Jak funguje Bitcoin

Jan Oupický

1. Úvod

1.1. Počátek Bitcoinu

Za počátek Bitcoinu se většinou považuje publikování dokumentu popisující, jak by mohla fungovat kryptoměna s názvem Bitcoin (slovem bitcoin s malým b budeme označovat samotnou jednotku měny). Tento dokument byl publikován v pdf prostřednictvím mailing listu[1] určeným pro lidi, zabývající se kryptografií, dne 31. 10. 2008[2].

Dokument je pojmenován „*Bitcoin: A Peer-to-Peer Electronic Cash System*“ a jeho autorem je Satoshi Nakamoto. Překvapivě má pouze 9 stran na to, jaký je jeho dopad na moderní svět. Tato „práce“ popisuje abstraktně, jak by kryptoměna s názvem Bitcoin mohla fungovat. Není to žádná odborná specifikace Bitcoinu. Je to spíše takový hrubý popis algoritmu.

Satoshi Nakamoto je pseudonym. Kolem tohoto jména se na internetu nachází mnoho teorií, o tom, kdo to ve skutečnosti je. Předpokládá se, že alespoň chvíli žil v Japosku a měl zájem kryptografii a programování. Není znám přesný důvod, proč Satoshi zatím (pravděpodobně nikdy) neukázal svou pravou tvář. Existuje mnoho rozumných vysvětlení, proč to neudělal. Nejlogičtější důvod, na který jsem narazil je ten, že hlavní idea Bitcoinu je založena na anonymitě a spolupráci mnoha lidí. Pokud by v tomto „ekosystému“ existoval jeden člověk, který by byl důležitější než všichni ostatní, bylo by to v rozporu s filosofií, se kterou byl Bitcoin vytvořen. Další možné vysvětlení je osobní bezpečnost Satoshiho. Předpokládá se, že Satoshi vlastní kolem 1 milionu bitcoinů[3]. Pokud použijeme dnešní kurz 1 BTC ~ 4316 $ (BTC je zkratka pro bitcoin), tak Satoshi má zhruba ve svém vlastnictví bitcoiny v hodnotě 4 316 000 000 $. Tedy přes 4 miliardy amerických dolarů.

Opravdový počátek Bitcoinu nastal 9. 1. 2009, když Satoshi zveřejnil první verzi softwaru Bitcoin. O vydání programu Satoshi opět informoval prostřednictvím emailu. Z počátku byl Bitcoin používán pouze nadšenci a neměl v podstatě žádnou reálnou hodnotu, jelikož za něj nebylo nic koupeno. První zaznamenaný „reálný“ obchod s bitcoiny byl proveden 22. 5. 2010 uživatelem „*laszlo“*, jenž

[1] Mailing list obvykle funguje tak, že uživatel se přihlásí k jeho odběru a poté dostává emaily od ostatních uživatelů na této „síti“.

[2] Některé zdroje uvádějí 1. 11. 2008

[3] Jelikož jsou všechny transakce veřejné, lze dohledat i úplně první transakce, ve kterých pravděpodobně hlavně figuruje Satoshi. Zdroj [18]

koupil za 10 000 BTC 2 pizzy s odhadovanou cenou 25 $. Tato transakce dala tedy 1 BTC hodnotu kolem 0,0025 $.

1.2. Co je vlastně bitcoin?

Důležitá otázka ohledně Bitcoinu je „Co to vlastně znamená reálně vlastnit např. 1 bitcoin?“. Odpověď na tuto otázku je poněkud neintuitivní, ale po bližším zamyšlení se to moc neliší od „reálných peněz“.

Zásadním konceptem pro pochopení je fakt, že neexistuje určitá fyzická (kus papíru, kterému říkáme bankovky) ani digitální věc (soubor bitů), kterou lze nazvat 1 bitcoin, 10 bitcoinů, 0.1 bitconu apod.

Vlastnit 1 BTC znamená mít „důkaz“[4], že mi někdo poslal[5] 1 BTC v minulosti. Pokud se zamyslíte, tak v „obyčejné“ měně tyto důkazy také máme a věříme jim. Mám-li ve svém vlastnictví bankovku, na které je napsáno 1000 Kč, mám vlastně důkaz, že jsem k ní nějak přišel. Vlastnictví dané bankovky je ten důkaz.

Ještě bližší analogie je s penězi na účtu v bance. Pokud mám na svém účtu 10 000 Kč, potom důkazem je v podstatě slovo banky. Jestliže chci zaplatit debetní kartou nákup v obchodě, tak obchod automaticky věří bance, že mám danou částku ve svém vlastnictví a mohu za ní směnit nákup, aniž bych fyzicky vlastnil něco jako peníze.

Vlastnit bitcoin je ideově to samé jako mít nějaké peníze v bance. Nepatrný rozdíl je, že banka ukládá aktuální bilanci. Banka má pravděpodobně v databázi uložený záznam ve smyslu „na účtu číslo 254874844 je aktuálně 15478,54 Kč“. Bitcoin toto nedělá, nikde není žádná hlavní databáze[6], ve které je uloženo, kolik má daný účet (adresa) bitcoinů.

2. Matematika v Bitcoinu

2.1. Asymetrická kryptografie

Nejhlavnějším důvodem, proč Bitcoin může správně fungovat je kryptografie, specifičtěji technologie digitálních podpisů. Digitální podpisy fungují díky asymetrické kryptografii.

Existují 2 hlavní typy šifrování

- symetrické
- asymetrické

Symetrické šifrování je to, co pravděpodobně napadne každého, když se řekne „šifrování“. Představme si nejjednodušší situaci. Řekněme že, Bob chce poslat Marii tajnou zprávu. Bob tuto zprávu zašifruje pomocí tajného klíče, který zná pouze on a Marie. Marie poté, co dostane zprávu od Boba dokáže pomocí

[4] Co to je za důkaz vysvětleno blíže v kapitole 4

[5] bitcoiny lze nejen získat od někoho, viz kapitola 5

[6] Tyto databáze existují, ale nejsou vůbec nutné k fungování Bitcoinu. Více v kapitole 7

stejného klíče zprávu dešifrovat. Dále Marie může zase Bobovi poslat zašifrovanou zprávu pomocí stejného klíče zpět a Bob jí zase může dešifrovat.

Způsob šifrování a co to je vlastně ten „klíč“ je pro tento případ nepodstatné. Hlavní je, že existuje jeden klíč a je znám pouze lidem, kteří spolu chtějí komunikovat. Tomuto se říká symetrická kryptografie, jelikož je stejný klíč používán pro šifrování i dešifrování.

Naopak u asymetrické kryptografie je klíčů více (v jednom směru obvykle 2). Jeden je tzv. privátní (soukromý) a druhý veřejný. Oproti případu se symetrickou kryptografií, nyní existují celkem 4 klíče (2 tajné) místo 1 tajného.

Bob i Marie mají vlastní rozdílný pár klíčů (1 veřejný, 1 privátní). Jestliže Bob chce Marii poslat tajnou zprávu, potřebuje k tomu znát pouze veřejný klíč Marie. Bob zašifruje zprávu pomocí veřejného klíče Marie a pošle jí zašifrovanou zprávu. Marie nyní dokáže dešifrovat tuto zprávu pouze svým privátním klíčem. Pokud Marie bude chtít odpovědět, tak také použije Bobův veřejný klíč k zašifrování své zprávy a Bob jí dešifruje svým privátní klíčem.

Znalost veřejného klíče Marie ani Boba nikomu nepomůže k rozluštění zašifrované zprávy tímto klíčem. To je hlavní výhoda asymetrické kryptografie oproti symetrické. V symetrické kryptografii bývá problém bezpečně předat soukromý klíč, který Bob i Marie spolu sdílí. V asymetrické kryptografii toto není vůbec potřeba.

Nejznámějším a nejpoužívanějším algoritmem, který funguje na tomto principu, je RSA. Bitcoin ale používá poněkud modernější algoritmus využívající tzv. eliptické křivky. Tyto dva algoritmy jsou založené na stejném principu. Využívají matematického problému, u kterého lze jednoduše zkontrolovat výsledek, zda je správný, ale je nesmírně těžké nalézt správný výsledek.

V RSA je tento problém tzv. faktorizace. Zvolme si například 2 prvočísla o délce 300 cifer v desítkové soustavě. Tato čísla vynásobíme a získáme produkt o cca 600 cifrách. Tato operace lze provést jednoduše a rychle. Pokud vám ale někdo dá pouze tento produkt a zeptá se vás na ta dvě původní prvočísla, neměli byste být schopni tento problém vyřešit v reálném čase pomocí dosud známých metod. Známe-li ale tato 2 prvočísla, je velice snadné ověřit, že jsou to řešení. Vynásobení a porovnání 2 velkých prvočísel není žádný problém. Toto je stručně problém faktorizace, na který spoléhá RSA. Zatím neexistuje důkaz, že by RSA šlo prolomit jinak, než právě výpočtem faktorů velkého čísla v „rozumném čase“.

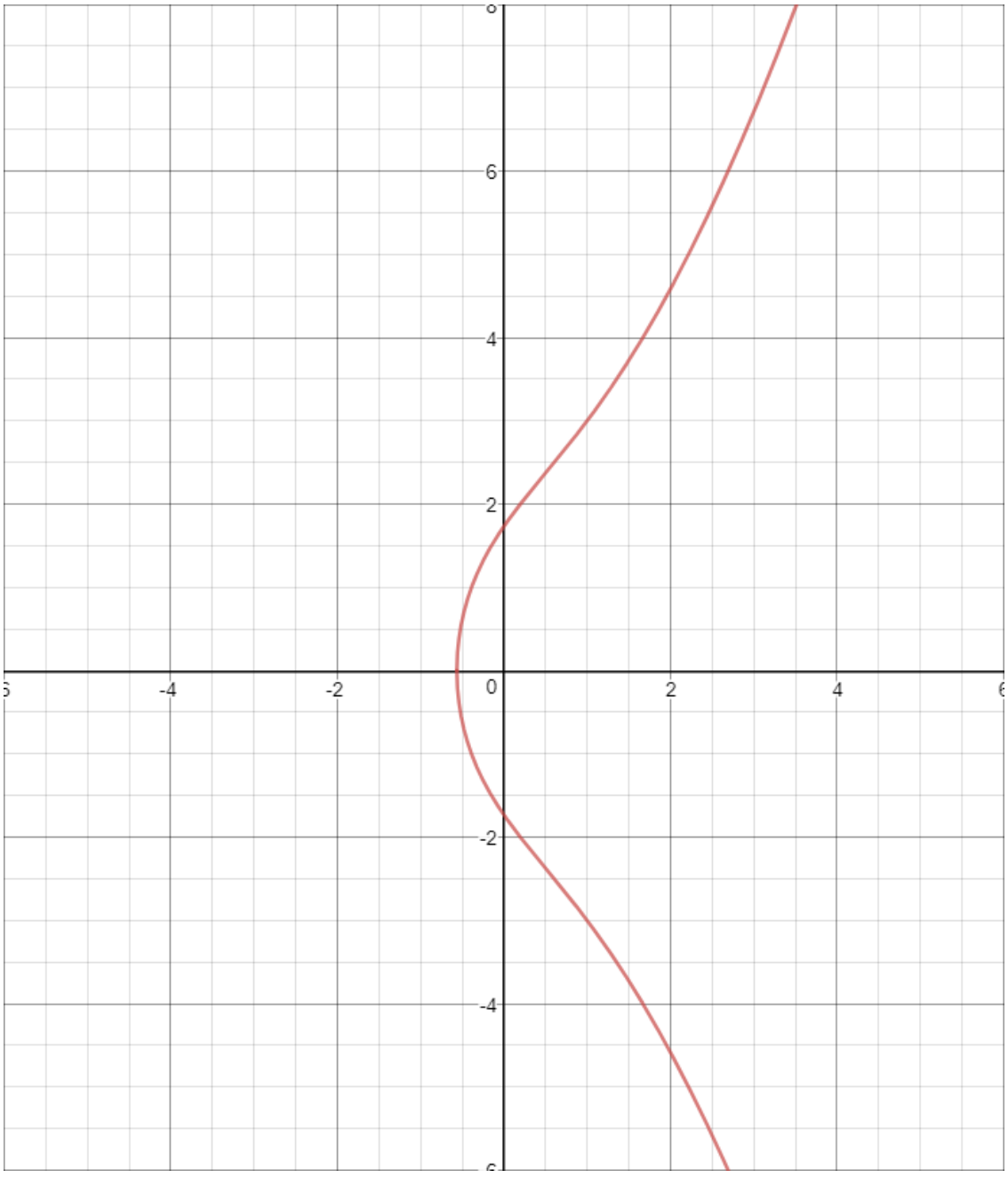
Eliptické křivky přinášejí podobný problém, jemuž se říká „nalezení diskrétního logaritmu na eliptických křivkách“[7].

2.2. Eliptické křivky

Eliptická křivka má poněkud jednoduchou definici.

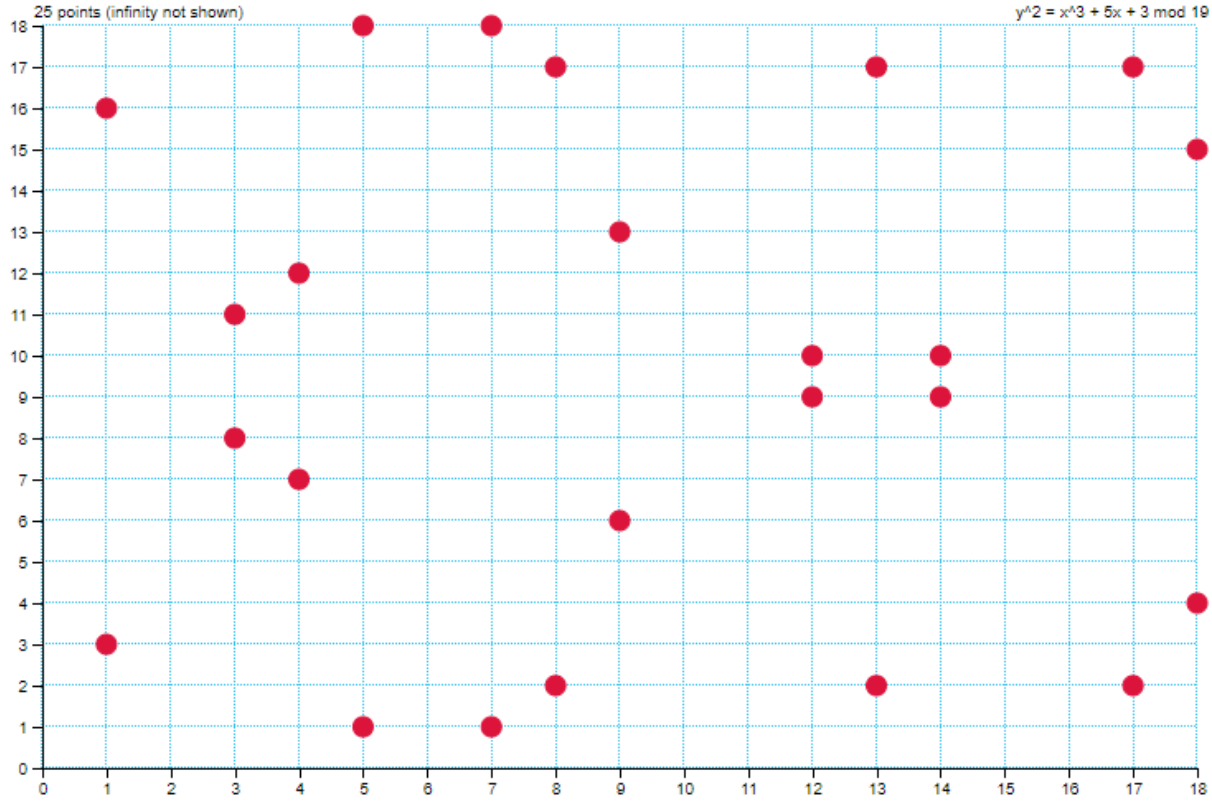
[7] Problém je blíže popsán v kapitole 2.3

Eliptická křivka nad tělesem X je množina dvojic bodů $[x, y] \in X \times X$ splňující rovnici tvaru $y^2 = x^3 + ax + b; kde\ x, y, a, b\ \in\ X$. Například křivka $y^2 = x^3 + 5x + 3$ nad tělesem reálných čísel vypadá takto



Eliptická křivka nad tělesem reálných čísel

V kryptografii se ale nepoužívá těleso reálných čísel ale konečná tělesa Z_p, kde p je prvočíslo. Stejná křivka akorát nad tělesem Z_{19} vypadá takto



Eliptická křivka nad konečným tělesem Z_{19}
Zdroj: http://graui.de/code/elliptic2/

Pro lepší pochopení dalších pojmů budeme ale uvažovat křivky nad reálnými čísly.

Eliptické křivky mají zajímavé vlastnosti, díky kterým je lze použít v kryptografii. Tyto vlastnosti budou popsány stručně a intuitivně.

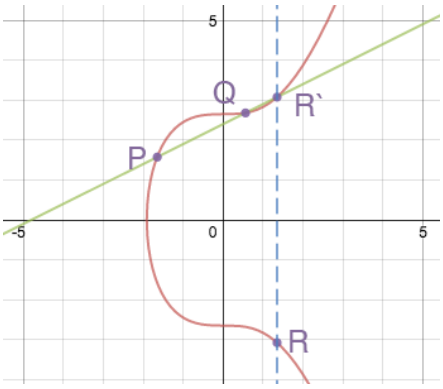
Eliptická křivka je osově souměrná podle osy x.

Jakákoliv přímka, která není rovnoběžná s osou y, protínající eliptickou křivku ve dvou bodech, které nejsou tečné body, vždy protne křivku ještě v jednom bodě.

Zvolíme-li tečnu křivky, která není rovnoběžná s osou y, vždy protne křivku ještě v jednom bodě.

Díky těmto vlastnostem, můžeme definovat operace jako sčítání 2 různých bodů na křivce a sčítání 2 identických bodů.

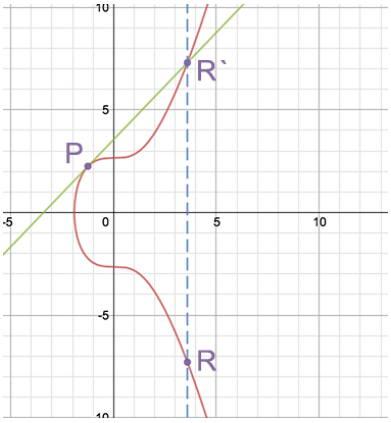
Mějme 2 různé body na křivce P, Q. Součtem P+Q rozumíme bod R, který je zrcadlením, podle osy x, bodu R', který vznikl jako průnik křivky a přímky spojující body P a Q. Z obrázku je to pochopitelnější.



Součet 2 různých bodů na eliptické křivce
Zdroj: https://media.coindesk.com/uploads/2014/10/point-addition.png

Součet 2 identických bodů P na křivce je také definován a je velice podobný. Součtem P+P rozumíme také bod R, který je zrcadlením, podle osy x (dále jen zrcadlení), bodu R', který vznikne ale jako průnik křivky s tečnou procházejícím bodem P. Viz další obrázek.

Množina bodů (označíme písmenem *X*) na eliptické křivce spolu s pomyslným bodem na křivce v nekonečnu (označíme 0) a operace sčítání (označíme „+“) na této množině tvoří tzv. grupu. Bod v nekonečnu, je definovaný tak, že ním prochází každá přímka rovnoběžná s osou Y. Každá grupa musí splňovat následující axiomy:



Součet 2 identických bodů na eliptické křivce
Zdroj: https://media.coindesk.com/uploads/2014/10/point-doubling.png

- uzavřenost: pro libovolné dva prvky z množiny *X* platí, že jejich součet je také prvkem množiny *X*
- existence neutrálního prvku: existuje prvek 0 z množiny *X*, pro který platí, že vezmeme-li libovolný prvek a z množiny *X*, tak $a + 0 = 0 + a = a$
- existence inverzního prvku: pro každý prvek a z množiny *X* existuje právě jeden inverzní prvek a^{-1} z množiny X a platí $a + a^{-1} = 0$
- asociativita: pro libovolné tři prvky a, b, c z množiny *X* platí $(a + b) + c = a + (b + c)$

Pokusíme se intuitivně vysvětlit, proč tyto vlastnosti platí.

- Uzavřenost plyne z definice sčítání, každý výsledný bod, je definován jako průnik křivky s nějakou přímkou.
- Neutrálním prvkem, je právě ten bod v nekonečnu. Přímka spojující libovolný bod b a bod v nekonečnu, je z definice přímka rovnoběžná s osou Y procházející bodem b. Jejím třetím průnikem s křivkou je právě zrcadlení bodu b, tedy výsledný součet $b + 0 = b$
- Inverzní prvek k libovolnému bodu b je právě ten bod, který je zrcadlením bodu b. Přímka, jenž spojuje tyto dva body, je rovnoběžná s osou Y, tedy třetí průnik s křivkou je bod v nekonečnu (zrcadlení v nekonečnu se neuvažuje).
- Asociativita se vysvětluje poněkud těžce, proto jí radši vynecháme.

Dále je dobré zmínit, že operace sčítání je také komutativní neboli pro libovolné prvky a, b platí $a + b = b + a$. To je jednoduché na představu, jelikož přímka z bodu P do bodu Q je ta samá, jako přímka z bodu Q do bodu P. Grupa, která má tuto vlastnost se nazývá Abelova grupa.

Definice výše je dobrá k představení o čem se vlastně bavíme, ale v praxi nelze použít (počítače asi těžko budou zbytečně rýsovat tečnu mezi dvěma body). Proto si ukážeme také algebraické definice (zdroj [17]) sčítání dvou bodů na eliptické křivce.

Nechť máme body P = (x_P, y_P), Q = (x_Q, y_Q), kde $P \neq Q$, a chceme jejich součet R. Spočítáme si tedy nejdříve směrnici s přímky, která prochází oběma body.

$$s = (y_P - y_Q) \cdot (x_P - x_Q)^{-1}$$

Souřadnice bodu R = (x_R, y_R) spočítáme

$$x_R = s^2 - x_P - x_Q$$

$$y_R = s \cdot (x_P - x_R) - y_P.$$

Tyto operace jsou obyčejné sčítání (odečítání) a násobení (inverzním prvkem) v tělese.

Jestliže $P = Q$, tak je postup podobný.

$$s = (3 \cdot {x_P}^2 + a) \cdot (2 \cdot y_P)^{-1}$$

kde a je parametr dané eliptické křivky. Souřadnice bodu R jsou

$$x_R = s^2 - 2 \cdot x_P$$

$$y_R = s \cdot (x_P - x_R) - y_P.$$

Jak tedy můžeme těchto vlastností využít v kryptografii?

2.3. ECDSA

ECDSA je zkratka pro „*Elliptic Curve Digital Signature Algorithm*“. Toto je algoritmus pro digitální podpis dat, využívající vlastnosti eliptických křivkách. Algoritmy pro digitální podpis jsou úzce spojeny s algoritmy pro šifrování pomocí asymetrické kryptografie. Využívají stejné problémy (faktorizace, diskrétní logaritmus na eliptických křivkách), akorát jejich „cílem“ není zašifrovat nějaká data, aby je někdo nemohl dešifrovat, ale zaručit původ dat. ECDSA zaručuje původ i integritu dat.

Jestliže Bob pošle Marii zprávu (klidně zašifrovanou), Marie by si chtěla ověřit, zda to opravdu je ta stejná zpráva, kterou Bob odeslal. Se zprávou se během cesty k Marii mohlo stát cokoliv. Bob proto pošle, spolu se zprávou, další „zprávu“ (podpis), který zaručí původ (zpráva opravdu pochází od Boba) i integritu (obsah zprávy se během cesty k Marii nezměnil) dat (zprávy). Podpis byl vytvořen pomocí privátního klíče Boba, ke kterému má pouze přistup on, jako v podobném případě se šifrování. Marie si pravost podpisu[8] může ověřit veřejným klíčem Boba.

[8] Jednak, že podpis je opravdu od Boba a také, že ta data jsou ta samá, co Bob odeslal.

Oba případy (šifrování a podpis) jsou velice podobné. Rozdíl je v tom, že Marie používá Bobův veřejný klíč pro kontrolu zprávy od Boba, nikoliv aby mu nějakou zprávu poslala.

Nyní se vraťme k tomu, jak to tedy může fungovat za pomoci eliptických křivek. Použijeme zjednodušenou verzi algoritmu ECDSA. Příklad je převzatý ze zdroje [11], liší se akorát proměnná data z důvodů náročnosti výpočtů.

Zvolme si tedy

- eliptickou křivku $y^2 = x^3 + 7$
- těleso = Z_{67} (67 je prvočíslo)
- bod *G* = (2,22) na eliptické křivce
- číslo *n* =79, pro které platí $n \cdot G = (0,0)$, v praxi se nejdříve volí n jako velké prvočíslo a k němu se volí bod
- privátní klíč *p* = 2, musí být z intervalu [1, n-1]

Veřejný klíč *V* se vypočítá jednoduše vzorcem $V = p \times G$. Použijeme-li tedy znalosti z předchozí kapitoly

$$V = G + G$$

$$s = (3 \cdot 2^2 + 0) \cdot (2 \cdot 22)^{-1}\ mod\ 67 = 12 \cdot 44^{-1}\ mod\ 67 = 12 \cdot 32\ mod\ 67 = 49$$

$$x_v = 49^2 - 2 \cdot 2\ mod\ 67 = \ 2401 - 4\ mod\ 67 = 52$$

$$y_v = 49 \cdot (2 - 52) - 22\ mod\ 67 = 49 \cdot 17 - 22\ mod\ 67 = 7.$$

Veřejným klíčem *V* k privátnímu klíči *p* je tedy bod (52, 7). V tomto je hlavní „problém“ kryptografie eliptických křivek. Je jednoduché sečíst bod několikrát se sebou (skalárně vynásobit). Ale zjistit kolikrát (*p*) byl bod *G* skalárně vynásoben, abychom získali bod *V* je nesmírně složité, zvlášť s velkými čísly. Problém hledání daného p se právě říká „hledání diskrétního logaritmu na eliptických křivkách“.

Všechno kromě čísla *p* je veřejné a každý tyto údaje může získat. Parametry eliptické křivky nejsou pro každý podpis náhodně zvolené, jak to bývá u RSA, kde se pokaždé se generují náhodná prvočísla. V praxi se používá jen pár eliptických křivek, které byly prokázány za bezpečné. Například Bitcoin používá křivku s označením Secp256k1. Jejími parametry jsou

- $y^2 = x^3 + 7$
- charakteristika tělesa $= 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$ v hexadecimální soustavě = $FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFC2F$
- bod *G*
  - $x = 0479BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81798$
  - $y = 483ADA7726A3C4655DA4FBFC0E1108A8FD17B448A68554199C47D08FFB10D4B8$
- číslo n = $FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEBAAEDCE6AF48A03BBFD25E8CD0364141$

Nyní použijeme naše předchozí parametry k podpisu nějakých dat a ukážeme, jak to vlastně funguje. Jako data si můžeme zvolit cokoliv. Například číslo 15. Nechť tedy *d* (data) = 15. Algoritmus podpisu dat funguje následovně.

1. Zvolíme náhodně číslo *k* z intervalu [1, n-1]
2. Spočítáme bod $W = k \times G$
3. Spočítáme $r = x_w \; mod \; n$
4. Jestliže $r = 0$ začneme znovu od kroku 1
5. Spočítáme $s = (d + r \cdot p) \cdot k^{-1} \; mod \; n$
6. Jestliže $s = 0$ začneme znovu od kroku 1
7. Dvojice (r,s) je náš podpis dat *d*.

Spočítejme podpis s našimi čísly.

1. k = 60
2. $W = (38{,}41)$
3. $r = 38 \; mod \; 67 = 38$
4. $r \neq 0$
5. $s = (15 + 38 \cdot 2) \cdot 60^{-1} \; mod \; 67 = (15 + 76) \cdot 19 \; mod \; 67 = 24 \cdot 19 \; mod \; 67 = 54$
6. $s \neq 0$
7. Náš podpis dat „15“ je tedy $(38{,}54)$

Tento podpis (38,54) můžeme nyní poslat společně s daty. Příjemce bude mít možnost zkontrolovat si, že ta data opravdu pochází od nás. Algoritmus pro kontrolu funguje následovně.

1. Zkontrolujeme že $r, s \in [1, n-1]$
2. Spočítáme $w = s^{-1} \; mod \; n$
3. Spočítáme $u = d \cdot w \; mod \; n$
4. Spočítáme $o = r \cdot w \; mod \; n$
5. Spočítáme bod $C = (x, y) = u \times G + o \times V$
6. Pokud platí $r = x \; mod \; n$ podpis je platný.

$n \times P$ je pouze zkratka pro $\underbrace{P + P + \cdots + P + P}_{n}$

Důkaz správnosti tohoto algoritmu není ani příliš komplikovaný.

Vezměme bod C z kroku 5.

Dle definice $C = u \times G + o \times V$. *V* je veřejný klíč, který jsme získali dle definice takto $V = p \times G$, kde *p* je náš privátní klíč. Po dosazení dostaneme

$$C = u \times G + o \cdot p \times G$$

$$C = (u + o \cdot p) \times G$$

Dále použijeme definice *u*, *o* a *w*.

$$C = (d \cdot w + r \cdot w \cdot p) \times G$$

$$C = (d \cdot s^{-1} + r \cdot s^{-1} \cdot p) \times G$$

$$C = ((d + r \cdot p) \cdot s^{-1}) \times G$$

Dle definice s z 5. kroku algoritmu pro vytváření podpisu platí $s = (d + r \cdot p) \cdot k^{-1} \; mod \; n$

$$C = ((d + r \cdot p) \cdot \left((d + r \cdot p) \cdot k^{-1}\right)^{-1}) \times G$$

$$C = ((d + r \cdot p) \cdot (d + r \cdot p)^{-1} \cdot k) \times G$$

$$C = (1 \cdot k) \times G = k \times G$$

Bod C je tedy roven bodu *W*, který jsme spočítali při vytváření podpisu v kroku 2.

Otestujeme tedy na příkladu tuto teorii. Máme tedy již spočítaný podpis pro data $d = 15$, který je $(r, s) = (38{,}54)$. Provedeme tedy kontrolu.

1. Platí $38 < 67$ a zároveň $54 < 67$
2. $w = 54^{-1} \; mod \; 67 = 36$
3. $u = (15 \cdot 36) \; mod \; 67 = 4$
4. $o = (38 \cdot 36) \; mod \; 67 = 28$
5. $C = 4 \times (2{,}22) + 28 \times (52{,}7) = (25{,}17) + (58{,}45) = (38{,}41)$
6. Platí $38 = 38 \; mod \; 67$. Podpis je tedy platný.

Takto zhruba funguje algoritmus ECDSA, který Bitcoin používá. V praxi se nepodepisují přímo data (např. velký soubor o několika MB) ale podepisuje se jejich tzv. hash, který v podstatě zaručuje jejich identitu.

2.4. Hašovací funkce

Hašovací funkce jsou pro Bitcoin stejně zásadní jako ECDSA. Stručně řečeno hašovací funkce je funkce, která přiřadí posloupnosti bitů jinou posloupnost bitů pevné délky. Hlavní vlastnosti hašovací funkce jsou

- výstup je pro jakákoliv vstupní data stejné délky
- nastání kolize (to, že pro 2 různé vstupy bude stejný výstup) je vysoce nepravděpodobné

Kryptografické hašovací funkce musí ještě splňovat tyto vlastnosti

- při nepatrné změně vstupu se výstup „zásadně liší“
  - například pokud použijeme hašovací funkci SHA-256 na řetězce „Bitcoin“ a „bitcoin“
  - SHA-256(„Bitcoin“) = b4056df6691f8dc72e56302ddad345d65fead3ead9299609a826e2344eb63aa4
  - SHA-256(„bitcoin“) = 6b88c087247aa2f07ee1c5956b8e1a9f4c7f892a70e324f1bb3d161e05ca107b
  - z výstupu tedy není možné poznat, že vstup byl skoro identický
- z výstupu nelze zrekonstruovat vstup

Příkladem kryptografické hašovací funkce SHA-256 (někdy nazývána SHA-2). Rozdíl mezi obyčejnou hašovací funkcí a kryptografickou hašovací funkcí bývá obvykle také rychlost. Kryptografické hašovací funkce musí být v praxi pomalejší, aby byl tzv. bruteforce (zkoušení všech možností) útok hůře proveditelný.

Hašovací funkce pracují s posloupnostmi bitů, tyto bity bývají obvykle převedeny do tzv. hexadecimální formy viz výstup funkce SHA-256 v příkladu výše. SHA-256 má toto označení, protože jejím výstupem je hash o délce 256 bitů. Každé 4 bitové číslo lze vyjádřit jedním hexadecimálním číslem, proto je výstup v příkladu délky 64.

Od hašovací funkce chceme, aby nenastala kolize. Teoreticky kolize musí někdy nastat, jelikož vstup je neomezený a výstup je omezený. Možných unikátní hashů, které tato funkce může vyprodukovat je $2^{256} \approx 11{,}5 \cdot 10^{76}$. Toto je nesmírně velké číslo. Pravděpodobnost, že při hašování 10^9 různých vstupů nastane kolize je zhruba 10^{-60}. Toto plyne z tzv. Narozeninového paradoxu, který říká následující. Vybereme-li ze sady $H(= 2^{256})$ $n = (10^9)$ hodnot, pravděpodobnost, že jsme vybrali alespoň jednu hodnotu více než jednou, je zhruba

$$1 - e^{\frac{-n^2}{2 \cdot H}} = 1 - e^{\frac{-10^{18}}{2 \cdot 2^{256}}} \approx 4{,}3 \cdot 10^{-60}.$$

Pro získání představy o tom, jak je to malá pravděpodobnost, pravděpodobnost toho, že do Země narazí právě v tuto sekundu asteroid, co dokáže vyhubit život na Zemi, je 10^{-15}. Tato pravděpodobnost je o 45 řádů větší. V praxi kolize v podstatě nenastane náhodným zkoušením vstupů.

Bitcoin používá funkce SHA-256 a RIPEMD-160. Funkce RIPEMD-160 se používá pouze pro vytváření adres, naopak SHA-256 je „srdce“ Bitcoinu.

3. Bitcoin adresy

Vytvoření Bitcoinové adresy je velice jednoduchý proces. Stačí vygenerovat privátní a veřejný klíč dle ECDSA. Bitcoinová adresa se poté získá jednoduchým zahašováním veřejného klíče.

Veřejný klíč v ECDSA je bod, tedy dvojice čísel. Aby z něho vzniklo jedno číslo, které můžeme zahašovat, jednoduše se za x zapojí y. Před toto spojené číslo se také dává prefix, který označuje, zda je toto číslo komprimované.

Pro určení bodu na eliptické křivce totiž nepotřebujeme znát obě souřadnice, ale stačí nám pouze souřadnice x a informace, zda y je kladné nebo záporné (v případě reálných čísel). Toto jednoduše plyne z toho, že pokud známe x, z rovnice eliptické křivky se stane obyčejná kvadratická rovnice. Komprese dat je v podstatě 50 %. Samozřejmě komprese dat je na úkor nutnosti spočítat danou kvadratickou rovnici. V Bitcoinu se používá komprimovaná verze veřejného klíče.

Pokud tedy máme veřejný klíč vyjádřený jako jedno číslo, adresu dostaneme jednoduchým aplikováním 2 hašovacích funkcí a jedné převodní funkce, která převádí bity na čitelný text.

$$adresa = base58\left(RIPEMD160\left(SHA256(veřejný\ klíč)\right)\right)$$

Druhá hašovací funkce se používá z důvodů zmenšení velikosti adresy z 256 bitů na 160 bitů. Tato funkce je méně bezpečná než SHA256, ale je prakticky nemožné získat původní SHA256 hash z RIPEMD160, natož veřejný klíč ze SHA256 hashe. Base58 je převodní funkce, která převádí 160 bitů na 58 charakterů. Zaručuje jednoduchou čitelnost adresy a eliminuje například záměnu charakteru „0" s „O". Tato funkce není hašovací, používá se pouze, aby člověk mohl adresy jednoduše číst.

Bitcoinová adresa vypadá například takto `3Ho27oYDZrdQz7zYjpCN74wduFZ1emiqhV`. Upřesnění odpovědi na otázku „Co vlastně člověk vlastní, pokud má nějaké bitcoiny?" je tedy: Člověk vlastní privátní klíče k dané adrese a existuje pravdivý veřejný záznam na tzv. blockchainu, že na této adrese jsou nějaké bitcoiny (někdo je na tuto adresu poslal).

4. Jak probíhá transakce v Bitcoinu

Jak to tedy funguje, pokud chce Bob poslat Alici např. 1,399 BTC[9]? Předpokládejme, že Bob již vlastní 1,399 BTC z předcházející transakce od někoho jiného. Bob potřebuje sestavit tzv. transakci což je v podstatě oznámení o změně vlastnictví. Obsahem transakce bude zpráva „Bob posílá ze své adresy `3NYdfynHuq4fj1Xn5Nr3tKYkNeUL1BDJZd` Alici na adresu `37mBxumFxt1TyjKRhvSTkfwVJdyK365X1o` 1,399 BTC. Bob poté tuto zprávu zahašuje funkcí SHA-256 a tento hash podepíše pomocí ECDSA. Dále stačí, aby Bob tuto zprávu „poslal do světa".

„Svět" pro Bitcoin znamená bitcoinová tzv. peer-peer (klient-klient) síť. V této síti každý klient šíří s ostatními informace. V našem případě Bob nemusí poslat přímo všem na bitcoinové síti informaci o této transakci, ale může jí poslat pouze zlomku, a ten to rozšíří mezi ostatní. Je to taková řetězová reakce.

Samozřejmě Alice nemůže Bobovi věřit jen proto, že takovou transakci vytvořil. Kde máme tu jistotu, že Bob opravdu vlastní 1,399 BTC? Na to existuje již zmíněný systém tzv. blockchain.

Stručně řečeno, blockchain je veřejná decentralizovaná distribuovaná databáze. „Decentralizovaná" a „distribuovaná" znamená, že neexistuje jeden hlavní server, na kterém je databáze uložena, ale každý má vlastní kopii této databáze a může jí šířit s ostatními za určitých podmínek (o těch později). Bitcoin blockchain[10] je tedy databáze všech transakcí, které kdy byly provedeny v bitcoinové síti.

Předpokládejme, že Alice má možnost projít předešlé transakce a zkontrolovat, že na adrese `3NYdfynHuq4fj1Xn5Nr3tKYkNeUL1BDJZd` stále je 1,399 BTC, a že tato adresa opravdu patří Bobovi, pomocí jeho podpisu. Jestliže Alice nenarazila

[9] Tato transakce s přibližně stejným počtem bitcoinů opravdu proběhla a detaily jsou k nahlédnutí zde https://blockchain.info/tx/4a4c25f1ed4202fc2ef75ce3e4202a5ffb44b60557b435aba8d8b2b8f65fdaa5

[10] Na prohlížení Bitcoin blockchainu lze použít například stránka https://blockchain.info/

na chybu, pravděpodobnost nepravosti transakce se zvýšila, ale stále si nemůže být jistá.

Bob mohl ve stejnou chvíli poslat do bitcoinové sítě druhou transakci, ve které posílal ty samé bitcoiny jiné osobě než Alici. Vzhledem k tomu, že blockchain je decentralizovaný a distribuovaný, se může jednoduše stát, že k Alici se tato zpráva o druhé transakci dostane později, než provede vlastní kontrolu se „zastaralou" databází. Aby takový systém mohl fungovat, musí se zavést protokol, podle kterého se každý bude řídit. Obvykle také ani Alice nemá prostředky danou transakci zkontrolovat, jelikož například velikost celé databáze (blockchainu) je, v době psaní tohoto dokumentu, přes 132 GB[11].

Hlavní součástí protokolu je to, že Alice i Bob se spoléhají na ostatní, že jejich transakci zkontrolují. Ten, kdo danou transakci zkontroluje, ale musí naopak něco dostat, aby měl nějakou motivaci tuto činnost vůbec provádět.

5. Mining (těžení) bitcoinů

Každý, kdo kdy něco slyšel o Bitcoinu jistě zaznamenal termín „mining" neboli „těžení" bitcoinů. Co to tedy znamená a jak to funguje? Bez miningu by Bitcoin nemohl existovat, je to totiž motor celého systému.

Ten kdo „minuje" vlastně jen potvrzuje transakce a dostává za to bitcoiny. Těžení probíhá následovně. Těžař[12] si z tzv. Mempoolu[13], což je seznam nepotvrzených transakcí, vybere určitý počet transakcí, aby jejich celková velikost byla menší než 1 MB[14].

Jak si mezi velkým množstvím transakcí těžař vybírá? V každé transakci je totiž část poslaných bitcoinů určena pro těžaře. Tento poplatek (anglicky „fee") se stále mění a není přímo pevně daný. Obvykle se počítá dle velikosti transakce. Průměrná velikost transakce je 226 bytů a poplatek za ní je 0,0003164 BTC ~ 1,4 $.

Jak velký poplatek za transakci zaplatíte je na vás, ale pokud dáte příliš malý, tak si vaší transakci nemusí nikdo vybrat, a tedy nikdy nemusí být potvrzená. Naopak můžete za svojí transakci zaplatit více a tím zajistíte její rychlejší potvrzení.

Těžař při vybírání transakcí také zkontroluje jejich platnost (provede kontrolu „zůstatku" na adrese a podpisu). Z transakcí sestaví tzv. blok. Těžař do svého bloku také přidá speciální transakci, která se nazývá „coinbase transakce".

Coinbase transakce vytváří bitcoiny z ničeho. Pouze říká, že se má na adresu těžaře poslat pevně daný počet bitcoinů. Tato odměna je v dnešní době 12.5 BTC

[11] Průběžná změna velikosti celé databáze je k nahlédnutí např. zde https://blockchain.info/charts/blocks-size?timespan=all

[12] Označení pro někoho, kdo těží (minuje) bitcoiny

[13] Počet transakcí čekajících na potvrzení v Mempoolu je k nahlédnutí např. zde https://blockchain.info/charts/mempool-count

[14] Toto je pevně stanovená velikost daná protokolem.

~ 53950$. Odměna se za každých 210 000 potvrzených bloků půlí. Další půlení nastane pravděpodobně někdy v roce 2020. Původní odměna při spuštění Bitcoinu byla 50 BTC. Během fungování Bitcoinu se odměna půlila zatím pouze 2x. Tento systém zajišťuje to, že maximální počet bitcoinů bude ~ 21 miliónů. Od určité doby (předpokládá se rok 2140)[15] již žádné nové bitcoiny generovány nebudou. Bitcoin proto nikdy nepodlehne inflaci.

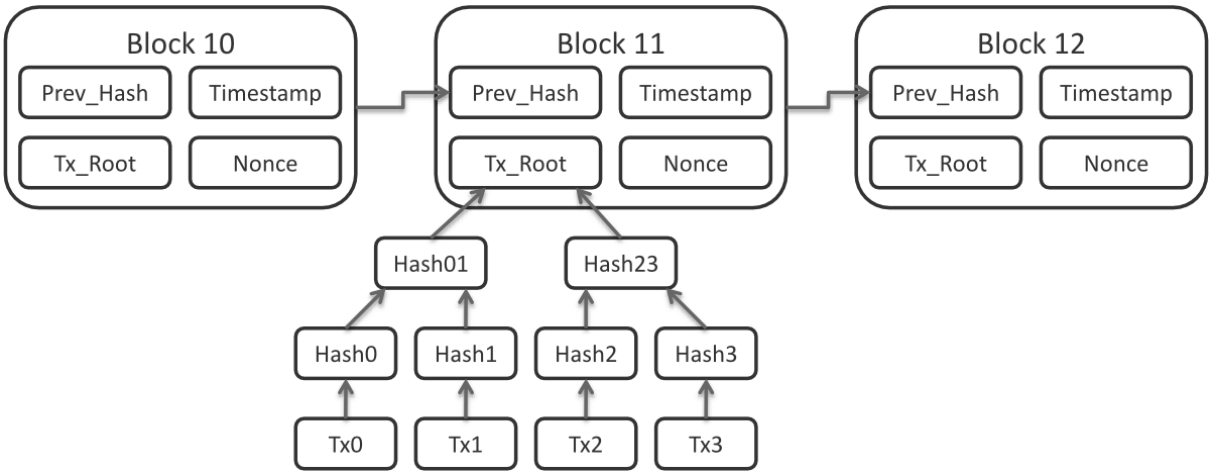
Těžaři od určité doby nebudou dostávat dostatek bitcoinů z coinbase transakce, aby se jim vyplatilo těžit. Právě proto dostávají i poplatky za každou transakci. Tento systém snad zaručí stálé fungování Bitcoinu.

Abychom zachovali řetězovou strukturu blockchainu, přidáme do našeho bloku ještě hash předchozího potvrzeného bloku. Nyní máme náš blok navázaný na předchozí a máme v něm transakce. Všechny transakce ale hašovat nechceme, způsobilo by to zbytečně náročný způsob kontroly určité transakce v bloku.

Proto z transakcí v bloku vytvoříme tzv. Merkle strom. Merkle strom je binární hašovací strom. Jeho listy jsou v našem případě všechny transakce v bloku. Strom je rekurzivně vytvořen zdola nahoru pomocí jednoduchého vzorce $otec = SHA256(SHA256(syn_1 + syn_2))$. V první iteraci jsou synové přímo transakce, v ostatních již jejich hashe. Výhoda Merkle stromu (s N listy) spočívá v tom, že pro kontrolu, zda obsahuje určitou transakci, je potřeba $2 \cdot \log_2 N$ operací.

Nyní konečně můžeme začít těžit tento blok. Algoritmus těžení je vcelku jednoduchý. Počítá se hash hlavičky bloku, jejíž hlavními komponenty jsou

- hash předchozího bloku
- kořen Merkle stromu všech transakcí
- tzv. „nonce“ (akronym pro „number used once“), což je obyčejné číslo
- timestamp neboli čas, kdy byl tento blok sestaven



Stuktura blockchainu

Celý proces těžení je hašování hlavičky bloku a změna hodnoty nonce, dokud tento hash nemá určitý počet nul v hexadecimální podobě. Těžař neustále

[15] V této době bude odměna menší než nejmenší jednotka bitcoinu tzv. satoshi. 1 satoshi = 0.00000001 BTC

zvyšuje hodnotu nonce dokud celý hash nevyhoví obtížnosti. Díky vlastnostem funkce SHA-256, každá změna hodnoty nonce následně výrazně a náhodně změní výsledný hash. Obtížnost se automaticky mění v závislosti na výkonnosti celé Bitcoin sítě. Obtížnost se mění každých 2016 vytěžených bloků, aby průměrný čas těžení byl kolem 10 minut.

V tuto chvíli obtížnost[16] je 922724699725,9628. Toto číslo přesně neznačí počet nutných nul. Počet nul nutných nul se spočítá dle jednoduchého vzorce.

$$počet\ bitů = 32 + \log_2 922724699725{,}9628$$

$$počet\ bitů \approx 71{,}7$$

Každé hexadecimální číslo reprezentuje 4 bity. Tedy potřebujeme minimálně $\frac{71,7}{4} = 18$ nul. Zde je například hash
`00000000000000000002de78311e4bb71d81c173c54db19bd113b0d01a985a5e`

bloku s číslem 485220, který vyhovoval obtížnosti (má dokonce 19 nul, ale stačilo by i 18), a tedy byl vytěžen neboli přidán do hlavního blockchainu.

Toto určení není úplně přesné, jelikož v realitě počítač přímo nepočítá nuly, ale snaží se, aby hash (což je vlastně hexadecimální číslo) byl menší než určitá hodnota spočítaná z obtížnosti. Počet nul je pouze hezká reprezentace tohoto problému pro člověka.

Nutno podotknout, že nalezení tohoto speciálního hashe (spíše tedy nalezení správné nonce) je nesmírně náročné. Naopak kontrola, zda určitá nonce funguje, je pouze jedno volání hašovací funkce SHA-256. Nalezení takové nonce je tzv. „proof of work“, tedy důkaz, že těžař provedl nějakou práci a pravděpodobně to není podvodník.

Ve chvíli, co těžař nalezne správný hash, pošle do sítě zprávu ve smyslu „Tento blok o těchto transakcích je mnou zkontrolován. Tady je důkaz: nonce“. Ostatní v síti tento důkaz zkontrolují. Jestliže je validní, napojí tento blok do své lokální kopie blockchainu (za blok, jehož hash je ve vytěženém bloku jako předchůdce) a začnou těžit na nový blok, navazující na tento poslední „vytěžený“ blok.

„Ostatní“ může být kdokoliv, kdo je zapojen do Bitcoinové sítě a staví si vlastní kopii blockchainu. Může to být někdo, kdo těží (tedy kontroluje transakce) nebo pouze někdo, kdo si akorát staví vlastní blockchain, aby mohl kontrolovat transakce.

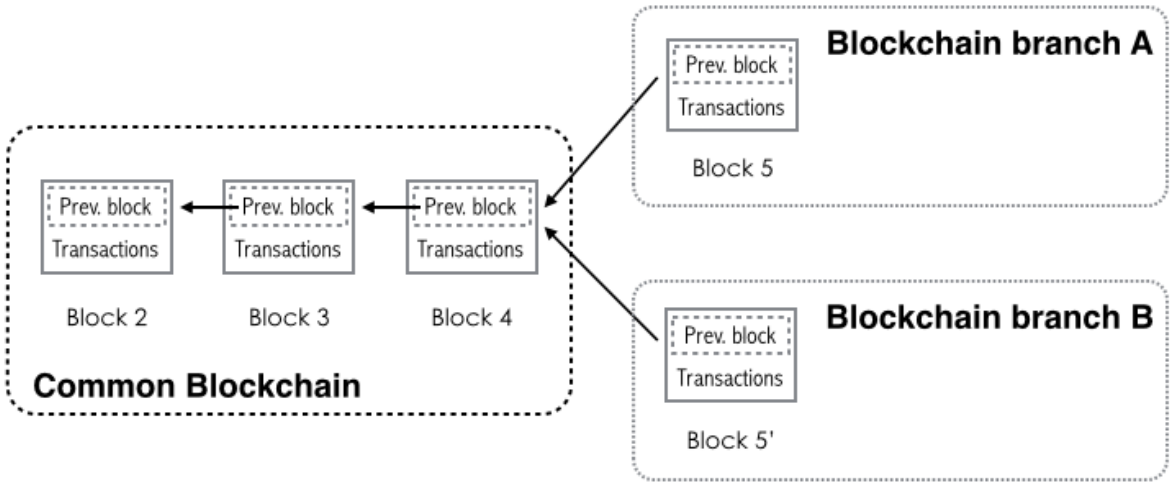
Znovu nastává pár potenciálních problémů kvůli architektuře blockchainu. Každý těžař si vybírá vlastní transakce, které chce těžit. Předchozí blok, ale ve většině případů bude stejný. Dva různí těžaři mohou vytěžit svůj vlastní blok v tu stejnou chvíli[17]. Oba tedy vyšlou do sítě zprávu, aby si každý updatoval svůj blockchain. Samozřejmě k ostatním na síti se jedna zpráva dostane dřív než

[16] Aktuální obtížnost je k dohledání např. https://blockexplorer.com/api/status?q=getDifficulty zde je graf, jak se měnila https://blockchain.info/charts/difficulty?timespan=all

[17] „Stejná chvíle“ je samozřejmě relativní pojem, ale pro ukázku problému intuitivní představa stačí.

druhá. Jeden těžař se může nacházet v Evropě a druhý v Asii. Tudíž většina lidí na síti v okolí Evropy dostane zprávu od těžaře v Evropě dříve než od toho v Asii. Máme tedy problém. Teoreticky oba tyto bloky jsou platné a nemůžeme jeden jen tak zahodit.

Bitcoin tento problém řeší tak, že blockchain rozdělí na 2 větve. Tomu se říká „fork“. Jako hlavní větev si každý vybere tu, na které se nachází blok, který k nim přišel nejdříve a začnou těžit. Zanedlouho (cca. 10 minut) by měl být vytěžený další blok, který musí navazovat na jednu ze dvou větví. To nastane, protože někteří si sestavili blok navazující na blok v jedné větvi (na obrázku Block 5) a někteří na blok (Block 5') v druhé. To znamená, že jedna větev se stane delší. Dle protokolu hlavní větev je ta, která je nejdelší, jelikož na ní byla provedena největší práce („proof of work“). Jistě se může stát stejný případ



Fork blockchainu

několikrát za sebou, ale to je málo pravděpodobné a s každým takovým případem pravděpodobnost exponenciálně klesá.

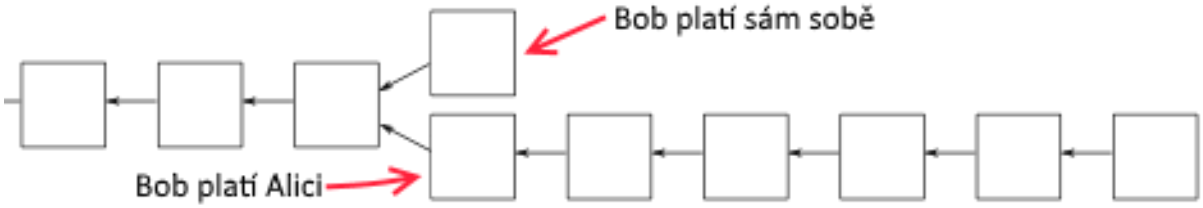
6. Ověřování transakce

Odpovíme tedy na otázku z předchozí případu. Kdy si Alice může být jistá, že transakce od Boba je platná?

Obvykle se považuje transakce za platnou, pokud se nachází v bloku, na který v blockchainu navazuje dalších 5 validních bloků a tyto bloky se nachází v nejdelší větvi. V případě 5 bloků, Alice bude muset počkat zhruba hodinu (10 + 5*10 minut), jelikož každý blok se kontroluje (těží) zhruba 10 minut. U každé transakce se ale doporučená doba liší v závislosti na hodnotě. Pokud půjde o velký počet bitcoinů, Bob bude jistě mít větší motivaci Alici podvést.

Řekněme, že Alice bude považovat transakci za platnou, jestliže je potvrzena 6krát[18]. Předpokládejme, že Bob vlastní 25 % výkonu celé bitcoinové sítě. Bob zveřejní transakci, ve které posílá Alici bitcoiny. Bob počká, až se tato transakce ověří. Alice v tuto chvíli považuje transakci za platnou a Bobovi tedy například pošle peníze.

Bob ale teoreticky může rozdělit blockchain (vytvořit novou větev) před posláním peněz Alici. Pokud by se mu teoreticky povedlo „předehnat" tu větev, podle které Alici poslal bitcoiny, celá síť by tu jeho delší větev začala považovala za hlavní. Alici by tedy chyběl důkaz, že již tyto bitcoiny vlastní a byla by tedy podvedena.



Naštěstí pravděpodobnost tohoto případu je také prakticky nemožná. Pravděpodobnost toho, že Bob vytěží 1 blok rychleji než zbytek sítě, je 1/4. Bob ale potřebuje 6krát za sebou vytěžit blok rychleji než zbytek sítě. Pravděpodobnost tohoto jevu je $(\frac{1}{4})^6 = 0{,}0002 = 0{,}02\ \%$. A to nebereme ani v potaz to, že hlavní větev se čím dál zvětšuje a Bob tedy musí teoreticky vytěžit více bloků.

Nejlepší argument, dle mého názoru, proč se ani tato situace nestane, není založen na tom, že je zcela nereálné dostat se k takové početní výkonnosti. Hlavní je zamyslet se proč by to někdo vůbec dělal? Bude člověk riskovat tuto malou pravděpodobnost a v podstatě hrát loterii? Nebo vloží svůj výkon na „dobré" účely? Dobré účely jsou nesmírně výhodné. Pravděpodobnost okradení (zisku), za těchto podmínek, o 25 BTC je nejvýše 0,02 %. Naopak pravděpodobnost vytěžení 2 náhodných bloků, za což je odměna 25 BTC + poplatky, je daleko vyšší. Bob v čestné situaci nepotřebuje x krát za sebou vytěžit blok, jemu je jedno, jaký blok vytěží. Bob také pravděpodobně vytěží každý 4. blok.

7. Anonymita Bitcoinu

Bitcoin je často chválen a také populární, hlavně kvůli tomu, že je vlastně anonymní. V jistém smyslu to je pravda. Není ale tak anonymní, jak by si většina lidí představovala. Všechny transakce provedené s bitcoiny jsou veřejně dostupné kdekoliv na internetu. Například na stránce https://blockchain.info/ si můžete prohlížet všechny bloky s transakcemi. Můžete si vyhledat libovolnou adresu, kolik na ní je bitcoinů, v jakých transakcích figuruje a mnoho dalších zajímavých informací.

[18] Na vytěžený blok obsahující „její" transakci navazuje další 5 bloků

Existují metody, jak těmto problémům předejít. Například problém možnosti určení, kolik má jaká adresa bitcoinů, většina peněženek, což je software pro správu vašich adres (tedy veřejných a privátních klíčů), umí vyřešit. Peněženka obvykle pro každou příchozí transakci vytváří novou adresu. To je možné, díky nesmírně velkému počtu možných adres. Existuje mnoho sofistikovaných způsobů, jak například používat jeden privátní klíč k více veřejným klíčům (adresám) apod. To jsou ale zbytečně komplikované věci, které nejsou důležité pro vysvětlení zásadního fungování Bitcoinu.

Samozřejmě také neexistuje jednoduchý způsob, jak najít vlastníka adresy. Úplně nemožné to ale není. Tyto způsoby nejsou přímo založené na (ne)fungování Bitcoinu, ale spíše na tzv. sociálním inženýrství, které využívá lidských chyb.

8. Budoucnost kryptoměn

Bitcoin zpopularizoval výše popsaný systém blockchainu. Proto se objevilo mnoho alternativ k Bitcoinu, které jsou ale v podstatě založené na stejném principu. Nejpopulárnějšími v dnešní době jsou Litecoin a Ethereum.

Litecoin funguje na stejném principu jako Bitcoin, akorát nepoužívá SHA-256 ale jinou hašovací funkci s názvem Scrypt. Výhoda této funkce oproti SHA-256 je, že by nemělo být snadné sestrojit speciální hardware[19], který jenom počítá tyto hashe.

Jeden z potenciálních problému Bitcoinu je, že se teoreticky stává lehce centralizovaný. Například v Číně, kde je levná elektřina, existují tzv. Bitcoin farmy, které právě těží bitcoiny pomocí tohoto specializovaného hardwaru. Více než 70 % bloků se v dnešní době vytěží právě v Číně.

Dále těžaři sami netěží samotné bloky. Místo toho se připojí do tzv. mining poolu. Mining pool funguje tak, že jeho administrátor sestaví blok a dá ho těžit všem, kteří jsou na tento mining pool napojeni. Pokud někdo vytěží tento blok, pošle ho administrátorovi. Administrátor tento blok pošle do bitcoinové sítě za sebe a dostane odměnu. Administrátor poté odměnu rozdělí mezi všechny, kteří se na těžení podíleli, v závislosti na jejich výkonu. Tento způsob je pro každého výhodnější než samostatně těžit svůj blok, protože pravděpodobnost vytěžení jednoho bloku, pomocí průměrně výkonného počítače, je v dnešní době o mnoho řádů menší než výhra v klasické loterii.

Ethereum je o hodně zajímavější a komplexnější než Bitcoin. Je také založené na blockchainovém principu. Ethereum se ale nemusí používat pouze jako virtuální měna. Celá síť Bitcoinu vydává svoji sílu na počítání hashů. Pomocí Ethereum můžete dělat v podstatě cokoliv. Jednoduchý popis Ethereum je „distribuovaný superpočítač“. Ethereum ale opravdu to nelze vysvětlit na pár

[19] tzv. ASIC (Application Specific Integrated Circuit). Obvykle se hashe počítají pomocí grafických kart pro počítače.

řádcích. Jen je důležité vědět, že je jeho potenciál mnohokrát větší než u Bitcoinu.

9. Závěr

Hodně lidem, kteří pochopí, jak Bitcoin funguje, musí vrtat hlavou skutečnost, že je to vlastně všechno založené důvěře. Aby Bitcoin fungoval správně, musí přes 50 % těžařů dodržovat protokol, aby nenastávaly situace popsané výše.

Spoléháme na vývojáře softwaru, že program funguje tak jak má, že v něm není žádná chyba. Většina softwaru Bitcoinu[20] je tzv. „open source“, neboli každý může nahlédnout do zdrojového kódu a software také vylepšovat. Obyčejný člověk ale na toto nemá čas ani dovednosti.

Spoléháme také na ostatní v bitcoinové síti, že nám naše transakce potvrdí.

Bitcoin protokol se stálé mění a vylepšuje. Přidávají se do něj různá vylepšení, jako například úspora velikosti bloku (tím pádem se do bloků vejde více transakcí). Tato vylepšení se nazývají „Bitcoin Improvement Proposals“ (zkráceně BIP) a jsou vymyšlena opět komunitou nadšenců. Schvalování BIP také závisí na „podpoře“ většiny na síti. Většina musí updatovat svůj software, který implementuje tyto změny, aby byl zbytek donucen také updatovat[21].

Bitcoin je podle mě nádherná ukázka spolupráce a důvěry nesmírného množství lidí na této planetě.

10. Zdroje

[1] http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/
[2] SCHNEIER, Bruce. Applied cryptography: protocols, algorithms, and source code in C. 2nd ed. New York: Wiley, c1996. ISBN 0-471-11709-9. (str. 258,259,260)
MENEZES, A. J., Paul C. VAN OORSCHOT a Scott A. VANSTONE. Handbook of applied cryptography. Boca Raton: CRC Press, c1997. ISBN 978-0849385230.
[3] https://github.com/bitcoinbook/bitcoinbook
[4] https://bitcoin.org/bitcoin.pdf
[5] https://bitcoin.stackexchange.com/questions/16687/solo-mining-just-for-luck-realistic
[6] https://bitcoin.stackexchange.com/questions/7724/what-happens-if-your-bitcoin-client-generates-an-address-identical-to-another-pe
[7] https://bitcointalk.org/index.php?topic=137.0
[8] https://en.wikipedia.org/wiki/History_of_bitcoin
[9] https://en.wikipedia.org/wiki/Elliptic_Curve_Digital_Signature_Algorithm
[10] https://blog.cloudflare.com/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/
[11] https://www.coindesk.com/math-behind-bitcoin/
[12] https://en.bitcoin.it/wiki/Secp256k1
[13] https://stackoverflow.com/questions/4014090/is-it-safe-to-ignore-the-possibility-of-sha-collisions-in-practice
[14] https://bitcoin.stackexchange.com/questions/9202/why-does-bitcoin-use-two-hash-functions-sha-256-and-ripemd-160-to-create-an-ad

[20] Ať již to je software pro mining, bitcoin peněženka apod.

[21] Tyto updaty se provádí také v principu forku neboli rozdělení blockchainu. Jestliže většina považuje updatovanou větev za hlavní, zbytek se musí přizpůsobit.

[15] https://cs.wikipedia.org/wiki/Blockchain
[16] https://samsclass.info/141/proj/BitDiff.htm
[17] https://cs.wikipedia.org/wiki/Eliptick%C3%A1_k%C5%99ivka
[18] https://bitslog.wordpress.com/2013/04/17/the-well-deserved-fortune-of-satoshi-nakamoto/
[19] https://www.andrew.cmu.edu/user//tnayak/papers/EllipticCurves.pdf
[20] https://cs.wikipedia.org/wiki/Grupa
[21] https://koclab.cs.ucsb.edu/teaching/ecc/project/2015Projects/Levy.pdf
[22] http://wstein.org/edu/2007/spring/ent/ent-html/node89.html
[23] https://cs.wikipedia.org/wiki/Narozeninov%C3%BD_%C3%BAtok