VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2021 Veronika Mlýnková

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

VIRTUÁLNÍ MĚNY
VIRTUAL CURRENCIES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

Veronika Mlýnková

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. Václav Zeman, Ph.D.

BRNO 2021

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

Studentka: Veronika Mlýnková ***ID:*** 202682

Ročník: 3 ***Akademický rok:*** 2020/21

NÁZEV TÉMATU:

Virtuální měny

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je poskytnout přehledný a komplexní obraz o virtuálních měnách založených na technologii „blockchain“. Na základě uvedeného přehledu navrhněte a realizujte výukovou aplikaci, která bude demonstrovat funkci „blockchainu“ a jejího využití pro virtuální měny. Výuková aplikace bude obsahovat vzorovou laboratorní úlohu zaměřenou na vytvoření vlastního „blockchainu“ a systému pro jeho využití v kryptoměnách.

DOPORUČENÁ LITERATURA:

[1] MOUGAYAR, William. The business blockchain: promise, practice, and application of the next Internet technology. Hoboken, New Jersey: John Wiley & Sons, 2016.

[2] TAPSCOTT, Don, Alex TAPSCOTT a Jeff CUMMINGS. Blockchain revolution: [how the technology behind bitcoin is changing money, business, and the world]. Unabridged. Grand Haven, MI: Brilliance Audio, 2016. ISBN 1511357665.

Termín zadání: 1.2.2021 ***Termín odevzdání:*** 31.5.2021

Vedoucí práce: doc. Ing. Václav Zeman, Ph.D.

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

Abstrakt

Tato bakalářská práce se zabývá virtuálními měnami, které v základě využívají technologii blockchain. Základy způsobu fungování jsou vysvětleny na kryptoměně bitcoin, která je z technologického hlediska rozebrána detailněji. Kromě obecného fungování bitcoinové sítě je pak blíže popsána například tvorba bitcoinových adres, složení transakcí a způsob jejich podepisování. Širší pohled na kryptoměny je obsažen ve čtvrté kapitole, kde je uvedena jejich klasifikace do generací a vysvětleny pojmy coin, token a stable coin. V kapitole Ethereum je pak vysvětlena funkce smart kontraktů a decentralizovaných aplikací. V rámci práce byla realizována laboratorní úloha, kde je popsán postup k naprogramování zjednodušeného blockchainu v jazyce Python. Pro podporu pochopení tématu byla dále vytvořena výuková aplikace formou webové stránky vytvořené na platformě Google sites, která shrnuje rozebrané téma.

Klíčová slova

Bitcoin, blockchain, kryptoměny, Ethereum, konsensuální algoritmy

Abstract

This bachelor thesis deals with the topic of virtual currencies based on the blockchain technology. The foundations of the cryptocurrency workings are explained on cryptocurrency Bitcoin, which is later described in a greater detail from technological point of view. In addition to the general workings of a bitcoin network, this thesis describes creation of bitcoin addresses, the composition of transactions and the way they are sign and verified. A broader view of cryptocurrencies is offered within the fourth chapter, with the classification into generations and explanation of terms coin, token and stable coin. Chapter Ethereum explains the meaning of smart contracts and decentralized applications. Within the thesis a laboratory task was performed, which describes the procedure for programming a simplified version of blockchain in Python. To summarize the discussed topic, an educational web application was created.

Keywords

Bitcoin, blockchain, cryptocurrencies, Ethereum, consensus algorithm

Bibliografická citace

MLÝNKOVÁ, Veronika. Virtuální měny [online]. Brno, 2021 [cit. 2021-05-16]. Dostupné z: https://www.vutbr.cz/studenti/zav-prace/detail/134508. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Václav Zeman.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta:	*Veronika Mlýnková*
VUT ID studenta:	*202682*
Typ práce:	*Bakalářská práce*
Akademický rok:	*2020/21*
Téma závěrečné práce:	*Virtuální měny*

Prohlašuji, že svou bakalářskou práci na téma Virtuální měny jsem vypracovala samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autorka uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

.

V Brně dne: 16. května 2021 ----------

podpis autora

Poděkování

Děkuji svému vedoucímu bakalářské práce panu doc. Ing. Václavovi Zemanovi Ph.D. za odborné vedení, konzultace a cenné poznámky při vypracovávání mé bakalářské práce.

V Brně dne: 16. května 2021 ----------

podpis autora

Obsah

SEZNAM ZKRATEK

Zkratky:

ASCII	American Standard Code for Information Interchange
ASIC	Application Specific Integrated Circuit
BTC	bitcoin
CSS	Cascading Style Sheets
DApps	Decentralized Applications
ECDSA	Eliptic Curve Digital Signature Algorithm
ERC	Ethereum Request for Comment
FPGA	Field Programmable Gate Array
HTML	Hypertext Markup Language
ICO	Initial Coin Offering
NOP	no operation
RIPEMD	RIPE Message Digest
Segwit	Segregated Witness
SHA	Secure Hash Algorithm
SPV	Significant Payment Verification
URL	Uniform Resource Locator

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

Úvod

Za posledních 10 let nastal ve světě kryptoměn obrovský průlom, který započal zveřejněním bitcoin protokolu, jakožto vůbec první plně decentralizované a transparentní virtuální měny. Veškeré předchozí pokusy o realizaci digitálních peněz různých konceptů dopadly neúspěchem, ať už z důvodu jejich krachu, zpronevěrou od samotných provozovatelů, nebo zásahem vlády, která z různých důvodů jejich tvůrce trestně stíhala. Všechny tyto měny totiž postrádaly základní podstatu úspěchu takového systému, a to právě jeho decentralizovanost a plnou transparentnost. Ve chvíli, kdy existovala jedna hlavní autorita spravující tento systém, bylo jednoduché ji napadnout. V tomto ohledu přinesl bitcoinový protokol na pole kryptoměn zcela inovativní řešení. Po něm pak následovaly další virtuální měny více či méně Bitcoinem inspirované, či s ním patrně korelující.

Tato bakalářská práce se bude zabývat technologií blockchain, podrobněji kryptoměnou Bitcoin, jakožto první kryptoměny využívající tuto technologii. Dále pak bude souhrnně pojednáno o dalších kryptoměnách, bude uvedeno třídění kryptoměn do jednotlivých generací, vytyčení pojmů jako coin, token, či stablecoin, které se v tomto odvětví hojně užívají. V praktické části práce bude realizována laboratorní úloha, kde bude popsán postup k naprogramování zjednodušeného blockchainu v jazyce Python. Pro lepší pochopení problematiky bude vytvořena výuková aplikace formou webové stránky.

V první kapitole budou pro lepší pochopení zbytku práce uvedeny a vysvětleny některé z mechanismů a principů použitých v bitcoinovém protokolu. Druhá kapitola se zaobírá samotnou technologií blockchain, principem jeho tvorby a nejpoužívanějšími typy konsensuálních algoritmů. Třetí kapitola se již věnuje samotné kryptoměně Bitcoin, kde na začátku krátce shrnuje celkový princip fungování bitcoinové sítě a následně je v jednotlivých kapitolách tento princip detailněji rozepsán. Čtvrtá kapitola pojednává o kryptoměnách v komplexnějším měřítku a poskytuje ucelenější přehled. Najdeme zde například rozdělení kryptoměn do jednotlivých generací, nebo s kryptoměnami často skloňované pojmy coin, token, či stablecoin. V páté kapitole s názvem Ethereum jsou pak vysvětleny často skloňované pojmy – smart kontrakty a decentralizované aplikace. V šesté kapitole je konceptuálně popsána výuková aplikace, která byla pro účely této bakalářské práce vytvořena na platformě Google sites. Nakonec následuje laboratorní úloha, kde je krok po kroku popsán způsob tvorby zjednodušeného blockchainu v jazyce Python.

1. VYMEZENÍ POJMŮ

Systém kryptoměn a blockchain sám o sobě využívá ve svých implementacích řadu kryptografických funkcí a protokolů. Na začátek tedy bude uveden základní popis nejdůležitějších z nich.

1.1 Asymetrická kryptografie

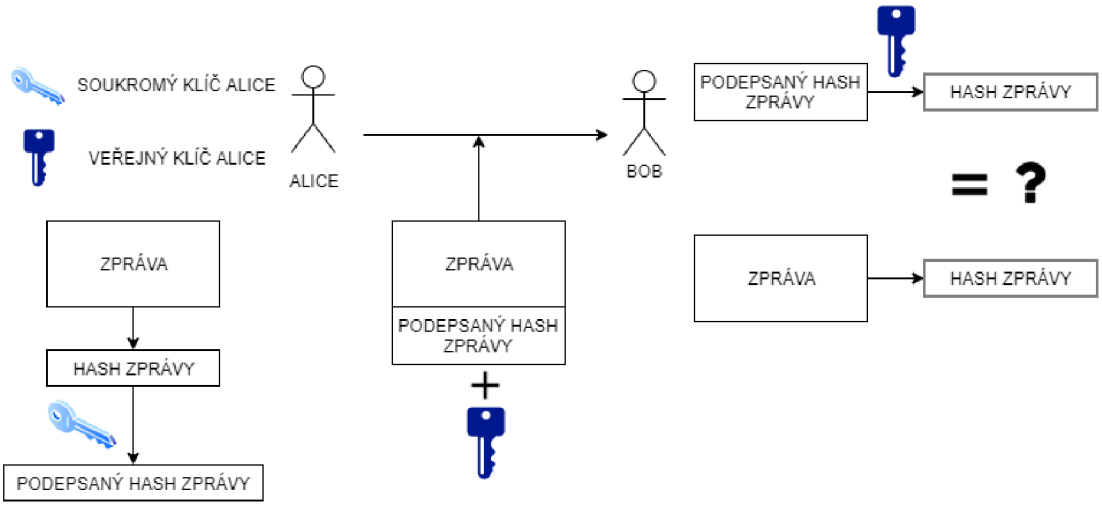
Kryptografie je věda zabývající se šifrováním, tedy převodem čitelných dat (data, která je počítač schopný zpracovat, případně převést na pro člověka čitelný text) na jejich nečitelnou šifrovanou podobu a opačně, za využití kryptografických klíčů. Kryptografie se dělí na dvě základní skupiny, a to symetrickou a asymetrickou. Symetrická kryptografie používá pro šifrování i dešifrování jeden a ten samý klíč, který je sdílen oběma stranami, zatímco asymetrická využívá dva rozdílné klíče, soukromý a veřejný, z nichž je pouze klíč veřejný, sloužící k šifrování a ověřování podpisů, známý oběma stranám. Na asymetrické kryptografii je postaven právě stěžejní mechanismus pro ověřování transakcí v blockchainu, a to digitální podpis. V kryptografii se pro vysvětlování principů ustálilo používání jmen Alice a Bob [1].

1.2 Digitální podpis

Bezpečnost transakcí v bitcoinové síti spočívá ve složitosti asymetrické kryptografie, konkrétně v technologii digitálních podpisů.

Každý majitel účtu disponuje párem unikátních kryptografických klíčů, veřejným a soukromým. Zatímco soukromý klíč slouží k podepisování transakcí a zná jej pouze vlastník, klíč veřejný se sdílí do sítě a slouží k ověřování autentičnosti transakce. Klíče jsou v podstatě dlouhá čísla, která na sobě matematicky závisí. Samotný proces podepisování pak vypadá následovně:

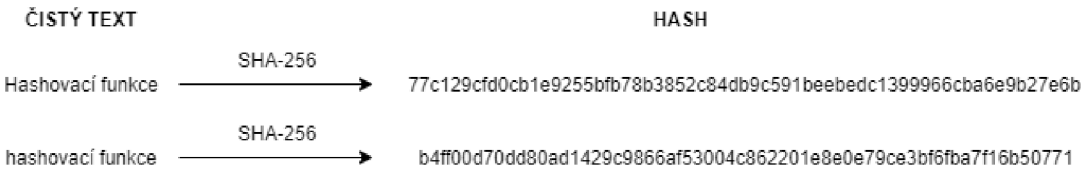
Alice vytvoří ze zprávy, kterou chce podepsat její hash, který následně podepíše svým soukromým klíčem. Podpisem se změní řetězec hashe zprávy. Takto podepsaný hash, spolu se zprávou a svým veřejným klíčem odešle Bobovi. Poté co Bob data obdrží, aplikuje na Alicí podepsaný hash zprávy veřejný klíč Alice a dostane tak hash zprávy původní. Poté vezme samotnou zprávu a aplikuje na ní stejnou hashovací funkci. Oba hashe poté porovná. Pokud se hashe rovnají, je to známka toho, že zpráva nebyla při průchodu sítí nikým dodatečně upravena a je tak ověřena její původnost [1]. Grafické znázornění principu digitálního podpisu je zobrazeno na obrázku 1.1.



Obrázek 1.1 Grafické znázornění principu digitálního podpisu.

1.3 Hashovací funkce

Je nereverzibilní matematickou funkcí, která transformuje data jakékoliv velikosti na řetězec znaků konstantní velikosti. Z takto hashovaných dat již nelze žádným způsobem získat zpět původní data v čisté podobě, ale pokud takto hashujeme znovu stejná data, dostaneme naprosto totožný řetězec. Další podstatnou vlastností je také to, že pokud hashujeme dvě zprávy, které se navzájem liší pouze nepatrně, jejich hashované řetězce budou diametrálně odlišné, což můžeme vidět v příkladu na obrázku 1.2. Pro získání hashového řetězce byl použit online konvertor dostupný na stránce [2]. Existují různé druhy hashovacích funkcí, z nichž se již některé nepovažují za bezpečné, jelikož je možné je s dnešními výkonnými procesory prolomit v dohledném čase. V bitcoinovém protokolu se používají hashovací funkce SHA-256, jejíž výstupem je řetězec znaků o délce 256 bitů a RIPEMD-160 (výstupní velikost řetězce je 160 bitů) [3].



Obrázek 1.2 Příklad výstupu z hashovací funkce SHA-256.

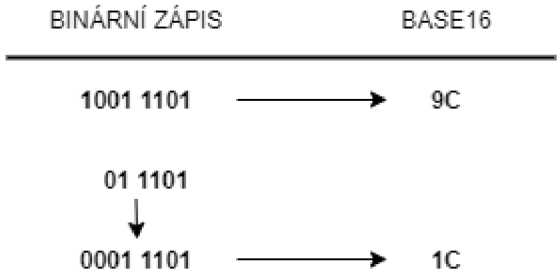
1.4 Binary-to-text kódování

Jak již název napovídá jedná se o kódování binárních dat do podoby tisknutelných znaků. Každý typ tohoto kódování má stanovený jiný mechanismus pro tvorbu tisknutelných znaků z binárních dat. Mění se jak počet různých vyskytujících se tisknutelných znaků,

tak délka bitového řetězce, pro jehož variace se ekvivalentní znaky stanovují. Jako jedno ze známějších a jednodušších můžeme uvést kódování Base16, které je v bitcoinovém protokolu hojně užito, což je ekvivalentní označení hexadecimální soustavy. Binární data jsou v něm rozdělována po 4bitových úsecích, z nichž každému tvaru náleží jeden ze šestnácti znaků uvedených v tabulce 1.1. Pokud počet bitů není dělitelný čtyřmi, je na začátek řetězce doplněn požadovaný počet nul [4][5]. Příklad konverze binárních dat do kódování Base16 je uveden na obrázku 1.3.

Tabulka 1.1 Zástupné symboly pro kódování Base16

BINÁRNĚ	0000	0001	0010	0011	0100	0101	0110	0111
BASE16	0	1	2	3	4	5	6	7
BINÁRNĚ	1000	1001	1010	1011	1100	1101	1110	1111
BASE16	8	9	A	B	C	D	E	F



Obrázek 1.3 Příklad konverze binárních dat do kódování Base16.

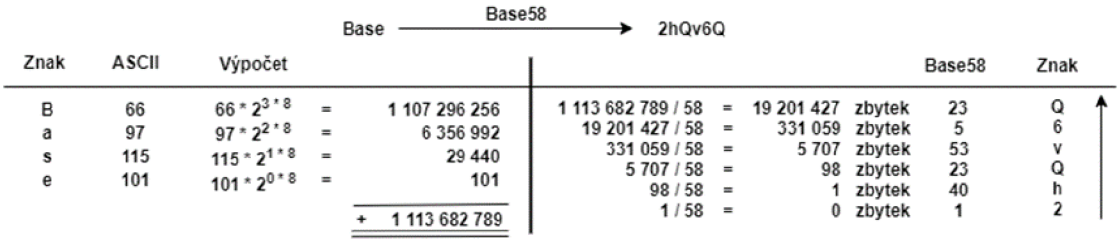
Kromě Base16 je v bitcoinovém protokolu použita funkce Base58, která se konkrétně využívá k úpravě tvaru bitcoinové adresy. Vychází ze základní funkce Base64 (všechny alfanumerické znaky a znaky + a /), oproti které neobsahuje 6 tisknutelných znaků, u kterých by v případě ručního opisování adres mohlo snadno dojít k záměně. Jedná se o znaky 0 (nula), O (velké o), l (malé L), I (velké i) a dále pak znak + a / (slash). Výčet znaků Base58 přiřazených k jejich decimální hodnotě můžeme vidět v tabulce 1.2.

Tabulka 1.2 Zástupné symboly pro kódování Base58

DECIMÁLNÍ HODNOTA	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
ZNAK	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	J	K	L
DECIMÁLNÍ HODNOTA	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**
ZNAK	M	N	P	Q	R	S	T	U	V	W	X	Y	Z	a	b	c	d	e	f	g
DECIMÁLNÍ HODNOTA	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**		
ZNAK	h	i	j	k	m	n	o	p	q	r	s	t	u	v	w	x	y	z		

V příkladu na obrázku 1.4 je znázorněn princip konverze řetězce znaků ze znakové sady ASCII na řetězec Base58. V první řadě se k danému znaku z kódování ASCII přiřadí jeho decimální hodnota pořadí z ASCII tabulky. Decimální hodnota pořadí jednotlivých

znaků se poté vynásobí exponenciálním číslem o základu 2, které má na pozici exponentu index pořadí daného znaku vynásobený číslem 8. Pro účely Base58 se jednotlivé znaky kódovaného řetězce indexují od konce řetězce a indexování započíná nulou. Takto získané součiny jsou následně sečteny a výsledné číslo postupně opakovaně děleno dělitelem 58, z čehož je podstatný právě zbytek po každém dělení. Právě zbytek udává pozici znaku v tabulce Base58. Výsledný zakódovaný řetězec jsou právě přiřazené znaky čtené od konce [6].



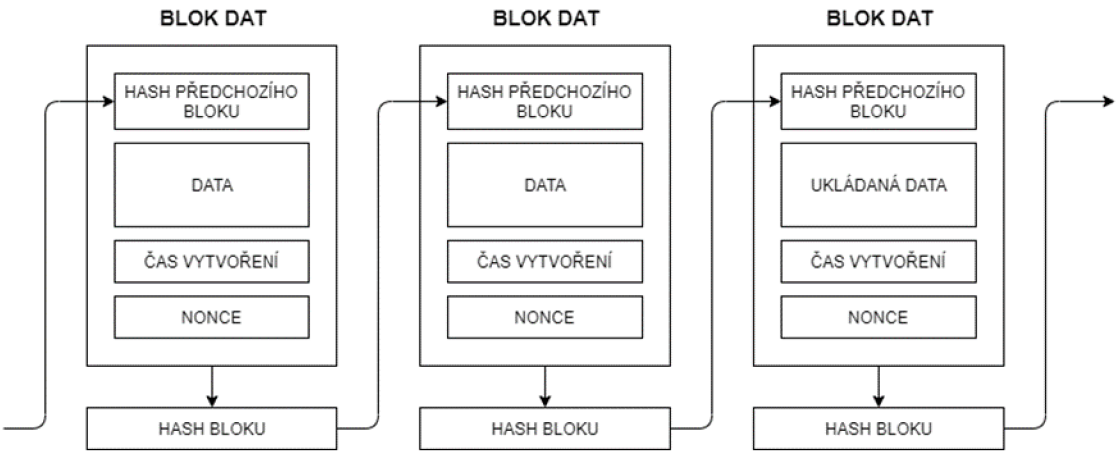
Obrázek 1.4 Příklad konverze binárních dat do kódování Base58.

2. BLOCKCHAIN

Blockchain, v české překladu tzv. "bločenka", je jedním z typů technologie distribuované knihy (distributed ledger technology), sloužící k bezpečnému uchovávání dat a zajištění jejich autentičnosti, transparentnosti a nepopiratelnosti. V podstatě je to forma distribuované databáze, ve které můžeme vyhledávat a číst již uložené informace, vkládat informace nové, ale je znemožněna jakákoliv zpětná úprava již uložených dat. K zajištění bezpečnosti blockchain používá hashovací funkce a mechanismus postupného navazování bloků dat do řetězce, kdy právě navazovaný blok obsahuje ve své hlavičce informaci o bloku předchozím, což zajišťuje jejich neměnné uspořádání. Celá, stále se rozrůstající, síť blockchainu je tedy složena z hashů bloků dat jdoucích za sebou v pevně ustanoveném pořadí. Nejvíce se o technologii blockchainu začalo hovořit až v roce 2009 s nástupem bitcoinu, avšak o první popis technologie zřetězování bloků se již v roce 1991 postarali Stuart Haber a W. Scott Stornetta ve svém dokumentu [7] kde představili způsob, jak zabránit manipulaci s časovými údaji u digitálních dokumentů a médií.

Součástí každého bloku je časový údaj o jeho vzniku, hash předchozího bloku dat, aktuálně ukládaná data a náhodné číslo označované jako nonce. Na takto vytvořený blok je poté aplikována hashovací funkce. Výsledný hash musí splňovat určité předem daná kritéria (u bitcoinu je to například velikost jeho hodnoty, která musí být menší než právě stanovená hranice). Celý proces tak spočívá v hádání náhodného čísla nonce a opakovaném aplikování hashovací funkce, dokud výsledný hash neodpovídá daným parametrům [8].

Blockchain je pouze název samotné technologie, která může být implementována různými způsoby a za využití rozličných kryptografických funkcí, každá z virtuálních měn tak může využívat svou konkrétní implementaci. Každá kryptoměna má pouze jeden blockchain, na jehož rozvoji, správě a distribuci se účastní tisíce uzlů. Je tak důležité, aby se všechny uzly shodly na správnosti řešení vždy pouze jednoho následného bloku v řetězci a tento stav blockchainu respektovaly. K dosažení shody mezi všemi uzly v distribuovaném systému se používají různé konsensuální algoritmy. Demonstrace tvorby jednotlivých bloků blockchainu je zobrazena na obrázku 2.1.



Obrázek 2.1 Znázornění principu zřetězování bloků dat.

2.1 Konsensuální algoritmy

2.1.1 Proof of work

Proof of Work je nejpoužívanějším konsensuálním algoritmem. K dosažení shody mezi uzly se dosahuje pomocí důkazu o vynaložené práci. Jednotlivé uzly mezi sebou soupeří o nalezení správného hashe bloku, a v důsledku toho spotřebovávají elektrickou energii potřebnou na vypočtení množství matematických operací. Největší šanci na to nalézt správné řešení dalšího bloku má pak uzel, který do systému dodá nejvíce výpočetního výkonu, s čímž souvisí i množství spotřebované elektrické energie. Nevýhodou tohoto systému je jeho značná neefektivita, kdy při soupeření uzly počítají stejné matematické operace. Tento algoritmus je využit například v Bitcoinu, Litecoinu, Moneru a mnoha dalších.

2.1.2 Proof of stake

Vznikl jako reakce na kritiku neefektivnosti Proof of Work. Jednotlivé uzly mezi sebou v hledání správného řešení nesoutěží, ale pro každý blok je vždy vybrán jeden uzel, který může najít řešení dalšího bloku. Způsobilý uzel je vybírán dle toho, kolik prostředků v dané měně vlastní. Nejvyšší pravděpodobnost k nalezení dalšího bloku tak mají účastníci disponující největším objemem dané kryptoměny. Tento algoritmus zakládá na faktu, že uživatelé držící největší množství měny mají největší zájem na tom, aby systém fungoval správně, a oni tak nepřišli o své prostředky. Za své úsilí při hledání dalšího bloku jsou účastníci odměněni získáním poplatků připojených k začleněným transakcím. Účastník musí vložit do systému jistý deposit, o který přijde v případě, že se pokusí do bloku začlenit neplatnou transakci. Je tak motivován k tomu jednat spravedlivě. Na tomto algoritmu pracují například kryptoměny Nxt, Potcoin a postupně na tento princip přechází Ethereum.

2.1.3 Delegated proof of stake

Funguje podobně jako Proof of Stake s tím rozdílem, že uzly, kterým bude dovoleno těžit další bloky jsou voleny všemi účastníky sítě. Účastníci sítě si tak volí několik desítek svých zástupců, kteří se podílejí na zapojování dalších bloků. Pokud jimi zvolený zástupce přestane disponovat potřebným množstvím výkonu, mohou svůj hlas vzít zpět a volit zástupce dalšího. Na tomto algoritmu pracuje například kryptoměna EOS [19].

3. BITCOIN

3.1 Obecný princip fungování

Bitcoin je první zcela decentralizovanou virtuální měnou. Vytvořen byl v roce 2008 člověkem nebo skupinou lidí skrývajících se pod pseudonymem Satoshi Nakamoto. Pravá identita jeho tvůrce zůstává dodnes neznámá. Stěžejní doména bitcoin.org byla zaregistrována 18. srpna 2008 a první průvodní dokument [10], napsaný samotným tvůrcem, podle kterého byl tento systém spuštěn, byl na doméně zveřejněn 31. října téhož roku. První blok bitcoinového blockchainu (pořadové číslo 0) označovaného jako "blok genesis", byl jeho tvůrcem vytěžen 3.1.2009. Toto datum je také považováno za jeho začátek [11].

Stejně jako máme běžné peněžní prostředky uložené na našich bankovních účtech a manipulujeme s nimi skrze čísla účtů a s nimi spojenými hesly a piny, tak i bitcoiny kterými disponujeme musíme někde uchovávat a mít možnost s nimi nakládat. Jako analogie bankovních účtů zde slouží bitcoinové adresy.

Bezpečnost transakcí v bitcoinové síti spočívá ve složitosti asymetrické kryptografie, konkrétně v technologii digitálních podpisů. Každý majitel účtu disponuje párem unikátních kryptografických klíčů, veřejným a soukromým. Samotnou transakci si můžeme představit jako zprávu o tom, že převádíme dané množství bitcoinů ze své adresy na adresu příjemce. Tuto zprávu podepíšeme svým soukromým klíčem a odešleme ke zpracování do bitcoinové sítě.

Samotné transakce jsou na sobě závislé podobně jako bloky v blockchainu. Každá transakce v sobě nese informaci o předchozích transakcích, díky kterým se bitcoiny ocitly až na konkrétním účtu. Společně tak vytváří rozsáhlou síť vzájemných referencí, díky nimž můžeme vysledovat pohyb jednotlivých mincí až k jejich vytvoření. Při schvalování (ověřování) transakcí dochází ke kontrole, zda účet opravdu disponuje daným počtem mincí neboli, jestli v databázi existují záznamy, které mapují cestu mincí na daný účet.

Transakce obsahuje vstupy a výstupy. Vstupem do transakce klasifikujeme soubor všech předchozích transakcí, díky nimž účet disponuje svou celkovou částkou. Vyplývá z toho tedy, že vstupů má transakce větší množství. Do transakce vstupuje vždy celková částka na účtu. Výstupy pak bývají zpravidla dva, jedna část bitcoinů vede na účet příjemce, a část druhá putuje zpět na účet vlastníka. Výstup z transakce může být mnohdy nižší než její vstup, rozdíl těchto dvou hodnot pak značí výši částky, přidanou jako poplatek těžaři za zapsání transakce do blockchainu.

Na zapisování transakcí do blockchainu, tzv. těžbě bloků, se podílejí uživatelé sítě označovaní jako těžaři (miners). Těžařem se může stát jakýkoliv uživatel (vlastník účtu), který má na tomto zájem a je schopen se vybavit technikou disponující dostatečnou výpočetní kapacitou pro zvládání náročných kryptografických výpočtů a schopnou svým výkonem konkurovat ostatním těžařům.

Motivací pro těžbu bloků, a tedy rozšiřování a udržování chodu celé sítě, je stanovená odměna za jejich vytěžení (připojení do blockchainu), která v dnešní době činí 6,25 BTC (zkratka pro bitcoin, podobně jako koruna česká CZK). Dále to pak jsou poplatky za v bloku zpracované transakce. Transakce čekající na zpracování se shromažďují v tzv. mempoolu, odkud si těžaři postupně vybírají transakce, které do bloku začlení. Velikost jednoho zpracovaného bloku má implementovanou maximální velikost na 1 MB, což znamená, že ne všechny čekající transakce je možné zpracovat v jediném bloku. V praxi jsou pak tedy přednostně zpracovávány transakce obsahující větší poplatek za zpracování, jelikož je tak těžař více finančně motivován. Výši poplatku si při zadávání transakce stanovuje každý sám (může být i nulový), ale pokud poplatek nestanovíme, je dost pravděpodobné, že naše transakce zůstane nezpracována.

Samotná těžba je velmi výpočetně náročná, z čehož vyplývá, že je i energeticky náročná kryptografická operace. Těžba spočívá v hledání správného hashe právě zpracovávaného bloku, který musí splňovat parametr, že velikost jeho hashe je menší než právě stanovená hranice. Zjednodušeně, že jeho hash musí začínat určitým počtem nul. Těžaři pak mezi sebou v tomto hledání soupeří, neboť prvnímu, kdo tento hash najde náleží odměna za vytěžení bloku. Vytěžení tedy znamená nalezení odpovídajícího hashe bloku. Po vytěžení se blok zapíše na konec blockchainu a těžaři sestaví blok nový, u kterého se opět hledá správný hash. Každý blok v sobě obsahuje i hash bloku předchozího, čímž je zaručena jejich provázanost v blokchainu. Právě díky této provázanosti jsou údaje v blockchainu zpětně neupravitelné, jelikož jakákoliv drobná změna v některém z předešlých bloků by znamenala změnu jeho hashe a tím pádem i změnu hashů všech po něm následujících bloků, které by se musely od tohoto bodu znovu přepočítat. Jelikož jsou záznamy o transakcích veřejné a jsou periodicky rozesílány celé síti, je dost nepravděpodobné, že by snaha o falzifikaci zůstala nepovšimnuta po tak dlouhou dobu, aby narušitel stihnul znovu přepočítat hashe všech bloků.

Bitcoin není měna v tradičním pojetí. Je to systém fungováním více podobný trhu s akciemi, kdy reálnou peněžní hodnotu bitcoinu stanovuje aktuální nabídka a poptávka a při podrobnějším zkoumání jeho vývojového grafu můžeme sledovat stejné zákonitostí jako při obchodování s cennými papíry. Jeho nejpřínosnější vlastností je jeho světová decentralizace, kdy jeho fungování neovlivňuje žádný stát, vláda ani konkrétní úřad, ale je plně v rukou všech členů bitcoinové komunity. S decentralizací souvisí i rychlost vyřizování plateb, kterou určuje pouze přidaná odměna pro těžaře, a nikoliv zda se platba posílá do zahraničí, jak je tomu u bankovních institucí. Další důležitou vlastností je jeho transparentnost, kdy veškeré záznamy o všech proběhlých operacích jsou sousledně a neměnně uloženy v blockchainu, bez možnosti je zpětně upravit a možnost do nich nahlížet nebo ověřovat jejich pravost pak může kdokoliv, kdo o to má zájem [3][8][12][13].

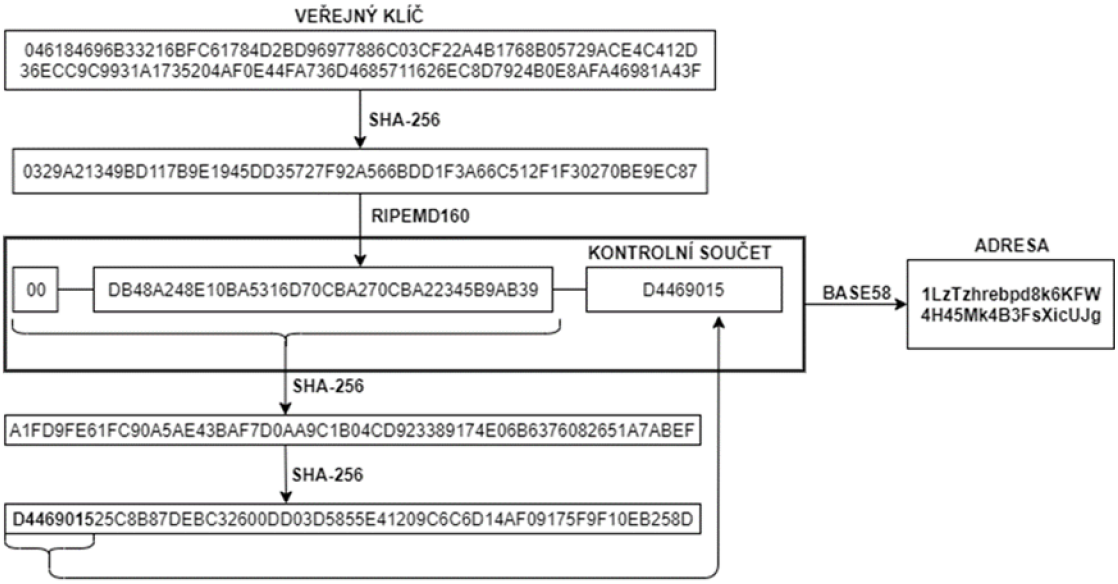
3.2 Kryptografické klíče

Pro ustanovení kryptografických klíčů se u Bitcoinu používá protokol ECDSA využívající matematické vlastnosti eliptických křivek. Pro tvorbu soukromého klíče se vygeneruje náhodné 32 bajtové číslo, na které se následně aplikuje hashovací funkce SHA-256. Výstupem z této funkce je náš soukromý klíč. Soukromý klíč má tedy velikost 32 bajtů.

Veřejný klíč se poté vypočítá na základě klíče soukromého právě pomocí algoritmu ECDSA. Výstupem z algoritmu je 65 bajtové číslo (což odpovídá 130 znakům hexadecimální soustavy), což je náš veřejný klíč [14]. K jednomu soukromému klíči můžeme vygenerovat až 2 miliardy veřejných klíčů, čehož se využívá pro ztížení sledovatelnosti pohybů na jednotlivých adresách, jelikož se naše adresa vypočítává právě z veřejného klíče a může se tak s každou transakcí změnit [8].

3.3 Bitcoinová adresa

Bitcoinovou adresu získáme odvozením z našeho veřejného klíče. Grafické znázornění tvorby bitcoinové adresy vykresluje obrázek 3.1. Pro získání řetězců v jednotlivých krocích byl použit konvertor dostupný na stránce [15]. Na 65 bajtů dlouhý řetězec veřejného klíče se aplikují postupně hashovací funkce SHA-256 a RIPEMD160. Vznikne nám tak pouze 20bajtový řetězec znaků. K tomuto řetězci na začátek přidáme předponu, díky které je poté ovlivněn znak, nebo několik znaků na začátku výsledné adresy. Většina obvyklých adres začíná číslem 1, v tomto případě je prefix přidávaný k řetězci 00 (v hexadecimální soustavě). Na konec řetězce je poté připojen ještě jeho kontrolní součet. Kontrolní součet získáme tak, že na výstup z funkce RIPEMD160 s připojenou předponou 2x aplikujeme funkci SHA-256. Kontrolní součet pak tvoří první 4 bajty (8 hexadecimálních znaků) z takto získaného řetězce. Výstup z funkce RIPEMD160 s připojenou předponou a kontrolním součtem na konci konvertujeme pomocí Base58 a získáme tak výslednou adresu. Výstupem z funkce je řetězec obsahující pouze 58 možných zástupných symbolů z ASCII tabulky. Jsou jimi arabské číslice a malá a velká písmena abecedy, vyjma znaků o (malé O), 0 (nuly), I (velké i), l (malé L), jelikož u těchto znaků dochází při opisování často k záměnám. Bitcoinová adresa pak může vypadat například takto: 1LzTzhrebpd8k6KFW4H45Mk4B3FsXicUJg [14][16][17].



Obrázek 3.1 Grafické znázornění používání digitálního podpisu v transakcích.

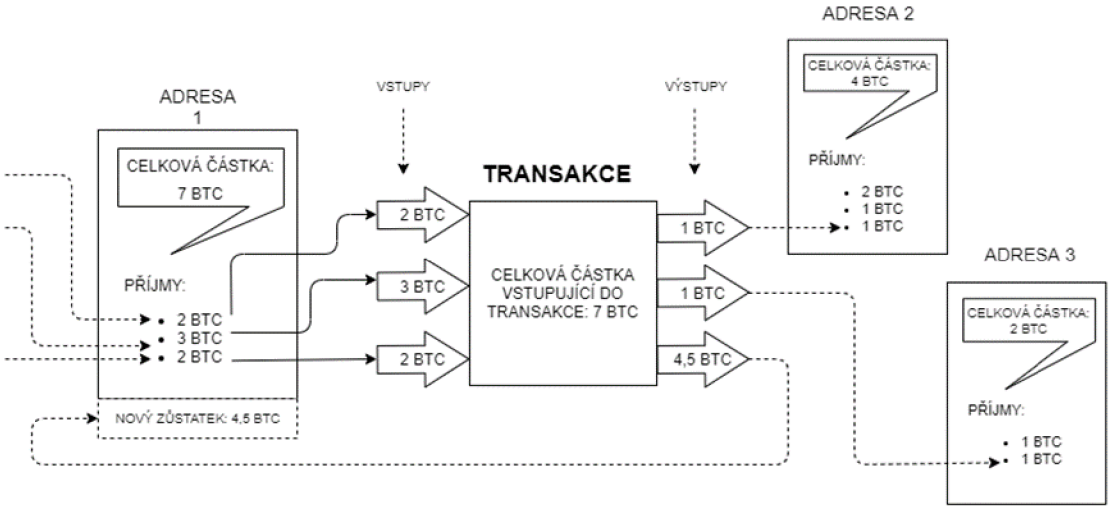
3.4 Transakce

U Bitcoinu máme 2 druhy transakcí. Jsou jimi všechny běžné transakce posílané uživateli a poté speciální generující transakce, která slouží pro vyplácení odměny těžaři za vytěžený blok. Každý blok obsahuje právě jednu generující transakci, tzv. coinbase, která jako jediná transakce v bloku neobsahuje žádné vstupy, jelikož ona sama vytváří v systému nové bitcoiny. Jako výstup obsahuje adresu těžaře, který daný blok vytěžil, a na kterou nově vytvořené mince poputují.

Všechny běžné transakce obsahují vstupy a výstupy. Jak vstupy, tak výstupy mohou obsahovat vícenásobné hodnoty, a zpravidla je také obsahují. Do transakce vstupuje vždy celková částka, která se na účtu nachází a jako vstupy jsou označovány všechny předešlé příchozí transakce, díky nimž účet právě touto částkou disponuje. Jako vstup tedy označujeme informaci o tom, z jaké adresy přišla konkrétní zlomková částka na náš účet.

Výstup je informace o částce, kterou zasíláme na účet příjemce. Do jedné transakce může být zahrnuto více plateb na různé účty. Specifickým výstupem je pak ten, který vrací zbytek po zaplacení zpět na náš účet. Bitcoiny tak takto s každou transakcí cirkulují systémem. Tento koloběh má výhodu týkající se zvýšení anonymity, jelikož s každou transakcí můžeme takto změnit adresu našeho účtu a souvislosti mezi transakcemi jednoho majitele se tak stanou mnohem hůře vysledovatelné. Graficky je princip tvorby transakcí znázorněn na obrázku 3.2. Z tohoto principu vyplývá, že by se souhrnná částka ze všech vstupů měla rovnat výstupům, v praxi to tak ale nebývá. Vstupy bývají zpravidla větší než výstupy a tento rozdíl je pak definován jako odměna pro těžaře, který zahrne naši transakci do bloku připojovaného do blockchainu.

Když vytvoříme novou transakci a pošleme ji do sítě, její první cesta směřuje do tzv. mempoolu, což je databáze transakcí čekajících na zpracování. Odtud si pak těžaři vybírají transakce, které do svého bloku začlení. Motivací pro těžbu je také odměna za zpracování transakce, z čehož vyplývá, že transakce s vyšším stanoveným poplatkem bývají zpracovávány přednostně, zatímco transakce poplatek neobsahující nemusí být zpracovány vůbec [3][14].

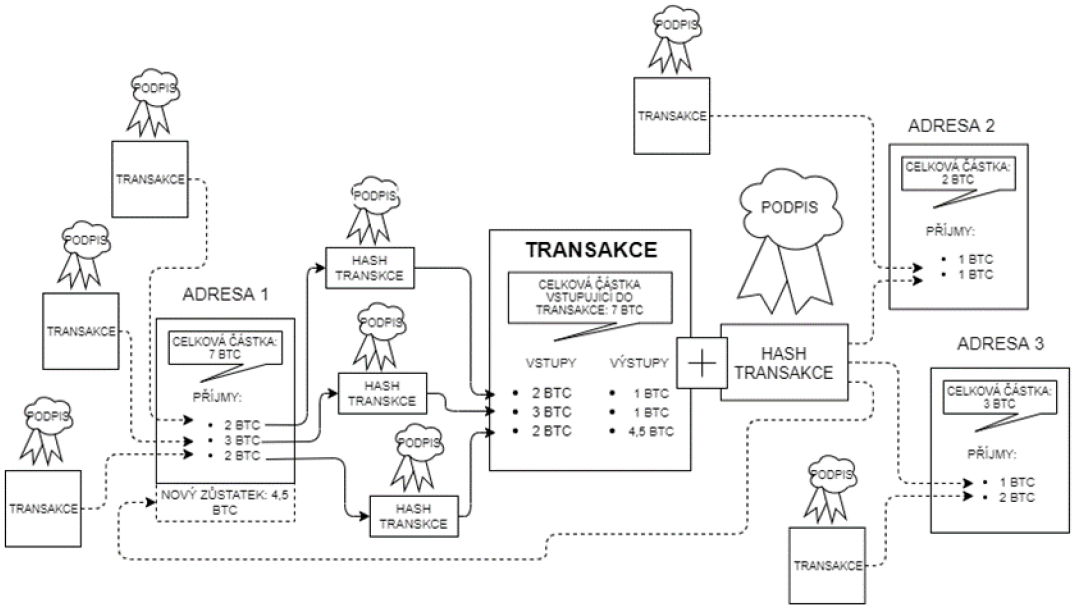


Obrázek 3.2 Grafické znázornění průběhu transakce, jejich vstupů a výstupů.

3.5 Podepisování transakcí

Každá řádná transakce musí být podepsaná svým zadavatelem. Ve skutečnosti to vypadá tak, že se z výsledné transakce vytvoří hash pomocí SHA-256, a až ten se poté podepíše. Takto podepsaná transakce je již plně připravena ke zpracování. Nicméně svým podpisem musíme stvrdit i vstupy do transakce. Celý proces podepisování pak vypadá následovně:

Na náš účet nám přijdou odesílateli podepsané transakce. Všechny dřívější přijaté transakce se podílejí na velikosti konečného zůstatku, a tedy všechny tvoří vstupy do transakce. Každý takový vstup je nutné individuálně podepsat. Z každé takové transakce, podepsané předchozím odesílatelem, se vytvoří hash, který následně stvrdíme svým vlastním podpisem. Všechny takto podepsané hashe společně vytváří veškeré vstupy do transakce. V transakci poté stanovíme množství peněž a adresy, na které peníze posíláme. Z tohoto konečného stavu transakce vytvoříme nový hash, opět jej podepíšeme a spolu s transakcemi odešleme do sítě [14]. Pro lepší pochopení je princip podepisování transakcí znázorněn na obrázku 3.3.



Obrázek 3.3 Grafické znázornění používání digitálního podpisu v transakcích.

3.6 Skladba bloku

Abychom lépe pochopili samotný proces těžby a to, jakým způsobem se bloky reálně připojují do blockchainu, je nutné si uvést, z jakých dat se celý blok skládá. Velikost jednoho bloku je bitcoinovým protokolem omezena na 1 MB.

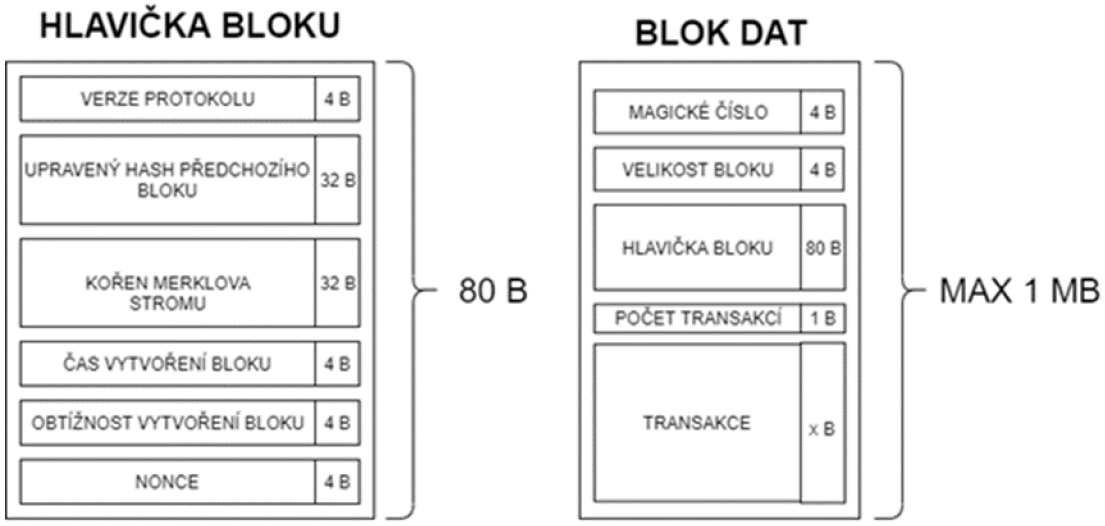
Každý blok začíná tzv. magickým číslem, které má v Bitcoinu tvar 0xD9B4BEF9. Toto číslo je konstantní a nese informaci o typu dat, jaké má program očekávat. Není výsadou bitcoinového protokolu, ale používá se u většiny typů souborů. V samotném procesu těžby nehraje nijak důležitou roli.

Za magickým číslem následuje 4bajtový údaj o celkové velikosti bloku, která je zapisována v hexadecimálním tvaru a nesmí překročit 1 MB, tato hodnota je ustanovená přímo v bitcoinovém protokolu. Do celkové velikosti není započteno prvních 8 bajtů tzn. magické číslo a místo rezervované právě pro tuto velikost. Pokud tedy máme na tomto místě například hodnotu 850 bajtů, reálná velikost bloku je 858 bajtů.

Za velikostí bloku následuje 80bajtová hlavička bloku. Ta je pro proces těžby naprosto zásadní, jelikož právě z hlavičky bloku se vypočítává hash, který zajišťuje bezpečnost blockchainu. Když tedy zjednodušeně mluvíme o tom, že každý blok obsahuje referenci na blok předchozí, myslíme tím, že v hlavičce aktuálního bloku je zaznamenán hash hlavičky bloku předchozího. Hlavička obsahuje celkem 6 parametrů. Postupně je to verze protokolu (4 bajty), hash hlavičky předchozího bloku (32 bajtů), transakce obsažené v bloku, upravené do struktury merklova stromu, z čehož právě kořen merklova stromu je v hlavičce uložen (32 bajtů), čas vytvoření bloku (4 bajty), vypočítaná

obtížnost vytvoření bloku (4 bajty) a hádané číslo nonce (4 bajty). Velikost 32 bajtů u hashe hlavičky a kořene merkelova stromu vyplývá z používané funkce SHA-256.

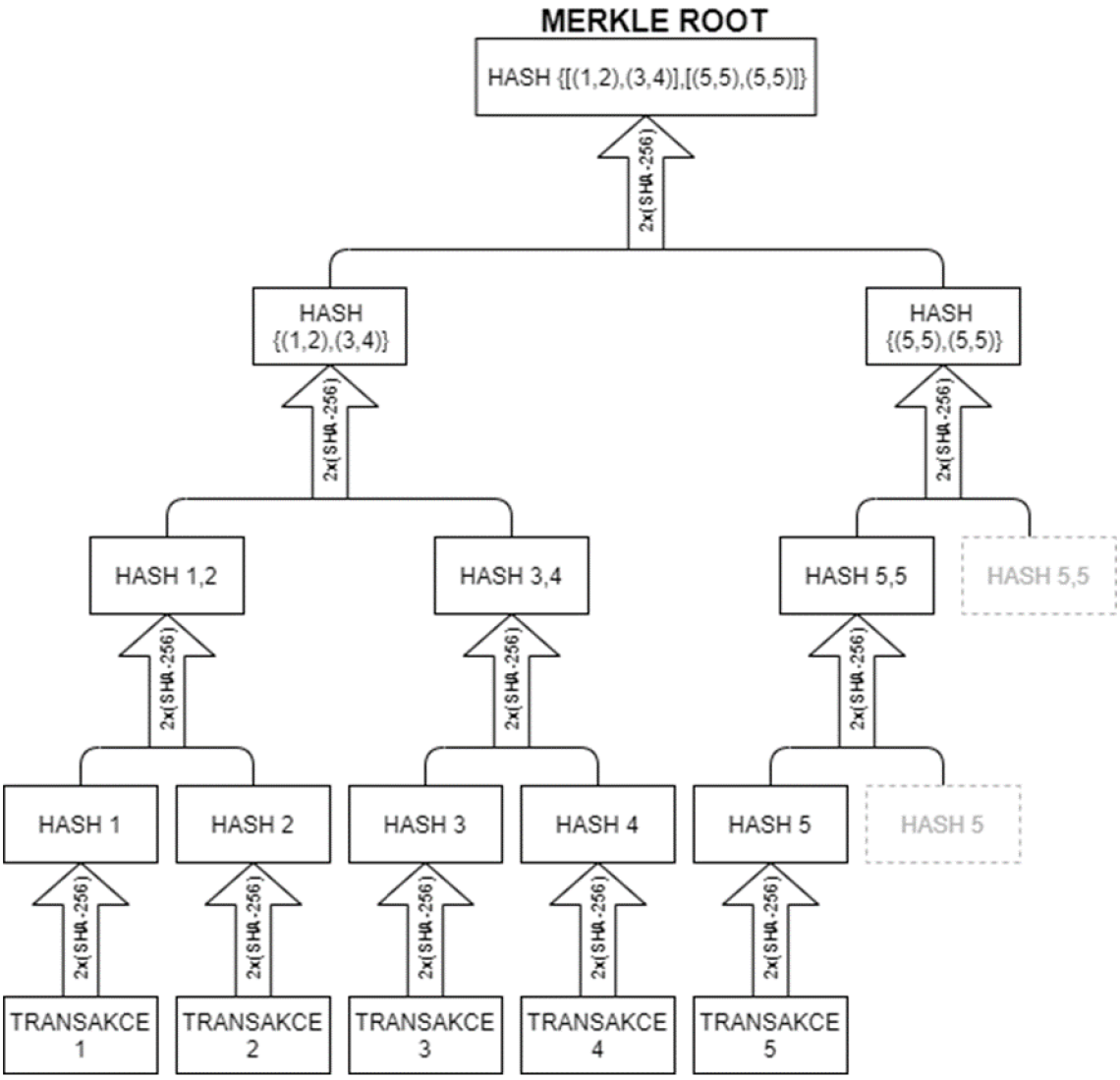
Za hlavičkou následuje číslo udávající počet transakcí začleněných do bloku (1 B) a poté již následují samotná data [8][13]. Stavba bloku dat a jeho hlavičky je znázorněna na obrázku 3.4.



Obrázek 3.4 Grafické znázornění prvků, ze kterých se skládá blok dat a hlavička bloku, s jim náležejícími velikostmi v bajtech.

3.7 Merklův strom

Merklův strom je binární hashovací datová struktura, která slouží k seskupení většího množství vstupů na jeden jediný výstup, tzv. kořen stromu (merkle root). U Bitcoinu se tato struktura používá k bezpečnému zapracování všech transakcí do bloku. Jak již bylo zmíněno hash transakcí se ukládá do hlavičky, z níž samotné se poté vytvoří hash, na který je odkazováno dalším blokem. Bylo by příliš objemné a nepraktické zaznamenávat do hlavičky hashe všech obsažených transakcí, a proto se tam zapíše pouze jeden konečný. Představme si zjednodušenou situaci, kdy vytváříme blok s pěti transakcemi. Na každou takovou transakci nejprve 2x aplikujeme SHA-256. Následně 2x aplikujeme tuto funkci vždy na 2 takto vytvořené hashe, dokud nedostaneme jediný výstup. Pokud je v patře stromu lichý počet hodnot, poslední hodnota se duplikuje. Výhodou tohoto přístupu je, že ať by byla, byť jen nepatrně upravena jakákoliv transakce v pyramidě, výsledný hash bude diametrálně odlišný [8][13]. Způsob tvorby merklova stromu vykresluje obrázek 3.5.



Obrázek 3.5 Grafické znázornění způsobu tvorby merklova stromu.

3.8 Typy uzlů v síti

Bitcoinová síť je složena z jednotlivých uzlů, které mezi sebou komunikují skrze peer-to-peer protokoly. V terminologii Bitcoinu používáme označení úplný uzel (full node) pro uzel, který udržuje kompletní historii blockchainu se všemi podrobnostmi o transakcích. V současnosti, v reakci na stále se zvyšující velikost dat blockchainu, však všechny uzly v síti nejsou uzly plnými. Uzel, který neudržuje plnou historii blockchainu nazýváme uzlem odlehčeným (lighweight nodes, SPV nodes (Significant Payment Verification)). Odlehčené uzly neuchovávají veškeré informace o transakcích, ale pouze hlavičky bloků.

Ne všechny uzly v bitcoinové síti vykonávají stejnou funkci. Obecně lze konstatovat, že každý uzel musí obsahovat pouze směrovací protokol, aby byla zajištěna jeho propojenost se sítí. Podle toho, jaké funkce uzly dále provozují je můžeme rozdělit na

uzly koncových uživatelů využívajících bitcoinovou sít pouze pro manipulaci s bitcoiny na jimi spravovaných adresách (bitcoinové peněženky), nebo uzly těžařské, které se aktivně podílejí na připojování nových bloků do blockchainu. Těžařský uzel může také zároveň plnit funkci peněženky.

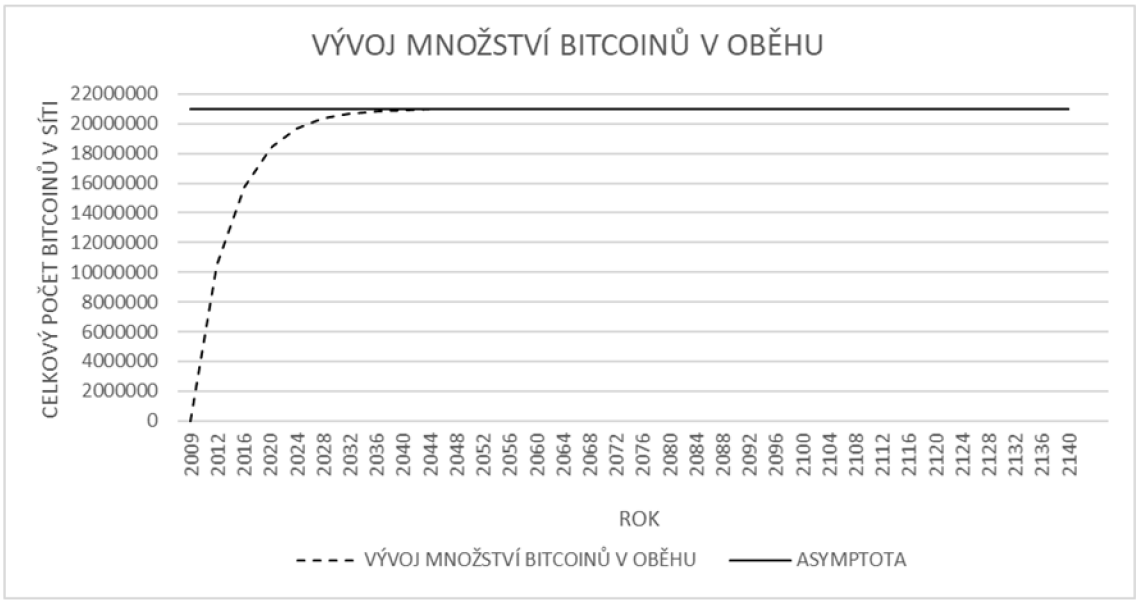
3.9 Těžba

Námi vytvořené transakce, musí někdo ověřit a začlenit do blockchainu. Členové sítě, kteří se na tomto podílí se nazývají těžaři. Těžba je tedy v konečném důsledku ověřování a vyřizování transakcí. Jelikož blockchain se skládá z následně za sebou jdoucích bloků, transakce musí být do některého z bloků začleněny. Celá komunita těžařů nepracuje pospolitě, ale vzájemně si konkurují a soupeří mezi sebou o to, kdo dříve vytěží další blok, jelikož jsou motivováni ziskem bitcoinů z vytěženého bloku a poplatků za transakce. Každý těžař nebo skupina vzájemně spolupracujících těžařů, tzv. pool, si sestavuje blok vlastní a záleží tedy pouze na něm, které transakce si do něj vybere, většinou jsou to však ty s nejvyšším poplatkem. Do blockchainu se však zapíšou pouze transakce obsažené v bloku těžaře, který zvítězil. Následně se sestaví zase blok nový a vše se opakuje.

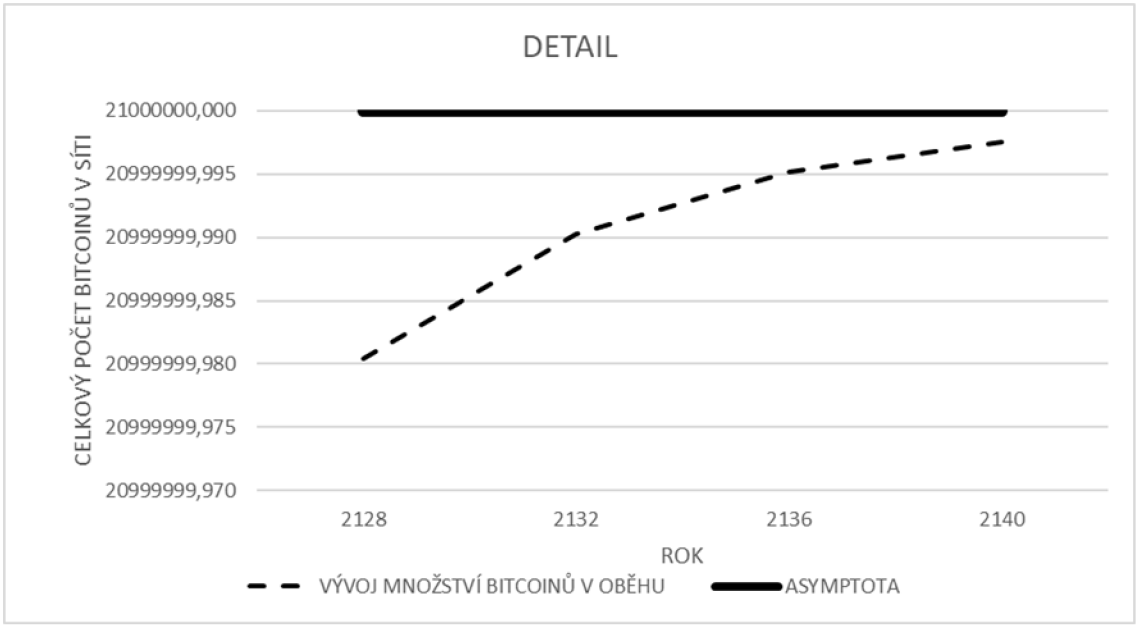
Těžba spočívá v hledání správného čísla nonce, což je jediný proměnlivý údaj v bloku, a tudíž pouze jím můžeme ovlivňovat tvar výsledného hashe. Výsledný hash pak musí splňovat kritérium, že jeho číslo je menší než právě stanovená hranice. Poté co těžař sestaví blok, dosadí do něj konkrétní 4 bajtové číslo nonce a vypočítá hash. Pokud hash toto kritérium nesplňuje, dosadí se nové číslo nonce a výpočet se takto opakuje až do doby, dokud není správné číslo tvořící odpovídající hash nalezeno.

Hranici velikosti hashe dynamicky stanovuje bitcoinový protokol, který tak může regulovat složitost výpočtu a s tím i nutný výpočetní výkon na tuto operaci a výsledný čas jejího provedení. Průměrný čas na vytěžení jednoho bloku je stanoven na 10 minut. Jelikož s vytěžením bloku souvisí i příliv nových bitcoinů do systému, je tak zamezeno náhlému nárustu počtu bitcoinů v síti. Každých 2016 vytěžených bloků kontroluje bitcoinový protokol průměrnou dobu těžby, pokud je nižší než 10 minut, zvýší složitost výpočtu tím, že sníží maximální hranici čísla, jakého může hash nově vytěženého bloku dosahovat. Pokud je naopak průměrná doba vyšší, protokol hranici zvýší. Co se týče omezování přílivu nových bitcoinů do sítě, omezuje jej ještě tzv. půlení (halving). Každých 210000 vytěžených bloků, což odpovídá zhruba 4letému intervalu, se odměna za nově vytěžený blok sníží na polovinu. Na začátku činila odměna 50 bitcoinů, která se postupně snížila na 25, 12,5 a v současné době je na 6,25 bitcoinech. Tímto postupným snižováním je také docíleno stanovení celkového množství bitcoinů, které kdy budou v oběhu a zaokrouhleně činí 21 miliónů. Toto číslo je získáno aritmetickým součtem nekonečné řady, jejíž asymptota je právě 21 miliónů. Znamená to, že množství bitcoinů přidávaných do oběhu se bude s každým půlením snižovat a čím dál tím pomaleji se přibližovat této hranici, kterou nikdy nedosáhne. Tuto situaci znázorňují grafy na

obrázcích 3.6 a 3.7. Konec přílivu bitcoinů je podle doby na vytěžení jednoho bloku vypočten na rok 2139, avšak už v roce 2040 bude vytěženo 99,9 % všech bitcoinů [3][18].



Obrázek 3.6 Grafické znázornění vývoje množství vygenerovaných bitcoinů.



Obrázek 3.7 Detailnější znázornění konce grafu 3.6.

Tento mechanismus má za cíl kopírovat chování trhu s drahými kovy, jejichž zásoba je omezena, a kdy s narůstajícím počtem již vytěžených je těžší a těžší získat nové.

To, jak rychle se jeden blok připojí do blockchainu souvisí s veličinou charakterizující výpočetní výkon bitcoinové sítě nazývanou hashrate. Její jednotkou je hash za sekundu (h/s) a udává, kolik různých hashů dokáže celá bitcoinová síť vypočítat za jednu sekundu. V současné době se hashrate pohybuje okolo 130 Eh/s (Exahash za sekundu = 10^{18} hash/s).

Jelikož celá síť spolu komunikujících těžařů je velice rozsáhlá, tak občas dochází k situacím, kdy konkurenční těžaři vytěží nezávisle na sobě každý svůj blok s minimálním časovým rozdílem a informace o tom, že byl první blok vytěžen, se nestihne rozšířit po celé síti. Dejme tomu, že 2 těžaři nacházející se na opačných stranách zeměkoule vytěží nový blok s časovým rozdílem v řádech milisekund. V tomto případě, vzhledem ke zpoždění při přenosu dat, ostatní těžařské uzly se sídlem na stejné polokouli dostanou informaci o tom, že blok vytěžil první těžař, a dále pokračují v těžbě s hashem tohoto bloku. Ovšem těžaři obývající druhou polokouli, společně se „svým“ těžařem, budou zase informováni o bloku, který vytěžil tento druhý těžař a své další bloky budou stavět na něm. Blockchainový řetězec se tak rozdělí na 2 ramena, kdy jedno je označováno jako hlavní a druhé jako vedlejší. Tento případ nechtěného rozčlenění blockchainu je označován jako „fork“. Blockchain může ale obsahovat pouze jeden hlavní řetězec. Informace o jednotlivých blocích se časem roznese celou sítí a hlavním a jediným pokračujícím řetězcem se pak stane rameno, ke kterému jako prvnímu přibude další blok. Vzhledem ke složitosti výpočtu hashe se nepovažuje za pravděpodobné, že by takto vzniklá ramena zvládli koexistovat nezávisle na sobě bez povšimnutí déle než 6 po sobě jdoucích bloků a bylo tedy dohodnuto, že za skutečně validní transakce budou označovány až ty v minimální hloubce 6 bloků. Druhé zaniklé rameno je pak označováno jako mrtvá větev. Toto se již dále nerozrůstá a všechny transakce v něm obsažené jsou stornovány, jako kdyby se nikdy nestaly.

V současné době složitost výpočtu dosáhla takové náročnosti, že se stává pro těžaře značně nevýhodné těžit individuálně, protože tak mají pouze nízkou procentuální šanci na úspěšné vytěžení bloku. Těžaři se tak začali uskupovat do skupin, tzv. těžařských poolů (mining pool). Komunikace mezi členem těžařského poolu a správcovským serverem probíhá pomocí speciálně navržených protokolů, z nichž nejvyužívanější je protokol Stratum.

3.9.1 Těžařský pool

Těžařský pool je složen ze správce a jednotlivých těžařů zapojených v daném poolu. Správce pomocí těžařských protokolů komunikuje s uzly jednotlivých těžařů a přerozděluje jim práci. Pokud konkrétní těžař nalezne správný hash bloku, odměna za vytěžený blok se nepřičítá jemu, ale jde na adresu jeho poolu, který ji poté spravedlivě rozdělí mezi všechny těžaře, kteří se na výpočtu podíleli, podle jimi dodaného výpočetního výkonu. Těžaři tak mají větší šanci na získání odměny za vytěžený blok (částečné), než když by v tomto vysoce konkurenčním prostředí těžili samostatně [3].

3.9.2 Těžební hardware

Jak již bylo v úvodu řečeno, těžba je velmi výpočetně náročná operace a složitost tohoto výpočtu se zvyšuje spolu s narůstajícím počtem aktivních těžařů. V dnešní době již není možné úspěšně těžit bitcoiny na běžných počítačích v důsledku vysoké konkurence a existenci specializovaných těžebních zařízení.

V počátcích Bitcoinu těžba probíhala na běžných výkonnějších počítačích, postupně se přešlo na těžbu pomocí grafických karet, což ale také přestalo být brzy dostačující. Krátkou éru pak zažila těžba pomocí programovatelných hradlových polí (FPGA), která byla brzy nahrazena specializovanými zařízeními ASIC (Application Specific Integrated Circuit, zákaznický integrovaný obvod), které nyní poskytují většinový objem těžebního výkonu [3].

4. KRYPTOMĚNY

V první části bakalářské práce jsme si popsali fungování kryptoměny Bitcoin jakožto hlavního představitele decentralizovaných kryptoměn. Ani zdaleka to ale již není jediný zástupce digitálních měn na trhu. Zveřejněním bitcoinového protokolu započal velký boom v rozvoji do té doby ne příliš rozvinutého trhu digitálních měn. V závislosti na bitcoinovém protokolu začaly vznikat další projekty virtuálních měn, z nichž první v základu na tomto protokolu stavěly, ale později začaly vznikat kryptoměny s protokoly vlastními, nebo zcela nové inovativní koncepty, kde kryptoměny neplní pouze zástupnou funkci peněz, ale začínají rozvíjet další potenciál, který sebou technologie blockchain přináší, a to programovatelné smart kontrakty. Mnoho z těchto pokusů nebylo úspěšných a nemělo dlouhé trvání, ale některé z nich si dokázaly získat důvěru, vytvořily si svou základnu uživatelů a uchytily se na kryptoměnovém trhu.

Kryptoměn jsou v dnešní době tisíce, a může být problém se v jednotlivých z nich vyznat. Nejznámější webovou stránkou udržující aktuální databázi kryptoměn je stránka Coinmarketcap.com [21], kde se dozvíme základní detaily o měně, jako její oficiální doménu, aktuální kurz k americkému dolaru, historii pohybu na burze nebo také burzy, na kterých je měna obchodována.

Důležité je také podotknout, že většinu kryptoměn není možné směňovat přímo za fiat měnu. Přímý nákup je možný pouze u Bitcoinu, Litecoinu, Ripple, Etherea a Bitcoin cash [22] (množství přímo směnitelných měn se postupem času stále zvyšuje). V minulosti bylo možné směňovat tyto měny pouze za americký dolar nebo euro, postupem času se ale stále rozrůstá soubor měn, za které lze vybrané kryptoměny směnit přímo, mezi něž patří i česká koruna. Přímou směnu za CZK nabízí například česká směnárna CoinMate nebo SimpleCoin.

4.1 Klasifikace kryptoměn

Jelikož jsou kryptoměny stále poměrně novým fenoménem, nebylo zatím ustanoveno žádné oficiální třídění nebo kategorie, do kterých by jednotlivé kryptoměny bylo možné podle určitých pravidel zařadit. Jako jedno z možných třídění se však zaběhlo třídění kryptoměn na generace, dle časového a funkcionálního hlediska. V poslední době tak v mnohých článcích můžeme najít například poznámku, že se daná kryptoměna řadí do 3. či 2. generace kryptoměn [26].

4.1.1 Nultá generace

Za nultou generaci se označuje pouze samotný Bitcoin.

4.1.2 První generace

Do 1. generace kryptoměn se řadí veškeré altcoiny, jejichž hlavním cílem je pouze plnit zástupnou funkci peněz. Mnoho z nich, především pak ty, které vznikaly krátce po

zveřejnění Bitcoinu, staví v základu na bitcoinovém protokolu, pouze s některými drobnými obměnami jako je změna hashovacího algoritmu, či rychlost zapisování do blockchainu. Mezi tyto můžeme zařadit například Litecoin, Namecoin, Peercoin a mnoho dalších. Tyto měny jsou stejně jako Bitcoin psané v jazyce C++. Některé z kryptoměn řazené do této kategorie však disponují zcela vlastními protokoly. Zřejmě nejpodstatnější je protokol CryptoNote zajišťující vysokou míru anonymity, který byl prvně představen při uvedení měny Bytecoin. Nejvýznamnější zástupce využívající protokol CryptoNote je však kryptoměna Monero vzniklá odvozením právě z Bytecoinu [23][24][25][26].

4.1.3 Druhá generace

O druhé generaci kryptoměn se začalo mluvit spolu se vznikem programovatelných smluv neboli smart kontraktů a s nimi souvisejícími decentralizovanými aplikacemi (zkráceně DAaps). Kryptoměny druhé generace tak neslouží pouze k běžné výměně peněz, nýbrž tvoří platformu, na které lze naprogramovat téměř jakýkoli požadavek, smlouvu, který bude posléze neměnně uložen do blockchainu dané kryptoměny a automaticky vykonán. Můžeme tak například mezi dvěma stranami vytvořit smlouvu o vyhotovení díla, kdy domluvený obnos bude díky smart kontraktu uložen v síti a vyplacen zhotoviteli až poté, co dílo dodá a obě strany se shodnou na správnosti. Celá tato dohoda se po prvotním zadání provede zcela automaticky a bez nutnosti zprostředkování třetí stranou. Tyto platformy otevřely dveře do světa nového využití kryptoměn, a proto se považují za další vyspělejší generaci. Nejznámější a nejpoužívanější platformou kryptoměny druhé generace je Ethereum, které bude samostatně rozebráno v následující kapitole, kde také budou detailněji vysvětleny pojmy smart kontrakt a decentralizovaná aplikace [23][24][25][26].

Kromě Etherea zde exisují i další platformy s vlastním blockchainem, kterými jsou NXT, EOS, NEO, Lisk, a další.

4.1.4 Třetí generace

Třetí generace kryptoměn se stále vyvíjí a množina funkcí kterými by měly disponovat není nijak pevně ohraničena. První kryptoměnou, o které se diskutuje jako o průkopníkovi nové generace je kryptoměna s názvem Cardano, jejíž mince se označují ADA. Očekávání, které od 3. generace vyvstávají, jsou v základu definovány tímto projektem a výčet nadstandardních přínosů, jaké by měla 3. generace přinést, bude tedy vztažen k této konkrétní kryptoměně.

Cardano

Cardano je inovativní projekt, který se začal vyvíjet v roce 2015 a jeho prvotní vývoj trval celé dva roky. Na burzy se pak kryptoměna dostala v roce 2017 a od té doby se stále vyvíjí a prochází různými etapami aktualizací. Cardano staví na poznatcích kryptoměn první i druhé generace a snaží se poučit z jejich nedostatků, navíc pak přidává své vlastní nové funkcionality, které jsou postupně uváděny v provoz. Při vývoji se na kryptoměnu

pohlíželo ze tří různých perspektiv, a to z pohledu škálovatelnosti, interoperability a udržitelnosti. Kryptoměna je založena na dvouvrstvovém modelu, který odděluje ukládání samotné hodnoty od výpočtu, který k ní vede. Hlavní vrstva – Cardano Settlement Layer – tak slouží k uchovávání hodnoty podobně jako kryptoměny první generace, zatímco na druhé vrstvě – Cardano Computation Layer – probíhají výpočty k hodnotám vedoucí a mohou na ní vnikat i decentralizované aplikace a smart kontrakty typické pro druhou generaci. Co se týká interoperability, tvůrci přicházejí s odvážnou myšlenkou komunikace s blockchainy různých kryptoměn. Cardano by tak bylo schopno rozpoznávat data pocházející z jiných blockchainů a pracovat s nimi. Bylo by tak schopné ověřit i legitimnost transakce uskutečněné v jiné kryptoměně. Dále se snaží najít vhodný model práce s metadaty, díky nimž by bylo schopno na určité úrovni komunikovat i s běžnými bankovními institucemi. Co se týče škálovatelnosti, snaží se vyřešit problém, kdy s rostoucí základnou uživatelů různých kryptoměn přestávají být dostačující jejich technické možnosti a síť se tak stává zahlcenou (například rychlost ověřování transakcí). Svou technologii vývojáři přirovnávají k rozšířenému nástroji pro přenos dat – BitTorrentu, kde se přenosové schopnosti sítě zvyšují s rostoucí základnou uživatelů. S ohledem na udržitelnost přichází Cardano s přístupem financování z více zdrojů, kdy počáteční finance byly vybrány formou ICO, stejně jako u velké části kryptoměn, v prvních etapách vývoje je výzkum financován převážně soukromými institucemi IOHK, Emurgo a Cardano Foundation, které také drží správu nad většinou uzlů a dá se tak mluvit o dočasné centralizaci, avšak v dohledné době se počítá s ústupem v řízení těmito firmami a přechodem na plnou decentralizaci [27][28][29][30].

4.2 Forky – rozdělní kryptoměny

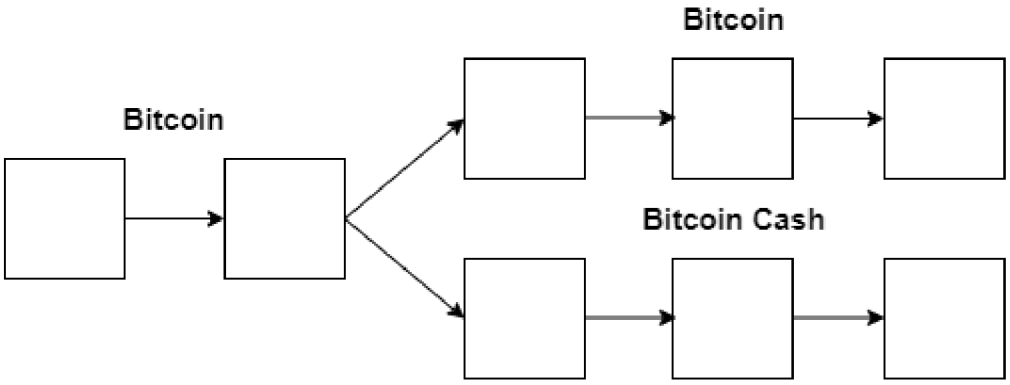
Fork, výraz do češtiny překládaný jako „vidlička“, značí nějakou dodatečnou úpravu v programovém kódu kryptoměny, která implementuje nové způsoby fungování. Podle míry dopadu, jaký má tato úprava na blockchainový řetězec rozlišujeme na Hard forky a Soft forky.

4.2.1 Hard fork

Hard fork kryptoměny často následuje potom, co dojde k rozepřím v uživatelské komunitě kryptoměny ohledně jejího dalšího technického vývoje. Zjednodušeně to znamená, že se blockchain jedné kryptoměny v určité chvíli rozpojí a vzniknou tak dvě na sobě nezávislá ramena jednoho původního blockchainu, z nichž ani jedno nezaniká, jako je tomu u nechtěného forku při vytěžení bloků různých těžařů ve shodném čase, ale na každé z nich se dále navazují nové bloky. Z toho vyplývá, že od této chvíle vzniká druhá samostatná kryptoměna. Děje se tak ve chvíli, kdy se početná část komunity rozhodne pro změnu některé zásadní funkcionality v protokolu, s níž druhá část komunity nesouhlasí. Decentralizovaný systém, na kterém jsou kryptoměny postaveny je tvořen mnoha samostatnými uzly, které musí být schopny spolu po datové stránce komunikovat

a docházet k vzájemnému konsenzu. Pokud se však část uživatelů rozhodne na svých uzlech pro upgrade protokolu, který s sebou nese zásadní změnu ve způsobu dosahování konsenzu, vzájemná shoda mezi uzly se starou verzí pak není možná. Rozdělení množství mincí pak funguje způsobem, že každý uživatel, který ve chvíli hard forku vlastnil nějaký obnos mincí původní měny, obdrží odpovídající množství mincí kryptoměny nové [13][31].

Jako příklad z praxe je možné uvést hard fork Bitcoinu, který nastal 1. srpna 2017. V uživatelské komunitě tehdy vznikla rozepře ohledně velikosti bloků. Část komunity byla pro zvýšení velikosti bloku z 1 MB na 8 MB, a neschopnost domluvy vedla až k hard forku na původní Bitcoin a odvozený Bitcoin cash [21].



Obrázek 4.1 Znázornění hard forku na Bitcoin cash.

4.2.2 Soft fork

Soft fork je oproti hard forku změna v protokolu, která je zpětně kompatibilní a uzly tedy nejsou nuceny na tuto verzi upgradovat, aby byla konektivita zachována. Nemodifikovaný klient vidí v tomto případě nově vzniklé bloky stále jako validní a je tak schopen pokračovat ve své práci v souladu s novými pravidly. Jako soft fork je však možné aplikovat pouze změny, které jsou pro bloky, jež jsou považovány za platné omezující, nikoliv ty, jež by bloky dříve označované za neplatné začaly považovat za platné, jako je tomu u hard forku. Ve skutečnosti tak k žádnému forku blockchainu nedochází, pouze k rozdělení uzlů, které upgrade přijali, a které ne.

Soft forky jsou obecně komunitou přijímány mnohem lépe než hard forky, napomáhá tomu i fakt, že již v původním kódu jsou zakomponovány volné instrukce, které slouží právě k účelu dodatečných nevynucovaných upgradů. Tyto instrukce označované jako NOP (no operation) nemají v původním kódu žádný účinek a neprovádějí se. Provádějí se až při dodatečné implementaci, která jim přidává na významu, a upgradu klientů. Pro klienty, kteří neupgradují na verzi obohacenou o tyto nové funkcionality zůstávají instrukce neprováděny. V zásadě to pak znamená že neupgradovaní klienti považují bloky za platné, aniž by je ověřili podle nově definovaných pravidel [13][23].

Jako příklad z praxe může být uveden soft fork Bitcoinu s mediálním označením Segwit (Segregated Witness) z roku 2017, který přišel s řešením, jak začlenit do bloku s omezenou velikostí větší množství transakcí než doposud tím, že transakci rozdělil na část se samotnými daty transakce a část podpisovou [32].

4.3 Coiny, Tokeny, Stablecoiny

Kryptoměny můžeme dále rozdělit podle toho, zda fungují na vlastním blockchainu, nebo pro svůj chod využívají blockchain jiné kryptoměny. Z tohoto pohledu se kryptoměny dělí na coiny a tokeny. Také lze na kryptoměny nahlížet s ohledem na volatilitu, kdy měny s nízkou volatilitou jsou označovány za stablecoiny.

4.3.1 Coiny

Jako coiny se označují kryptoměny, které běží na svém vlastním blockchainu. Můžeme tedy říci, že každá kryptoměna, která má vlastní blockchain je coin. Účelem coinů je především plnit zástupnou funkci peněz a být tedy prostředkem směny. Většina coinů se těží, avšak nemusí tomu tak být vždy, takže na toto kritérium nelze pohlížet jako na pravidlo. Coiny jsou i kryptoměny vzniklé hard forkem z jiného coinu, jelikož po hard forku každý z řetězců funguje nezávisle na sobě a s vlastními pravidly. Příkladem coinu je tedy Bitcoin, Ethereum, Bitcoin cash a mnoho dalších kryptoměn [27].

4.3.2 Tokeny

Tokeny na rozdíl od coinů nemají vlastní blockchain, a pro svůj chod využívají blockchain kryptoměnové platformy na které byly založeny. S tokeny se můžeme nejčastěji setkat jako s vnitřním platidlem některé decentralizované aplikace, nebo smart kontraktu. Na rozdíl od coinů je lze užít i k jiným účelům než pouze jako platidlo. Tokeny lze naprogramovat téměř k jakýmkoliv účelům a podle funkce jakou zastávají je můžeme rozdělit do 2 hlavních kategorií:

- **Bezpečnostní tokeny** – Mají podobný charakter jako běžné cenné papíry s tím rozdílem, že tyto jsou zapsány v blockchainu. Vlastnictví takového tokenu může opravňovat k podílu na zisku, nebo dávat majiteli jistá hlasovací práva v projektu, ke kterému byly vytvořeny. Nabídka a prodej takovýchto tokenů podléhá platným zákonům o cenných papírech. Bezpečnostní tokeny mohou být také navázány na hmotný majetek a dokazovat například vlastnictví obrazu.
- **Utility tokeny** – Často označované také jako užitkové, aplikační či uživatelské tokeny. Jejich vlastnictví opravňuje k přístupu k určité službě v projektu, ke kterému byly vydány. Utility tokeny jsou často využívány k platbám uvnitř decentralizovaných aplikací a smart kontraktů. Nejznámější platformou pro tvorbu tokenů je Ethereum, na které jsou většinou zakládány pod standardem ERC-20 [23][33][34][35].

4.3.3 Stablecoiny

Jako stablecoiny jsou označovány coiny nebo tokeny, které se snaží svou volatilitou přiblížit takzvaným tvrdým měnám (světové měny u nichž se předpokládá, že jejich cena zůstane dlouhodobě stabilní, například americký dolar, švýcarský frank…). Volatilita je neodmyslitelnou vlastností kryptoměn a je tak náročné ji stabilizovat. U stablecoinů je to řešeno krytím některou světovou měnou nebo komoditou, na kterou jsou navázány. Nejčastěji jsou stablecoiny vázány na dolar (Tether, TrueUSD), z komodit pak například na zlato (DigixGold). Spolu s krytím přichází otázka, zda lze tyto stablecoiny skutečně považovat za decentralizované kryptoměny, jelikož za jejich provozem a s tím spojenou stabilitou musí stát správce, který drží deklarovaný obnos peněz (nebo komodit) na účtu [36][37][38].

5. ETHEREUM

Ethereum je kryptoměnová platforma, která se vývojově řadí do 2. generace kryptoměn. Byla vyvíjena skupinou programátorů, z nichž nejznámější jsou Joseph Lubin, Gavin Wood, který vytvořil skriptovací jazyk Solidity pro tvorbu inteligentních smluv, a Vitalik Buterin. V médiích nejskloňovanějším jménem ohledně platformy Ethereum se však stalo především jméno Vitalika Buterina, který se pravidelně účastní mediálních rozhovorů.

Vitalik Buterin je rusko-kanadský programátor, a právě díky jeho iniciativě se začalo Ethereum vyvíjet. Vitalik měl vizi toho, že blockchain jako pear-to-pear nástroj nemusí sloužit kryptoměnám pouze pro jednoduchý přenos digitálních aktiv, ale má mnohem větší potenciál využitelný pro vytváření smart kontraktů a decentralizovaných aplikací. Původně s tímto návrhem konfrontoval bitcoinovou komunitu vývojářů, a byl pro rozšíření Bitcoinu o tyto funkcionality. Setkal se však s nepochopením a odmítavým postojem, jelikož aby tyto funkcionality byly schopné provozu, musel by Bitcoin přijmout daleko komplexnější turingovsky kompletní skriptovací jazyk, který by s sebou podle komunity nesl jistá bezpečnostní rizika. Proto byla roku 2014 uvedena na trh zcela nová kryptoměna a inovativní platforma pro tvorbu smart kontraktů Ethereum [35][39][40].

Ethereum je název kryptoměnové platformy, v níž se používá měna ether. Ether je dále dělitelný a jeho nejmenší jednotka nese jméno wei. Pro převodní hodnotu pak platí, že 1 ether = 10^18 wei. V běžném provozu se často používá označení gwei (giga wei) což je 10^9 wei. Pokud chceme vytvářet smart kontrakty, musíme určité množství etherů vlastnit, jelikož veškeré poplatky síti za provedení smart kontraktu jsou účtovány v etherech [41].

5.1 Smart kontrakty

Smart kontrakt, do češtiny překládaný jako inteligentní smlouva, můžeme definovat jako drobný počítačový program, který slouží k převodu digitálních aktiv způsoby v něm definovaném. Po svém spuštění, které nastává zapsáním na blockchain je plně transparentní a prováděný zcela automaticky. Lze o něm říci, že zastává funkci běžně uzavíratelných smluv, s tím rozdílem, že po zapsání na blockchain je již neměnný a k vykonání dohod v něm uložených nepotřebuje dohled žádné třetí strany. Ve světě kryptoměn používáme smart kontrakty především k výměně peněžních prostředků, ale jejich potenciál přesahuje běžné peněžní či jiné převody a svou formou by mohly být využity například při ukládání záznamů, u nichž je vyžadována neměnnost a transparentnost, například pro uložení volebního hlasování, správě majetku a jiné [35][42][43].

Pokud využijeme některou z připravených šablon pro smart kontrakty, které nabízejí právě platformy kryptoměn druhé generace, je vytvoření kontraktu velmi jednoduché. V principu se ale při komplexním pohledu mohou vyskytnout problémy související s tím,

že smart kontrakty zatím nejsou zadefinovány v právním systému žádného státu, a tudíž nejsou vymahatelné. Také je třeba vzít v úvahu složitost prvotního naprogramování smart kontraktu a neomylnost lidského faktoru, což pak může mít při nemožnosti opravy již spuštěného kontraktu velké následky [43].

5.2 Decentralizované aplikace

Oficiální definice, která by přesně definovala decentralizované aplikace není stanovena. Můžeme však říci, že jsou to počítačové programy, které pro svůj chod využívají decentralizovanou síť a běží na blockchainu. Vnitřní běh aplikace je zabezpečen pomocí smart kontraktů. Decentralizovaná aplikace je tak tvořena souborem smart kontraktů, které jsou nadefinovány k rozličným funkcím. Decentralizované aplikace disponují pár základními vlastnostmi. Jejich kód musí být ve formě open-source, nemají centrální autoritu a pro platby a poplatky uvnitř aplikace často využívají vlastní token [35][44][45].

5.3 Poplatky v síti

Chod decentralizované sítě Etherea zajišťují stejně jako u Bitcoinu těžaři. Aby byli motivováni tak činit a práce se jim vyplatila, je každý úkon zapisovaný na blockchain zatížen poplatkem. Poplatek je v zásadě tím vyšší, čím větší množství práce musí být pro zapsání požadavku na blockchain provedeno. Tedy čím větší množství dat na zpracování transakce obsahuje, tím větší množství práce musí být vykonáno a úměrně s objemem vykonané práce roste i poplatek. Pro usnadnění orientace v objemu vykonané práce byla stanovena jednotka gas (jednotka plynu). Jeden gas je elementární jednotkou a udává nejmenší možné množství vykonané práce. Gas je pouze jednotkou kvantifikující vykonanou práci a nemá žádnou peněžní hodnotu. V tomto systému jsou nejpodstatnější 2 hodnoty:

- **Gas price** – cena za jednotku gas. Cenu, jakou je za jednotku ochoten zaplatit si stanovuje sám zadavatel transakce. Čím vyšší tato cena bude, tím rychleji bude transakce zpracována, jelikož je pro těžaře lákavější. Cena za jeden gas se udává v jednotkách gwei.
- **Gas limit** – udává, kolik je zadavatel za transakci ochoten spálit jednotek gas. Je to v zásadě odhad, jakého množství práce je potřeba pro zapsání transakce na blockchain. Tyto odhady bývají dosti složité, proto je pro mnohé typy transakcí stanoveno přibližné množství.

Celkový poplatek se pak vypočítá jako součin těchto dvou hodnot. Pokud bude hodnota gas limit stanovena níže, než kolik jednotek je skutečně potřeba, dojde ve chvíli jejího vyčerpání k zastavení prováděné transakce, a zadavatel přijde o částku za poplatek, aniž by byla jeho transakce skutečně dokončena. Pokud však bude hodnota stanovena výše, bude zadavateli přeplatek vrácen [46][47].

6. VÝUKOVÁ APLIKACE

Zadáním praktické části této bakalářské práce bylo realizování výukové aplikace, na v práci rozebrané téma blockchainu a virtuálních měn, na vhodné blíže nespecifikované platformě a sestavení laboratorní úlohy zaměřené na vytvoření vlastního „blockchainu“. Tato kapitola popisuje tvorbu výukové aplikace, na níž je poté odkazováno v úvodu níže popsané laboratorní úlohy.

Výuková aplikace byla konceptuálně pojata jako webová stránka a umístěna na adrese https://sites.google.com/vutbr.cz/virtualni-meny. Jako platforma pro vytvoření webové stránky bylo použito rozhraní nabízené od společnosti Google, Google sites. Google sites je volně dostupný nástroj a šablona pro snadnou tvorbu webových stránek poskytovaný držitelům účtu google, který zároveň plní i funkci webového hostingu, a není tak třeba mít vlastní doménu. Disponuje velice jednoduchým a intuitivním ovládáním s možností vkládání různých funkcionálních prvků, jejichž výčet sice není převratný, ale pro účely této výukové aplikace plně dostačující.

6.1 Použité nástroje

Na stránce se nachází množství obrázků, které byly vytvořeny ve volně dostupném online editoru určenému pro tvorbu diagramů, diagrams.net, který je propojitelný s účtem google a vytvořené diagramy lze tak ukládat přímo do cloudu google disku. Z obrázků s vhodnou strukturou byly vytvořeny jednoduché animace, pro jejichž tvorbu byl použit editor pro tvorbu gif animací dostupný na stránce https://ezgif.com/.

6.1.1 Google sites

Jak již bylo zmíněno, Google sites je bezplatný nástroj pro tvorbu webových stránek a aplikací, který uživateli nabízí řadu možností k vytvoření vlastní webové aplikace i bez znalosti programování. Při tvorbě webu má uživatel možnost si zvolit některý z předpřipravených motivů, kdy grafická šablona motivu má již v sobě zakomponovány veškeré detaily, co se použitých fontů či barevného schématu týče. Toto se může jevit jako jistá nevýhoda, protože uživatel je tak omezen v základním přizpůsobení grafického vzhledu stránky a musí přijmout šablonu s jejím celkovým vzhledem bez možnosti rozsáhlejších úprav.

Tvorba samotné aplikace probíhá pomocí vkládání jednotlivých prvků do stránky. Tyto předpřipravené prvky jsou uživateli nabízeny z jednoduchého a přehledného menu. K tvorbě vlastních prvků stránky je možné využít funkcionality vložení tzv. „embedded“ kódu. Přičemž má uživatel na výběr, zdali chce do stránky vložit vlastní HTML kód, či přes URL adresu vložit již existující web. Takto vložený HTML dokument samozřejmě podporuje i použití CSS a JavaScriptu.

Velkou výhodou Google sites je, že veškeré weby vytvářené na této platformě jsou již v základu konceptuálně pojaty jako „mobile first“, a není tak potřeba řešit uspořádání stránky zvlášť pro případné zobrazení mobilními zařízeními.

6.2 Struktura aplikace

Webová aplikace byla rozvržena do kapitol, které obsahově odpovídají tématům rozepsaným v teoretické části bakalářské práce a doplněna o kapitolu „Cvičení“, ve které je navržen krátký test. Názvy kapitol tak obsahují:

- ***Základní pojmy***
- ***Blockchain***
- ***Konsensuální algoritmy***
- ***Obecně o kryptoměnách***
- ***Bitcoin***
- ***Ethereum***
- ***Cvičení***

Kapitoly „Bitcoin“ a „Základní pojmy“ jsou tvořeny formou delšího souvislého textu na jedné stránce rozčleněného na podkapitoly. Aby byla orientace na takto vytvořených stránkách snazší, obsahují tyto kapitoly v menu přidružené rozbalovací seznamy, jejichž položky odkazují na pozici příslušné podkapitoly na stránce. Jelikož takovýto typ odkazování platforma přímo nenabízí, bylo toto realizováno pomocí vložení externích odkazů (odkazů na externí URL adresy) na jednotlivé části stránek.

Ke kapitole „Obecně o kryptoměnách“ je také přiřazen rozbalovací seznam. Jelikož však témata v ní uvedená spolu nejsou tak úzce spjata jako u předchozích kapitol, jsou podkapitoly vážící se k tomuto tématu řešeny samostatnými podstránkami.

6.3 Grafický návrh aplikace

Při návrhu výukové aplikace bylo potřeba si nejprve ujasnit rozvržení grafických prvků, při respektování omezení plynoucích z použitého prostředí Google Sites. Základní rozložení aplikace je totiž předem definováno a uživateli nabízí možnosti customizace jen ve velmi omezené míře. Toto se netýká jen barevných schémat a fontů, jak již bylo zmíněno výše, ale zejména primární navigace v aplikaci. Ta je omezena jen na horní lištu, případně levý postranní panel. Stejně tak záhlaví webu může být implementováno jen dle předem daných pravidel. Každá ze stránek aplikace by sice mohla mít záhlaví pojato jinak, nicméně toto by narušovalo celistvost aplikace a pro uživatele působilo rušivě. Z výše zmíněného tak vyplývá, že při návrhu aplikace bylo nutné zvážit zejména rozmístění prvků v obsahu jednotlivých stránek.

6.4 Implementace aplikace

Aplikace byla vytvořena převážně použitím standardních prvků poskytnutých vývojovým prostředím. Textové prvky jsou v aplikaci vytvořeny vložením prvku „Sbalitelný text“ a nastavením možnosti „nesbalitelné“ na hlavičce prvku. Do podkapitoly „Hashovací funkce“ je pro názornou ukázku fungování hashovacích funkcí vložen externí web ze stránky https://emn178.github.io/online-tools/sha256.html. Toto bylo implementováno pomocí funkcionality vloženého kódu. Po vložení URL se v rámci tvorby aplikace tato externí stránka jeví a chová jako standardní prvek a je tak možné s ní ve stránce manipulovat (přesouvat, či měnit její velikost).

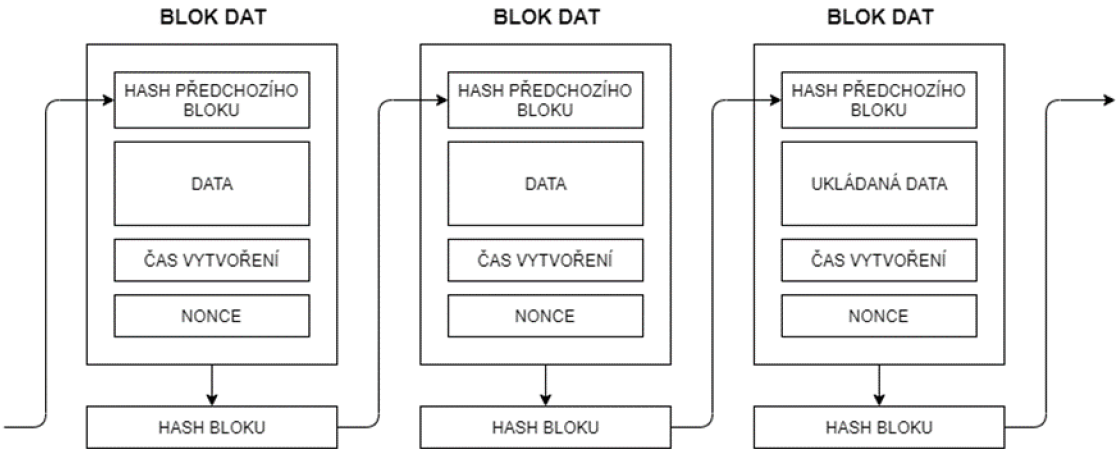
7. LABORATORNÍ ÚLOHA

7.1 Demonstrace blockchainu v jazyce Python

V tomto cvičení vytvoříme zjednodušený model blockchainu, který k přidávání nových bloků využívá konsensuálního algoritmu Proof-of-Work.

Blockchain je řetězec na sebe navazujících bloků, který slouží k bezpečnému uchovávání dat a zajištění jejich autentičnosti, transparentnosti a nepopiratelnosti. V podstatě je to forma distribuované databáze, ve které můžeme vyhledávat a číst již uložené informace, vkládat informace nové, ale je znemožněna jakákoliv zpětná úprava již uložených dat. K zajištění blockchain se používá hashovací funkce a mechanismus postupného navazování bloků dat do řetězce, kdy právě navazovaný blok obsahuje ve své hlavičce informaci o bloku předchozím (jeho hash), což zajišťuje jejich neměnné uspořádání.

Součástí každého bloku je časový údaj o jeho vzniku, hash předchozího bloku dat, aktuálně ukládaná data a náhodné číslo označované jako nonce. Na takto vytvořený blok je poté aplikována hashovací funkce. Výsledný hash musí splňovat určitá předem daná kritéria, v tomto cvičení to bude stejně jako u bitcoinu počet nul na začátku hashe v hexadecimální soustavě.



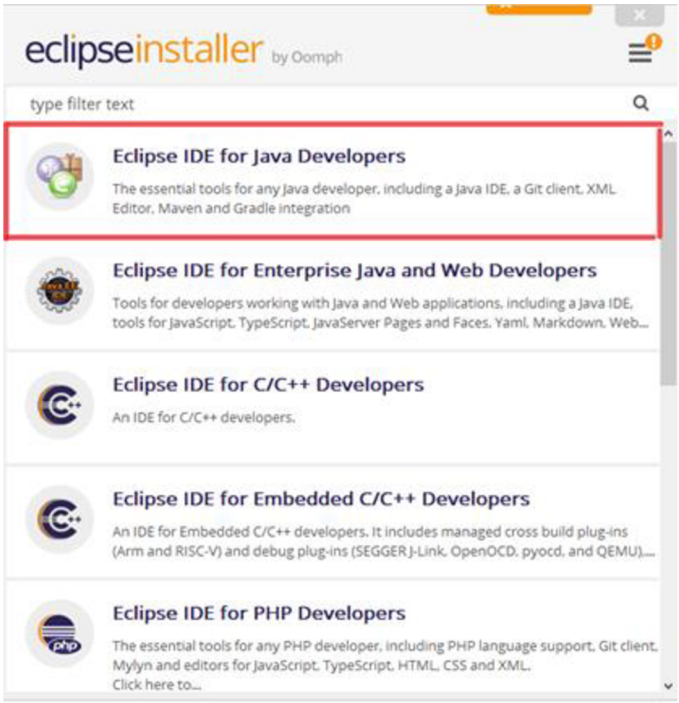
Obrázek 7.1 Znázornění principu zřetězování bloků dat.

Pro detailnější seznámení s fungováním kryptoměn můžete navštívit webovou stránku vytvořenou k tomuto cvičení k prohloubení teoretické znalosti o fungování kryptoměn na adrese: https://sites.google.com/vutbr.cz/virtualni-meny. Stránka je vytvořená pod doménou školy, proto je k bezproblémovému přístupu nutné být v prohlížeči přihlášen do školního účtu VUT.

7.1.1 Příprava pracoviště

Jako vývojové prostředí bylo zvoleno Eclipse, které je možné volně stáhnout z jeho oficiálních stránek. Odkaz pro stažení verze z března 2021: https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2021-03/R/eclipse-inst-jre-win64.exe

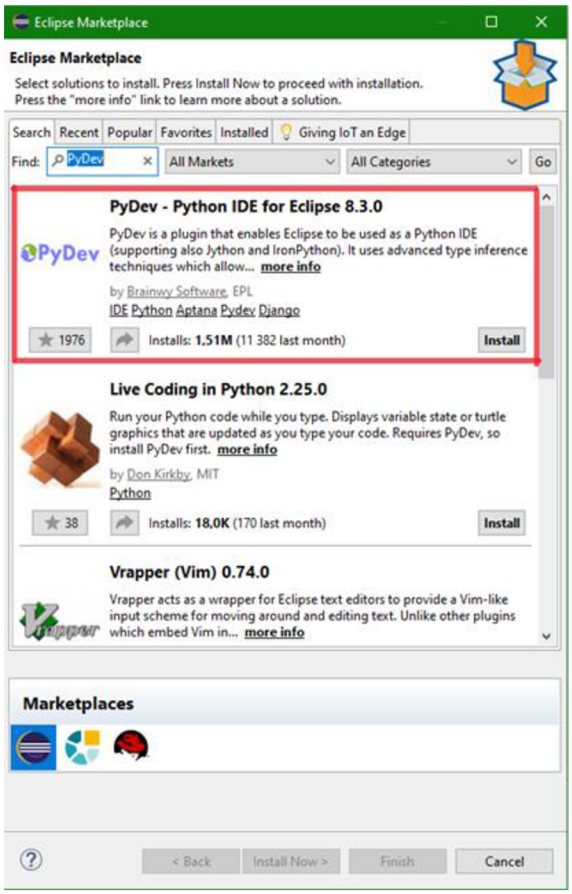
Pro účely úlohy je doporučeno při instalaci zvolit první možnost viz obrázek 7.2.



Obrázek 7.2 Doporučená volba při instalaci vývojového prostředí Eclipse.

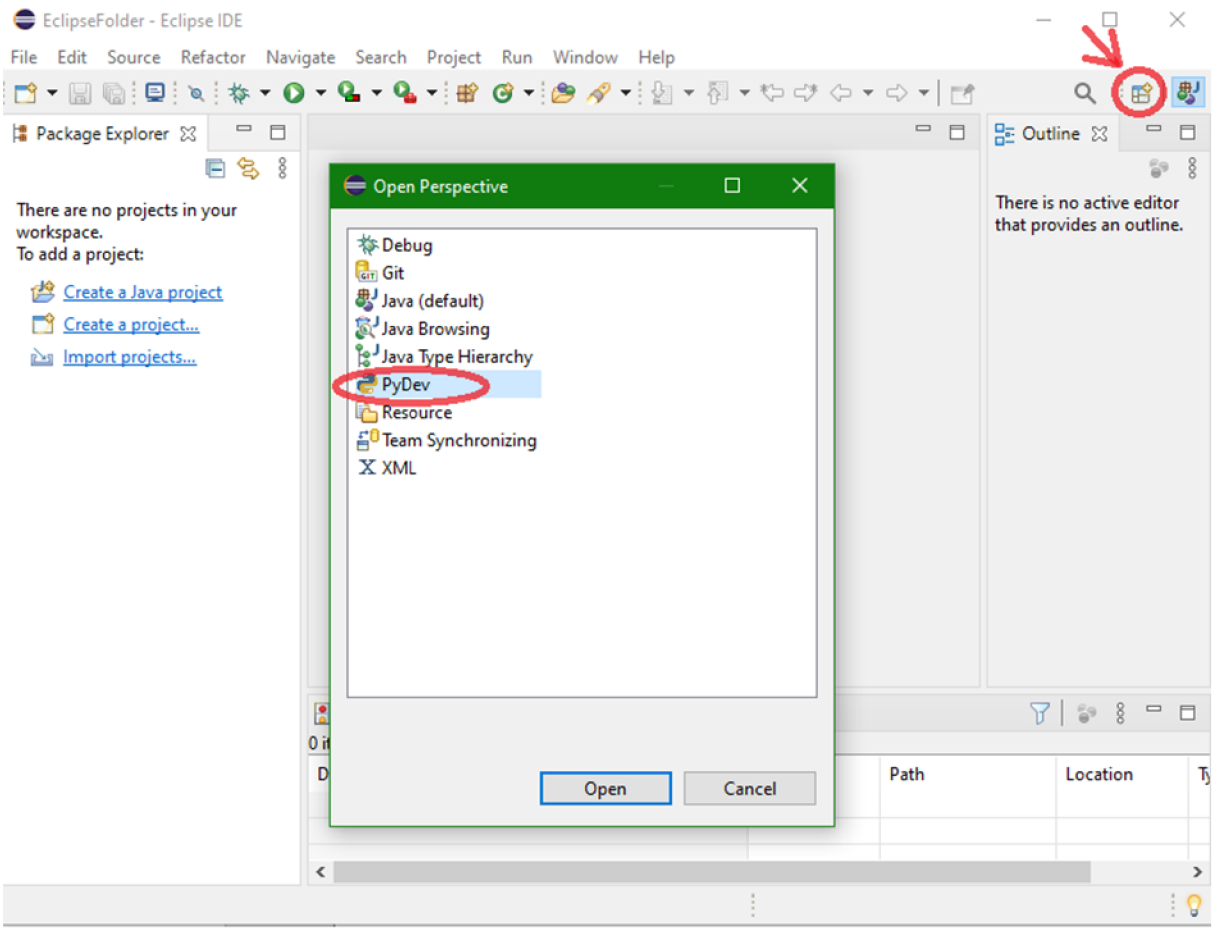
Dále je nutné na PC nainstalovat podporu pro programovací jazyk Python v aktuální verzi, která je ke stažení z oficiálních stránek: https://www.python.org/downloads/

Do vývojového prostředí Eclipse je zapotřebí přidat plugin, který nám umožní Eclipse využívat jako vývojové prostředí pro jazyk Python. Pro potřeby této práce bude využito doplňku PyDev, který lze stáhnou přímo z oficiálního Eclipse Marketplace (viz obrázek 7.3). V horní liště je potřeba pod záložkou Help vyhledat možnost Eclipse Marketplace. Po kliknutí se nám Marketplace otevře v novém okně. Pak jen stačí do vyhledávače zadat "PyDev" a provést instalaci stiskem tlačítka Install. Poté je potřeba Eclipse restartovat.



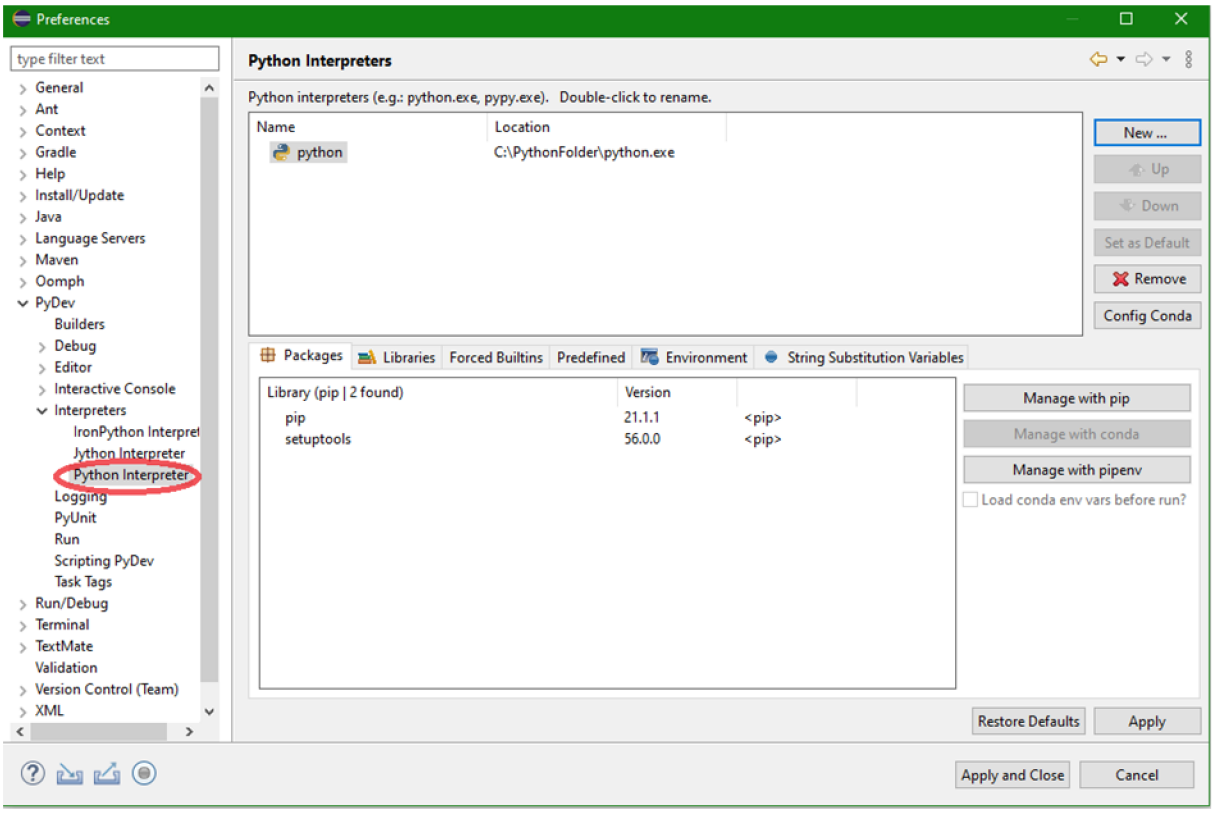
Obrázek 7.3 Instalace pluginu PyDev v Eclipse.

Dále pro snažší navigaci při práci s Pythonem nastavíme Perspektivu PyDev. V pravém horním rohu klikneme na ikonu „Open Perspective“ a vybereme rozložení PyDev viz obrázek 7.4.



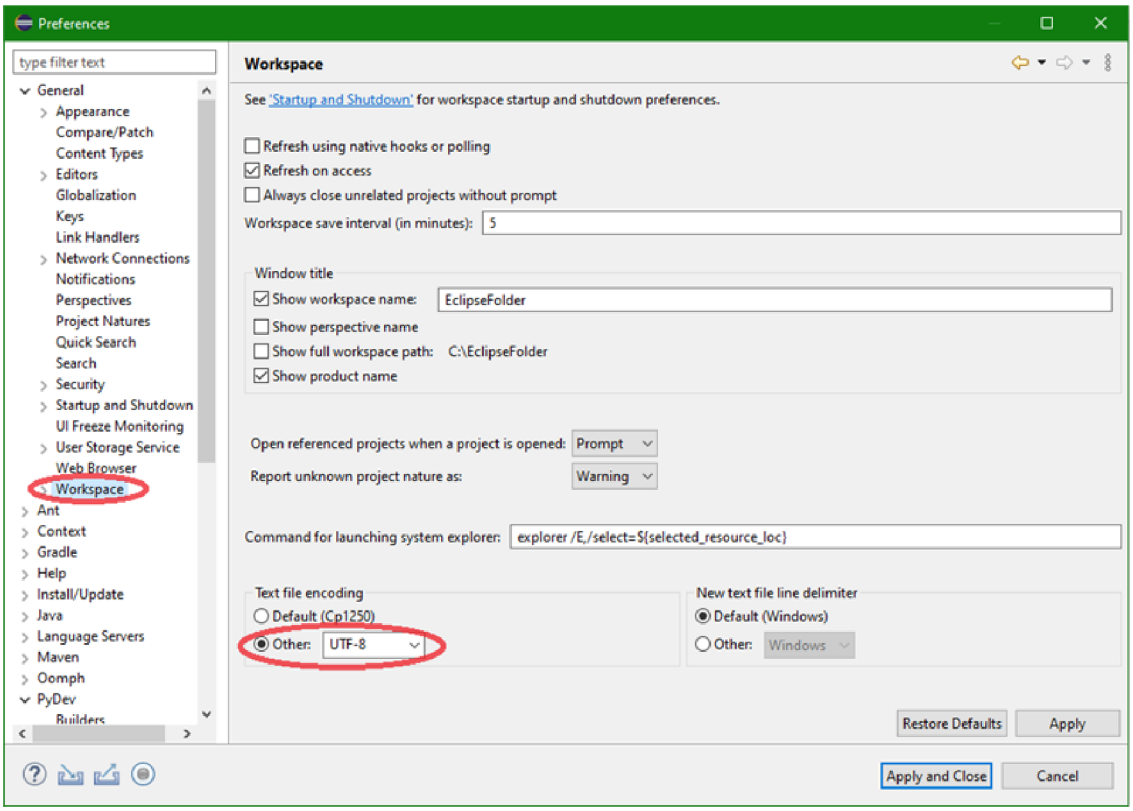
Obrázek 7.4 Nastavení Python perspektívy v Eclipse.

V dalším kroku přidáme cestu k python interpreteru, který jsme výše stáhli ze stránek https://www.python.org/downloads/. V horní liště klikneme na Window → Preferences. Tam rozklikneme záložku PyDev → Interpreters → Python Interpreter. V pravém rohu klikneme na New a zadáme cestu k umístění, kde je nainstalovaný soubor python.exe. Tlačítkem „Apply“ aplikujeme změny.



Obrázek 7.5 Nastavení cesty k Python interpreteru.

V Preferences ještě zůstaneme a nastavíme podporu kódování UTF-8 viz obrázek 7.6, které není defaultně nastaveno a je potřebné pro bezproblémový překlad českých znaků. V záložce General → Workspace dole zatrhneme kódování Other a vybereme UTF-8. Změny aplikujeme a můžeme uzavřít.



Obrázek 7.6 Nastavení kódování UTF-8.

Nyní již můžeme vytvořit nový Python projekt: File → New → PyDev Project, který libovolně pojmenujeme. V něm vytvoříme PyDev modul: New → PyDev module, ve kterém budeme program vytvářet.

7.1.2 Stavba bloku

Na úvod naimportujeme do vytvořeného modulu knihovny, jejichž funkce budeme v programu používat. Z knihovny *time* budeme používat funkce pro určení času. (součástí bloku je i informace o čase jeho vytvoření). Dále naimportujeme knihovnu *json*. Tato knihovna nám díky metodě *json.dumps()* umožní konverzi Python objektů do textových řetězců. Tohoto bude využito zejména u hashovacích funkcí. Dále budeme využívat hashovací funkci sha256, která je připravená v modulu hashlib.

```
from hashlib import sha256;
import json;
import time;
```

Jako první vytvoříme třídu *Block*. Všechny objekty demonstrující bloky blockchainu pak budou tohoto typu. Třída *Block* bude obsahovat inicializační metodu *__init__*, kde definujeme všechny parametry, které by měl blok obsahovat.

```
class Block(object):

    def __init__(self, height, transactions, timestamp, previousHash,
nonce = 0):

        'výška (pořadí) bloku v řetězci'
        self.height = height
        'data/transakce obsažené v bloku'
        self.transactions = transactions
        'čas vytvoření bloku'
        self.timestamp = timestamp
        self.previousHash = previousHash
        self.nonce = nonce
        'hash bloku, k vytvoření se zavolá metoda calculateHash()'
        self.hash = self.calculateHash()
```

Dále vytvoříme metodu *toDict*, která vrátí reprezentaci bloku jako Python objektu. Tato metoda se potom využívá při vytváření hashe, jelikož funkce sha256 potřebuje jako vstupní parametr string.

```
    def toDict(self):
        return {
            'height' : self.height,
            'transactions' : self.transactions,
            'timestamp' : self.timestamp,
            'previousHash' :  self.previousHash,
            'nonce' : self.nonce
            }
```

Poslední metodou třídy *Block* bude metoda, která na blok aplikuje hashovací funkci sha256 a vrátí hodnotu hashe. Abychom dostali výslednou hodnotu v hexadecimálním tvaru, použijeme funkci hexdigest.

```
    def calculateHash(self):
        'json.dumps je utilita Pythonu vracející string řetězec.'
        'Funkce hexdigest vrátí výstup v hexadecimálním zápisu'
        return                          sha256(json.dumps(self.toDict(),
sort_keys=True).encode()).hexdigest()
```

Úkol č. 1

Vytvořte python *main* metodu a napište příkazy, které vytisknou do konzole první dva bloky blockchainu obsahující patřičné parametry (zatím bez ohledu na složitost těžby, tj. nonce = 0)

Pozn. Nultý blok nemůže obsahovat hash předchozího bloku.

Možné řešení:

```
if __name__ == '__main__':
    zeroBlock = Block(0, 'první transakce', time.time(), [])
    firstBlock    =    Block(1,    'další    transakce',    time.time(),
zeroBlock.calculateHash())
    print("nultý blok: ", zeroBlock.toDict())
    print('Hash nultého bloku je: ', zeroBlock.calculateHash())
    print("první blok: ", firstBlock.toDict() )
    print('Hash prvního bloku je: ', firstBlock.calculateHash())
```

7.1.3 Těžba – tvorba blockchainu

Další třídou v modulu bude třída *Blockchain*. Tato třída bude obsahovat metody, které zjednodušeně demonstrují tvorbu blockchainu blok po bloku. Inicializační metoda bude obsahovat tyto globální proměnné. Obtížnost se inicializuje při vytvoření objektu blockchain.

```
class Blockchain(object):

    def __init__(self, difficulty):

        'pole transakcí, které čekají na "vytěžení" (zatím nebyly
přiřazeny do žádného bloku'
        self.transaction = []
        'blockchain řetězec, pole objektů třídy Block'
        self.chain = []
        'obtížnost  těžby,  inicializuje  se  při  vytvoření  objektu
Blockchain'
        self.difficulty = difficulty
        'při vytvoření objektu Blockchain se automaticky vytvoří nultý
(Genesis) blok řetězce'
        self.createGenesisBlock()
```

Dále vytvoříme metodu *proofOfWork* s while cyklem pro hledání čísla nonce takového, aby výsledný hash svým tvarem odpovídal zvolené obtížnosti. Jako parametr budeme předávat objekt třídy block, nad kterým se tato metoda bude provádět.

```
    'předání objektu třídy blok jako parametru metody'
    def proofOfWork(self, block):
        'blockHash jako lokální proměnná v metodě proofOfWork'
        blockHash = block.hash
        'startswith  -  defaultní  metoda  na  string  řetězci,  vrací
návratovou hodnotu boolean'
        while not blockHash.startswith('0' * self.difficulty):
            'inkremetace'
            block.nonce += 1
            blockHash = block.calculateHash()
        return blockHash
```

V dalším kroku vytvoříme metodu *createGenesisBlock*, která se provádí při vytváření objektu třídy blockchain. V metodě inicializujeme objekt třídy Blok *genesisBlock,* nad kterým poté provádíme metodu *proofOfWork*.

```
    'metoda přidá do pole chain v inicializační metodě na začátek nultý
blok blockchainu'
    def createGenesisBlock(self):
        'nová instance třídy block a přidá se do pole'
        'import time, genesis nemá předchozí hash bloku proto 0'
        genesisBlock = Block(0, [], time.time(), "0")
        genesisBlock.hash = self.proofOfWork(genesisBlock)
        'funkce append zařadí blok na konec řetězce chain'
        self.chain.append(genesisBlock)
```

Přidáme metodu, která vrátí poslední blok v řetězci…

```
    def getLastBlock(self):
        'záporný index vrátí poslední objekt v poli'
        return self.chain[-1]
… metodu pro kontrolu, zda je poslední vytěžený blok validní…
    'kontrola zda je právě vytěžený blok validní'
    def isActualHashValid(self, block, actualHash):
        return (
            actualHash.startswith('0' * self.difficulty) and actualHash
== block.calculateHash()
            )
```

… metodu pro kontrolu validity celého řetězce

```
    'kontrola zda je celý blockchain validní'
    def isChainValid(self):
        for block in self.chain[1:]:
            if  block.previousHash  !=  self.chain[  block.height  -
1].calculateHash() : return False
            if block.hash != block.calculateHash() : return False

        return True
```

O připojování vytěženého bloku do řetězce se bude starat metoda *addBlock*, ve které se ověří, pomocí dříve vytvořené metody *isActualHashValid*, jestli je hash bloku, který chceme přidat platný a následně se připojí k řetězci.

```
    def addBlock(self, block, hash):
        if self.getLastBlock().hash != block.previousHash : return False
        if not self.isActualHashValid(block, hash) : return False
        block.hash = hash
        self.chain.append(block)
        return True
```

Jako poslední metodu vytvoříme metodu *mine*. Tato metoda slouží k samotné těžbě bloku. V těle metody dochází k vytvoření nového bloku, do kterého se zařadí všechny

nevytěžené transakce. Metoda *addBlock* je popsána výše. Nakonec dojde k inicializaci pole nevytěžených transakcí.

```
    def mine(self):
        block = Block(self.getLastBlock().height + 1, self.transaction,
time.time(), self.getLastBlock().hash)
        self.addBlock(block, self.proofOfWork(block))
        self.transaction = []
```

Ve spustitelné metodě poté vytvoříme objekt typu *Blockchain* s parametrem složitosti těžby. Na tomto objektu pak zavoláme metodu *mine*. Pokud bychom chtěli zjistit dobu trvání těžby bloku, můžeme k tomu využít například metody *perf_counter* z knihovny *time*.

```
if __name__ == '__main__':
    blockchain = Blockchain(2)
    print("Těžím 1. blok: ")
    startTime = time.perf_counter()
    blockchain.mine()
    print (f"Blok vytěžen za {time.perf_counter() - startTime:0.4f}
vteřiny")
    print("Těžím 2. blok: ")
    startTime = time.perf_counter()
    blockchain.mine()
    print (f"Blok vytěžen za {time.perf_counter() - startTime:0.4f}
vteřiny")
    print (f"Je blockchain validní? " + str(blockchain.isChainValid()))
    for block in blockchain.chain:
        print(block.__dict__)
```

Úkol č. 2

Vytvořte ve třídě Blockchain metodu, která umožní přidávání transakcí do dalšího vytěženého bloku.

Řešení:

```
    def addNewTransaction(self, transaction):
        self.transaction.append(transaction)
```

výstup pak může vypadat takto:

```
if __name__ == '__main__':
    blockchain = Blockchain(2)
    blockchain.addNewTransaction("První transakce")
    blockchain.addNewTransaction("Druhá transakce")
    print("Těžím 1. blok: ")
    startTime = time.perf_counter()
    blockchain.mine()
    print (f"Blok vytěžen za {time.perf_counter() - startTime:0.4f}
vteřiny")
    blockchain.addNewTransaction("Třetí transakce")
    blockchain.addNewTransaction("čtvrtá transakce")
```

```
    print("Těžím 2. blok: ")
    startTime = time.perf_counter()
    blockchain.mine()
    print (f"Blok vytěžen za {time.perf_counter() - startTime:0.4f}
vteřiny")
    print (f"Je blockchain validní? " + str(blockchain.isChainValid()))
    for block in blockchain.chain:
        print(block.__dict__)
```

ZÁVĚR

Tato bakalářská práce pojednávala o tématu virtuálních měn. Jejím cílem bylo poskytnout přehledný a komplexní obraz o virtuálních měnách založených na technologii blockchain a vytvořit výukovou aplikaci podtrhující rozebranou látku. Ačkoliv se technologie decentralizované blockchainové sítě stojící za fungováním kryptoměn zrodila v poměrně nedávné době, jsou kryptoměny velice rozsáhlým a rychle se vyvíjejícím odvětvím. Bylo by velice složité, ba téměř nemožné, pokud bychom chtěli podrobněji prozkoumat detaily všech konkrétních kryptoměn a orientovat se v jejich tisících zástupcích a nově vznikajících projektech. Nicméně ačkoliv je vlastních kryptoměn velmi mnoho, většina z nich pracuje na podobném základu a liší se pouze v detailech. Dá se tak říci, že pokud pochopíme obecný princip fungování, získáme tím celkový vhled do tohoto systému. Tato bakalářská práce se tak z většinové části zaobírala právě základními principy fungování, které byly rozebrány na kryptoměně Bitcoin.

V první kapitole byly pro lepší pochopení problematiky vysvětleny některé z často využívaných kryptografických mechanismů. Následovala kapitola Blockchain, kde byla tato technologie stručně popsána, byl znázorněn proces napojování bloků a uvedeny tři nejvyužívanější algoritmy pro dosahování konsenzu. Třetí kapitola zaobírající se Bitcoinem obsahovala nejprve celkový popis činnosti celého systému a v následných podkapitolách pak byly detailněji popsány jednotlivé mechanismy a výklad byl doplněn o nákresy. Čtvrtá kapitola se již dívala na komplex kryptoměn z větší perspektivy. Byla zde uvedena možná kategorizace kryptoměn na generace a vysvětleny v souvislosti s kryptoměnami často užívané pojmy jako coin, token a stablecoin. Pátá kapitola Ethereum byla vložena jako základ pro vysvětlení pojmů smart kontrakty a decentralizované aplikace. Dále v ní bylo také nastíněno, jakým způsobem jsou smart kontrakty zpoplatňovány.

V rámci praktické části bakalářské práce byla vytvořena výuková aplikace formou webové stránky a sestavena laboratorní úloha na vytvoření zjednodušeného blockchainu v jazyce Python. Popis tvorby výukové aplikace a použité nástroje byly uvedeny v šesté kapitole. Sedmou kapitolu pak tvoří zmíněná laboratorní úloha.

LITERATURA

[1] DOHNÁLEK, Libor, Marta VOHNOUTOVÁ a Miroslav KNOTEK. *Velký průvodce infrastrukturou PKI*. 2. aktualizované vydání. Brno: Computer Press, a.s, 2009. ISBN 978-80-251-2619-6.

[2] SHA256 Hash Generator. *Passwordsgenerator* [online]. [cit. 2019-12-02]. Dostupné z: https://passwordsgenerator.net/sha256-hash-generator/

[3] STROUKAL, Dominik a Jan SKALICKÝ. *Bitcoin peníze budoucnosti*. Praha: Ludwig von Mises Institut CZ&SK, 2015. ISBN 978-80-87733-26-4.

[4] Binary-to-text encoding. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-12-02]. Dostupné z: https://en.wikipedia.org/wiki/Binary-to-text_encoding

[5] How to convert binary to hex. *Rapidtables* [online]. [cit. 2019-12-02]. Dostupné z: https://www.rapidtables.com/convert/number/how-binary-to-hex.html

[6] Blockchain tutorial 13.1: Base-58 encoding. In: Youtube [online]. 21.7. 2017 [vid. 2019-12-11]. Kanál uživatele Mobilefish.com. Dostupné z: https://www.youtube.com/watch?v=GedV3S9X89c

[7] HABER, Stuart a W. Scott STORNETTA. How to Time-Stamp a Digital Document. *Anf* [online]. [cit. 2019-12-01]. Dostupné z: https://www.anf.es/pdf/Haber_Stornetta.pdf

[8] FURNEAUX, Nick. Investigating cryptocurrencies: understanding, extracting, and analyzing blockchain evidence. Idianapolis, IN: Wiley, [2018]. ISBN 978-1-119-48058-7.

[9] ANWAR, Hasib. *Consensus Algorithms: The Root Of The Blockchain Technology* [online]. 25.8.2018 [cit. 2019-12-21]. Dostupné z: https://101blockchains.com/consensus-algorithms-blockchain/#prettyPhoto/0/

[10] NAKAMOTO, Satoshi. Bitcoin: A Peer-to-Peer Electronic Cash System. Bitcoin [online]. 2009 [cit. 2019-12-01]. Dostupné z: https://bitcoin.org/bitcoin.pdf

[11] Domain Name Registration Data Lookup [online]. [cit. 2019-11-03]. Dostupné z: https://lookup.icann.org/lookup

[12] OUPICKÝ, Jan. Jak funguje bitcoin [online]. [cit. 2019-12-17]. Dostupné z: http://www.karlin.mff.cuni.cz/~tuma/Aplikace17/Prace/btc_oupicky_oprava.pdf

[13] ANTONOPOULOS, Andreas M. *Mastering Bitcoin: programming the open blockchain*. Second edition. Sebastopol, CA: O'Reilly, 2017. ISBN 14-919-5438-8.

[14] MUKHI, Vijay. THE UNDOCUMENTED INTERNALS OF THE BITCOIN ETHEREUM AND BLOCKCHAINS. India: BPB Publications, 2018. ISBN 978-93-8655-130-6.

[15] *TP's Go Bitcoin Tests - Addresses* [online]. [cit. 2019-12-05]. Dostupné z: https://gobittest.appspotbitti druí.com/Address

[16] List of address prefixes. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-12-18]. Dostupné z: https://en.bitcoin.it/wiki/List_of_address_prefixes

[17] How do I check the checksum of a Bitcoin address? In: STACK EXCHANGE NETWORK [online]. [cit. 2019-12-05]. Dostupné z: https://bitcoin.stackexchange.com/questions/32353/how-do-i-check-the-checksum-of-a-bitcoin-address

[18] In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-12-21]. Dostupné z: https://en.bitcoin.it/wiki/Controlled_supply

[19] ANWAR, Hasib. *Consensus Algorithms: The Root Of The Blockchain Technology* [online]. 25.8.2018 [cit. 2019-12-21]. Dostupné z: https://101blockchains.com/consensus-algorithms-blockchain/#prettyPhoto/0/

[20] Marty. Princip Bitcoinové sítě (teorie I.). *Bitcoin v Čechách* [online]. 24.2.2017 [cit. 2019-12-05]. Dostupné z: https://bitcoincz.cz/index.php/2017/02/24/princip-bitcoinove-site-teorie-i/

[21] CoinMarketCap [online]. MarkMonitor, 2013 [cit. 2020-07-06]. Dostupné z: https://coinmarketcap.com/

[22] SHEVCHENKO, ANDREY. Binance Adds 15 Fiat Currency Options for Purchasing Crypto. Cointelegraph.com [online]. Creo Creatives Limited, 2013-2020, 14.2.2020 [cit. 2020-08-06]. Dostupné z: https://cointelegraph.com/news/binance-adds-15-fiat-currency-options-for-purchasing-crypto

[23] KALISKÝ, Boris. *Bitcoin a ti druzí: nepostradatelný průvodce světem kryptoměn*. [Praha]: IFP Publishing, 2018. ISBN 978-80-87383-71-1.

[24] SMITH, Bryan. What are the three generations of blockchain, and how are they similar to the web? Coininsider.com [online]. 12.7.2018 [cit. 2020-08-11]. Dostupné z: https://www.coininsider.com/three-generations-of-blockchain/

[25] SMITH, Bryan. What are the three generations of blockchain, and how are they similar to the web? Coininsider.com [online]. 12.7.2018 [cit. 2020-08-11]. Dostupné z: https://www.coininsider.com/three-generations-of-blockchain/

[26] KALISKÝ, Boris. *Bitcoin a ti druzí: nepostradatelný průvodce světem kryptoměn*. [Praha]: IFP Publishing, 2018. ISBN 978-80-87383-71-1.

[27] GALVÁNEK, Matěj. Cardano (VŠE, CO CHCETE VĚDĚT) – Kryptoměna s velkým potenciálem, ale i riziky. Alza.cz [online]. Alza.cz a.s, © 1994 - 2020, 3.3.2020 [cit. 2020-07-06]. Dostupné z: https://www.alza.cz/cardano-ada

[28] IOHK | Cardano whiteboard; overview with Charles Hoskinson. In: Youtube [online]. 26.10.2017 [vid. 2020-07-11]. Kanál uživatele IOKH. Dostupné z: https://www.youtube.com/watch?v=Ja9D0kpksxw&feature=emb_logo

[29] Why we are building Cardano. Cardano.org [online]. Panama: WhoisGuard, 2014 [cit. 2020-08-06]. Dostupné z: https://cardano.org/why/

[30] OLSZEWIC, Josh. Cardano Price Analysis - Centralized testnet slated for move to decentralized mainnet later this year. Bravenewcoin.com [online]. Auckland, NZ: Techemy, 2013 [cit. 2020-08-06]. Dostupné z: https://bravenewcoin.com/insights/cardano-price-analysis-centralized-testnet-slated-for-move-to-decentralized

[31] FiliM. Nejen Bitcoin fork: co je hard fork nebo soft fork. Cryptosvet.cz [online]. 2020, 21.8.2018 [cit. 2020-08-06]. Dostupné z: https://cryptosvet.cz/nejen-bitcoin-fork-o-co-jde-kdyz-se-rekne-hard-fork-nebo-soft-fork/

[32] FRANKENFIELD, Jake. Segwit (Segregated Witness). Investopedia.com [online]. MarkMonitor, 2020 [cit. 2020-08-06]. Dostupné z: https://www.investopedia.com/terms/s/segwit-segregated-witness.asp

[33] KYTKA. Rozdělení kryptoměn – čím se liší coin a token. Kryptomagazin.cz [online]. Praha, 2017, 19.1.2019 [cit. 2020-08-06]. Dostupné z: https://kryptomagazin.cz/rozdeleni-kryptomen-rozdil-mezi-coinem-a-tokenem/

[34] SEKERKOVÁ, Veronika. Tokeny. Zralocisobe.cz [online]. 2017, 2017, 11.11.2019 [cit. 2020-08-06]. Dostupné z: https://www.zralocisobe.cz/tokeny-222.html

[35] ANTONOPOULOS, Andreas M. a Gavin WOOD. Mastering Ethereum [online]. 1005 Gravenstein Highway North, Sebastopol: O'Reilly Media, 2018 [cit. 2020-08-06]. ISBN 978-1-491-97194-9. Dostupné z: https://github.com/ethereumbook/ethereumbook

[36] SAM. What are Stablecoins? USDT, TUSD, PAX, USDC, GUSD, EURS Explained. Saturn.network [online]. Prosinec [cit. 2020-08-11]. Dostupné z: https://www.saturn.network/blog/what-are-stablecoins-usdt-tusd-pax-usdc-gusd-eurs-explained/

[37] KOLÁŘ, Martin. Stablecoiny – ostrůvek stability v rozbouřeném moři kryptoměn. Btctip.cz [online]. 15.12.2019 [cit. 2020-08-11]. Dostupné z: https://btctip.cz/stablecoiny-ostruvek-stability-v-rozbourenem-mori-kryptomen/

[38] TĚTEK, Josef. Stable coins (VŠE, CO CHCETE VĚDĚT) – Svatý grál kryptoměn? Alza.cz [online]. 4.8.2018 [cit. 2020-08-11]. Dostupné z: https://www.alza.cz/stable-coins?layoutAutoChange=1

[39] HERTIG, Alyssa. Coindesk.com [online]. 30.3.2017 [cit. 2020-08-10]. Dostupné z: https://www.coindesk.com/learn/ethereum-101

[40] Ethereum (VŠE, CO CHCETE VĚDĚT). Alza.cz [online]. 21.5.2018 [cit. 2020-08-11]. Dostupné z: https://www.alza.cz/ethereum

[41] Gwei [online]. 2019 [cit. 2020-08-11]. Dostupné z: https://gwei.io/

[42] KUDLÁČEK, Patrik. Smart contracts (Chytré kontrakty) – Co jsou a jak fungují? Finex.cz [online]. 22.6.2019 [cit. 2020-08-11]. Dostupné z: https://finex.cz/chytre-kontrakty-smart-contracts-co-jsou-a-jak-funguji/

[43] HOODA, Parikshit. Blockchain | Smart Contracts. Geeksforgeeks.org [online]. 1.2.2020 [cit. 2020-08-11]. Dostupné z: https://www.geeksforgeeks.org/smart-contracts/

[44] FRANKENFIELD, Jake. Decentralized Applications - dApps. Investopedia.com [online]. 6.2.2018 [cit. 2020-08-10]. Dostupné z: https://www.investopedia.com/terms/d/decentralized-applications-dapps.asp

[45] MAKOVSKÝ, Jiří. Diit.cz [online]. 11. 6. 2020 [cit. 2020-08-11]. Dostupné z: https://diit.cz/clanek/co-jsou-decentralizovane-aplikace-cim-se-vyznacuji-k-cemu-slouzi

[46] GAS ISN'T A TOKEN BUT UNDERSTANDING IT CAN SAVE YOU MONEY AND FRUSTRATION. Ethgas.io [online]. Radar Relay Inc. 2019, 2019 [cit. 2020-08-06]. Dostupné z: https://ethgas.io/

[47] RIETH, Yulia. What Are Gas, Gas Limit, and Gas Price in the Ethereum Network? Decenter.org [online]. Domain Protection Services, 2017-2020, 19.3.2018 [cit. 2020-08-06]. Dostupné z: https://decenter.org/en/what-are-gas-gas-limit-and-gas-price-in-the-ethereum-network

[48] Dumbcoin - An educational python implementation of a bitcoin-like blockchain. Github [online]. San Francisco, 2008, [cit. 2021-5-20]. Dostupné z: https://github.com/julienr/ipynb_playground/blob/master/bitcoin/dumbcoin/dumbcoin.ipynb?fbclid=IwAR24TmSacJM9aA4cTTxcoT6ci3DwYzvDJms_pcNfrZjkC0TwxTxpVAtAGRE

[49] TIWARI, Abhishek. Create simple Blockchain using Python. Geeksforgeeks.org [online]. 4.8.2020 [cit. 2021-5-20]. Dostupné z: https://www.geeksforgeeks.org/create-simple-blockchain-using-python/?fbclid=IwAR0XFAlRYndbC5w-spVn7-lkcaYzKeB0WN2q8YBr0Y-HtkUYW4o0nBbWfSw

[50] SBLENDORIO, Dante. How To Build A Blockchain In Python (Get Pre-Built Runtime). Activestate.com [online]. 6.10.2020 [cit. 2021-5-26]. Dostupné z: https://www.activestate.com/blog/how-to-build-a-blockchain-in-python/?fbclid=IwAR3DMcMsX5Iks7zYk3N9Yn1HJHflN_H3YbvismGikQnb7cbqJM8IHJNR45M

[51] MOUJAHID, Adil. A Practical Introduction to Blockchain with Python. Adilmoujahid.com [online]. 14.3.2018 [cit. 2021-5-26]. Dostupné z: http://adilmoujahid.com/posts/2018/03/intro-blockchain-bitcoin-python/?fbclid=IwAR3oflSNm6AAdWpgxP8uqcBSKdiHadjaIla00pfJdGLxANazMSmRpa-wWDY