10. Blockchain: zdecentralizowane zaufanie

Stanisław Barański

Katedra Architektury Systemów Komputerowych
Wydział Elektroniki, Telekomunikacji i Informatyki
Politechnika Gdańska, Gdańsk
stanislaw.baranski@pg.edu.pl

Streszczenie
Bitcoin wprowadził innowację na wielu płaszczyznach. Jako pierwszy rozwiązał problem osiągania konsensusu w sieciach otwartych, stworzył zagadkę ekonomiczną w postaci globalnej waluty deflacyjnej, pozwolił na transfer pieniędzy tym, którzy wcześniej byli wykluczeni bankowo, ale przede wszystkim stworzył fundamenty pod platformę zdecentralizowanego zaufania. Zapoczątkował technologie zdecentralizowanych aplikacji, które dotychczas nie mogły istnieć bez zaufanej trzeciej strony. W pracy opisana została ta ostatnia płaszczyzna.

Słowa kluczowe: blockchain, zaufanie, kryptowaluty, Bitcoin, Ethereum

10.1 Wprowadzenie

Zaufanie jest fundamentalnym komponentem relacji społecznych. Mówimy, że ufamy komuś, kiedy spodziewamy się uczciwego zachowania, czyli postępowania według ustalonych zasad.

Korzystając z mediów społecznościowych, poczty elektronicznej, przestrzeni w chmurze, ufamy, że nasze dane są odpowiednio zabezpieczone oraz że dostawca nie używa ich do innych celów niż świadczenie nam usług. Poddając się badaniom genetycznym, ufamy, że nasze dane nie są dostępne dla pracowników laboratorium. Pożyczając koledze samochód, ufamy, że odda nam go w nienaruszonym stanie. W sytuacji gdyby się tak nie stało, ufamy, że sąd uczciwie rozwiąże konflikt na naszą korzyść. Sprzedawca, który sprzedaje nam swój towar, ufa nam, że pieniądze które otrzymał nie są podrobione oraz że w przyszłości będzie mógł kupić za nie inny towar. Jeśli płatność otrzymał przelewem, ufa bankowi, że ten nie cofnie przelanych mu pieniędzy. Ufa również, że bank nie zablokuje mu jego środków. Oddając głos w wyborach, ufamy, że nasz głos zostanie uwzględniony podczas liczenia oraz że zostanie poprawnie zaliczony dla zaznaczonego kandydata. Ufamy, że tylko uprawnieni wyborcy mogą oddać głos oraz że każdy wyborca może oddać tylko jeden głos. Na koniec, ufamy, że wyniki, które zostaną ogłoszone nie zostały zafałszowane.

Zaufanie jest spoiwem łączącym grupy społeczne. Pozwala się rozwijać i sprawia, że nasze życie staje się prostsze. Socjologowie są zgodni co do tego, że

"codzienne życie społeczne, które uznajemy za naturalne, jest bez zaufania po prostu niemożliwe" [14], zaufanie jest również "podstawową potrzebą transakcyjną" [21] (cyt. według [23]). Co jednak, jeśli chcielibyśmy osiągnąć ten sam, a nawet większy poziom zaufania, ale bez instytucji zaufania publicznego?

10.2 Zaufanie do waluty

Zacznijmy od waluty, bo to od niej zaczyna się historia zdecentralizowanego zaufania. Aby jakakolwiek waluta miała wartość, musi być deficytowa — jej ilość lub chociaż dostęp do niej musi być ograniczony. Pierwszą walutą tego typu było złoto, jego naturalne ograniczenie oraz koszt wydobycia idealnie spełniały te założenie. Złoto było walutą, która pozwalała na transakcje bez zaufanej trzeciej strony, było jednak niepraktyczne w transporcie oraz niewielkich płatnościach. System bankowy rozwiązał ten problem wprowadzając papiery wartościowe; każdy mógł zdeponować w banku złoto w zamian za certyfikat (banknot) pozwalający w późniejszym czasie na odzyskanie zdeponowanego złota, tworząc w ten sposób "papierowe" złoto, które było łatwiejsze w transporcie oraz pozwalało na mniejsze transakcje. Bank gwarantował, że liczba banknotów zawsze będzie odzwierciedlać ilość złota w depozycie, a ludzie *ufali*, że faktycznie tak będzie. O ile zagwarantować, że liczba złota nie przewyższy liczby banknotów, nie jest trudno, o tyle zagwarantować, że liczba banknotów nie będzie większa niż ilość rezerw złota, nie jest tak łatwo. Tworzenie banknotów jest o wiele mniej kosztowne niż wydobywanie złota.

Podczas I wojny światowej większość krajów Europy popadła w ogromne długi spowodowane wydatkami na działania wojenne. Banki pożyczały pieniądze, jednak przez standard złota ich zasoby były ograniczone. Kraje jednak potrzebowały kredytu, aby móc dalej się rozwijać. Rozwiązaniem tego problemu miało być tymczasowe zerwanie ze standardem złota, pozwalającym na dodruk pieniądza bez pokrycia kruszcem, co za tym idzie sztuczne pobudzenie gospodarki. Niestety, tymczasowe rozwiązania często okazują się ponadczasowe. W normalnej sytuacji odejście od parytetu złota spowodowałoby niepokoje społeczne. Wyjątkowa sytuacja jaką jest depresja, wojna lub kryzys pozwoliła jednak na wprowadzenie takich zmian na stałe. Do dziś większość krajów nie powróciła do standardu złota, a waluty fiducjarne — które bazują na zaufaniu do emitenta — nie mają pokrycia w żadnym surowcu. Jej wartość bazuje na monopolu jako legalnego środka płatniczego, popycie generowanym przez podatki oraz na zaufaniu pomiędzy dwoma stronami transakcji [11, 20].

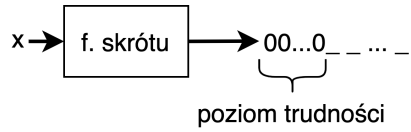
10.3 Historia kryptowalut

Po odejściu od parytetu złota oraz po powstaniu Internetu pojawiła się dodatkowa motywacja do stworzenia waluty, która posiadałaby deficytowe cechy złota, nie była kontrolowana przez żaden rząd, była anonimowa, której transfer byłby tak prosty i szybki jak wysyłanie e-maila.

W 1998 Wei Dai zaproponował walutę **b-money** [1], której wytwarzanie wymaga rozwiązywania zagadek kryptograficznych. Zagadka powinna być w klasie problemów NP, tak aby znalezienie rozwiązania było trudne — wymagało poświęcenie mocy obliczeniowej, a zweryfikowanie poprawności było łatwe. Przykładem może być znajdowanie liczby, która podana do funkcji skrótu zwróci liczbę z pewnego przedziału. Formalnie taką zagadkę możemy zapisać:

$$\text{znajdź } x \in \mathbb{N}, \text{takie że, } H(x) < c, \tag{10.1}$$

gdzie H jest funkcją skrótu np. SHA-256, a c jest liczbą definiującą poziom trudności — im mniejsza, tym ciężej znaleźć x spełniające nierówność; c można również interpretować jako liczba wymaganych zer w wyniku funkcji skrótu (patrz rysunek 10.1).



Rysunek 10.1. Zagadka kryptograficzna typu dowód pracy (ang. *proof-of-work*)

Szansa na znalezienie skrótu zaczynającego się od jednego zera wynosi 50%, od dwóch zer 25%, trzech 12,5%; poziom trudności rośnie wykładniczo wraz ze wzrostem wymaganych zer. Rozwiązywanie zagadki konsumuje moc obliczeniową, która jest deficytowym zasobem. Ilość wytworzonej waluty odpowiada ilości pracy włożonej w rozwiązanie zagadki. W b-money, ilość otrzymanej waluty obliczana jest poprzez stosunek kosztu rozwiązania zagadki na najbardziej optymalnym komputerze do ceny statystycznego koszyka zakupowego.

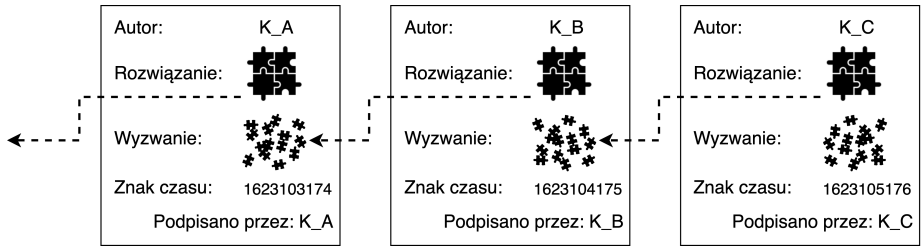
Każdy, kto rozwiąże taką zagadkę, może wymienić dowód (wartość x) na jednostki b-money. Dowód rozgłaszany jest w sieci *peer-to-peer* (p2p) które po otrzymaniu poprawnie rozwiązanej zagadki, aktualizują stan konta autora dowodu pracy (ang. *proof-of-work*). Podmioty w sieci identyfikowane są za pomocą pseudoanonimowych adresów (kluczy publicznych), a weryfikacja ich tożsamości jest możliwa za pomocą podpisów cyfrowych stworzonych przy pomocy klucza prywatnego.

Aby uniemożliwić podwójne tworzenie waluty, każdy węzeł w takiej sieci p2p zapisuje w swojej bazie danych wszystkie dowody, które otrzymał, nie pozwalając na duplikaty.

Załóżmy sytuację, w której Alice identyfikująca się kluczem publicznym K_A chce przesłać pieniądze do Boba identyfikującego się kluczem publicznym K_B. Transfer odbywa się przez stworzenie transakcji «Ja K_A przekazuję x jednostek z konta K_A na konto K_B. Podpisano K_A» i rozesłanie jej wszystkim węzłom w sieci. Po otrzymaniu wiadomości, każdy węzeł sprawdza poprawność podpisu oraz czy stan jednostek przypisanych do konta K_A jest nie mniejszy niż x, następnie obciąża konto K_A kwotą x i dodaje tą wartość do konta K_B.

Pomysł b-money był niestety mało praktyczny, zakładał istnienie *idealnego rozgłoszenia*, gdzie każda transakcja bezproblemowo i natychmiastowo zostaje odebrana przez każdego uczestnika sieci, zakładał, że każdy węzeł będzie zawsze dostępny oraz nie będzie zachowywał się szkodliwie (nie będzie węzłem bizantyjskim[16]). B-money nigdy nie został wcielony w życie.

W 2005 r. Nick Schabo oficjalnie opublikował koncept zdecentralizowanej cyfrowej waluty **BitGold** [2]. Podobnie jak Wei Dai, chciał jak najbardziej odzwierciedlić charakterystyczną cechę złota — deficytowość, jednak w świecie cyfrowym, na dodatek w sposób zdecentralizowany, bez zaufanej trzeciej strony. Podobnie jak b-money nowe jednostki tworzone są przez rozwiązanie zagadki kryptograficznej. Dowód pracy oznaczany jest znakiem czasowym (ang. *timestamp*) przez rozproszony system serwisów, a następnie dodawany do — również rozproszonego — rejestru tytułów własnościowych. Każda zagadka kryptograficzna bazuje na poprzednim rozwiązaniu zagadki, tworząc łańcuch dowodów (patrz rysunek 10.2).



Rysunek 10.2. Łańcuch dowodów w BitGold

Podobnie jak b-money, BitGold nie był odporny na węzły bizantyjskie, a rozproszona sieć serwerów rejestru tytułów własnościowych oraz serwerów znakujących czas nie miała motywacji do zachowywania się poprawnie.

10.4 Bitcoin

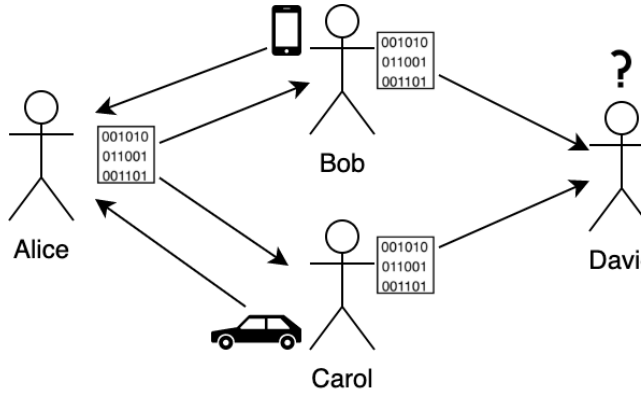
Dopiero w 2008 r. anonimowy autor pod pseudonimem Satoshi Nakamoto opublikował dokument pod tytułem "Bitcoin: A Peer-to-Peer Electronic Cash System", wyjaśniający jak osiągnąć otwarty, bezpieczny i zdecentralizowany system płatności *peer-to-peer* [18]. Biorąc pod uwagę b-money oraz BitGold, Bitcoin wprowadził niewiele innowacji. Jednak wystarczająco dużo aby zrewolucjonizować system bankowy, a później również inne systemy oparte na zaufanej trzeciej stronie.

10.4.1 Problem podwójnego wydawania

Bitcoin tak samo jak jego poprzednicy użył algorytmu *proof-of-work* [9], jednak nie tylko do wytwarzania nowych jednostek, ale przede wszystkim do osiągnię-

cia globalnego konsensusu — rozwiązując w ten sposób problem podwójnego wydawania (ang. *double spending problem*). Problemu, który dotychczas rozwiązywany był przez centralny podmiot — lub jak w przypadku BitGold przez konsorcjum podmiotów. Problem podwójnego wydawania polega na trudności stwierdzenia, czy dana jednostka waluty nie została już wcześniej wydana, innymi słowy, czy wcześniej nie została opublikowana transakcja, która wydaje te same jednostki waluty.

Rozważmy przykład z rysunku 10.3; Alice jest w posiadaniu ciągu bitów, które reprezentują pewną ilość waluty, Alice chcąc kupić od Boba telefon, wysyła mu ciąg bitów, Alice jednak jest nieuczciwa i zawczasu zrobiła sobie kopie tych bitów, następnie wydaje je ponownie, tym razem kupując od Carol samochód. Przewinienie to zostaje niezauważone do momentu, gdy zarówno Bob i Carol będą chcieli wydać swoje środki u jednego sprzedawcy — Davida, który nie będzie chciał przyjąć ciągu bitów, które już wcześniej otrzymał.



Rysunek 10.3. Problem podwójnego wydawania, w którym Alice wydaje dwa razy swoje środki, a David nie jest w stanie określić, kto jest ich faktycznym właścicielem

W przypadku banknotów problem ten jest rozwiązany przez fizyczne przemieszczenie, oraz nielegalność i trudność w kopiowaniu ich. W cyfrowym świecie nic nie stoi na przeszkodzie, aby zduplikować ciąg bitów i przesłać go do wielu osób, wydając jedną jednostkę wiele razy. W systemach z zaufaną trzecią stroną na ten problem odpowiada centralna jednostka (np. Visa, Mastercard lub sam bank), która autorytatywnie odpowiada, która transakcja miała miejsce wcześniej, unieważniając kolejną. Na taką centralną jednostkę można patrzeć jak na zegar, który znakuje czasem każdą transakcję.

10.4.2 Serwer znakowania czasem

Problem znakowania czasem jest łatwy dla pojedynczego serwera, ponieważ jednoznacznie może stwierdzić, która transakcja pojawiła się jako pierwsza, a która

jako druga. Dla systemów rozproszonych nie jest to już tak oczywiste, ponieważ każda jednostka może obserwować inną kolejność zdarzeń. Problem spotykany pod nazwą rozproszonego znakowania czasem, ma swoje korzenie w teorii względności[15] — dwie informacje wysłane dwoma różnymi kanałami (np. zapchanym łączem i wolnym łączem) pojawiają się w różnym czasie u odbiorcy, zakłamują faktyczną kolejność zdarzeń, tj. pierwsza wiadomość, która została wysłana wolniejszym łączem, może dotrzeć później do odbiorcy, niż druga wiadomość wysłana szybszym łączem. Sytuacja dodatkowo komplikuje się, gdy odbiorców jest wiele i każdy z nich obserwuje inną kolejność. Rozwiązaniem problemu jest *total-order broadcast*, w którym węzły uzgadniają między sobą wspólną kolejność, tworząc w ten sposób wirtualny zegar.

10.4.3 Konsensus

Ustalanie wspólnej kolejności jest szczególnym przypadkiem problemu rozwiązywanego przez algorytm konsensusu. Węzły muszą dojść do porozumienia odnośnie pewnej wartości, którą w tym przypadku jest kolejności zdarzeń.

Tradycyjne algorytmy konsensusu, jak Paxos[17] czy Raft[19] rozwiązują ten problem, jednak nie są odporne na węzły bizantyjskie (węzły, które mogą wysyłać sprzeczne informacje do różnych węzłów w sieci). Dzięki pracy Lessiego Lamporta [16] wiemy, że algorytm konsensusu może tolerować n węzłów bizantyjskich, jeśli łączna liczba węzłów w sieci wynosi co najmniej $3n + 1$, czyli aby sieć tolerowała co najwyżej jeden węzeł bizantyjski, musi składać się z łącznie czterech węzłów, dla dwóch węzłów bizantyjskich minimalna liczba węzłów wynosi siedem itd. Algorytmem tego typu jest PBFT [13], jednak nawet on nie sprawdzi się w sieciach p2p.

Wszystkie dotychczas wymienione algorytmy bazują na głosowaniu większościowym, w którym wspólnie uznaną wersją jest ta, która otrzymała większość głosów.

W sieciach otwartych — gdzie węzły mogą dowolnie dołączać i odłączać — liczba członków nie jest stała, dlatego określenie większości głosów jest problematyczne. Sytuacja staje się jeszcze gorsza, gdy uwzględnimy atak *sybila*, w którym to jeden podmiot dołącza do sieci wieloma tożsamościami osiągając zakłamaną opinie większościową. Przez długi czas problem osiągania konsensusu w otwartych sieciach p2p pozostawał nierozwiązany.

10.4.4 Dowód pracy jako rozwiązanie konsensusu na globalną skalę

Bitcoin rozwiązał ten problem w genialny sposób. Słuszną kolejnością zdarzeń jest ta kolejność, której została poświęcona największa ilość pracy obliczeniowej, bez znaczenia, czy pokrywa się z fizyczną kolejnością zdarzeń. Co istotne, taki mechanizm pozwala każdemu na uczestniczenie w algorytmie konsensusu, przy jednoczesnej odporności na atak *sybila*; dzieje się tak, ponieważ w algorytmie konsensusu z wykorzystaniem *proof-of-work*, na wynik nie wpływa liczba głosów — jak ma to miejsce w algorytmach bazujących na głosowaniach ilościowych — lecz sumaryczna moc obliczeniowa, która jest taka sama przy podawaniu się za

jeden, jak i wiele podmiotów. Co więcej, protokół Bitcoin gwarantuje poprawne działanie tak długo, jak większość sieci (liczona w mocy obliczeniowej) zachowuje się poprawnie, t.j. conajmniej 51% jest uczciwa.

Aby zrozumieć, jak Bitcoin to osiągnął, musimy spojrzeć na każdą część z osobna.

10.4.5 Transakcje

W Bitcoinie transfer pieniędzy pomiędzy Alice i Bob odbywa się poprzez stworzenie transakcji «Ja K_A zmieniam właściciela bitcoinów które otrzymałam w poprzedniej transakcji o skrócie h na konto o adresie K_B. Podpisano K_A.» Każdy może zweryfikować taki transfer cofając się w łańcuchu transakcji dochodząc do miejsca w którym dana jednostka została stworzona. Jeśli w którymś miejscu skrót transakcji się nie zgadza bądź podpis cyfrowy jest niepoprawny, transakcja jest odrzucana.

Wciąż pozostaje jednak problem podwójnego wydawania. Co jeśli Alice stworzy dwie transakcje,

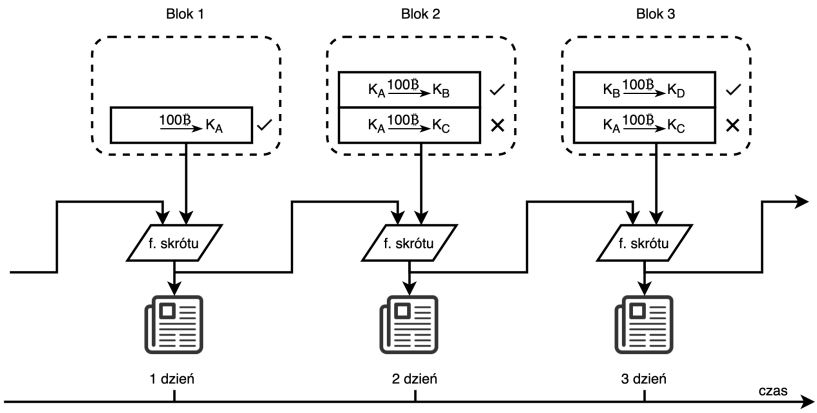
1. «Ja K_A zmieniam właściciela bitcoinów które otrzymałam w poprzedniej transakcji o skrócie h na konto o identyfikatorze $\mathbf{K_B}$. Podpisano K_A».
2. «Ja K_A zmieniam właściciela bitcoinów które otrzymałam w poprzedniej transakcji o skrócie h na konto o identyfikatorze $\mathbf{K_C}$. Podpisano K_A».

Jedną transakcję wyśle do Boba, a drugą do Carol. Obydwaj pomyślnie przeprowadzą weryfikację. W konsekwencji Alice dwa razy wyda środki o skrócie h.

Bitcoin znalazł rozwiązanie tego problemu bez zaufanej trzeciej strony.

Transakcje grupowane są w bloki; następnie skrót takiego bloku publikowany jest w miejscu, gdzie nie da się go usunąć ani zaprzeczyć, że istniał, np. zamieszczając w gazecie codziennej. Tysięcy wydrukowanych egzemplarzy nie da się zmienić, co za tym idzie nie da się zmienić zamieszczonego w nich skrótu. W konsekwencji każdy może zaufać, że dany skrót zostanie tam na zawsze. Takie powiązanie kryptograficznego skrótu z gazetą działa jak znak czasowy. Transakcje zawarte w bloku, którego skrót ukazał się we wczorajszej gazecie, są uznawane za wcześniejsze od transakcji zawartych w bloku, którego skrót ukazał się w dzisiejszej gazecie. Dodatkowo, aby zachować integralność w sytuacji, gdy gazeta wyda dwa egzemplarze jednego dnia, lub jakiegoś dnia nie wyda wcale, do funkcji skrótu podajemy zarówno blok, jak i skrót z poprzedniego bloku, otrzymując w ten sposób kryptograficzne powiązany łańcuch bloków (ang. *blockchain*).

Wracając do przykładu z Alice, która chciałaby wydać dwa razy swoje bitcoiny. Rysunek 10.4 przedstawia hipotetyczny blockchain oparty o gazetę codzienną. Pierwszego dnia, Alice otrzymała 100 jednostek bitcoinów (dla uproszczenia pomijamy ich pochodzenie), transakcja trafia do pierwszego bloku, którego skrót trafia do gazety z pierwszego dnia. Alice uzgadnia z Bobem, że prześle mu 100 bitcoinów w zamian za telefon, następnie z Carol że również prześle jej 100 bitcoinów w zamian za samochód. Bob i Carol zgadzają się pod warunkiem, że ich transakcja pojawi się w bloku w gazecie z drugiego dnia. Alice próbuje



Rysunek 10.4. Blockchain oparty o gazetę codzienną

opublikować dwie transakcje, pierwszą do Boba, drugą do Carol. Bob akceptuje transakcję i wydaje telefon dla Alice; Carol widząc, że środki Alice zostały już przelane, nie akceptuje transakcji. Alice może próbować kolejnego dnia w gazecie z dnia trzeciego, lecz Carol przechodząc po łańcuchu rozpozna, że konto Alice, po transakcji z Bobem, jest puste. David przyjmie transakcje od Boba, ponieważ nakładając wszystkie (poprawne) transakcje otrzyma stan w którym Bob jest w posiadaniu 100 bitcoinów.

Nietrudno się jednak nie zgodzić, że waluta cyfrowa oparta o gazety nie jest najlepszym rozwiązaniem. Bitcoin osiągnął jednak ten sam mechanizm znakowania czasem przy użyciu zdecentralizowanej sieci p2p, oraz innowacyjnego podejścia do osiągania konsensusu za pomocą algorytmu dowodu pracy.

Dowód pracy w Bitcoinie polega na poszukiwaniu liczby, która w połączeniu z blokiem i podaniu do funkcji skrótu SHA-256 zwróci liczbę która zapisana binarnie rozpocznie się od zadanej długości zer. Każde dodatkowe zero wykładniczo zwiększa trudność zagadki. Wymagana liczba zer zależy od sumy mocy obliczeniowej całej sieci; jeśli kolektywna móc obliczeniowa rośnie, wzrasta liczba wymaganych zer, jeśli moc spada, wraz z nią liczba wymaganych zer na początku skrótu. Protokół Bitcoin stara się produkować bloki co 10 min — innymi słowy, co 10 min ten "globalny zdecentralizowany zegar" znakuje czasem aktualny blok.

Zmiana transakcji w bloku jest możliwa, ale wymaga ponownego obliczenia dowodu pracy dla zmodyfikowanego bloku. Co więcej, ponieważ każdy kolejny blok wskazuje na skrót poprzedniego bloku, dla każdego kolejnego bloku również trzeba policzyć nowy dowód pracy. Jednak to nie koniec, aby sieć przyjęła taką zmianę musi ona "wyprzedzić" aktualny łańcuch bloków; wyprzedzenie jest możliwe jeśli podmiot jest w stanie produkować nowe dowody pracy szybciej niż reszta sieci, stąd też założenie, że sieć jest bezpieczna tak długo, jak większość mocy obliczeniowej sieci jest kontrolowana przez uczciwe podmioty.

10.4.6 Motywacja ekonomiczna

Węzły chcą wykonywać dowody pracy i konsumować energię ponieważ za znalezienie zagadki czeka ich nagroda w postaci bitcoinów. Dopóki wartość nagrody jest większa niż koszt energii elektrycznej dopóty liczenie dowodów pracy — nazywane również kopaniem (ang. *mining*) — jest opłacalne i znajdą się ludzie którzy będą chcieli taka inwestycje podjąć.

Nagroda za każdy blok zmniejsza się w czasie, początkowo wynosiła 50 BTC, i co każde 210 tys. bloków (około cztery lata) zostaje zmniejszona o połowę, następuje tak zwane dzielenie (ang. *halving*). W latach 2012-2016 nagroda wynosiła 25 BTC, od 2016 do 2020 o połowę mniej — 12,5 BTC, aktualnie wynosi 6,25 BTC, a po roku 2140 wynosić będzie zero. Jak w takim razie węzły będą zmotywowane do liczenia dowodów pracy i zabezpieczania sieci? Poprzez system prowizji. Każda transakcja w sieci Bitcoin może zawierać "napiwek" dla górnika, jest on dobrowolny, jednak górnicy zachowując się racjonalnie wybierają transakcje, z których otrzymają największe wynagrodzenie. Im bardziej sieć obciążona, tym większe opłaty transakcyjne, tym większe wynagrodzenie dla górników. Mechanizm opłat transakcyjnych jest również istotny ze względu na ochronę sieci przed atakiem odmowy dostępu (ang. *deny-of-service*). Gdyby transakcje nie wymagały opłat, z łatwością można byłoby zalać sieć mikrotransakcjami. Opłaty transakcyjne sprawiają, że takie działanie jest nieopłacalne.

10.4.7 Podsumowanie

W ten sposób Bitcoin jako pierwszy osiągnął system płatności bez zaufanej trzeciej strony. W tym systemie sprzedawca nie musi ufać kupującemu, że ten nie wydał wcześniej swoich jednostek gdzie indziej. Sam może to zweryfikować w swojej lokalnej kopii blockchaina. Ma też pewność, że nikt nie wykona obciążenia zwrotnego, ponieważ jest to trudne obliczeniowo, co za tym idzie, bardzo kosztowne do wykonania.

Bitcoin zagwarantował, że liczba wszystkich jednostek w systemie nigdy nie przekroczy 21 mln bitcoinów. Jest jednak bardzo prawdopodobne, że liczba ta będzie malała; spowodowane jest to nieustanną utratą środków przez zgubione klucze prywatne oraz przelewów na nieistniejące konta. W teorii zmniejszająca się podaż i zwiększający się popyt prowadzić musi do wzrostu ceny. W praktyce jednak istnieje wiele czynników z powodu których nie musi się tak dziać.

10.5 Ethereum

Bitcoin osiągnął coś więcej niż system płatności w modelu p2p. Zbudował system bezstronnej, otwartej, neutralnej, motywowanej prawami wolnego rynku oraz zdecentralizowanej zaufanej trzeciej strony, której zaufanie nie leży w wierze w uczciwość, lecz w prawa matematyki.

Po kilku latach Vitalik Buterin, Gavin Wood oraz kilku innych twórców zauważyli, że potencjał technologii stojącej za Bitcoinem jest dużo większy niż sam transfer pieniędzy [12, 22].

Dotychczas, gdy ktoś chciał zmodyfikować lub rozszerzyć funkcjonalność Bitcoina, musiał zmodyfikować protokół i postawić całkowicie nową sieć, następnie przekonać górników, aby poświęcili swoją moc obliczeniową na liczenie dowodów pracy (co za tym idzie zabezpieczali sieć) dla jego projektu, a nie Bitcoina. Powodowało to dwa problemy, po pierwsze waluta taka musiała mieć już jakąś wartość, aby kopanie jej było opłacalne. Po drugie górnik, który przenosił swoją moc obliczeniową na kopanie innej kryptowaluty, zabierał ją z kopania Bitcoina, co za tym idzie obniżał jego bezpieczeństwo.

Ethereum rozwiązało ten problem tworząc platformę wykonawczą dla programowalnych transakcji, tzw. smart kontraktów. Podczas gdy transakcja w Bitcoinie stanowi o zmianie właściciela bitcoinów, smart kontrakt może być dowolnym programem wykonywanym w ramach transakcji. Transakcja może — podobnie jak w Bitcoinie — być zwykłym przelewem waluty, może mieć też inne znaczenie jak na przykład oddanie głosu w wyborach, zablokowanie środków w ramach lokaty, wzięcie pożyczki, wynajem krótkoterminowy samochodu, zmiana właściciela nieruchomości, nadanie uprawnień lekarzowi do wykonania obliczeń na naszych danych medycznych, obstawienie zakładu bukmacherskiego, i wiele innych.

Transakcje wykonywane są na wirtualnej maszynie Ethereum (ang. *Ethereum Virtual Machine*, EVM), która jest maszyną *quasi-Turing-skończoną* — pozwala na implementację dowolnego algorytmu wraz z rozgałęzieniami oraz pętlami; *quasi* oznacza, że nie występuje problem stopu — maszyna nie może się zapętlić w nieskończoność, gdyż każda instrukcja procesora konsumuje kryptowalutę; jeśli transakcja zużyje wszystkie dostępne środki, zostaje przerwana.

Na sieć Ethereum można patrzeć jak na publiczny klaster węzłów. na których uruchomiony jest JVM (*Java Virtual Machine*), do którego każdy może wysłać *bytecode* do wykonania. Zadanie wykonywane jest przez wszystkie węzły w klastrze; trwały stan (zapisywany między wykonaniami smart kontraktu) przechowywany jest w łańcuchu bloków; algorytm konsensusu zapewnia, że każdy węzeł w sieci posiada tą samą wersję łańcucha; użytkownik wysyłający *bytecode* musi również "podpiąć swoją kartę płatniczą", z której węzły konsumują środki za każdą instrukcje procesora. EVM w porównaniu do JVM, nie posiada dostępu do interfejsu sieciowego, ani innego interfejsu systemowego. Aby uniknąć wyścigów, EVM nie obsługuje wielowątkowości, a wykonywany smart kontrakt nie może komunikować się z niczym innym, niż inne smart kontrakty. EVM może zapisywać swój stan do dedykowanej dla niego przestrzeni w łańcuchu bloków, która również jest dodatkowo płatna.

Wykonanie smart kontraktu jest transparentne i możliwe do zweryfikowania. Ethereum stworzyło platformę do aplikacji, które posiadają wszystkie cechy bezpieczeństwa, transparentności, oraz odporności na cenzurę, jakie posiadał Bitcoin. Aplikacje tworzone w oparciu o Ethereum zostały nazwane Decentralized Applications (DApps), a całe podejście do tworzenia całkowicie zdecentralizowanych aplikacji zostało okrzyknięte mianem web 3. Aplikacje tego typu cechuje decentralizacja wszystkich komponentów: backendu, frontendu, przechowywania danych, wymiany wiadomości oraz systemu nazw. Zamiast tworzyć aplikację,

która komunikuje się z centralnym serwerem, który korzysta z prywatnej bazy danych, można stworzyć aplikację, która komunikuje się ze smart kontraktami, a dane przechowuje w sieciach p2p (np. IPFS [10] lub Ethereum Swarm [6]).

Takie podejście oferuje wiele zalet niedostępnych dla zcentralizowanych aplikacji:

1. Dostępność — sieci Ethereum działa jak ogromny klaster, awaria nawet wielu węzłów nie powoduje braku dostępności aplikacji. Co więcej, każdy użytkownik może uruchomić swój własny lokalny węzeł Ethereum przeprowadzając interakcje bezpośrednio z nim, uniezależniając się od jakiegokolwiek dostawcy.
2. Transparentności — każda transakcja, stan konta, smart kontrakt oraz jego stan w blockchainie jest dostępny publicznie dla każdego.
3. Niezmienialność — każda interakcja z blockchainem zostaje zapisana w nim na zawsze, zapewniając niezaprzeczalność. Każdy smart kontrakt zawsze będzie miał ten sam kod, zapewniając determinizm wykonywania. Każda treść będzie miała ten sam skrót, zapewniając jednoznaczność nazewnictwa.
4. Odporność na cenzurę — sieć Ethereum działa na globalną skalę, nie jest kontrolowana przez żadną organizację ani żaden kraj. Każdy węzeł jest finansowo motywowany do przetwarzania wszystkich transakcji bez względu na ich pochodzenie i przeznaczenie. Zatrzymanie sieci Ethereum jest tak samo ciężkie jak zatrzymanie każdej innej sieci p2p.

Wszystkie te zalety mają jednak swoją cenę. Tworzenie zdecentralizowanych aplikacji wymaga zmiany podejścia w projektowaniu, wytwarzaniu oraz wdrażaniu. Raz opublikowany smart kontrakt nie może być zmieniony, co powoduje duży nacisk na wytwarzanie kodu wysokiej jakości. Tworzenie smart kontraktów można porównać do tworzenia sprzętu komputerowego, naprawa każdego błędu wymaga kosztownych akcji serwisowych oraz wpływa negatywnie na renomę marki. Niektóre błędy są katastrofalne w skutkach i pociągają za sobą utratę środków, których nie da się odzyskać [3].

Używanie aplikacji tego typu oferuje dużo wolności, ale w zamian wymaga wiele odpowiedzialności. Użytkownicy są właścicielami swoich danych, w przypadku utraconego klucza prywatnego nie ma możliwości przywrócenia go, nie ma żadnego wsparcia technicznego, do którego można się zgłosić.

10.6 Otwarte problemy

Wiele aplikacji nie może istnieć w paradygmacie web 3 z powodu (1) słabej skalowalności (mierzonej w liczbie transakcji na sekundę), (2) wysokich opłat transakcyjnych, (3) braku regulacji prawnych, (4) słabej prywatności, (5) ograniczonej interoperacyjności pomiędzy systemami, (6) braku standaryzacji.

Skalowalność jest problematyczna, ponieważ jest związana z trylematem skalowalności [5, 7] który cechują trzy właściwości. System blockchainowy chciałby posiadać każdą z nich, ale można wybrać tylko dwie:

1. Skalowalność — kolektywna przepustowość przetwarzania transakcji całej sieci większa niż przepustowość jednego węzła w sieci.
2. Decentralizacja — węzeł sieciowy powinien być możliwy do uruchomienia na przeciętnym laptopie konsumenckim.
3. Bezpieczeństwo — sieć powinna być odporna na znaczną część (idealnie 50%) szkodliwych węzłów partycypujących w algorytmie konsensusu.

Większość blockchainów wybiera bezpieczeństwo, pozostawiając wybór między skalowalność a decentralizację. Tradycyjne blockchainy, jak Bitcoin czy Ethereum, wybierają decentralizację, kosztem skalowalności osiągając jedynie 3-15 transakcji na sekundę. Te które stawiają na skalowalność, podnoszą wymagania sprzętowe oraz wprowadzają "superwęzły", stając się mocno zcentralizowane.

Zwiększenie wydajności i obniżenie kosztów transakcyjnych próbuje się osiągnąć na kilka sposobów: wprowadzeniem *shardingu*, który sprawi, że każdy węzeł w sieci będzie przetwarzał nie wszystkie, lecz podzbiór transakcji w ramach swojego *shardu*; kolejnym pomysłem jest wprowadzenie rozwiązań drugiej warstwy, na której to węzły nie będą musiały komunikować się nieustannie z blockchainem, lecz wysyłać mu zbiorcze transakcje co jakiś czas. Rozwiązania drugiej warstwy można implementować na wiele sposobów, jednym z nich (najbardziej obiecującym [4]) jest technologia *zk-rollup* [8], która pozwala na weryfikację wykonanych transakcji bez konieczności wykonywania ich na każdym węźle oraz zapewnia wysoki poziom prywatności poprzez zastosowanie dowodów wiedzy zerowej (ang. *zero-knowledge proofs*).

Różne projekty blockchanowe rozwiązują każdy z tych problemów w nieco inny sposób, sprawiając, że środowisko to jest chaotyczne i brak mu standaryzacji; prawnikom natomiast przysparza wyzwanie precyzyjnego zdefiniowania technologii w kontekście prawnym.

10.7 Wnioski końcowe

Fenomenem technologii blockchain nie jest rozproszona i bezpieczna baza danych, nie jest nim również możliwość uruchamiania smart kontraktów; fenomenem jest platforma zdecentralizowanego zaufania umożliwiająca tworzenie aplikacji, które wcześniej mogły istnieć tylko w modelu z centralną zaufaną jednostką.

Kryptowaluty są jedynie pierwszą — a zarazem fundamentalną — aplikacją tej technologii. Są dowodem dojrzałości, wiarygodności i rzetelności. Pozwalają na finansowanie dalszych prac — elementu, bez którego wiele projektów open-source upada. Co najistotniejsze, zapewniają model bezpieczeństwa bazujący na teorii gier i prawach wolnego rynku.

Kolejnymi aplikacjami są usługi rządowe, internetowe wybory, tokenizacja udziałów w organizacjach (tzw. ICO), katalogi zasobów cyfrowych, instrumenty finansowe (tzw. DeFi), zakłady bukmacherskie, rynek energetyczny w modelu *peer-to-peer*, systemy nazw domen, a wiele z nich wciąż czeka na odkrycie.

Bibliografia

1. b-money. http://www.weidai.com/bmoney.txt, (Accessed on 06/08/2021).
2. Bit gold | satoshi nakamoto institute. https://nakamotoinstitute.org/bit-gold/, (Accessed on 06/08/2021).
3. The dao attack: Understanding what happened – coindesk. https://www.coindesk.com/understanding-dao-hack-journalists, (Accessed on 06/14/2021).
4. A rollup-centric ethereum roadmap. https://ethereum-magicians.org/t/a-rollup-centric-ethereum-roadmap/4698, (Accessed on 10/06/2021).
5. Sharding-faqs | ethereum wiki. https://eth.wiki/sharding/Sharding-FAQs, (Accessed on 06/14/2021).
6. Swarm. https://www.ethswarm.org/, (Accessed on 06/13/2021).
7. Why sharding is great: demystifying the technical properties. https://vitalik.ca/general/2021/04/07/sharding.html, (Accessed on 06/13/2021).
8. Zk-rollups - ethhub. https://docs.ethhub.io/ethereum-roadmap/layer-2-scaling/zk-rollups/, (Accessed on 10/06/2021).
9. Back, A., et al.: Hashcash — a denial of service counter-measure (2002).
10. Benet, J.: Ipfs-content addressed, versioned, p2p file system. arXiv preprint arXiv:1407.3561 (2014).
11. Bordo, M.D., Kydland, F.E.: The gold standard as a rule: An essay in exploration. Explorations in Economic History (4), 423–464 (1995). https://doi.org/10.1006/exeh.1995.1019, https://www.sciencedirect.com/science/article/pii/S0014498385710194.
12. Buterin, V., et al.: Ethereum white paper. GitHub repository **1**, 22–23 (2013).
13. Castro, M., Liskov, B., et al.: Practical byzantine fault tolerance. In: OSDI. vol. 99, pp. 173–186 (1999).
14. Good, D.: Individuals, interpersonal relations, and trust. Trust: Making and breaking cooperative relations pp. 31–48 (2000).
15. Lamport, L.: Time, clocks, and the ordering of events in a distributed system. In: Concurrency: the Works of Leslie Lamport, pp. 179–196 (2019).
16. Lamport, L., Shostak, R., Pease, M.: The byzantine generals problem. In: Concurrency: the Works of Leslie Lamport, pp. 203–226 (2019).
17. Lamport, L., et al.: Paxos made simple. ACM Sigact News (4), 18–25 (2001).
18. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. Tech. rep. (2019).
19. Ongaro, D., Ousterhout, J.: In search of an understandable consensus algorithm. In: 2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14). pp. 305–319 (2014).
20. Song, H., Sierakowski, T.: Wojna o pieniądz: prawdziwe źródła kryzysów finansowych. Wydawnictwo "Wektory"(2010), https://books.google.pl/books?id=jUuVpwAACAAJ.
21. Turner, J.H.: Face to face: Toward a sociological theory of interpersonal behavior. Stanford University Press (2002).

22. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger .
23. Zemła, Ł., et al.: Zaufanie. fundament społeczeństwa. Studia Politicae Universitatis Silesiensis (4-5), 322–328 (2009).