monero a Ethereum.

Fitcoin se bude snažit konkurovat ve srozumitelnosti a jednoduchosti implementace. Tyto vlastnosti nám nepomohou nezajistit úspěch na trhu s kryptoměnami, ale poslouží zájemcům o kryptoměny jako jednoduché uvedení do světa kryptoměn na kterém si mohou prakticky a bezpečně vyzkoušet práci s kryptoměnou aniž by riskovali ztrátu peněz. Pro programátora jsou užitečné ke snazšímu proniknutí do technologie.

Většina kryptoměn funguje na stejných principech, které v práci popíši. Fitcoin sice bude mít s Bitcoinem společnou architekturu, ale o fork nepůjde. Fitcoin začíná na zelené louce a měl byl mít hlavně minimalistickou strukturu a také fungování. Nadbytečné věci by měly být vypuštěny. Práce nejprve popíše klíčové pojmy a koncepty, které budeme potřebovat ke konstrukci libovolné kryptoměny a v posledních kapitolách se postupně dostanu k samotné implementaci blockchainu pro Fitcoin.

První kapitola popisuje kryptografii použitou v kryptoměnách. V druhé kapitole se zabývám analýzou digitálních peněz, kde si ujasním vhodné vlastnosti virtuálních peněz a později z nich odvodím podobu transakcí Fitcoinu. Třetí část je věnovaná blockchainu a ve čtvrté se popisuje fungování kryptoměny na peer-to-peer síti. Na závěr je z těchto poznatků navrhnuta implementace. Praktická část obsahuje zdrojové kódy blockchainu ve Fitcoinu napsané v jazyce C.

Kapitola 1

Kryptografie

Kryptoměny, jak napovídá název, využívají kryptografii, tím nejsou nijak speciální oproti jiným platebním systémům. Bez kryptografie by on-line platby prakticky neexistovaly. Kvalita digitálních peněz je závislá na tom, jaké kryptografické systémy zvolíme a jakým způsobem systém funguje dohromady. Kryptoměny využívají zejména hashovací funkce a asymetrické šifrování, konkrétní kryptografické funkce a algoritmy se v různých kryptoměnách liší. Přece jen nechceme zabezpečit nic menšího než peníze, které chráníme i ve fyzickém světě. Cílem měny je zabezpečit následující funkcionality:

- Problém dvojího utrácení, nebo-li double-spending problem. Zduplikovat zlatou minci nebo zfalšovat bankovku není tak triviální jako kopírovat digitální informaci. Kopírování nelze zabránit, o čemž se opakovaně přesvědčuje hudební a filmový průmysl. Platba, která utrácí jednotky, které už byly utraceny by měla být odmítnuta. Kontrola probíhá oproti historii všech provedených transakcí (účetní knize).

- Emise. v reálném světě opatřují státy bankovky ochrannými prvky proti padělání. Zvětšování objemu peněz znehodnocuje měnu. Paradoxně státní instituce peníze znehodnocují nejvíc tiskem nových bankovek a multiplikací peněz prostřednictvím úvěrů. Měna by měla mít jasně daná pravidla, za kterých se emitují peněžní jednotky. Kryptoměny mají vlastní mechanismy monetární kontroly.

  V Bitcoinu má existovat maximálně 21 milionů bitconů[2], nové jednotky se emitují prostřednictvím těžby. Ještě nedávno většina sítě používala klienta Bitcoin Core ve verzi, která obsahovala zranitelnost CVE-2018-17144, tzv. *Inflation Bug*, který mohl zapříčinit vznik více mincí kvůli špatné kontrole duplicit ve vstupu transakce. Ke zneužití naštěstí nedošlo.

[2] https://github.com/bitcoin/bips/blob/master/bip-0042/inflation.png

- Vlastnictví prostředků. Banky, či jiná centrální autorita spravuje v databázích zůstatky klientů na účtech a vede evidenci příchozích a odchozích plateb.

  Prokazování vlastnictví tokenů měny je v kryptoměnách zajištěno asymetrickou kryptografií. Vlastnictví tokenů znamená mít přístup k adresám, na které směřují výstupy neutracených transakcí. Utrácet mince je pak možné pouze se znalostí soukromého klíče.

- Historie transakcí. Měla by být nezměnitelná. Neměnnost záznamů zabezpečuje blockchain.

- Převod. Peníze se nesmí znehodnotit po cestě k novému příjemci. Převod peněz musí být spolehlivý, bezpečný a nesmí při něm uniknout citlivá data. Kryptoměny neshromažďují informace o totožnosti svých uživatelů, vhodnými datovými strukturami zabezpečují integritu dat a účastníci sítě pečlivě validují každou transakci.

Protože se obecně nedoporučuje snažit se implementovat vlastní kryptografické funkce, využíváme ve Fitcoinu kryptografickou knihovnu. Ve Fitcoinu se používá primárně open-source knihovna LibreSSL, která je forkem knihovny OpenSSL. Fork vzniknul v dubnu 2014 v reakci na odhalení bezpečnostní chyby Heartbleed (CVE-2014-0160). Od původní knihovny se liší také výrazným pročištěním kódu a větší srozumitelností. LibreSSL i OpenSSL poskytují všechny druhy kryptografických primitiv, které potřebujeme ke konstrukci kryptoměny. Kryptografická primitiva jsou dobře prozkoumané algoritmy, které se používají jako základní stavební prvky složitějších kryptografických systémů.

1.1 Hash

Hashe mají v kryptoměně nezastupitelnou roli: jsou hlavní součástí datových struktur, které tvoří blockchain a transakce (hash pointer a Merkleův strom), používají se v těžícím algoritmu Proof of Work, vytváříme nimi adresy v peněženkách, využíváme je v digitálním podpisu a slouží k běžné kontrole integrity dat, která načítáme ze sítě nebo z lokálního úložiště.

Hashovací funkce je matematická funkce, která na vstupu přijímá libovolně dlouhou binární zprávu $m \in \{0,1\}^*$ a vrátí binární řetězec o pevně definované délce n, kde $n \in N$.

$$h : \{0,1\}^* \to \{0,1\}^n \tag{1.1}$$

Kryptografické hashovací funkce mají navíc vlastnosti bezkoliznosti a jednosměrnosti a v ideálním případě se chová jako náhodné orákulum.

- Jednosměrnost: funkce je jednosměrná, pokud je snadné z jakékoli hodnoty m vypočítat hodnotu $y = h(m)$. Avšak nalézt vstupní vzor m pro náhodně zvolený obraz $y \in h(m)$ je výpočetně nemožné ($\in$ třída NP)

i když teoretický možné. K hashování se využívá typ jednosměrné funkce, u které se věří v to, že ji nelze invertovat.

- Náhodné orákulum: výstup této funkce nejde snadno uhádnout a působí jako náhodný výběr prvku z celého oboru hodnot. Při každém zavolání se chová deterministicky, avšak změna byť jednoho bitu na vstupu bude mít za následek absolutně odlišnou hodnotu hashe.

- Bezkoliznost: funkce je bezkolizní, pokud je výpočetně náročné, ideálně nemožné, nalézt libovolnou kolizi. Rozlišujeme dva typy bezkoliznosti:

  – Odolnost proti nalezení kolize. Nalezneme dvojici různých vstupů m_1 a m_2 mající stejný obraz, tedy $h(m_1) = h(m_2)$. Pravděpodobnost náhodného nalezení této kolize je známá jako *„narozeninový paradox“*. Kolizi dvou prvků tedy najdeme průměrně po $2^{\frac{n}{2}}$ pokusech.

  – Odolnost proti nalezení druhého vzoru: pro vzor m_1 nalézáme druhý různý vzor m_2, který dává stejný obraz a tedy platí $h(m_1) = h(m_2)$. Pokud se bude chovat funkce jako náhodné orákulum, pak nelze hledat druhý vzor rychleji než hrubou silou přes všech 2^n možností.

Důležitým parametrem při volbě vhodné hashovací funkce je délka hashe. Pokud je velikost definičního oboru hašovací funkce větší, než je velikost oboru hodnot, pak hashovací funkce nemůže být bezkolizní (princip holubníku). Například 256bitová kryptografická hashovací funkce může vygenerovat až 2^{256} unikátních čísel a to nám dává velkou jistotu, že kolize nenastane.

Při výběru vhodného algoritmu hashovací funkce pro kryptoměnu můžeme také zvážit, jak je hashovací funkce rezistentní proti těžbě na hardware typu FPGA a ASIC. Obvykle tyto algoritmy mají větší prostorovou složitost. Tato vlastnost ovlivňuje, jakým způsobem budou nové tokeny distribuovány mezi uživatele.

Hashovací funkce	**Kryptoměna**
Scrypt	Litecoin, Dogecoin
Equihash	Zcash
Blake256	Decred
CryptoNight (CryptoNote)	Monero

Tabulka 1.1: Alternativních hashovací algoritmy v kryptoměnách.

Bitcoin používá dvě hashovací funkci, první je SHA256, která se pro zvýšení bezpečnosti aplikuje na výstup ještě jednou a vytváří *dvojitý SHA256*. Druhou je funkce RIPEMD-160, která slouží pro zkrácení adresy peněženky z 256 bitů na 160 bitů.

1.1.1 Hash pointer

Hash pointerem nazýváme speciální odkaz, který má podobu hashe struktury na kterou odkazujeme. Tímto libovolným datům vytvoříme jednoznačnou identitu a zajistíme i jejich integritu. Těchto vlastností bohatě využijeme v implementaci samotného blockchainu, kde se hash pointery odkazujeme na jednotlivé bloky a transakce.

1.1.2 Proof of Work

V roce 1997 zveřejnil Adam Back[3] systém Proof of Work (zkratka POW) jménem Hashcash, který měl původně sloužit k ochraně proti spamu a DDoS útokům. Skládá se ze dvou částí. První je předložení důkazu o provedené práci, druhá část je ověření tohoto důkazu. V kryptoměně se systém POW využívá pro vytváření validních bloků.

Hashcash funguje na principu hledání částečných kolizí hashů. Hledané hashe začínaly číslem 0 a měly stanovenou obtížnost udanou počtem počátečních nul. Hodnotu hashe můžeme reprezentovat jako číslo. Čím více počátečních nul nalezený hash obsahuje, tím je obtížnost vyšší, protože hledáme číslo ze stále menšího intervalu a potřebujeme generovat větší množství pokusů k nalezení vhodného hashe.

Na rozdíl od Hashcashe není v kryptoměně obtížnost specifikována pouze počtem nul, ale maximální hodnotou čísla T tzv. *targetu*. Čím nižší je hodnota target, tím vyšší je složitost těžby bloku. Tím se mění obtížnost rovnoměrněji a ne skokově po násobcích mocniny dvou. Těžař se snaží změnou parametru nonce (parametr bloku, který je rozebrán v sekci 3.2.1.4) měnit hash bloku a hledá právě takovou hodnotu, která je menší než target. Pravděpodobnost nalezení hashe, který je menší než target je pak:

$$P[H \leq T] = \frac{T}{2^{256}} \tag{1.2}$$

V Bitcoinu[3] je maximální target bloku stanoven na hodnotu hashe 0x00000000ffffffffffffffffffffffffffffffffffffffffffffffffffffffff, který odpovídá obtížnosti $d_0 = 1$. Obtížnost se pravidelně mění, aby se dodrželo stanovené pravidlo konsensu: vytvářet blok průměrně každých 10 minut (tzv. Target Spacing). Pokud během určitého časového období vytěží více bloků, než je průměr, pak se obtížnost zvyšuje, pokud méně, pak se snižuje.

Target se přepočítává každých 14 dní (tzv. Target Timespan), to znamená po vytěžení 2016 bloků (14 dní x 24 hodin x 6 bloků za hodinu). Mějme

[3] https://github.com/bitcoin/bitcoin/blob/master/src/chainparams.cpp

hodnoty $t_c = 14 \cdot 24 \cdot 60 \cdot 60$ (počet sekund ve 14 dnech), t_n je čas v sekundách, kdy byl vytvořen blok s výškou n. Pro $k, n \in N$ a $n = 2016k$, T_k vypočítáme:

$$\begin{aligned} \Delta t &= t_n - t_{(n-2016)} \\ T_k &= T_{k-1} \cdot \frac{\Delta t}{t_c} \end{aligned} \tag{1.3}$$

Těžba je často nepochopené téma mezi uživateli kryptoměn. Často zaznamenávám dotazy [4], co se stane s měnou, až těžaři přestanou těžit, když se většině z nich s narůstající složitostí přestane těžba ekonomicky vyplácet. Neuvědomují si, že se složitost přizpůsobuje aktuálnímu stavu sítě a také může klesat. Už v návrhu se zohledňoval Moorův zákon, který předpokládá zvyšování výkonu hardware v čase. Výkon těžby se určuje údajem hash rate, jednotkou je hash/s.

1.1.3 Merkleúv strom

Nebo také *Merkle tree* si v roce 1979 nechal patentovat Ralph Merkle. Jde datovou strukturu, kterou může být obecně libovolný n-ární strom, ale pro naše potřeby je nejvhodnější úplný binární strom. Struktura se běžně využívá k ověření integrity dat. V kryptoměně se osvědčil pro ověřování transakcí, podpisů v SegWitu a redukci úložného místa.

V listech stromu se nachází hash pointery na jednotlivé transakce zahrnuté do bloku. Merkleúv strom se vytváří zdola od listů rekurzivním hashováním potomků v každé generaci (výšce) stromu, tak dlouho dokud zůstane pouze jeden hash a tím je kořen. Kořen stromu se nazývá *Merkle koren* nebo *Merkle kořen.*

Pokud je počet listů lichý, pak se hash poslední položky zduplikuje a tím se doplní na sudý. V Bitcoinu kvůli tomu vznikla zranitelnost CVE-2012-2459, která spočívala v tom, že poslední duplicitní transakce neovlivňovala hodnotu kořene Merkleova stromu.

Příklad vytvoření kořenu Merkleova stromu se čtyřmi transakcemi:

$$\mathrm{Hash}_1 = H(\mathrm{Tx}_1),\ \mathrm{Hash}_2 = H(\mathrm{Tx}_2),\ \cdots\ , \mathrm{Hash}_4 = H(\mathrm{Tx}_\mathrm{n})$$

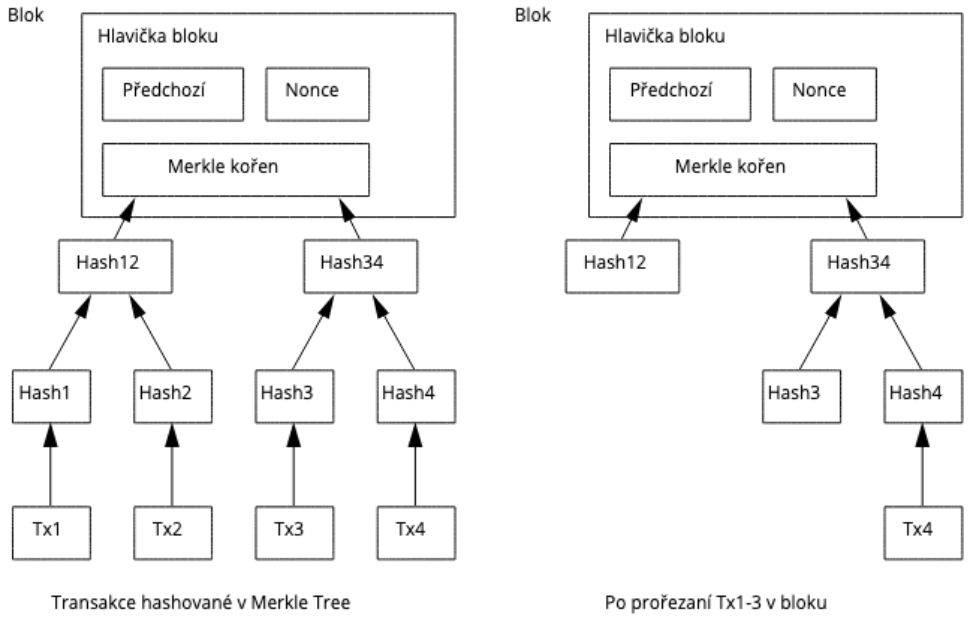
$$\mathrm{Hash}_{1,2} = H(\mathrm{Hash}_1 + \mathrm{Hash}_2),\ \cdots\ , \mathrm{Hash}_{3,4} = H(\mathrm{Hash}_3 + \mathrm{Hash}_4)$$

$$\mathrm{MerkleRoot} = \mathrm{Hash}_{1,2,3,4} = H(\mathrm{Hash}_{1,2} + \mathrm{Hash}_{3,4})$$

Výhodou Merkleova stromu je ověření existence prvku ve stromu v logaritmickém čase. V Merkleově stromu také platí, že každý rodič ověřuje korektnost dat svých potomků.

Na obrázku 1.1 je vpravo znázorněn strom, který pro ověření existence transakce nepotřebuje mít k dispozici všechny ostatní transakce bloku, ale stačí mu znát jen hashe na cestě od listu ke kořeni. Této vlastnosti se využívá při optimalizaci úložného místa, kdy můžeme utracené transakce *zapomenou.*

[4] hlavně před halvingem



Obrázek 1.1: Podoba struktury Merkleova stromu v bloku

1.2 Asymetrická kryptografie

Asymetrická kryptografie je kryptografický systém, ve kterém se používá k šifrování místo jednoho klíče dvojice klíčů. První klíč je veřejný, který můžeme zveřejnit ostatním účastníkům a slouží pro zašifrování zprávy a druhý klíč je soukromý, známý pouze adresátovi zprávy a používá se pro rozšifrování zprávy. Asymetrická kryptografie poskytuje celou řadu schémat, ale bezpečnost většiny stávajících implementací je postavena na třech matematických problémech, které nejsou na klasickém (ne kvantovém) počítači vypočítatelné v polynomiálním čase. Jde o problém faktorizace čísel, problém diskrétního logaritmu a problém diskrétního logaritmu eliptických křivek. Použitá knihovna OpenSSL/LibreSSL nabízí implementace schémat RSA, DSA a pro eliptické křivky pak ECDSA (digitální podpis) a ECDH (Diffieho–Hellmanův protokol využívající eliptické křivky). Obě využívají programovací jazyk C.

1.2.1 Problém faktorizace čísel

Máme dvě velká prvočísla p a q, pro které snadno vypočítáme jejich součin $n = p \cdot q$. Pokud však známe pouze složené číslo n, pak je výpočetně náročné ($\in$ třídě NP) nalézt oba jeho faktory p a q, pokud jsou vhodně zvoleny. Na tomto principu funguje kryptosystém RSA.

Konstrukce RSA se provádí následujícím algoritmem:

1. Adresát si náhodně zvolí dvě prvočísla p a q

z nichž vypočítá modul $n = (p \cdot q)$ a číslo $v = (p-1) \cdot (q-1)$.

2. Zvolí celé číslo $e \in (0, n)$, sloužící jako veřejný exponent. Toto číslo musí být nesoudělné s číslem v.

3. Vypočítá soukromý exponent $d \equiv e^{-1}\ mod\ v$.

4. Dvojice $VK = (e, n)$ je veřejný klíč adresáta a dvojice $SK = (d, n)$ je soukromý klíč adresáta.

Šifrování zprávy M, která se převede na číslo m poté probíhá jako: $c \equiv m^e\ mod\ n$, kde c je výsledná šifrovaná zpráva. Dešifrování poté probíhá obdobně: $m \equiv c^d\ mod\ n$.

1.2.2 Problém diskrétního logaritmu

Pokud známe kladná čísla $x, g \in Z$ a velké prvočíslo p, pak můžeme snadno vypočítat mocninu $y \equiv g^x\ mod\ p$. Avšak pokud známe pouze čísla y, g a p, pak nalezení exponentu $x \equiv log_g y\ mod\ p$ je výpočetně náročné a je v obecném případě prakticky nemožné v polynomiálním čase.

Nejznámější kryptosystém postavený na problému diskrétního algoritmu je El-Gamalův algoritmus.

1.2.3 Eliptické křivky

Kryptografie na bázi eliptických křivek byla nezávisle navržena kryptografy Nealem Koblitzem a Victorem Saulem Millerem v roce 1984 [4]. Kryptografický systém eliptických křivek je založen na problému diskrétního logaritmu přeneseného nad množinu bodů dané eliptickou křivkou. Praktickou výhodou tohoto systému, je dobrá úroveň bezpečnosti s mnohem kratším klíčem. Základem je křivka E nad tělesem K, která je definovaná obecnou Weierstrassovou rovnicí:

$$E/K : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$

$$a_1,\ a_2,\ a_3,\ a_4,\ a_6,\ x,\ y \in K.$$

Eliptická křivka E představuje množinu všech bodů $[x, y]$, vyhovujících Weierstrassově rovnici. Bodem eliptické křivky je také bod $\mathcal{O}$, který nazýváme bod v nekonečnu [4]. Abychom mohli sestrojit eliptickou křivku, musí platit, že diskriminant D nesmí být roven nule.

Nad eliptickými křivkami lze provádět algebraické operace negace bodu, sčítání dvou různých bodů, sčítání dvou stejných bodů a násobení bodů skalárem[4]. V praxi používá jednodušší tvar rovnice eliptické křivky:

$$y^2 = x^3 + ax + b, (x, y)\ \in R^2 \tag{1.4}$$

pro kterou se diskriminant vypočítá jako:

$$D = 4a^3 + 27b^2.$$

Pro záporný diskriminant získáme jednu spojitou křivku a pro kladný získáme křivku ze dvou spojitých částí.

1.2.4 Operace nad eliptickými křivkami

- Negace bodu. Ke každému bodu $P = (x, y)$, jenž leží na eliptické křivce E, a který není bodem v nekonečnu $\mathcal{O}$, lze sestrojit opačný bod $-P = (x, -y)$ [4].

- Sčítání dvou bodů. Pro body P a Q, pro které platí, že nejsou sobě navzájem negací, sestrojíme jejich součet R' tak, že je proložíme přímkou. Bod, ve kterém se přímka znovu protne s eliptickou křivkou E označíme jako bod R. Výsledný bod R' získáme negací bodu R, jak je znázorněno na obrázku 1.2 [5] v části 1.

- Sčítání stejného bodu (zdvojnásobení). V případě, že platí $P = Q$ zkonstruujeme tečnu k eliptické křivce E v bodě P. Dále postupujeme obdobně jako u sčítání dvou různých bodů. Tečna protne eliptickou křivku v bodě R a jeho negací získáme výsledný bod R'.

- Násobení bodu skalárem. Násobek bodu P skalárem $n \in N, n > 1$, získáme postupným sečtením bodu P se sebou samým n-krát.

$$\underbrace{P + P + \cdots + P}_{n \text{ krát}} = n \cdot P \tag{1.5}$$

- Zdvojnásobení bodu. Přičtení bodu P k sobě samému. Pokud by souřadnice byla nulová, pak tečna protne body v nekonečnu a platilo by $2P = \mathcal{O}$. Situace je znázorněna na obrázku 1.2 v části 4.

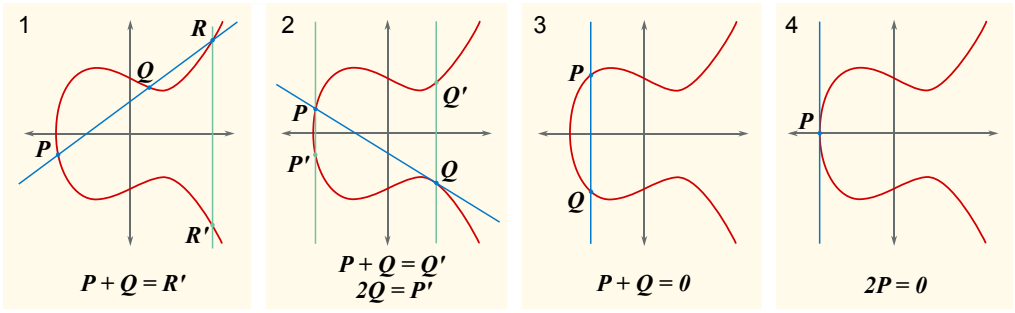
1.2.5 Eliptické křivky v kryptoměně

V praxi se využívá k šifrování několik eliptických křivek, které byly prokázány za bezpečné a pracuje se výhradně s tělesy prvočíselnými $GF(p)$ a $GF(2^m)$.

V Bitcoinu se používají křivky v konfiguraci secp256k1[5] doporučenou konsorciem SECG (Standards for Efficient Cryptography Group). Číslo za „secp“ znamená velikost klíče v bitech, p se odkazuje na konečné těleso prvočísel $GF(p)$ a písmeno k na Koblitzovy křivky. Parametry křivky jsou následující:

- Velikost modulu $p = 2^{256} - 2^{32} - 977$.

[5] https://commons.wikimedia.org/w/index.php?title=File:ECClines.svg



Obrázek 1.2: Příklady geometrického významu operací sčítání s body P a Q na eliptických křivkách

- Volené koeficienty $a = 0, b = 7$, čímž dostáváme rovnici.

$$E_{secp} : y^2 \equiv x^3 + 7 \; mod \; p.$$

- Volba základního bodu (generátoru) $G = (x, y)$ x=0x79be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798, y=0x483ada7726a3c4655da4fbfc0e1108a8fd17b448a68554199c47d08ffb10d4b8.
- Řád křivky n=0xfffffffffffffffffffffffffffffebaaedce6af48a03bbfd25e8cd0364141.
- Kofaktor je 1, protože řád křivky je prvočíslo.

1.2.6 Generování klíčů

Vygenerujeme si náhodnou a nepredikovatelnou hodnotu soukromého klíče číslo $k \in (1, n-1)$. Veřejný klíč je odvozen od soukromého klíče. Vypočteme si veřejný bod Q, vynásobením předem stanoveným základním bodem $G \in E$. Rovnice je následující:

$$Q_x = k_x G \tag{1.6}$$

Veřejný klíč tvoří čtveřice (E, G, n, Q_x). Hodnoty E, G, n jsou známé např. z uvedeného standardu secp256k1.

1.3 Digitální podpis

Digitální podpis je obvykle formou asymetrického kryptografického schématu, jeho účelem však není dešifrovat zprávu, ale ověřit pravost a integritu. Od podpisů požadujeme, aby plnil dvě následující funkce:

1. Podobně jako u ideálního papírového podpisu můžete svůj podpis vytvořit pouze vy a někdo jiný může ověřit jeho platnost.
2. Podpis je pevně spojen s konkrétním dokumentem (daty) a nikdo tento podpis nedokáže přenést na jiný dokument.

Podepisování je postaveno na třech funkcích:

1. Generátor klíčů, potřebujeme veřejný VK a soukromý klíč SK.
2. Operace podpis s := $sign(SK, m)$
3. Operace verifikace v := $verify(VK, m, s)$

1.3.1 Algoritmus ECDSA

ECDSA (Elliptic Curve Digital Signature Algorithm) je algoritmus používaný pro digitální podpis dat, který využívá eliptické křivky. ECDSA vychází z normy ANSI X9.62. Pracuje se slovem délky 160 až 256 bitů a garantuje stejnou bezpečnost jako 1024 až 3072 bitové RSA. Najdeme je například v podepisovacím skriptu *scriptSig*, který obsahuje digitální podpis a veřejný klíč.

1.3.2 Serializace

SECG specifikuje také normu pro serializaci veřejného ECDSA klíče. Používají se dva formáty komprimovaný (33 bitů) a nekomprimovaný (65 bitů). Oba formáty najdeme v knihovně OpenSSL/LibreSSL a typ reprezentující bod na křivce je EC_POINT. Daný bod P = (x,y) se interpretuje v nekomprimovaném formátu jako:

Název	**Typ**	**Velikost**
prefix 0x04	const	1 bajt
souřadnice x	integer big-endian	4 bajty
souřadnice y	integer big-endian	4 bajty

Tabulka 1.2: Nekomprimovaný SEC formát

Komprimovaný formát uvádí pouze souřadnici x a jeden znaménkový bit souřadnice y. Tento formát využívá toho, že pro určení bodu na eliptické křivce nepotřebujeme znát obě souřadnice. Pro x má rovnice křivky dvě řešení a znaménkem upřesňujeme o které jde.

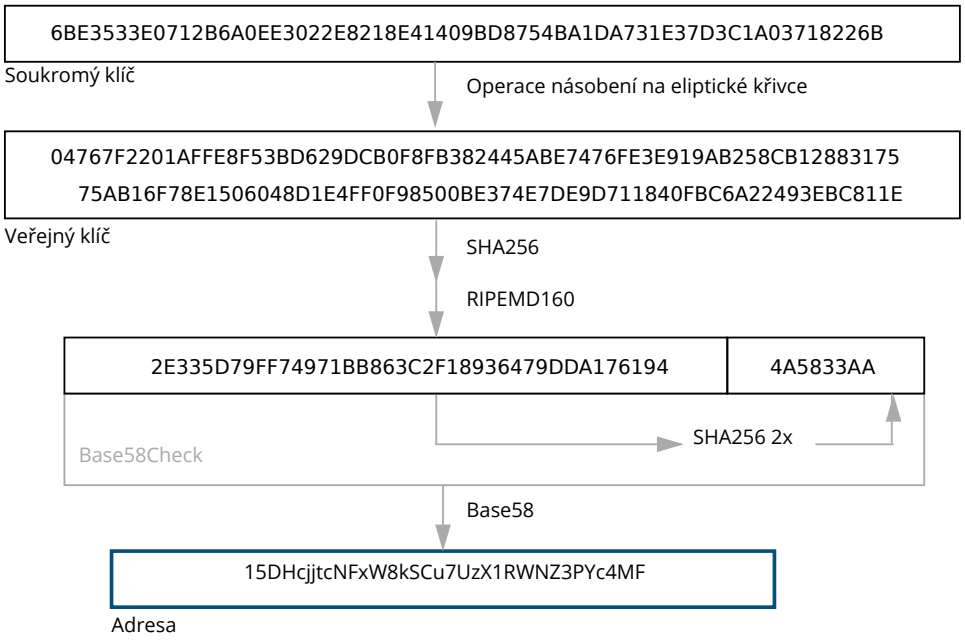
1.3.3 Slepé podpisy

V roce 1982 David Chaum představil návrh protokolu *slepých podpisů*[6] (známější pod původním názvem *blind signatures*). Jde o digitální podpis založený na RSA, který umožňuje signatáři podepsat zprávu, jejíž obsah mu není znám.

1.4 Adresa

Podobně jako používáme číslo bankovního účtu pro příjem a odeslání peněz, může si uživatel kryptoměny vygenerovat unikátní adresu, na kterou směruje

platby. Vytvoření adresy funguje tak, že si uživatel pomocí ECDSA vygeneruje dvojici klíčů - soukromý a veřejný.



Obrázek 1.3: Schéma konstrukce adresy v Bitcoinu

Adresou je hash veřejného klíče, který je dodatečně upraven, tak aby se běžnému uživateli s adresou pracovalo lépe. V Bitcoinu se nejprve klíč zkracuje na 160 bitový hash, pak je převeden z binárních dat na řetězec alfanumerických znaků pomocí kódovací funkce Base58. Base58 je modifikace Base64, která eliminuje znaky 0,O a I,l, které jsou lehce zaměnitelné a pak znaky +,/.

Kódování Base58 s kontrolním součtem se nazývá Base58Check. Kontrolní součet má délku 4 bajty.

$$\text{publicKeyHash} := RIPEMD160(SHA256((Q_x))$$

$$\text{checksum} := SHA256(SHA256(\text{version} + \text{publicKeyHash}))$$

$$\text{adresa} := Base58(\text{version} + \text{publicKeyHash} + \text{checksum}[0][1][2][3])$$

Kapitola 2

Analýza digitálních měn

Peníze jsou jednou z nejstarších technologií lidstva a mnohokrát změnily svoji formu. Od barterové ekonomiky přešlo lidstvo k primitivním penězům, komoditním penězům a postupně dospělo k dnešní fiat měně. Peníze nikdy v minulosti nebyly stvořené pro primární použití v digitální formě, ačkoliv je dnes v peněžním oběhu více digitálních peněz než fyzických. Fiat peníze si do digitálního prostředí nesou značnou těžkopádnost a naopak nevyužívají možnosti, které jim technologie umožňují.

Peníze v počítačích nemusí být jen statické jednotky určující hodnotu, kterou diktují a kontrolují centrální instituce. Kryptoměny jsou programovatelné peníze, které umožňují zefektivnit transakce a snížit transakční náklady. Kryptoměny neznají hranice, oproti tomu tradiční měna neplatí za hranicemi jiného území. Snahy o vybudování digitálních peněz tu byly minimálně dvacet let před vytěžením prvního bitcoinového bloku. Tehdy se internet začal šířit do celého světa a začal vstupovat do všech oblastí našich životů a to také vedlo k potřebě mít spolehlivý platební systém, který zprostředkoval směnu finančních prostředků mezi účty v bance.

V této kapitole definuji, co jsou peníze a měna, do kterých kategorií patří kryptoměny, jaké platební systémy přecházely kryptoměnám a jaké vlastnosti by měl použitelný platební systém mít.

2.1 Druhy peněz a aktiv

Pro konkrétní srovnání kryptoměn s fiat penězi je jako zástupce kryptoměn použit Bitcoin, který má největší likviditu, kapitalizaci a komunitu.

2.1.1 Měna

Měna představuje určitou formu či druh (podmnožinu) peněz. Základní rozdíl mezi měnou a penězi spočívá v tom, že peníze jsou kategorií ekonomickou a měna kategorií právní. Status oficiální měny zatím Bitcoinu uznalo jen Ja-

ponsko. Naše legislativa řadí kryptoměny mezi virtuální měny a ze strany státních orgánů jsou považovány za nehmotný movitý majetek. Dle České národní banky *„některé kryptoměny (tzv. stablecoins, navázané na určitou měnu v poměru 1:1) mohou naplnit právní definici elektronických peněz“*[7].

2.1.1.1 Virtuální měna

V roce 2012 termín *virtuální měna* definovala Evropská centrální banka (ECB)[6] jako *typ neregulovaných, digitálních peněz, které jsou vydávány a obvykle kontrolovány jejich vývojáři, a jsou používány a přijímány mezi členy určité virtuální komunity.* V roce 2016 ECB doporučila vymezit *virtuální měny* tak, aby bylo jasné, že virtuální měny nejsou zákonnými měnami nebo penězi, které emitují státy.

2.1.1.2 Fiat měna

Jde o peníze, které emituje a kontroluje státní autorita a jejichž hodnota je stanovena zákonem. Její používání je vynucováno státem — také se jí říká měna s nuceným oběhem drahými kovy nekrytá. V České republice je touto měnou koruna česká. Za fiat měnu se považují i tzv. *elektronické peníze*, které Evropská komise vymezuje jako digitální ekvivalent „skutečných peněz“[7].

2.1.1.3 Cenné papíry

Podle rozhodnutí Komise pro kontrolu cenných papírů Spojených států (SEC) Bitcoin nesplňuje podmínky pro zařazení do kategorie cenný papír, tak jak je stanovuje Howeyův test. Tokeny kryptoměny vůbec nemusejí reprezentovat pouze peníze. Jen malá skupina kryptoměn slouží jako alternativní platidla (Payment Token). Velké množství obchodovaných kryptoměn jsou nechvalně proslulé Initial Coin Offering (ICO): tokeny nabízené společnostmi jako forma crowdfundingu. Další kategorií jsou Security Token Offering (STO) sloužící jako obdoba cenných papírů, se všemi regulačními dopady a kontrolami příslušných státních orgánů. Zakoupené STO tokeny bývají kryté buď majetkem, ziskem nebo výnosem společnosti, která je vydává. V ČR zatím není legislativa připravena na to, aby mohli být STO považovány za cenné papíry.

2.1.1.4 Stable coins a národní digitální měny

V současné době při manipulaci s bezhotovostními platbami pozorujeme rozdíl mezi tím, jestli nakládáme s *fiat měnou* nebo *virtuální měnou.* Stable coiny jsou speciální kryptoměny se stabilní hodnotou. Cena jejich tokenu je navázána

[6] https://www.ecb.europa.eu/pub/pdf/other/virtualcurrencyschemes201210en.pdf

[7] Fiat peníze nejsou ničím kryté. Neustále se tisknou nové bankovky a emitují nové peníze ve formě multiplikace prostřednictvím úvěrů v systému částečných rezerv.

na cenu určitého platidla či aktiva a díky tomu pak nevykazuje o moc vyšší volatilitu. Čím dál více vlád uvažuje o zavedení svých národních digitálních měn tzv. *Central bank digital currencies*. Mělo by jít o digitální měny spravované centrálními bankami.

2.1.2 Funkce peněz

Většina ekonomických učebnic rozlišuje tři základní funkce peněz. Peníze by měly sloužit jako prostředek směny, účetní jednotka a uchovatel hodnoty. Měny nesplňují všechny tyto funkce současně, vždy nastává kompromis mezi stabilitou a efektivitou. Na to, do jaké míry tyto funkce plní Bitcoin nemají jednotný názor ani ekonomové a svoje argumenty často opírají o to, jak ho nyní používají lidé. Národní měny mají výhodnější pozici v tom, že jejich akceptace je vynucená zákonem, vlády si je uznávají navzájem a konkurenční měny jsou v některých legislativách kriminalizovány. Bitcoin má potenciál se zlepšit v mnoha aspektech a už nyní v některých úlohách obstojí lépe, než tradiční peníze.

- Vlastnosti prostředku směny: za kryptoměny lze dnes koupit téměř vše, co za fiat peníze. Uživatelé chtějí posílat transakce levně a rychle. Tady v současné době trochu nastává problém, možnost škálování je často slabina současných kryptoměn.

- Schopnost plnit funkci uchovatele hodnoty se v průběhu historie u každého statku mění. Bitcoinu tuto vlastnost kazí zejména volatilita. Tržní kapitalizace má vliv na cenovou volatilitu. Čím bude tržní kapitalizace Bitcoinu vyšší, tím nižší výkyvy ceny lze očekávat.

- Kryptoměna jako účetní jednotka. Účetní jednotky v České republice mají povinnost v souladu se zákonem o účetnictví vést účetnictví v korunách českých. Národní banka nevypisuje kurz kryptoměn ve svém kurzovním lístku a firmy také musí platit daně v českých korunách. Společnosti jsou tak vystaveny měnovým rizikům z kurzových pohybů. Jednotky bitcoinu jsou dělitelné a nejmenší jednotce se říká satoshi (1 sat = 0,00000001 btc).

2.2 Historie virtuálních měn

2.2.1 Vlastnosti měny

V roce 1996 formulovali Okamoto a Ohta v článku[8] šest vlastností, kterými by měl disponovat ideální digitální platební systém.

1. Bezpečnost: peníze by neměly jít okopírovat nebo opakovaně utratit (problém dvojího utracení).

2. Nezávislost: peníze by měly existovat na počítačové síti a neměly by být závislé na fyzickém umístění (např. platební kartě).

3. Anonymita: ochrana soukromí uživatele. Nelze propojit zákazníka s jeho nákupem. Obchodník ani jiný uživatel by neměl mít šanci zjistit identitu zákazníka.

4. Okamžitá kontrola: uživatel nepotřebuje mít spojení s bankou (nebo jiným centrálním bodem). Platbu lze vykonat off-line.

5. Dělitelnost: mince digitálních peněz lze rozměňovat. Atomická jednotka měny je rozumně velká.

6. Směnitelnost: peníze si uživatelé mohou posílat mezi sebou navzájem.

Autoři popsali vylepšený platební systém, který podporovat dělitelnost jednotek. Kryptoměny tyto vlastnosti splňují.

2.2.2 DigiCash

V 90. letech společnost DigiCash Inc. provozovala eCash[9]: první funkční implementaci digitálních peněz. Zakladatelem společnosti byl David Chaum, který navrhnul slepé podpisy (sekce 1.3.3), které se používají v mnoha altcoinech a virtuálních měnách.

Peníze zde byly vytvořeny jako digitální certifikát (Digital Bearer Certificate). Software eCashe vygeneroval veřejný a soukromý klíč.

V tomto systému neexistovala databáze transakcí, pouze databáze sériových čísel reprezentující digitální certifikáty. Ty vydával server a využíval princip slepých podpisů. Převod digitálních certifikátů probíhal tak, že server starý certifikát zneplatnil a pro příjemce vydal nový.

Systém rozlišoval dva aktéry: prodejce a uživatele. Výhodou těchto peněz byla anonymita uživatelů, prodejci však anonymní nebyli. Systém eCash splňoval dobře první čtyři vlastnosti definované Okamotou a Ohtou. Jednotky měny byly nedělitelné a uživatelé si nemohli platit navzájem.

2.2.3 Bit gold

Nick Szabo navrhnul koncept chytrých kontraktů a v roce 1998 rovněž popsal fungování virtuální měny nazvané Bit gold[10]. Satoshi Nakamoto potvrdil, že implementace Bitcoinu se inspiruje návrhem Bit Goldu[8].

[8] https://satoshi.nakamotoinstitute.org/posts/bitcointalk/249/

2.2.4 Centralizace u privátních měn

Většina prvních virtuálních peněz nebyla špatná po technické stránce, ale mezi běžné lidi se příliš nerozšířily a když ano, byl tento pokus po zásluze potrestán.

Podobným pokusem, jako eCash byla digitální měna E-gold vytvořená v roce 2001. Umožňovala držet účet se zůstatky krytými gramy zlata. V roce 2008 službu využívaly miliony uživatelů a spravovala několik tun zlata. V důsledku boje proti terorismu v USA a přijetí zákona Patriot Act se systém dostal do hledáčku státních úřadů a zakladatelé byli obviněni z praní špinavých peněz a provozu směnárny bez licence.

Měna Liberty Dollar byla tvořená zlatými a stříbrnými mincemi, papírovými certifikáty a virtuálními jednotkami. Mince na sobě měly vyraženou nominální hodnotu v amerických dolarech. V roce 2011 byl zakladatel Bernard von NotHaus souzen za terorismus a padělání amerického dolaru, ačkoliv mince Liberty Dollaru měly větší hodnotu.

Podobný osud jako E-gold a Liberty Dollar potkal i soukromé měny E-Bullion, Liberty Reserve a další. V roce 2019 představila společnost Facebook svůj záměr vytvořit centrální virtuální měnu Libra. Kvůli regulatorním problémům však z projektu prozatím sešlo.

Ukázalo se, že nelze vytvořit konkurenci fiat penězům a centralizované projekty postupně selhaly. Až decentralizace mohla pomoci digitálním měnám na výsluní. Bylo potřeba vytvořit systém, který by ideálně nešel zničit útokem na jednu společnost a zároveň by nespoléhal na to, že nějaká entita bude jednat vždy čestně.

2.2.5 B-money a decentralizace

V roce 1998 Wei Dai navrhnul distribuovaný elektronický peněžní systém B-money[11], který by nebylo možné regulovat státními zásahy.

V prvním protokolu navrhuje, aby každý účastník spravoval vlastní databázi zůstatků uživatelů, dále navrhuje použít POW[9] mechanismus pro vytváření peněz. Takové peníze by mohl vytvářet kdokoliv, kdo vyřeší výpočetně náročný problém. Druhou variantou je druhý protokol, ve kterém by účetní zůstatky spravovala podmnožina účastníků, kteří by byly náhodně vybráni. Důležitým prvkem mělo být soukromí uživatelů platebního systému a využívání digitálních podpisů.

2.3 Transakce

Veškeré změny stavu blockchainu jsou realizovány prostřednictvím transakcí. V principu jsou dva druhy transakcí: *coinbase* a běžná transakce. *Coinbase transakce* je mincetvorná transakce, ve které se emitují nové mince, které slouží

[9] https://cypherpunks.venona.com/date/1998/12/msg00203.html

zároveň jako odměna těžařům/validátorům. V kryptoměnách se využívají dva typy modelů pro uchovávání záznamů transakcí: účetní model a UTXO model.

2.3.1 Účetní model

Jednodušší model, který koresponduje s podobou učetnictví jak ho známe. Každá transakce má podobu účetního zápisu ve formě má dáti, dal. To znamená, z jedné adresy jsou tokeny odečteny a na druhou adresu jsou přičítány. Pro systém to znamená držet si globální stav zůstatků na jednotlivých adresách. Transakce je validní, pokud je na účtu ze kterého odebíráme dostatek prostředků.

Má dáti (MD)	**Dal (D)**
z účtu	na účet
debit	kredit

2.3.2 Model UTXO

UTXO (Unspent Transaction Output – neutracené výstupy transakcí). UTXO jsou často nazývány mince nebo tokeny. Transakce obsahuje dva seznamy: vstupy a výstupy. Každý vstup je výstupem jiné transakce, která je obsažena v některém již dříve potvrzeném bloku. Hash transakce se nazývá *txId*. Vstupní utracené transakce jsou smazány ze seznamu UTXO a nové výstupy jsou vloženy do UTXO. Coinbase transakce obsahuje vždy jednu vstupní transakci, jejíž hash je nulový.

- Vstupy transakce (*txIn*) jsou označeny jako *utracené jednotky*.
- Výstup transakce (*txOut*) nesou *neutracené jednotky* určující adresu nového příjemce daných tokenů.

Funkční příklad UTXO modelu najdeme přímo v Bitcoinu, jehož transakce obsahují navíc i skripty chytrých kontraktů. Podoba transakce je popsána v tabulce 2.3.2.

- Kontrakt *scriptPubkey* obsahuje instrukce, po jejichž úspěšném provedení se uvolní prostředky.
- Podepisovací skript *scriptSig* jehož součástí je secp256k1 podpis a veřejný klíč.

Položka		Typ	Popis
#in		integer	počet vstupních transakcí.
in[]	hash	char[32]	txid dřívější transakce
	scriptSigLen	varint	délka scriptSig
	scriptSig	Script	podepisovací skript
	n	integer	index výstupu transakce
#out		integer	počet výstupních transakcí.
out[]	hodnota	integer	množství jednotek
	scriptPubKeyLen	varint	délka scriptPubKey
	scriptPubKey	Script	instrukce kontraktu

Tabulka 2.1: Struktura transakce v Bitcoinu

2.3.3 SegWit

Myšlenku Segwitu představil Pieter Wuille na konferenci Scaling Bitcoin pořádanou v Hongkongu v roce 2015[10]. Segwit je úprava protokolu, která řeší podpisy.

Tradičně je součástí vstupní transakce, ze které se počítá hash i podepisovací skript (scriptSig). Řešením Segwitu bylo oddělení digitálního podpisu od ostatních údajů transakce. V SegWitu jsou podpisy „segregovány“ mimo transakci ve vlastní struktuře a kořen Merkleova stromu s podpisy se uchovává v coinbase transakci. Původní scriptSig zůstává prázdný. Podpis obvykle tvoří větší část (až 65%) transakce a tak se dosáhlo i uspory místa.

SegWit tímto způsobem řeší zranitelnost *transaction malleability*. Jde o chybu v původním návrhu Bitcoinu, kdy ve scriptSig je možné udělat změny, které způsobí, že výsledný hash transakce je jiný a zároveň nový podpis zůstává stále platný.

Uživatelé Bitcoinu měli rozdílný názor na Segwit a další škálování Bitcoinu a proto proběhl 1. srpna 2017 významný hardfork. Zatímco většina komunity přijala SegWit, menšina šla cestou zvětšení velikosti bloku a z větve tohoto forku vznikla kryptoměna *Bitcoin Cash*.

2.4 Fiat měny a kryptoměny

Kryptoměny nemají otevírací dobu jako banky. V kontrastu s možnostmi dnešních technologií stále nemůžeme provést bankovní transakci o víkendu nebo o svátcích a na její zpracování čekáme i několik dní. Transakce fiat měny jsou obvykle zpracovány třemi způsoby:

- Uvnitř banky. Banka přepíše ve své databázi zůstatky svých klientů.

[10] https://hongkong2015.scalingbitcoin.org

Podle zákona[11] musí být příjemci připsaná do konce téhož (pracovního) dne.

- Mezibankovně. Při převodu peněz mezi dvěma různými bankami musí platba projít přes clearingové (zúčtovací) centrum. V České republice je to systém mezibankovního platebního styku CERTIS a vede ho Česká národní banka. Platba musí příjemci podle zákona dorazit do druhého pracovního dne.

- Mezinárodně. V roce 1973 vznikla mezinárodní platební síť SWIFT, která zpracovává přeshraniční platby. Provedení transakce trvá nejméně několik dní.

Většina dnešních platebních metod není dostupná například v zemích třetího světa a miliony lidí po celém světe nemají ani bankovní účet.

2.4.1 (Ne)zabavitelnost

Během 20. století lidé zažili mnoho inflačních a hyperinflačních epizod, nebo na ně dopadly měnové reformy (ČSSR 1953, Kypr 2013, ...). Lidé mohou skrz kryptoměny vlastnit své peníze a mít je plně pod kontrolou. ČNB připouští za *legitimní používání (kryptoměn) obyvatelstvem zemí, kde měnové autority a vlády selhávají v zajištění stabilní domácí měny*[7].

2.4.2 Anonymita

Problém Fiat peněz v on-line prostředí oproti hotovosti je především úplná ztráta anonymity (zákony Know Your Customer[12], Anti Money Laundering[13]). Od roku 2018 je navíc v EU de facto zrušeno bankovní tajemství.

Oproti platbě kartou, kde má zákazník prostor si nákup rozmyslet jsou transakce kryptoměn nevratné. Obchodníci musí být při platbě kartou obezřetní vůči svým zákazníkům a požadovat od nich více informací než je nezbytně nutné.

Kryptoměny nejsou automaticky anonymní, jsou tzv. *pseudonymní*. Všichni účastníci mohou vidět všechny proběhlé transakce v blockchainu a zjistit tak zůstatky na každé adrese. Pouze v některých altcoinech je obsah blockchainu skryt (např. v Moneru jsou záznamy transakcí šifrované dalším klíčem). Co nemusí být zjevné, je vlastnictví adresy. Pokud zabráníte propojení adresy s vaší identitou v reálném světě, pak jste anonymní. Propojení adres, mezi kterými proběhly transakce se často nazývá transakční graf. Na trhu jsou společnosti (např. Chainalysis nebo CipherTrace), které se věnují analýze blockchainu

[11] Lhůty pro provádění platebních transakcí upravuje Zákon o platebním styku č. 284/2009 Sb. § 109.

[12] Povinnost subjektu zjišťovat totožnost osob, se kterými vstupuje do obchodního vztahu.

[13] Zákon proti praní špinavých peněz, upravuje zákon č. 253/2008 Sb.

a vytváří tyto transakční grafy a snaží se rozkrýt, kdo si s kým vyměňoval tokeny. Poptávka po anonymitě je také velká a pracuje se i na rozšířeních Bitcoinu, které zajistí větší soukromí.

2.4.3 Zaměnitelnost (fungibility)

Zaměnitelnost představuje vlastnost peněz, která znamená, že každá jednotka dané měny má přesně stejnou hodnotu jako jakákoliv jiná jednotka stejného druhu. Blockchain uchovává veřejnou transakční historii. Některá platba může být spojena s nelegální nebo trestnou činností, což v konečném důsledku může snížit hodnotu těchto mincí. Tzn. za nově vytěženou minci budou ochotni lidé platit více. Stopování transakcí může být ztížené mixováním. Přes speciální peněženky lze vytvořit tzv. *CoinJoin* transakci, což je způsob kombinování transakcí od více uživatelů do jedné, takže je mnohem náročnější určit, kdo komu platil.

2.4.4 Škálování

Schopnost systému zvládnout požadavky většího počtu uživatelů. Do jednoho bloku se vejde omezený počet transakcí. Jeden blok se vytěží v průměru za 10 minut a velikost bloku je omezena na 1 MB a tím pádem zvládne síť zpracovat maximálně desítky transakcí za sekundu. V kontrastu s tím např. karetní společnosti Visa a MasterCard zpracovávají tisíce transakcí za sekundu. Škálování je v Bitcoinové komunitě velké téma: bezesporu existuje velká poptávka po mikrotransakcích, avšak Bitcoin není škálovatelný na úrovni jednotlivých transakcí a spíše se podobá clearingové vrstvě mezi bankami.

Díky SegWitu je nyní možné do jednoho bloku vložit více transakcí, nicméně do budoucna bude mít větší vliv na škálování implementace nových protokolů a off-chain řešení.

2.4.5 Off-chain a Lightning Network

Řešení off-chain (mimo blockchain) počítá s tím, že se do blockchainu nebudou zapisovat všechny platby. Část plateb bude probíhat ve druhé vrstvě a po určité době se stav zapečetí do blockchainu. Slibným příkladem je Lightning Network[14], který popisuje protokol pro platební kanály. Dvě strany mezi sebou otevřou platební kanál, průběžně si přes tento kanál posílají přepisy stavu účtu a na konci se platba vyrovná zápisem do blockchainu.

[14] http://lightning.network/lightning-network-paper.pdf

Kapitola 3

Správa Blockchainu

V předchozí kapitole jsem nastínila problematiku digitálních peněz a podobu transakcí, nyní se zaměřím na uložení transakcí. V kryptoměně potřebujeme uchovávat záznamy o tom, kolik bylo odesláno a přijato tokenů na kterou adresu. Tím se jeho úloha blockhainu podobá účetní knize. Namísto provozování jediné centralizované databáze účtů a zůstatků, která je spravována jednou institucí, jsou tyto záznamy replikovány mezi několik účastníků sítě tzv. *nodů* různé důvěryhodnosti. Každý tento node si udržuje svoji kopii účetní knihy a tím si i dovede odvodit zůstatky na jednotlivých adresách. Také dokáže zabránit dvojímu utracení a může kontrolovat jednotlivé transakce.

Satoshi Nakamoto jej propagoval slovy „*The proof-of-work chain is the solution to the synchronisation problem, and to knowing what the globally shared view is without having to trust anyone.*“ Blockchain nemusí uchovávat jenom transakce, ale může zapečetit libovolná data. Jeho původní využití se podobalo notářské službě.

3.1 Definice blockchainu

Definice blockchainu se v různé literatuře nepatrně liší. Někdo za blockchain považuje pouze datovou strukturu podobnou timestamp serveru (rozebráno níže v sekci), někdo jiný do definice zahrnuje celý ekosystém technologií včetně protokolu konsenzu, peer-to-peer komunikace atd. Než přejdeme k definici blockchainu, definujeme si dva příbuzné koncepty.

3.1.1 Distribuovaná účetní kniha

Z angličtiny *Distributed Ledger Technology* nebo zkráceně DLT, je databáze transakcí distribuovaná na více počítačích. Jediná povolená operace, je vkládání záznamů. Ne každá DLT má podobu blockchainu, některé mají podobu orientovaného acyklického grafu (DAG). Obecně je možné DLT rozdělit na veřejné a soukromé.

3.1.2 Timestamp server

V roce 1991 Stuart Haber a Scott Stornett ve svém článku [12] popsali Timestamp server, ve zkratce TSS. Tento systém měl sloužit jako základní schéma notářské služby pro ověřování digitálních dokumentů. Jeho hlavní úlohou bylo poskytnout informaci a zároveň důkaz o tom, kdy a v jaké podobě soubor existoval v daný čas. Jednotkou záznamu byl blok. Bloky se ukládaly v pořadí, v jakém se na server nahrávaly soubory a každý následující záznam byl propojen referencí na předešlý záznam, čímž se zachovávalo pořadí. Integrita se zajistila pomocí hashovací funkce. Tato struktura tvoří kostru blockchainu.

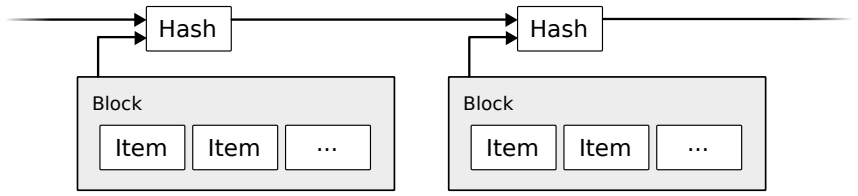
TSS funguje následujícím způsobem:

1. Klient pošle na TSS *dotaz* v podobě páru hodnot $d = (y_n, id_n)$, které představují

   - y_n: hash dokumentu, který chceme podepsat.
   - id_n: identifikátor uživatele

2. TSS pošle klientovi *odpověď* v podobě certifikátu $s := (C_n)$. Certifikát je uspořádaná 5-tice $C_n := (n, t_n, id_n, y_n, L_n)$

   - n: pořadové číslo dokumentu
   - t_n: časové razítko momentu, kdy server zaznamenal dokument
   - L_n: $\mathrm{H}(\mathrm{C_{n\text{-}1}})$ je hash předchozího vystaveného certifikátu - stane se tak digitální verzí pečeti.



Obrázek 3.1: TimeStamp server

3.1.3 Blockchain

Můžeme blockchain definovat jako decentralizovanou účetní knihu uchovávající nezměnitelné záznamy, které jsou kryptograficky chráněné.

Slovo „blockchain“ vzniklo spojením slov „block“ a „chain“, tedy v překladu řetězec bloků, což koresponduje s podobou zřetězených bloků. Základní strukturu si můžeme nejsnadněji představit jako spojový seznam. Ten je tvořen jednou nebo více položkami stejného typu – v tomto případě je jimi blok.

3.2 Datová struktura

Aby byla synchronizace transakcí mezi účastníky efektivnější, nepracujeme s jednotlivými transakcemi zvlášť, ale pracujeme s balíkem transakcí, které vkládáme do bloku.

3.2.1 Blok

Blok je základní stavební jednotka blockchainu. Každý blok je identifikovaný svým hashem a proto nelze jednotlivé bloky měnit. Blok s pořadím n si můžem představit jako uspořádanou čtveřici $B_n := (P_n, t_n, c_n, d_n)$:

- P_n, prev: $H(B_{n-1})$ hash pointer na předchozí blok.
- t_n, timestamp: časové razítko momentu, kdy byl blok vytvořen.
- c_n, consensus: slouží pro ověření konsenzu a validaci bloku.
- d_n, data: data nebo hash dat, které má blockchain zapečetit.

3.2.1.1 Prev

Bloky jsou navzájem propojeny pomocí *hash pointerů* (sekce 1.1.1), díky tomu nelze bloky libovolně přeskládat. Pokud začneme číst řetězec bloků od konce pomocí odkazu na předchozí blok, pak se iterativně dostaneme až k nultému bloku a tím projdeme celou historií. Blockchain svou strukturou zabezpečuje neměnnost dat a bloků, pokud uděláme libovolný zásah do historie, zneplatníme v tomto bodě všechny následující hash pointery a tím pomyslně přetrhneme řetěz bloků. Pořadí bloků je chronologické a neměnné a poskytuje nám představu o tom v jakém pořadí byla data do blockchainu přidávána.

3.2.1.2 Timestamp

Tato hodnota se nabízí k využití blockchainu pro další užitečnou činnost – notářskou službu. Aby blockchain fungoval jako tzv. Timestamp Server, stačí nám každý blok opatřit aktuálním časový razítkem. Tím dostaneme k dokumentu důkaz, že data existovala v době vytvoření bloku.

To ostatně Satoshi Nakamoto demonstroval už na genesis bloku, ve kterém naznačil svůj vztah k systému postaveném na bankovnictví s částečnými rezervami a do první transakce zapsal zprávu inspirovanou článkem v britském deníku The Times *„The Times 03/Jan/2009 Chancellor on brink of second*

bailout for banks“. The Times toho dne psal o tom, že britský ministr financí je jen krůček od druhého záchranného programu určeného bankám, které byly zasaženy probíhající světovou finanční krizí. Satoshi často dával najevo na on-line fórech (např. při oznamování nové verze Bitcoin Core[15]), že mu vadí fungování ekonomiky řízené centrálními bankami.

Na peer-to-peer síti dochází k latenci a vytěžit blok trvá těžařům různou dobu a proto hodnota všech časových razítek není nutně monotónní. Ověřuje, zda se nový blok nachází v platném intervalu. V Bitcoinu je jako validní čas brán interval větší než medián předchozích 11 bloků a menší než je aktuální čas na síti + 2 hodiny.

Důkazem jsou platné bloky v Bitcoinu:

- blok #180966 s časem 20. května 2012 23:02:53.
- blok #180967 s časem 20. května 2012 23:02:13.

3.2.1.3 Data

Většinou jde o hash pointer na složitější objekt nebo strukturu. Mohli bychom ukládat do blockchainu celé datové soubory, ale pokud nám jde primárně o ověření integrity nějakých dat, pak postačí hashe. Autoři TSS[12], oceňují vlastnost hashů reprezentovat libovolně velké soubory krátkými otisky. Díky tomu pro tento typ služby není nutné velké úložiště dat. Další výhodou ukládání hashů je, že pokud současně chceme naše soubory utajit, neriskujeme, že by mohl z TSS datový soubor uniknout.

3.2.1.4 Konsenzus

Konsenzus vzniká na základě vzájemného souhlasu uživatelů sítě. Součástí protokolu konsenzu je stanovení pravidel, za kterých se přidávají bloky do blockchainu. Pokud se řídíme konsenzem Proof of Work, pak do hlavičky bloku ukládáme ještě trojici parametrů: verzi, nonce a target.

- verze - verze pravidel, kterými byl blok validován.
- target - nalezený hash bloku musí být menší nebo roven targetu.
- nonce - zkratka *number used only once*. Změnou hodnoty těžař generuje různou hodnotu hashe bloku.

V Bitcoinu se pracuje s parametrem bits reprezentujícím hodnotu targetu ve 32-bitovém v kompaktním formátu místo 256 bitového hashe. Podobá se vědecké notaci, ale používá dvojkový základ místo desítkového.

$$N := (-1^{sign}) * mantissa * 256^{exponent-3}$$

[15] https://satoshi.nakamotoinstitute.org/quotes/banks/

Skládá se z exponentu, znaménkového bitu a mantisy. Hodnota obtížnosti bloku musí být stejná jako v předchozím bloku, výjimkou jsou bloky, jejichž výška je dělitelná 2016. Jeho reprezentace vychází z OpenSSL implementace ve funkcích `BN_bn2mpi()` a `BN_mpi2bn()`.

Existují varianty privátních blockchainů, které bývají buď přístupné vybrané skupině uživatelů, nebo přidávat bloky mohou jen pověření uživatelé. O pravidlech rozhoduje nějaká autorita nebo instituce. Tokeny se nevytváří těžbou a proto vznikly alternativní protokoly konsenzu, jejichž motivací byly zejména nižší nároky na výpočetní výkon a energetickou spotřebu. Příkladem může být Proof of Stake, kde má právo na vytvoření nového bloku *validátor*, který složí depozit a v závislosti na velikosti depozitu a případně i délce držení je oprávněn ověřovat transakce. Pokud validátor validuje nevalidní transakce, o peníze v depozitu přijde. V této práci se nebudu zabývat alternativními protokoly. V zásadě by jiný protokol neměl mít vliv na skládání bloků v blockchainu.

3.2.2 Data a transakce

V kryptoměně se za data považují transakce. Každá validní transakce provede změnu stavu neutracených tokenů v blockchainu, transakce je záznam o změně vlastnictví mezi dvěma stranami. Blok obsahuje záznamy proběhlých transakcí, které ještě dříve nebyly zapsány do blockchainu.

Transakce bloku musí být neprázdné a první transakcí musí být mincetvorná tzv. *coinbase transakce*, při které se vytváří nové mince (tokeny) měny. Hash kořene Merkleova stromu v bloku musí odpovídat kořenu, který lze vytvořit z transakcí. Hodnota transakcí na vstupu a výstupu se musí rovnat (těžaři si přebytky pošlou sami sobě jako poplatek).

3.2.3 Přidávání bloku

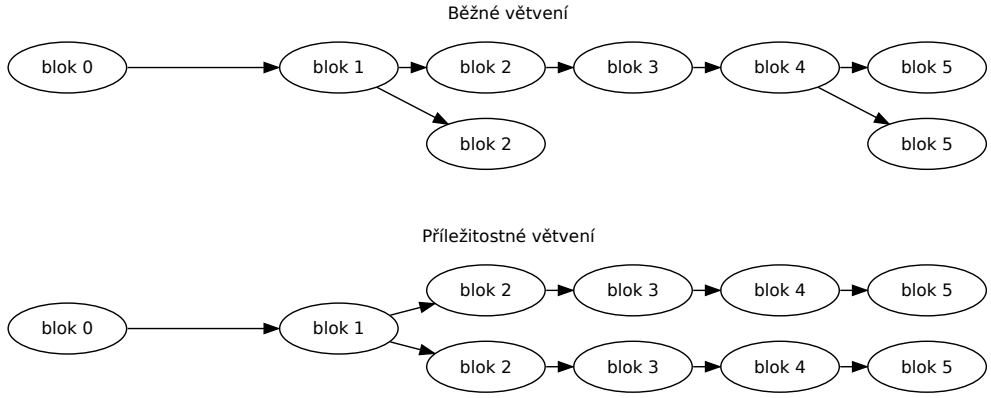
Pokud se řídíme konsenzem Proof of Work pak tyto bloky vytvářejí těžaři procesem „těžení“. Těžaři seskupují data do bloků: v případě transakcí vytvoří strukturu Merkleova stromu (sekce 1.1.3) a hodnotu kořene stromu zapečetí do bloku. Těžení spočívá ve vyřešení výpočetně náročné úlohy, aby nebylo snadné manipulovat s historií blockchainu. Těžaři nepálí výpočetní výkon pro dobrý pocit, ale jsou ekonomicky motivováni - za vytěžený blok dostávají odměnu a poplatky ze zahrnutých transakcí.

Bezprostředně po vytěžení validního bloku se tento blok rozešle ostatním účastníkům sítě. Aby byl nový blok blockchainu sítí přijat, musí splňovat dohodnutá pravidla konsenzu. Ostatní příjemci, kteří budou tento blok považovat za validní si jej uloží do své lokální databáze a následně může být zahrnut do blockchainu.

3.3 Větvení

Větvení znamená, že se v určitém bodě odloučíme od současné linie historie a pokračujeme v jiné. V této situaci může dojít k dvojímu utracení, proto se doporučuje považovat transakci za potvrzenou, až když je obsažena v dostatečné hloubce blockchainu. K větvení dochází přirozeně při těžení nových bloků.

V kryptoměně také může dojít k větvení z důvodu aktualizace protokolu, které se říká *hard fork*, při které dojde k takové změně pravidel protokolu, která není kompatibilní s větší částí sítě.



Obrázek 3.2: Větvení blokchainu

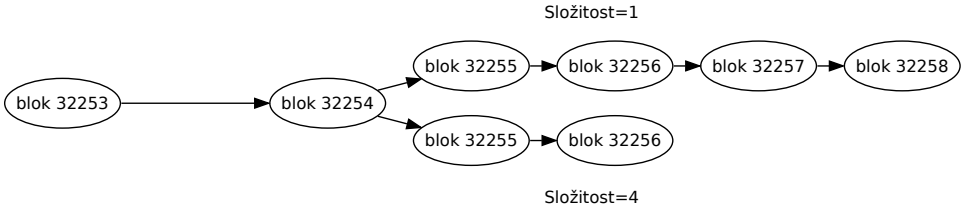
3.3.1 Hlavní větev

Jde o větev, která je krytá největší výpočetní silou, vyjádřenou množstvím spočítaných hashů. Pomocí targetu bloku lze spočítat minimální množství potřebných hashů pro nalezení tohoto bloku $2^{256}/(target + 1)$. Množství vykonané práce na větvi je pak součtem těchto hodnot.

$$\sum_{i=0}^{n} = \frac{2^{256}}{target_i + 1} \tag{3.1}$$

V původním článku[2] se operuje s termínem „nejdelší větev“ a dokonce v prvních verzích Bitcoin Core se výška bloku používá k určení hlavního řetězce s vírou, že jde o větev, kterou stálo „nejvíc úsilí“ vytvořit. Obtížnost těžení se však průběžně mění. Kontrola výšky větve nereflektuje problém, kdy by se v nedávné minulosti zformovala větev s nižší obtížností, na které by bylo snadnější vytvářet bloky a tak by mohla rychleji přerůst hlavní.

Od verze Bitcoin Core 0.3.3 vydané v červenci 2010[16] se ve zdrojovém kódu používá k určení hlavní větve množství práce, tzv. „chainwork“.



Obrázek 3.3: Větev s větší výškou a jinou složitostí

Aktuálně je součástí každé vydané verze Bitcoin Core i definice minimálního množství vykonané práce na hlavní větvi blockchainu, aby klient při stahování bloků ze sítě poznal, jestli je správně synchronizován.

3.3.2 Vedlejší větev

O bloky soutěží několik těžařů současně a tak se často stává, že se vytěží dva různé bloky ve stejnou chvíli a díky latenci na síti dojde ke krátkodobému rozvětvení. V danou chvíli existují dva bloky, které jsou validním kandidátem na další blok. Každý node si může individuálně rozhodnout, kterou větev bude následovat. Většinou zvolí tu, kterou ze sítě stáhnul dřív. Při zahrnování bloku do blockchainu mohou nastat tři případy:

1. blok rozšiřuje hlavní větev.
2. blok rozšiřuje vedlejší větev, která není krytá větší výpočetní silou.
3. blok prodlužuje vedlejší větev a ta se stává hlavní větví.

3.4 Správa blockchainu

V blockchainu je cílem udržovat hlavní řetězec. Samotný blok žádné informace, díky kterým by se dalo rozhodnout o tom, jestli ho lze zahrnout do blockchainu neposkytuje. To platí také pro validaci. Blok neustále ověřujeme a některá pravidla konsenzu nemůžeme ověřit dokud nemáme k dispozici celou historii.

Blockchain se v průběhu času rozvětvuje a tím se spíš podobá datové struktuře strom. Udržování hlavního řetězce se proto provádí ve dvou strukturách. První z nich je strom bloků a druhou je samotný hlavní řetězec. Každou cestu od kořene k listu nazýváme větví.

- Strom bloků: uchovává všechny validní bloky u kterých známe předka, až ke genesis bloku. Poskytuje nám informaci o výšce bloků a množství

[16] //github.com/bitcoin/bitcoin/commit/40cd0369419323f8d7385950e20342e998c994e1

vykonané práce na jednotlivých větvích. My pak určíme jednu větev za hlavní.

- Hlavní řetěz: jde o větev stromu bloků, která je krytá největší prací.

3.4.1 Validní blok

Jde o blok, který má

- Validní hlavičku, která současně splňuje konsenzus.
- Validní transakce. První v bloku je coinbase transakce, mezi ostatními se neobjevuje pokus o dvojí utracení. Neutrácí více mincí, než je na adrese k dispozici.
- Hash pointer na předka ukazuje na validní blok, který se nachází na hlavní větvi.

Poslední dvě vlastnosti často nejsme schopni ověřit hned, protože v danou chvíli nemáme k dispozici celý blockchain nebo předek bloku ještě nebyl stažen ze sítě.

Z tohoto důvodu, rozlišujeme čtyři typy bloků.

1. Validní blok. Platný blok, který je součást hlavního řetězce.
2. Neplatný blok. Nemá validní hlavičku, transakce a nebo nesplňuje konsenzus. Tento blok je zahozen.
3. Osamocený (osiřelý) blok. Blok u kterého neznáme jeho předky.
4. Zastaralý blok. Vytěžený platný blok, který není součástí hlavního řetězce.

3.4.2 Vytvoření blockchainu

Prvním blokem blockchainu je tzv. „genesis blok“, tento blok bývá zapsaný ve zdrojovém kódu. Je předkem všech bloků v blockchainu.

3.5 Bezpečnost blockchainu

3.5.1 51% útok

K tomuto útoku může dojít v situaci, kdy organizovaná skupina těžařů získá více než polovinu výpočetního výkonu (hash rate) celé sítě a mohou tak určovat, co je pravda. Pokud by těžba nebyla nákladná, bylo by snadné provést vrácení (rollback) řetězce. Nebezpečnost útoku tkví především v možném provedení dvojího utracení a ztráty důvěry v to, že je historie nezměnitelná.

Síť Bitcoinu je co do množství výpočetního výkonu nejbezpečnější mezi altcoiny a dosáhla výše, kterou v tuhle chvíli nedisponuje žádná jiná entita. K útoku jsou náchylnější altcoiny s menším objemem výpočetní síly[17].

3.5.2 Sybil útok

V tomto typu útoku vytváří útočník řadu identit, za které se vydává. Reputační systém je zranitelný vůči tomuto typu útoku v závislosti na tom, jak náročné je generování identit. Různé algoritmy konsenzu se s tímto typem útoku vypořádávají jinak. V POW systému se hlasuje o pravdě pomocí CPU a je jedno jestli vystupujete jako jeden node na síti nebo milion.

[17]Na stránce https://www.crypto51.app/ je přehledná tabulka odhadu ceny útoku pro jednotlivé kryptoměny

Kapitola 4

Síť kryptoměny

Kryptoměny fungují na peer-to-peer síti a tak jediný požadavek pro provoz kryptoměny je připojení k internetu a spuštění aplikace. V tomto případě je node/peer reprezentován běžícím programem, který komunikuje s ostatními nody podle protokolu dané kryptoměny. Síť kryptoměny je pak tvořena množinou těchto nodů. Každý node si může prohlédnout obsah blockchainu, potvrzovat transakce, vyměňovat si vzájemně zprávy.

V peer-to-peer síti si jsou jednotliví účastníci rovni, každý však může poskytovat jinou funkcionalitu. Řada kryptoměn má nejméně dva typy klientů: full-node a lightweight node. Full node si uchovává celou historii blockchainu. Stahuje každou transakci, kontroluje každý blok oproti pravidlům konsenzu. Lightweight node má omezenou funkcionalitu a získává informace od full-nodu. Fitcoin je zatím jen full-node klient.

Fitcoin je tvořen dvěma aplikacemi: `ftcd` (Fitcoin Deamon), který reprezentuje síťový node, který komunikuje pomocí zpráv s ostatními nody na síti. Druhou aplikací je utilita `ftctl` (Fitcoin Control), který odesílá přes socket příkazy `ftcd`.

4.1 Připojení k síti

Node se při spuštění potřebuje připojit k některému peerovi. Fitcoin Deamon naslouchá na portu 8888 a když se s námi některý peer pokusí navázat spojení, přidá se do seznamu peerů. Node se může dotázat svých peerů na adresy jiných peerů a rozšírit si tak „kontakty“.

Při spuštění deamona jsou načteny IP adresy ze souboru peers, který obsahuje adresy ostatních nodů. Do této databáze lze přidat další peery svépomocí příkazem

```
$ addpeer do ftctl
```

4.1.1 Výměna dat

Výměna dat mezi mezi účastníky sítě probíhá pomocí zpráv. Význam zprávy určuje typ.

Název	Typ	Velikost	Popis
type	uint32	4 bajty	typ zprávy
size	uint32	4 bajty	velikost zprávy
data	char[32]	n bajtů	obsah zprávy

Tabulka 4.1: Struktura zprávy ve Fitcoinu

Každý node si ověřuje příchozí bloky. Provádí posloupnost testů, předtím než si blok uloží do lokální databáze a odešle jej dalším svým známým peerům. To zajišťuje, že se sítí snadněji šíří platné bloky a nodům se vyplatí chovat čestně.

K výměně bloků a transakcí se využívají zprávy těchto typů:

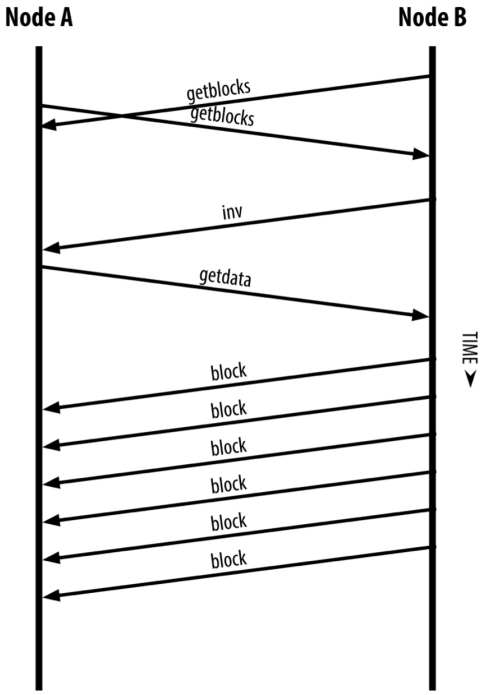
- `getblock`
- `gettx`
- `getindex`

4.1.2 Prvotní synchronizace

Při prvním spuštění klient spustí proces prvotní synchronizace (Initial Block Download). Předtím, než klient může validovat nepotvrzené transakce a nově vytěžené bloky, musí si stáhnout index bloků a všechny bloky blockchainu od genesis bloku po aktuální konec řetězce. Tyto bloky jsou uloženy na lokální uložiště. Tento proces může trvat dlouho, všechno závisí na rychlosti internetového připojení, disku a výkonu počítače.

Bitcoin používá dvě metody prvotní synchronizace:

- Blocks-First. Nový node pošle zprávu getblocks druhému nodu s hodnotou hashe z posledního bloku na hlavní větvi. Na to dostane odpověď typu `inv` se seznamem hashů následujících (maximálně 500) bloků v blockchainu adresáta. Node tuto odpověd využije a pošle mezi své peery požadavky zprávou `getdata`, kterými žádá o poslání celého bloku.
- Headers-First. Nejprve stáhne hlavičky blockchainu, které částečně validuje. Místo zprávy `inv` používá `getheaders`. Cílem je zjistit hash posledního bloku blockchainu. Následně pak žádá různé peery o stažení celých bloků včetně transakcí.



Obrázek 4.1: Synchronizace bloků [1]

Kapitola 5

Implementace Fitcoinu

V předchozích kapitolách byly představeny základní koncepty používané v kryptoměnách fungujících na konsensu Proof of Work. V této kapitole popíši konečnou implementaci blochainu, tak jak byl popsán v kapitole 3. Fitcoin je psaný v jazyku C.

5.1 Cizí knihovny a použitá kryptografie

Ve Fitcoinu používáme převážně standardní knihovny jazyka C. Jediná použitá externí knihovna, je kryptografická knihovna LibreSSL, alternativně OpenSSL. Obě knihovny jsou vzájemně kompatibilní a uživatel si může nalinkovat, kterou uzná za vhodnou.

Z kryptografické knihovny potřebujeme hlavně kryptografickou hashovací funkci a funkci pro asymetrické šifrování. Vybraná knihovna LibreSSL/OpenSSL nabízí hashovací funkce `HMAC`, `MD4`, `MD5`, `RIPEMD160`, `SHA1` a `SHA256`. Dnes ještě hojně používané funkce `MD5` ani `SHA1` nejsou považovány za bezpečné (ani jejich předchůdci). Pro Fitcoin byla vybrána funkce `SHA256` tvořící 256bitové hashe. Fitcoin používá pro digitální podpisy algoritmus `ECDSA` v konfiguraci secp256k1, popsanou v sekci 1.2.5.

Ve Fitcoinu je také potřeba se vypořádat s velkými čísly, která jsou až 256bitová. Na to nám nestačí standardní primitivní datové typy. Naštěstí knihovna OpenSSL/LibreSSL nabízí i dostatečnou podporu pro velká čísla (typ `BIGNUM`) a aritmetické operace na velkých číslech.

Adresa je ve Fitcoinu vytvořená méně sofistikovaným způsobem, než v Bitcoinu (sekce 1.4). Sestavuje se pouze ve dvou krocích: nejprve je potřeba vygenerovat soukromý a veřejný klíč pomocí eliptických křivek a následně adresou je hash tohoto veřejného klíče: adresa := $SHA256(Q_x)$. Uživatel si novou adresu vygeneruje příkazem:

```
$ echo "newaddr" | ./ftctl
```

5.2 Instalace

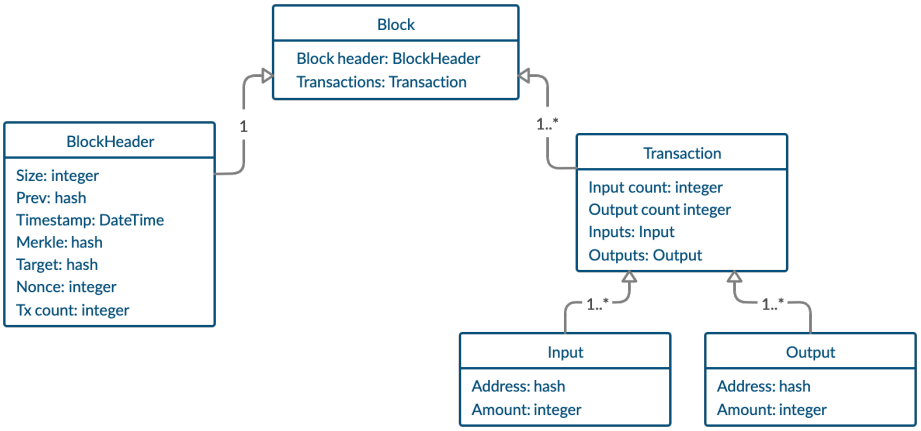
V řadě linuxových distribucí je standardně dostupný kompilátor jazyka C, linker a nástroj make. V macOS jsou tyto nástroje součástí Apple Developer Tools, takže je pravděpodobné, že uživatel nebude muset instalovat další vývojové prostředí.

Pokud máme nainstalovanou potřebnou kryptografickou knihovnu, můžeme přejít k samotné instalaci Fitcoinu. Proces kompilace je řízen utilitou Make. Instalace probíhá jednoduchým příkazem do terminálu:

```
$ make install
```

5.3 Architektura

Koncept kryptoměny se skládá ze dvou druhů záznamů: transakcí a bloků. Blok se skládá z hlavičky bloku a seznamu transakcí (model na obrázku 5.1).



Obrázek 5.1: Doménový model Fitcoinu

V této sekci popíši jak jsou implementovány konkrétní struktury blockchainu. Struktura bloku se zásadně neliší od podoby bloku popsaného v sekci 3.2.1, obsahuje navíc atributy `size` a `numtx`. Size značí celkovou velikost bloku spolu se všemi transakcemi bloku. Hlavička má sice pevnou velikost 112 bitů, ale počet transakcí (`numtx`) se liší a každá transakce má variabilní počet vstupů a výstupů. `Prev`, `merkle` a `target` jsou 256 bitové hashe z funkce SHA256 a jsou v bloku uloženy jako pole 32 znaků. Pro zaznamenání času se používá vyjádření v klasickém unix timestampu, který zaznamenává počet sekund od 1. ledna 1970 UTC. Blok je před uložením a posláním po síti vždy převeden na formát big-endian.

Název	Typ	Velikost	Popis
size	uint32	4 bajty	velikost bloku a transakcí
prev	unsigned char[32]	32 bajtů	hash předchozího bloku
merkle	unsigned char[32]	32 bajtů	hlavička merkle stromu
ts	uint32	4 bajty	unix timestamp
target	unsigned char[32]	32 bajtů	target limit
nonce	uint32	4 bajty	číslo validního bloku
numtx	uint32	4 bajty	počet transakcí
data	unsigned char[]		seznam transakcí

Tabulka 5.1: Struktura bloku ve Fitcoinu

Struktura transakce obsahuje informaci o své velikosti, počtech vstupů, výstupů a nakonec pole těchto vstupů a výstupů. Model transakcí ve fitcoinu odpovídá jednoduššímu účetnímu modelu (sekce 2.3.1), kde ze vstupů odebíráme mince a na výstupy přičítáme mince. Každý typ `io` nese dvojici informací: adresu a počet mincí.

Název	Typ	Velikost	Popis
size	uint32	4 bajty	velikost transakce
numi	uint8	1 bajt	počet vstupů transakce
numo	uint8	1 bajt	počet výstupů transakce
io	struct io[]	36*n bajty	io

Tabulka 5.2: Struktura transakce ve Fitcoinu

Název	Typ	Velikost	Popis
addr	unsigned char[32]	32 bajtů	adresa
amount	uint32	4 bajty	částka

Tabulka 5.3: Struktura io ve Fitcoinu

5.4 Správa bloků

Každý blockchain začíná genesis blokem, tento blok je ve Fitcoinu napevno zapsán ve zdrojovém kódu ve funkci `genesis()`. Pozorný čtenář může v této práci najít soukromý klíč k peněžence, obsahující mince vytěžené v genesis bloku Fitcoinu.

Abychom mohli zformovat hlavní větev blockchainu, potřebujeme nejdřív umět platné bloky zařadit do stromu bloků. Až u těchto bloků můžeme teprve rozhodnout, jestli je lze zahrnout do blockchainu.

5.4.1 Strom bloků a index

Vrcholy stromu bloků můžeme transformovat na index bloků. Index obsahuje metainformace o poloze bloků ve stromu. Těmito informacemi jsou výška (vzdálenost od genesis bloku), množství provedené práce na dané větvi a aktuální stav validace. Tyto informace mi pomáhají určit, která větev je hlavní. Obecně je index struktura sloužící k rychlému přístupu k datům podle klíče. V kryptoměně má dvě reprezentace.

- Fyzickou, kde meta informace jsou uloženy v key-value databázi. Klíčem je hash bloku a hodnotou je krátký blob metadat. Index se taky stará o uložení bloků na úložiště. V tuto chvíli se jednotlivé bloky ukládají na file systém, avšak drobnou modifikací funkcí `rblock()` a `wblock()` by se mohly bloky ukládat do libovolné databáze.

- Datovou, reprezentovaný hashovací tabulkou v paměti. Klíčem je opět hash bloku a hodnotou instance bloku. K nalezení jednotlivých instancí bloku slouží funkce `indexget()`.

Struktura indexu bloku vypadá následovně:

Název	**Typ**	**Velikost**	**Popis**
hash	unsigned char[32]	32 bajtů	hash bloku
height	uint32	4 bajty	výška
valid	uint32	4 bajty	validace
work	long double	10 bajtů	množství práce
bindex	struct bindex*	4-8 bajty	pointer na předka
blk	struct block*	4-8 bajty	pointer na blok

Tabulka 5.4: Struktura indexu bloku ve Fitcoinu

Bloky se do indexu dostanou třemi způsoby:

- Načtením z disku. Voláním funkce `indexload()` se obnoví z databáze stav bloků, jaký byl při posledním běhu programu.

- Vytěžením. Blockchain nejprve zkontroluje hlavičku bloku spolu s transakcemi, následně tento blok zařadí do indexu voláním `indexadd()`. V ideálním případě bude těžař prodlužovat pouze hlavní větev.

- Stažením ze sítě. Stejně jako v případě vytěženého bloku se volá funkce `indexadd()`, která se po kontrole bloku pokusí blok napojit do stromu. Může se stát, že předka bloku ještě neznáme, v tomto případě zařadím blok do seznamu osiřelých bloků.

5.4.2 Hlavní větev

Hlavní větev je větev stromu bloků, která má největší hodnotu vykonané práce. Hlavní větev je ve Fitcoinu implementovaná jako pole blokindexů pojmenovanou `chain`. Index pole odpovídá výšce bloku tzn. genesis je jako `chain[0]`.

- `chainadd()` validuje blok, zařazuje bloky do indexu. Zaúčtovává transakce a přepíná hlavní větev.
- `chainrun()` inicializuje hlavní větev blockchainu.
- `orphansadd()` seznam osiřelých bloků.
- `findfork()` pomocná funkce k nalezení nejbližšího předka, který se nachází na blockchainu. Využívá se při reorganizaci bloků, když měníme hlavní větev. Nejprve je potřeba odpojit bloky, které se nenachází na nové hlavní větvi a pak připojit bloky nové hlavní větve.

5.5 Těžba

Do Fitcoinu jsem zaintegrovala funkci těžaře od Filipa Volfa a doplnila jsem ho o funkce které jsou potřeba v blockchainu. Emise bloků je nastavitelná konstantami TARGETTIMESPAN a TARGETWINDOW. První z nich je doba v sekundách, po kterou platí konkrétní obtížnost těžení. Druhou je počet bloků, které se během tohoto období má vytěžit. Výchozí rychlost těžení nového bloku je v průměru pět minut. Funkce těžaře mající vliv na blockchain jsou následující:

- `mineblock()` vytváří blok z daného seznamu transakcí. Rozšiřuje aktuálně známou hlavní větev vytvořením validního bloku hledáním vhodné nonce. Návratovou hodnotou je pak nalezený blok. Funkce volající `mineblock()` odešle blok peerům a předá ho blockchainu voláním funkce `chainadd()`. Blockchain si nejprve zkontroluje hlavičku a transakce a pak blok zařadí do stromu bloků. Následně provede zaúčtovaní transakcí. Pokud vše proběhne korektně, aktualizuje hlavní větev.
- `blockproof()` vrací hodnotu provedené práce na bloku k určení hlavní větve podle rovnice (3.1).
- `adjusttarget()` vypočítá další hodnotu target podle rovnice (1.3)
- `validtarget()` kontroluje pravidlo, zda targetu bloku je menší než hash.
- `settarget()` nastaví hodnotu targetu aktuálního bloku podle targetu předchozího bloku.

5.6 Fyzická vrstva

Velikost blockchainu v čase narůstá a dřív nebo později přeroste velikost operační paměti. Proto je nezbytné ukládat bloky na pevný disk. Při prvním spuštění daemon ftcd vytvoří složku `/.ftc` v domovském adresáři. Struktura je následující:

- `chain` ..... Bloky blockchainu
  - `index` ..... Index databáze
- `coins` ..... Stav mincí na jednotlivých adresách
- `fresh` ..... Transakce, které nebyly vytěženy
- `wallet` ..... Soukromé klíče k adresám
- `peers` ..... IP adresy peerů

5.6.1 Index bloků

Obsah indexu je uložen v adresáři chain. Databáze udržuje meta informace o každém bloku blockchainu a při spuštění klienta je načte do paměti. Prvních 32 bajtů v indexu je rezervováno pro hodnotou hashe bloku, který určuje list posledně známé hlavní větve.

V Bitcoinu index obsahuje navíc informaci o umístění bloků na filesystému. Ve Fitcoinu tohle zatím není nutné, protože k nalezení souboru bloku nám stačí pouze znalost hashe bloku a adresáře ve kterém jsou bloky umístěny. Bitcoin seskupuje několik bloků do jednoho souboru blk*.dat a do indexu si zapíše název souboru ve kterém který blok leží a jeho pozici v tomto souboru. Tento způsob je šetrnější pro souborový systém (nezabíráme tolik bloků, inodů).

Původní Bitcoin Core do verze 0.8 využíval pro index databázi Berkeley DB vyvinutou společností Oracle. Ta se ukázala nevhodná pro využití v blockchainu kryptoměny z několika důvodů: pomalu zapisuje větší dávky dat, pomalu přistupuje k záznamům a má složitou konfiguraci. V datové struktuře b-tree používané BerkeleyDB jsou potřeba 2 zámky při aktualizaci indexového záznamu. Databáze také vyžaduje, aby uživatel nastavil limit[18] na maximální počet zámků, který může být současně proveden. Pokud je hodnota příliš malá, dotaz selže. Pokud je hodnota velká, uzamykací subsystém spotřebuje více zdrojů, než je nutné. Bitcoin Core do verze 0.7 měl nastaven limit na 10 000 a pokud vytěžený blok obsahoval větší množství transakcí, nebyl takový klient schopen zpracovat a akceptovat jinak validní blok.

Nyní se používá v Bitcoinu LevelDB od Googlu. LevelDB žádnou podobnou restrikci nemá a navíc je i rychlejší pro často využívané operace. Během migrace[19] z Berkeley DB na LevelDB došlo k jednomu z nejdelších rozvětvení blockchainu v historii, kdy část sítě se starší verzí klienta nebyla schopna akceptovat blok číslo 225430, který obsahoval transakce, které překračovaly limit zámku a zbytek sítě formoval několik dalších hodin větev s problema-

[18] https://web.stanford.edu/class/cs276a/projects/docs/berkeleydb/ref/lock/max.html
[19] https://github.com/bitcoin/bips/blob/master/bip-0050.mediawikiroot-cause

tickým blokem. Nakonec se vývojáři a těžaři s majoritou výpočetní síly rozhodli dočasně pozastavit těžbu na klientech s novější verzí a nechali vyhrát větev, která byla kompatibilní se staršími klienty.

Kapitola 6

Testování

Testování je nedílnou a stále důležitější součástí vývoje softwaru. Ověřuje zda softwarový produkt odpovídá definovaným požadavkům.

6.1 Scénáře

Blockchain by měl zvládnout následující scénáře:

6.1.1 Spuštění

Pokud není možné provést následující operace, blockchain se nespustí. Problém bude na straně uživatele, který nám nedopřál práva k vytvoření adresářů a souborů.

- Vytvoření potřebných adresářů: `./chain` a `./coins`, pokud ještě neexistují.
- Vytvoří se soubor databáze indexu.
- Zapíše se genesis blok do adresáře s bloky.

6.1.2 Synchronizace

Node se dotazuje ostatních peerů, na jejich stav blockchainu a pokusí se od nich stáhnout chybějící bloky (sekce 4.1.2).

- `indexadd()` zařazuje bloky do indexu. Kontroluje předka bloku.
- `addchain()` přijímá bloky a pomocí funkce `indexadd()` je přidává do stromu bloků. Pokud blok zařadit nejde, je zařazen do seznamu sirotků.

6.1.3 Obnovení

Není žádoucí abychom po každém vypnutí a zapnutí aplikace prováděli synchronizaci znovu.

- `chainrun()` dokáže obnovit stav jaký byl před vypnutím aplikace.

6.1.4 Akceptace bloků

Tato situace se příliš neliší od synchronizace, jen se v tomto scénáři počítá se situací, že se bude často měnit hlavní větev a blockchain musí umět mezi větvemi přepínat.

- `chainadd()` zajišťuje přepínaní na větev s největší odvedenou prací, zaúčtování transakcí a jejich vrácení.

6.2 Unit testy

Spolu s blockchainem jsou k dispozici unit testy pro hashovací tabulku, index bloků, blockchain a provedení transakcí.

- test-hashtable.c
- test-block-index.c
- test-chain.c
- test-block-validation.c

6.3 Testování platforem

Fitcoin byl ručně testován na různých platformách. Testování se zaměřovalo na to, zda se aplikace zkompiluje, spustí, zda funguje korektně a zda vytvořené soubory fungují stejně na architektuře big endian a little endian. Fitcoin byl testován na následujících operačních systémech:

- MacOSX, x86_64, Little endian
- GalliumOS, x86_64, Little endian
- Xubuntu, amd64, Little endian
- OpenBSD, amd64, Little endian
- OpenBSD, macppc, Big endian
- OpenBSD, armv7, Big endian

Proces kompilace nebyl vždy bez problémový. Problémy způsobovali např. starší verze OpenSSL, odlišné definice funkcí v hlavičkových souborech.

Testovala jsem na různých platformách podobu uloženého genesis bloku a podobal se na všech platformách stejně jako je uvedeno níže:

```
$ hexdump -C ~/.ftc/chain/000000b5f003360555043affb39bd
1c513321169d92a64d779aed1c2100d5663

00000000  00 00 00 bc 00 00 00 00  00 00 00 00 00 00 00 00
00000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000020  00 00 00 00 5e b4 a4 e8  06 6e 35 db 00 00 00 ff
00000030  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff
00000040  ff ff ff ff ff ff ff ff  ff ff 00 00 cd 5b 66 44
00000050  10 9e 6a fc ac ba c9 6f  10 67 de eb 81 2d 54 67
00000060  ff 49 ee 68 c7 94 c4 96  01 43 38 b1 00 00 00 01
00000070  00 4c 01 01 00 00 00 00  00 00 00 00 00 00 00 00
00000080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000090  ff ff ff ff 00 00 04 00  53 ac 51 1e 7f c6 19 4c
000000a0  5d 18 67 7d aa 1f 7d c2  f5 ac 9f 68 69 b4 3a 0a
000000b0  03 e4 8e 78 23 c4 73 06  00 00 04 00
000000bc
```

Závěr

Cílem této práce bylo seznámit se s konceptem kryptoměn a navrhnout funkční implementaci blockchainu pro Fitcoin. To se doufám do velké míry povedlo naplnit. Implementovala jsem blockchain pro Fitcoin. Nepodařilo se otestovat peer-to-peer synchronizaci blockchainu mezi uživateli, nicméně blockchain je schopen přijmout libovolný blok voláním funkce `chainadd()` ať je volána odkudkoliv.

Kryptoměny jsou široké téma, které nabízí spoustu výzev a také poukazují na nedostatky finančního systému. Detailně jsem prozkoumala implementaci Bitcoinu a jeho zdrojový kód byl pro mě hlavní zdroj informací, neboť podrobnější informace o fungování Bitcoinového blockchainu je obtížné dohledat. Velké množství konceptů v kryptoměnách se dá rozvést do mnoha širších témat a bylo pro mě chvílemi náročné držet se pouze blockchainu.

Tato práce může programátorovi usnadnit orientaci v technických konceptech kryptoměn a zejména pomoci mu pochopit fungování blockchainu.

Literatura

[1] Antonopoulos, A. *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*. O'Reilly Media, 2017, ISBN 9781491954386.

[2] Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008, přístupné: 2020-02-20. Available from: `https://bitcoin.org/bitcoin.pdf`

[3] Back, A. Hashcash - A Denial of Service Counter-Measure. 2002, přístupné: 2020-05-01. Available from: `http://www.hashcash.org/papers/hashcash.pdf`

[4] Oulehla, M.; Jašek, R. *Moderní kryptografie*. IFP Publishing s.r.o., 2017, ISBN 9788087383674.

[5] Recommended Elliptic Curve Domain Parameters. 2010, přístupné: 2020-05-01. Available from: `http://www.secg.org/sec2-v2.pdf`

[6] Chaum, D. Blind Signatures for Untraceable Payments. In *Advances in Cryptology*, edited by D. Chaum; A. T. Rivest, Ronald L. nd Sherman, Springer US, 1983, pp. 199–203.

[7] Belling, V. Úloha krypto„měn“, hlavní funkce, historie, budoucnost, využití, investice, rizika v ČR i ve světě. 2019, přístupné: 2020-05-25. Available from: `https://www.cnb.cz/export/sites/cnb/cs/verejnost/.galleries/pro_media/konference_projevy/vystoupeni_projevy/download/beling_20190307_praha.pdf`

[8] Watanabe, O.; Yamashita, O. An improvement of the digital cash protocol of Okamoto and Ohta. In *Algorithms and Computation*, edited by T. Asano; Y. Igarashi; H. Nagamochi; S. Miyano; S. Suri, Springer Berlin Heidelberg, 1996, pp. 436–445, doi:10.1007/BFb0009520.

[9] Chaum, D.; Fiat, A.; et al. Untraceable Electronic Cash. In *Advances in Cryptology — CRYPTO' 88*, 1990, pp. 319–327, doi:10.1007/0-387-34799-2_25.

[10] Szabo, N. Bit gold. 2005, přístupné: 2020-04-14. Available from: `http://unenumerated.blogspot.com/2005/12/bit-gold.html`

[11] Dai, W. B-Money-an anonymous, distributed electronic cash system. 1998, přístupné: 2020-05-01. Available from: `http://www.weidai.com/bmoney.txt`

[12] Haber, S.; Stornetta, W. S. How to time-stamp a digital document. *Journal of Cryptology*, 1991: pp. 99–111, doi:10.1007/BF00196791.

Acronyms

POW Proof of Work

TTS Timestamp server

DLT Distributed Ledger Technology

P2P Peer-to-peer síť

OSS Open source software

BTC Bitcoin

ČNB Česká národní banka

ECB Evropská centrální banka

ASIC Integrovaný obvod navržený a vyráběný pro specifickou aplikaci

FPGA Programovatelné hradlové pole

Obsah souborů na přiloženém médiu

```
readme.md ..... soubor s popisem obsahu
fitcoin.zip ..... soubor archívu s Fitcoinem
src ..... adresář se soubory kódu latexu
    thesis ..... adresář se zdrojovým kódem práce LaTeX
text ..... adresář se závěrečnou prací
    thesis.pdf ..... závěrečná práce v PDF formatu
```