ივანე ჯავახიშვილის სახელობის თბილისის სახელმწიფო უნივერსიტეტი
ზუსტ და საბუნებისმეტყველო მეცნიერებათა ფაკულტეტი
მათემატიკის დეპარტამენტი

მამუკა სახელაშვილი

ბლოკჩეინი, ბიტკოინი და მათი მათემატიკური საფუძვლები

სამაგისტრო პროგრამა: მათემატიკა

ნაშრომი შესრულებულია მათემატიკის მეცნიერებათა მაგისტრის აკადემიური ხარისხის მოსაპოვებლად

სამაგისტრო ნაშრომის ხელმძღვანელი: ბაჩუკი მესაბლიშვილი, ფიზიკა-მათემატიკის მეცნიერებათა კანდიდატი, ასოცირებული პროფესორი.

თბილისი

2023

სარჩევი

ანოტაცია

ნაშრომში განხილულია ბლოკჩეინ ტექნოლოგიები და მათი წარმოშობის მიზეზი. ახსნილია რა საჭიროების გამო შეიქმნა ბლოკჩეინი და რა უპირატესობები აქვს მას არსებულ საბანკო სისტემასთან შედარებით. ასევე წარმოდგენილია ის მათემატიკური პრობლემები, რომლებზეც დაფუძნებულია ბლოკჩეინ ტექნოლოგიების გამართულად და უსაფრთხოდ მუშაობა. ნაშრომში ასევე ახსნილია რა არის ბიტკოინი და როგორ გამომუშავდება იგი ბლოკჩეინ სისტემის გამოყენებით. ნაშრომის ბოლოს მოყვანილია ბლოკჩეინ სისტემის სხვადასხვა გამოყენებები.

Summary

In this paper blockchain technologies and their origins are considered. It explains why the blockchain was created and what advantages does it have compared to the existing banking system. Also it explains the mathematical problems, which is a foundational blocks for blockchain to work safely and correctly. This paper also explains what is bitcoin and how bitcoin mining works. In the end it contains some examples how blockchain can be used to solve different every day problems.

შესავალი

ბლოკჩეინ ტექნოლოგიები და ბიტკოინი რევოლუციური სიახლეა, რომელიც მთლიანად ცვლის აქამდე არსებულ სტანდარტებს და რეგულაციებს საბანკო სფეროში. ეს არის დეცენტრალიზებული სისტემა, რომელსაც არ ყავს მაკონტროლებელი. შესაბამისად, იგი მთლიანად თავისუფალია რომელიმე ქვეყნის მიერ დადგენილი წესების, შეზღუდვებისა თუ გადასახადებისგან.

ბლოკჩეინ სისტემაზეა დაშენებული ბიტკოინი, რომელიც ციფრული ვალუტაა. ამ დროისთვის ბაზარზე 19 მილიონი ბიტკოინი არსებობს. თუმცა, ახალი ბიტკოინები ისევ გენერირდება და ეს გაგრძელდება იქამდე, სანამ ბიტკოინების საერთო რაოდენობა 21 მილიონს არ მიაღწებს. ამის შემდეგ კი ბაზარზე იარსებებს მუდმივი რაოდენობის ბიტკოინები, მყარი ღირებულებით. იქამდე კი ბიტკოინის ღირებულება განიცდის დიდ რყევებს, 20 ათასი დოლარიდან 60 ათას დოლარამდე.

ბლოკჩეინ ტექნოლოგიები და ბიტკოინის არსებობა დაფუძნებულია მათემატიკური პრობლემების გადაჭრასთან. ტრანზაქციის შექმნიდან, მის დადასტურებამდე, ბლოკჩეინ სისტემა უამრავ ნაბიჯს გადის და მრავალ მათემატიკურ პრობლემას წყვეტს. ამ ნაშრომში მოყვანილია ეს პრობლემები და ასევე ახსნილია მათი გადაჭრის გზები.

თავი 1

1.1 საბანკო სისტემები და გადარიცხვები

ტრადიციული საბანკო გადარიცხვები გლობალური ფინანსური სისტემის ხერხემალს წარმოადგენს. მისი საშუალებით ბიზნესებსა და ინდივიდუალურ მომხმარებლებს მარტივად შეუძლიათ თავიანთი ფინანსების მართვა სახლიდან გაუსვლელად. გადარიცხვების შესრულებისას მომხმარებლებს ურთიერთობა აქვთ ისეთ ორტანიზაციებთან, როგორებიცაა ბანკები, საკრედიტო დაწესებულებები და სხვადასხვა ფინანსური ერთეულები. ყოველი მათგანი უზრუნველყოფს გადარიცხვების უსაფრთხოებას და გადარიცხვაში არსებული მონაცემების უცვლელობას.

თანხის გადარიცხვისთვის საჭიროა მომხმარებელმა მიუთითოს თანხის მიმღების მისამართი და თანხის ოდენობა. თუმცა თითოეული გადარიცხვა მრავალ შემოწმებას გადის, სანამ გადარიცხული თანხა დანიშნულების ადგილს მიაღწევს. ფინანსურმა ინსტიტუტმა, რომლის ფარგლებშიც ხორციელდება გადარიცხვა, უნდა დაადგინოს მომხმარებლის ვინაობა, დარწმუნდეს რომ თანხის გადამრიცხავი და ასევე მიმღები ნამდვილად არსებობენ, ასევე შეამოწმოს, რომ გადამრიცხავის ანგარიშზე ნამდვილად არსებობს გადასარიცხი თანხა. შემოწმებების გავლის შემდეგ, დაამუშაოს ტრანზაქცია და განაახლოს ბალანსზე არსებული თანხის ოდენობა გამომგზავნისა და მიმღებისათვის.

ტრადიციული საბანკო სისტემა დამყარებულია ცენტრალიზებულ მოდელზე, სადაც ფინანსური ინსტიტუტი მოქმედებს როგორც ცენტრალური ავტორიტეტი. იგი თავად ინახავს ყველა მონაცემს ყველა მომხმარებლის შესახებ ცენტრალურ მონაცემთა ბაზაში. იგი ასევე პასუხისმგებელია გადარიცხვის შემოწმებასა და ავთენტურობაზე. ხშირად საბანკო სისტემებს ასევე უწევთ ქვეყანაში არსებული რეგულაციების გათვალისწინება და მათი მიხედვით გადარიცხვების მართვა.

გადარიცხვები ხშირად დაკავშირებულია შესაბამის ხარჯთან. ხარჯი შეიძლება იყოს როგორც ანგარიშის გახსნის საკომისიო, ასევე ერთი ვალუტიდან მეორეში კონვერტაციის ხარჯი. იმ შემთხვევაში, თუ გადარიცხვა ხდება ერთი ქვეყნიდან მეორეში, მაშინ ამას კიდევ ემატება საერთაშორისო გადარიცხვის საკომისიო.

ფულადი ხარჯის გარდა გადარიცხვა ასევე ხასიათდება ხანგრძლივობით, ხანგრძლივობა კი შეიძლება მერყეობდეს რამდენიმე წამიდან (მაგალითად, როცა გადარიცხვა ხდება ერთ ბანკში არსებული ერთი ანგარიშიდან მეორეზე) რამდენიმე დღემდე (როცა გადარიცხვა ხდება ერთი ქვეყნის ბანკიდან მეორე ქვეყნის ბანკში).

ყოველივე ზემოთ თქმულიდან გამომდინარე, შეგვიძლია განვიხილოთ ტრადიციული საბანკო სისტემის შემდეგი ნაკლოვანებები:

- ნაკლები გამჭირვალობა: რადგან ბანკი ფლობს მთლიან სისტემას, მომხმარებლებს ნაკლები ხილვადობა აქვთ იმის შესახებ თუ როგორ ხდება თითოეული ტრანზაქციის

დამუშავება ბანკის მიერ. ასევე ნაკლები წვდომა აქვთ ინფორმაციაზე, თუ როგორ ინახება თითოეული მომხმარებლის პირადი და მათ ბალანსზე არსებული თანხის შესახებ მონაცემები.

- ცენტრალიზებული სისტემა და უსაფრთხოება: მთელი ინფორმაცია გადარიცხვებისა და თითოეული მომხმარებლის ბალანსზე არსებული თანხის შესახებ ინახება ბანკის ცენტრალიზებულ მონაცემთა ბაზაში, რაც ჰაკერებისთვის ცენტრალიზებულ სამიზნეს წარმოადგენს. ერთ ცენტრალიზებული ადგილზე, არალეგალურად, წვდომის მოპოვებით, შეუძლიათ წვდომა მოიპოვონ ყველა მომხმარებლის მონაცემზე. ასევე, ბანკები ტექნიკური უზრუნველყოფისთვის იყენებენ შუამავალ კომპანიებს, რაც ზრდის ხარჯებს მათთვის და ასევე ზრდის შეცდომის დაშვების ალბათობას, ტრანზაქციის დამუშავების პროცესში.
- პროცესის ხანგრძლივობა: საბანკო გადარიცხვის განხორციელებისთვის ზოგჯერ საჭიროა გარკვეული დოკუმენტების წარდგენა გარკვეული დეტალის დადასტურებისთვის. დოკუმენტების მოძიება კი მოიცავს დამატებით შუამავალ რგოლებთან ურთიერთობას, რაც ძალზე ზრდის გადარიცხვის განხორციელებისთვის საჭირო საერთო დროს. მაგალითად, როცა გადარიცხვა ხდება ერთი ქვეყნიდან მეორეში, გადარიცხვის ლეგალურობის დასამტკიცებლად შესაძლოა საჭირო გახდეს დამატებითი დოკუმენტების წარდგენა, რაც გარკვეულ დროს და დამატებით ენერგიას მოითხოვს.
- გადასახადი: საბანკო გადარიცხვები დაკავშირებულია მაღალ საფასურთანაც. საფასური იცვლება იმისდა მიხედვით, თუ რა ტიპის გადარიცხვა ხორციელდება ერთი ბანკიდან მეორეში, ერთი ქვეყნიდან მეორეში, ერთი ვალუტიდან მეორეში, და ასე შემდეგ.

1.2 საბანკო სისტემები და ბლოკჩეინი

ბლოკჩეინი არის ტექნოლოგია, რომლითაც ყველა ზემოთ ჩამოთვლილი პრობლემის გადაჭრაა შესაძლებელი. ეს არის დეცენტრალიზებული სისტემა, რომელიც მომხმარებლებს სთავაზობს უსაფრთხო და გამჭირვალე სერვისს სხვადასხვა ფინანსური ოპერაციების შესასრულებლად.

დეცენტრალიზებული სისტემა უზრუნველყოფს, რომ მომხმარებელთა მონაცემები სამუდამოდ იქნება შენახული. თუ რომელიმე ერთი კომპონენტი გაითიშება, მონაცემები მარტივად აღდგება სხვა კომპონენტების გამოყენებით, რადგან სისტემა დეცენტრალიზებულია და მონაცემები შენახულია მრავალ ადგილას. კრიპტოგრაფიული ალგორითმების გამოყენება კი უზრუნველყოფს მონაცემების უსაფრთხოებასა და დაცულობას.

მოდით უფრო კონკრეტულად განვსაზღვროთ ბლოკჩეინის ტექნიკური თვისებები:

- სრული გამჭირვალობა: ბლოკჩეინის დეცენტრალიზებულობა იმაში მდგომარეობს, რომ მონაცემები განაწილებულია მრავალ სხვადასხვა კომპიუტერში, რომელთაც თავიანთი მფლობელები ყავთ. ნებისმიერ მომხმარებელს შეუძლია წვდომა ქონდეს ბლოკჩეინში არსებულ მონაცემებზე. მათ ასევე შეუძლიათ შეამოწმონ მონაცემების ვალიდურობა და ინფორმაციის სისწორე. ეს კი უზრუნველყოფს იმას, რომ ბლოკჩეინში მონაცემის ჩაწერის შემდეგ, მუდამ უცვლელად შეინახება.
- დეცენტრალიზებული სისტემა და უსაფრთხოება: ბლოკჩეინი არის დეცენტრალიზებული სისტემა, რაც ნიშნავს იმას, რომ არ არსებობს ინფორმაციის ცენტრალური წყარო. მთელი მონაცემები განაწილებულია მრავალ მონაცემთა ბაზაში, რაც უზრუნველყოფს ინფორმაციის სტაბილურობას და მეტ უსაფრთხოებას.
- პროცესის ხანგრძლივობა: ბლოკჩეინის გადარიცხვები არ არის რეგულირებული რომელიმე სახელმწიფოს ან უწყების მიერ. შესაბამისად, გადარიცხვები არ გადის შემოწმების პროცესს რომელიმე რეგულაციის მიმართ, რომელიც სახელმწიფოს მიერ არის დადგენილი. ერთი ბლოკჩეინიდან მეორე ბლოკჩეინში გადარიცხვა არ ხდება (ერთი ბანკიდან მეორეში გადარიცხვის მგავსად), რადგან ბლოკჩეინ სისტემა გლობალურია, მთელ მსოფლიოს მოიცავს და ნებისმიერი ქვეყნიდან მეორე ქვეყანაში გადარიცხვა იმავე დროში სრულდება, რა დროშიც ერთ ბანკში არსებული ერთი ანგარიშიდან მეორეზე გადარიცხვა.
- გადასახადი: ბლოკჩეინ სისტემაში გადარიცხვების განხორციელება ან სრულიად უფასოა, ან მინიმალური ფასის მქონეა.

1.3 ბლოკჩეინში შემავალი ერთეულები

სანამ უშუალოდ ბლოკჩეინის მუშაობის პრინციპებს და მათ მათემატიკურ საფუძვლებს გავეცნობოდეთ, ჯერ განვიხილოთ ბლოკჩეინის მონაწილე ერთეულები. ბლოკჩეინ სისტემაში გვაქვს სხვადასხვა მონაწილე ერთეული, რომლებსაც განსაზღვრული აქვთ სხვადასხვა როლი და დანიშნულება:

- მაინერი: მაინერები არიან კომპონენტები, რომლებიც სისტემას ამარაგებენ გამოთვლითი რესურსით. მათ საქმიანობაში შედის გადარიცხვის ვალიდაცია და ბლოკჩეინისთვის ახალი ბლოკის გენერირება. მათი შრომის სანაცვლოდ, ისინი შესაბამის ანაზღაურებას იღებენ შესაბამისი კრიპტოვალუტის სახით.
- კვანძები: ქსელში არსებული კვანძი ეს არის კომპიუტერი ან მოწყობილობა, რომელიც ინახავს ბლოკჩეინის ინფორმაციის სრულ კოპირებულ ვერსიას. ყოველ კვანძს აქვს სრული ისტორია ბლოკჩეინში არსებული მთლიანი ჩანაწერების: გადარიცხვების, მომხმარებლის მონაცემებისა და ასე შემდეგ.
- მომხმარებლები: მომხმარებლები არიან ადამიანები, რომლებიც იყენებენ ბლოკჩეინ ქსელს, რათა გადარიცხონ ან მიიღონ კრიპტოვალუტა.

1.4 გადარიცხვები ბლოკჩეინში

ახლა განვიხილოთ რა ნაბიჯებს გადის გადარიცხვა ბლოკჩეინ სისტემაში. აქვე აღვნიშნოთ, რომ ეს ნაბიჯები სხვადასხვა აპლიკაციაში შეიძლება რაღაც დეტალებით ერთმანეთისგან განსხვავდებოდეს, თუმცა ზოგადი ნაბიჯები უცვლელია და არის შემდეგი:

- გადარიცხვის წამოწყება: მომხმარებელი, რომელიც არის გადარიცხვის ინიციატორი, ქმნის ახალ გადარიცხვას. იგი ვალდებულია წარმოადგინოს გადარიცხვის დეტალები, როგორებიცაა მიმღების ვინაობა და გადასარიცხი თანხის ოდენობა.
- გადარიცხვის გავრცელება: გადარიცხვა და მისი დეტალება გადაეცემა ქსელში არსებულ ბლოკჩეინის კვანძებს. ყოველი კვანძი იღებს შეტყობინებას ახალი გადარიცხვის შექმნასთან დაკავშირებით, რომელიც ასევე შეიცავს გადარიცხვის ყველა დეტალს. ამ ინფორმაციის მიღებისას, თითოეული კვანძი ინახავს ახალ ინფორმაციას ლოკალურად თავის გადარიცხვების ბაზაში.
- გადარიცხვის ვალიდაცია: ქსელში არსებული მაინერები იღებენ ახალ ტრანზქციას და ამოწმებენ მის ვალიდურობას. ტრანზაქცია ვალიდურია, თუ ის შეიცავს ვალიდურ ციფრულ ხელმოწერას, გადარიცხვის ინიციატორს, ანგარიშზე, აქვს გადარიცხვისთვის საკმარისი თანხა და გადარიცხვა ექვემდებარება შესაბამის ფორმატს. ასევე მაინერები ამოწმებენ ტრანზაქციის უნიკალურობას(ეგრედწოდებული, ორმაგად დახარჯვის პრობლემა).
- ახალი ბლოკის დამატება ბლოკჩეინში: მას შემდეგ რაც, მაინერების მიერ, დადასტურდება ტრანზაქციის ვალიდურობა, ეს ტრანზაქცია ემატება ახალ ბლოკში სხვა ვალიდურ ტრანზაქციებთან ერთად და იწყება ახალი ბლოკის შექმნა. ამ ეტაპზე მაინერები ერთმანეთს ეჯიბრებიან ახალი ბლოკის შექმნაში. ბლოკის შექმნა ასევე მოიცავს გარკვეული მათემატიკური ამოცანის გადაწყვეტას (Proof of work). ის მაინერი, რომელიც პირველი ამოხსნის მათემატიკურ ამოცანას, შექმნის ახალ ბლოკს და გამოაცხადებს მის შესახებ ქსელში. გაწეული შრომისთვის კი ეს მაინერი მიიღებს შესაბამის ანაზღაურებას კრიპტოვალუტის სახით.
- ახალი ბლოკის გავრცელება: როცა მაინერი წარმატებით გადაჭრის კრიპტოგრაფიულ პრობლემას და შექმნის ახალ ბლოკს, ის ამ ბლოკს გააგზავნის ქსელში. სხვა კვანძები ახალი ბლოკის მიღებისას, პირველ რიგში, ამოწმებენ ამ ბლოკის ვალიდურობასა და შემდეგ ამატებენ თავიანთ ლოკალურ ბლოკჩეინში.
- კონსესუსი და ახალი ბლოკის დადასტურება: კვანძების ქსელი კონსესუსს აღწევს ახალი ბლოკის ვალიდურობის შესახებ და როცა ყველა კვანძი შეთანხმდება, რომ ეს ბლოკი ვალიდურია, მხოლოდ ამის შემდეგ ხდება თითოეული მათგანის მიერ ახალი ბლოკის ლოკალურ ბლოკჩეინში ჩამატება. კონსესუსის მექანიზმი უზრუნველყოფს იმას, რომ ქსელში არსებული ყველა კვანძი თანხმდება ბლოკჩეინის მიმდინარე მდგომარეობასა და მასში შესულ მონაცემებზე. ამის შედეგად კი ყველა კვანძს, ყოველთვის, განახლებადი ინფორმაცია აქვს ლოკალურად შენახული.

- ბლოკის ჩამატება ბლოკჩეინში: ახალი ბლოკის ბლოკჩეინში ჩამატება ნიშნავს გრძელი ჯაჭვის ბოლოში კიდევ ერთი რგოლის დამატებას. ახალი ბლოკი, აქამდე ბლოკჩეინში არსებულ ბოლო ელემენტს უკავშირდება კრიპტოგრაფიული ჰეშით. ახალი ბლოკის ჩამატებისას, გამოითვლება აქამდე არსებული ბოლო ელემენტის ჰეში და ინახება ახალ ბლოკში. შედეგად კი მიიღება, რომ ბლოკჩეინში არსებული ბლოკები ერთმანეთზე გადაბმულია კრიპტოგრაფიული ჰეშით.
- გადარიცხვის დასრულება: როცა ახალი ბლოკის დამატება წარმატებით დასრულდება, გადარიცხვა დასრულებულად ცხადდება. ანუ შესაბამისი გადამრიცხავისა და მიმღების ბალანსზე არსებული თანხა განახლდება. ამის შემდეგ, ამ გადარიცხვის შეცვლა ან წაშლა შეუძლებელია, იგი სამუდამოდ არის ჩაწერილი ბლოკჩეინში.

ახლა განვიხილოთ თითოეული ეს ნაბიჯი დეტალურად და ვნახოთ რა პრობლემები წარმოიშვება თითოეული მათგანის დროს და ასევე როგორ ხდება მათი გადაწყვეტა.

თავი 2

2.1 გადარიცხვის წამოწყება ბლოკჩეინში

ბლოკჩეინის სისტემები უმეტეს წილად დაფუძნებულია კრიპტოგრაფიულ ალგორითმებსა და ფუნქციებზე. ბლოკჩეინში შემავალ ყველა მომხმარებელს გააჩნია საკუთარი დახურული და ღია გასაღები. ღია გასაღები შეგვიძლია მივიჩნიოთ როგორც მომხმარებლის უნიკალური იდენთიფიკატორი, მაგალითად როგორც იმეილი, რომელსაც მომხმარებელი ყველას უზიარებს და ეს მონაცემი არის მომხმარებლის იდენთიფიკატორი მთელს ქსელში. ღია გასაღები გამოიყენება მომხმარებლის იდენთიფიკაციისთვისა და გადარიცხვის წამოწყებისას მიმღების ვინაობის მითითებისას. ხოლო დახურული გასაღები არის მხოლოდ მომხმარებლისთვის ცნობილი, როგორც პაროლი.

გადარიცხვის შექმნისას მომხმარებელი უთითებს მიმღების ვინაობასა და გადასარიცხი თანხის ოდენობას. თუმცა, ამას გარდა, გადარიცხვა შეიცავს დამატებით მონაცემებს, რომელიც საჭიროა გადარიცხვის ვალიდაციისთვის. დამატებითი მონაცემებიდან ერთ-ერთი არის მონაცემების ჰეში. მის მისაღებად გამოიყენებენ ჰეშირების ალგორითმს (როგორც წესი ეს ალგორითმია SHA256). ჰეშირების ალგორითმს გადაეცემა ყველა მონაცემი გადარიცხვის შესახებ, რის შედეგადაც თითოეული გადარიცხვისთვის მიიღება 256 ბიტიანი უნიკალური იდენთიფიკატორი, რომელიც ასევე გადარიცხვის ნაწილი ხდება.

ჰეშის მიღების შემდეგ, მომხმარებელი, თავისი დახურული გასაღების გამოყენებით, ციფრულად ხელს აწერს ამ მონაცემებს. ამის შემდეგ მომხმარებელი ქსელში გზავნის გადარიცხვის შესახებ მონაცემებს ღია ტექსტით, ხელმოწერას, რომელიც დაშიფრულია დახურული გასაღებით და ღია გასაღებს. ციფრული ხელმოწერის და ღია გასაღების

გამოყენებით ნებისმიერ მაინერს შეუძლია დაადასტუროს, რომ ეს გადარიცხვა ნამდვილად ეკუთვნის ტრანზაქციაში მითითებულ მომხმარებელს და არ არის რომელიმე სხვა მომხმარებლის მიერ შექმნილი. შედეგად მივიღებთ, რომ ელისისთვის შეუძლებელია შექმნას გადარიცხვა, სადაც გადამრიცხავი იქნება ბობი, რეალურ ცხოვრებაში კი ეს ნიშნავს, რომ მხოლოდ ანგარიშის მფლობელს შეუძლია ამ ანგარიშიდან გადარიცხვის განხორციელება.

2.2 ციფრული ხელმოწერა

ციფრული ხელმოწერა არის კრიპტოგრაფიის ერთ-ერთი ფუნდამენტური ამოცანა, რომლის მიზანია უზრუნველყოს გადაგზავნიდან მიღებამდე მონაცემების უცვლელობა, გადარიცხვაში მონაწილე პირების იდენთიფიკაცია და ავტორიზაცია.

ციფრული ხელმოწერის წყალობით ნებისმიერ ტექსტზე, დოკუმენტზე ან რაიმე ციფრულ ფაილზე შეგვიძლია დავადგინოთ ავტორის ავთენტურობა. თუ მასზე მომხმარებლის ხელმოწერა ფიქსირდება, ე.ი. მომხმარებელს ზუსტად ეს მონაცემები გამოუგზავნია ზუსტად ამ სახით.

ახლა განვიხილოთ უშუალოდ ციფრული ხელმოწერის ალგორითმი და ვნახოთ როგორ მუშაობს იგი.

- პირველი ნაბიჯი არის გასაღების წყვილის გენერირება: ღია და დახურული გასაღებები. ეს გასაღებების წყვილი ირჩევა შემთხვევითობის პრინციპით. დახურული გასაღები ინახება საიდუმლოდ და ცნობილია მხოლოდ მომხმარებლისთვის, ხოლო ღია გასაღები ვრცელდება მთელს ქსელში და ყველამ იცის, რომ ეს ღია გასაღები ეკუთვნის კონკრეტულ მომხმარებელს.
- მეორე ნაბიჯი არის გადასაგზავნი მონაცემების ჰეშის დათვლა. აქ როგორც წესი იყენებენ ზემოთ აღნიშნულ SHA256 ალგორითმს. შედეგად მიიღება 256 ბიტიანი ჰეში, რომელიც ნებისმიერ მონაცემს, ფაილს, ტექსტს, აუდიო ან ვიდეო ფაილს უნიკალურად აიდენთიფიცირებს.

$$\textbf{Generate hash}(\textbf{data}) \Rightarrow hash$$

- მესამე ნაბიჯია, ზემოთ დათვლილი ჰეშის და დახურული გასაღების გამოყენებით, ხელმოწერის შესრულება და შედეგის გამოთვლა. ხელმოწერის ალგორითმი იღებს დახურულ გასაღებს, ჰეშირების შედეგს, და აგენერირებს ხელმოწერას, რომელიც ასევე 256 ბიტიანი მონაცემია.

$$\textbf{Generate signature}(\textbf{hash}, \textbf{private key}) \Rightarrow signature$$

შესაბამისად, დაგენერირებული ხელმოწერა დამოკიდებულია როგორც დახურულ გასაღებზე, ასევე საწყის მონაცემზე, რაზეც ხდება ხელმოწერა.

- ხელმოწერის გამოთვლის შემდეგ ქსელში იგზავნება მონაცემები ხელმოწერასთან ერთად, (data, signature)-ის წყვილი.
- ქსელში გადაგზავნის შემდეგ ნებისმიერ ადამიანს შეუძლია შეამოწმოს და დაადასტუროს გადაგზავნილი მონაცემების ავთენტურობა. (data, signature) წყვილიდან, იგივე ჰეშირების ალგორითმის გამოყენებით, მარტივად გამოითვლება მონაცემების ჰეში, რის შედეგადაც მიიღება hash, რომლის გამოყენებითაც მონაცემის ავტორმა გამოთვალა ხელმოწერა signature. ღია გასაღების გამოყენებით კი გადმოგზავნილი signature-დან შესაძლებელია საწყისი hash-ის გამოთვლა, იმ ჰეშის, რაზეც ხელი მოაწერა გადმომგზავნელმა დახურული გასაღებით. თუ ეს ორი ჰეში ერთმანეთს დაემთხვა, ე.ი. გადმოგზავნილი ინფორმაცია უცვლელია და გადმომგზავნელის ვინაობაც დადასტურებულია. წინააღმდეგ შემთხვევაში, ვერიფიკაცია უარყოფილია.

ბლოკჩეინ ტექნოლოგიებში გამოიყენება სხვადასხვა ხელმოწერის ალგორითმები. ქვემოთ განვიხილავთ ყველაზე ცნობილ და პოპულარულ ალგორითმს.

2.3 RSA ალგორითმი

RSA ალგორითმი ფართოდ გამოიყენება ღია გასაღების კრიპტოგრაფიაში. ის უზრუნველყოფს კომუნიკაციის დაცულობას და ასევე გამოიყენება ციფრული ხელმოწერისას. აქ მოვიყვანთ მარტივ განმარტებას იმის შესახებ, თუ როგორ მუშაობს ეს ალგორითმი:

- გასაღების გენერაცია:
    - შეარჩიოთ ორი დიდი მარტივი რიცხვი p და q
    - გამოვთვალოთ მათი ნამრავლი, რომელსაც გამოვიყენებთ მოდულის დასათვლელად n = p * q
    - გამოვთვალოთ ეილერის ფუნქია ამ რიცხვებზე
      $$\boldsymbol{\varphi(n) = (p-1) * (q-1)}$$
      რიცხვების რაოდენობა, რომლებიც n-ზე ნაკლებია და მასთან თანამარტივია.
    - შევარჩიოთ e ისეთი რომ
      $$\boldsymbol{1 < e < \varphi(n)}$$
    - შევარჩიოთ d ისეთი რომ
      $$\boldsymbol{d \bmod \varphi(n) = 1}$$
      გამოითვლება შემდეგი ფორმულით
      $$\boldsymbol{d = e^{-1} \, mod \, (p-1)(q-1)}$$
    - შედეგად მიიღებთ ღია გასაღებს, რომელიც არის (n, e) რიცხვების წყვილი და დახურულ გასაღებს, რომელიც არის (n, d) რიცხვების წყვილი
- გადასაცემი ტექსტის დაშიფვრა და გაშიფვრა

- თუ m გადასაცემი ტექსტია, ხოლო c მისი დაშიფრული ვერსია, მაშინ m დან c გამოითვლება შემდეგი ფორმულით

$$\mathbf{c = m^e \bmod n}$$

- C-ს მიღების შემდეგ კი მისი გაშიფვრა მოხდება შემდეგი ფორმულით

$$\boldsymbol{m = c^d \bmod n}$$

ამავე ალგორითმით ხდება გადასაცემ დოკუმენტზე ხელმოწერის შესრულება. ამ შემთხვევაში ხელმომწერი იყენებს თავის დახურულ გასაღებს ხელმოწერის შესასრულებლად და მისი გამოყენებით აგენერირებს ციფრულ ხელმოწერას. ხოლო მიმღები იყენებს გამომგზავნის ღია გასაღებს, რომელიც ყველასთვისაა ცნობილი, რათა მოახდინოს დაშიფრული მონაცემების გაშიფვრა და ამგვარად დაადგინოს გამომგზავნის ავთენტურობა.

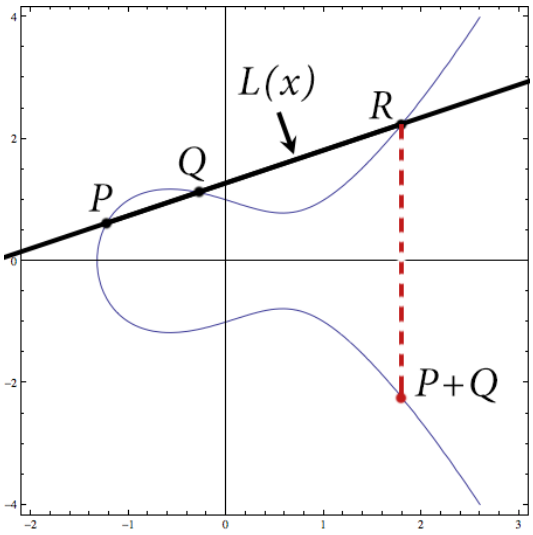
RSA ალგორითმის საიმედოობა დაფუძნებულია მათემატიკური გამოთვლადობის სირთულეზე, რომელიც მოითხოვს დიდი რიცხვების მამრავლებად დაშლას. მცირე რიცხვებზე ეს მარტივი გასაკეთებელია, თუმცა დიდი რიცხვების შემთხვევაში, ეს ამოცანა დიდ გამოთვლით რესურსს მოითხოვს და დიდ პრობლემას წარმოადგენს. აქედან გამომდინარე RSA ალგორითმის უსაფრთხოება, პირდაპირაა დაკავშირებული გასაღებების სიგრძესთან. საიმედოობის გამო დღეს უკვე იყენებენ 1024 ბიტიანი სიგრძის გასაღებებს RSA ალგორითმებში, რადგანაც 256 ბიტიანი გასაღებების გასატეხად საკმარისად დიდი გამოთვლითი რესურსები არსებობს.

2.4 ელიფსური წირების ალგორითმი

ელიფსური წირების ალგორითმი ასევე ფართოდ გამოიყენება ციფრული ხელმოწერისთვის. ამ ალგორითმში გამოიყენება ელიფსური წირი, და მასზე განსაზღვრული შეკრების და გამრავლების ოპერაციები. ამ ალგორითმის უკეთ გასაცნობად, ჯერ ვნახოთ როგორ განისაზღვრება ელიფსურ წირზე შეკრება და გამრავლება.

დავუშვათ წირზე მოცემული გვაქვს ორი P და Q წერტილი. ამ ორი წერტილის შეკრება გამოითვლება შემდეგნაირად:

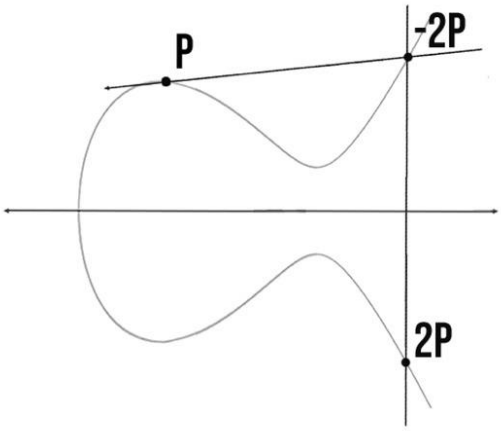
- მოცემულ ორ წერტილზე გავავლოთ წრფე და ვიპოვოთ მესამე წერტილი სადაც ეს წრფე კვეთს ელიფსურ წირს. დავარქვათ ამ წერტილს R.
- ვიპოვოთ R წერტილის მოპირდაპირე წერტილი X ღერძის მიმართ. ეს წერტილი არის P + Q შეკრების შედეგი.



ახლა როცა განსაზღვრება ორი წერტილის შეკრების შესახებ გავარკვიეთ, მოდით განვსაზღვროთ წერტილის საკუთარ თავთან შეკრება, რასაც შემდგომში ელიფსურ წირზე არსებული წერტილის ნატურალურ რიცხვზე გამრავლების განსაზღვრისათვის გამოვიყენებთ.

დავუშვათ მოცემული გვაქვს P წერტილი და გვინდა P + P შეკრების გამოთვლა:

- P წერტილზე გავავლოთ მხები, რომელიც ელიფსურ წირს რომელიღაც -2P წერტილში გადაკვეთს.
- ამის შემდეგ ვიპოვოთ -2P წერტილის სიმეტრიული 2P წერტილი X ღერძის მიმართ. მიღებული წერტილი იქნება P+P=2P წერტილი.



რადგან წერტილის თავის თავზე შეკრება განვსაზღვრეთ, ახლა შეგვიძლია წერტილის რიცხვზე გამრავლება განვსაზღვროთ.

ვთქვათ n ნატურალური რიცხვია, ხოლო P რაღაც წვეროა ელიფსური წირიდან. მაშინ n*P = P + ((n - 1) * P). რადგან 2P უკვე გამოვთვალეთ, ახლა გამოვთვალოთ 3–ზე გამრავლება. დავუშვათ მოცემული გვაქვს რაღაც წერტილი A, რომლისთვისაც 2A უკვე ვიპოვეთ. მაშინ 3A გამოითვლება შემდეგნაირად:

- ავიღოთ A და 2A წერტილები, როგორც ორი სხვადასხვა წერტილი და ვიპოვოთ ამ ორი წერტილის შეკრების შედეგად მიღებული 3A წერტილი, ზემოთ მოყვანილი ორი განსხვავებული წერტილის შეკრების წესის მიხედვით

2.5 ციფრული ხელმოწერა ელიფსური წირების გამოყენებით

ციფრული ხელმოწერის შექმნის და დადასტურების პროცესი რამდენიმე ნაბიჯისგან შედგება. ორივე მათგანის შემთხვევაში გარკვეული პარამეტრები წინასწარ შეთანხმებული და ცნობილია. მაგალითად, ალგორითმის კონკრეტული ვერსიისთვის, ელიფსური წირის პარამეტრები ცალსახაა და ყველასთვის წინასწარ ცნობილია. ელიფსური წირების გამოყენებით, განვიხილოთ ხელმოწერის შექმნისა და დადასტურების პროცესი, მიმღების მიერ.

2.5.1 ციფრული ხელმოწერის შექმნა

ციფრული ხელმოწერის შექმნა შემდეგი ნაბიჯებისგან შედგება:

- საიდუმლო გასაღების შექმნა: საიდუმლო გასაღებს ირჩევს მომხმარებელი და არის ცნობილი მხოლოდ მისი მფლობელისთვის. ეს არის დიდი რიცხვი, რომელიც გამოიყენება ელიფსურ წირებში მამრავლად. აღვნიშნოთ ეს გასაღები როგორც SK.
- გენერაციის წერტილი (V): საწყისი წერტილი ელიფსური წირიდან, რომელზეც ხდება გადამრავლების ოპერაცია. ეს წერტილი წინასწარ ცნობილია და უცვლელია კონკრეტული ელიფსური წირის ალგორითმისთვის.
- ვერიფიკაციის ღია გასაღები: ეს გასაღები მიიღება გენერაციის წერტილისა და საიდუმლო გასაღების გამოყენებით. რადგან V გენერაციის წერტილია ელიფსურ წირზე, ხოლო SK დიდი რიცხვია, ელიფსურ წირებზე განსაზღვრული გამრავლების ოპერაციის გამოყენებით, ღია გასაღები გამოითვლება შემდეგნაირად:

  $$\boldsymbol{VK = V * SK}$$

  VK გამოიყენება ხელმოწერის ვერიფიკაციისთვის. ეს არის ღია გასაღები, რომელზე წვდომაც ყველას შეუძლია ქონდეს.
- შეტყობინების წერტილი: შეტყობინება გადაგზავნამდე კონვერტირდება M რიცხვში SHA256 ჰეშირების ალგორითმის გამოყენებით. ხოლო M*V არის შეტყობინების შესაბამისი წერტილი ელიფსურ წირზე, სადაც V ელიფსური წირის გენერაციის წერტილია.
- შემთხვევითი რიცხვი: ყოველი შეტყობინების გაგზავნამდე შემთხვევითობის პრინციპით ირჩევა a რიცხვი და გამოითვლება a * V შემთხვევითი წერტილი ელიფსურ წირზე რაღაც (x, y) კოორდინატებით.
- N რიცხვი: ეს რიცხვი ასევე არის კონსტანტა, რომელიც წინასწარ შეთანხმებული და ყველასთვის ცნობილია. N არის ძალიან დიდი მარტივი რიცხვი.
- ზემოთ მოცემული პარამეტრების გამოყენებით გამოითვლება შემდეგი მონაცემები:

- $R = x \bmod N$
- $S = \left((M + R * SK) * \left(a^{-1} \bmod N\right)\right) \bmod N$

- ხელმოწერა კი არის (R, S) რიცხვების წყვილი. ეს წყვილი იგზავნება ინტერნეტში და ხელმისაწვდომია ნებისმიერი მომხმარებლისთვის.

2.5.2 ციფრული ხელმოწერის ვალიდაცია

ახლა ვნახოთ, როგორ მოხდება ხელმოწერის შემოწმება და გადმოცემული შეტყობინების ავტორის ავთენტურობის დადგენა:

- ჯერ გამოვთვალოთ გადმოცემული მონაცემების ჰეში. რადგან ჰეშირების ალგორითმი წინასწარ ცნობილია, ამიტომ იგივე ალგორითმის გამოყენებით გამოვთვალოთ M.
- N ცნობილი კონსტანტას გამოყენებით გამოვთვალოთ W შემდეგი ფორმულით $W = S^{-1} \bmod N$
- ზემოთ გამოთვლილი M და W მონაცემებით, გამოვთვალოთ P და Q

$$P = (M * W) \bmod N, Q = (R * W) \bmod N$$

- ახლა გამოვთვალოთ (x, y) წერტილი შემდეგი ფორმულის გამოყენებით

$$Point(x, y) = P * V + Q * VK$$

- ამის შემდეგ შევამოწმოთ x = R ტოლობის ჭეშმარიტება. თუ ტოლობა ჭეშმარიტია, ე.ი. ხელმოწერა ვალიდურია და გადმომგზავნილ შეტყობინებაში არაფერია შეცვლილი. წინააღმდეგ შემთხვევაში, ავთენტურობა და შეტყობინების შინაარსი დარღვეულია.

2.5.3 RSA და ელიფსური წირების ალგორითმების შედარება

დღესდღეობით ბლოკჩეინ ტექნოლოგიებში ძირითადად ელიფსური წირების ალგორითმი გამოიყენება, რადგან ის უფრო დაცული ალგორითმია. როგორც ზემოთ აღვნიშნეთ, დაცულობა პირდაპირ პროპორციულია საიდუმლო გასაღების დადგენისთვის ჩასატარებელი გამოთვლების სირთულესთან. რაც უფრო რთულია გასაღების გატეხვა, მით უფრო დაცულია მონაცემები. RSA-ს შემთხვევაში დაცულობის მისაღწევად იყენებენ 1024 ბიტიან გასაღებებს. ელიფსური წირების შემთხვევაში კი, იმავე დონის უსაფრთხოების მისაღწევად, საკმარისია გაცილებით მცირე ზომის გასაღებების გამოყენება. მცირე ზომის გასაღებები გაცილებით ხელსაყრელია, რადგან გადაგზავნის მომენტში საჭირო გამოთვლები ნაკლებ დროსთან და გამოთვლით რესურსთან არის დაკავშირებული, ხოლო უსაფრთხოების

მხრივ, ისევე უსაფრთხოა როგორც უფრო დიდი ზომის გასაღების მქონდე სხვა ალგორითმები. შედარებისთვის მოვიყვანოთ მაგალითები, მოცემული RSA-ს მოცემული სიგრძის გასაღებისთვის, რა სიგრძის გასაღებია ელიფსური წირების ალგორითმში საჭირო იმისათვის, რომ მივიღოთ იგივე დონის დაცულობა:

RSA	ელიფსური წირების ალგორითმი
3072 ბიტი	256 ბიტი
7680 ბიტი	384 ბიტი

როგორც ვხედავთ, ელიფსური წირების ალგორითმი გაცილებით ეფექტურია.

თავი 3

3.1 გადარიცხვის გავრცელება

მას შემდეგ, რაც გადარიცხვა შეიქმნება, იგი ეგზავნება ბლოკჩეინის ქსელში არსებულ ყველა კვანძს. ყველა კვანძს გააჩნია თავისი მეხსიერების აუზი (ეგრედ წოდებული mempool), რომელიც არის დროებითი მეხსიერების უბანი, სადაც თითოეული კვანძი ინახავს მის ხელთ არსებულ ისეთ ტრანზაქციებს, რომლებიც ჯერ ბლოკჩეინის ნაწილი არ გამხდარა. მეხსიერების ეს უბანი გამოიყენება როგორც მოცდის უბანი, მომხმარებლების გამოგზავნილი ტრანზაქციებისთვის, მაგრამ ჯერ არ არის დამუშავებული მაინერების მიერ. კვანძები განუწყვეტლივ რეჟიმში ანახლებენ თავიანთ მეხსიერებებს ახალი ტრანზაქციების დამატებით, ან ძველი ტრანზაქიების წაშლით. ტრანზაქცია, რომელიც დიდი ხნის გამნავლობაში არის მემორი პულში, ავტომატურად იშლება გარკვეული დროის გასვლის შემდეგ, რათა გამოთავისუფლებული მეხსიერების ადგილზე შესაძლებელი გახდეს ახალი ტრანზაქციის მიღება.

როდესაც კვანძი მიიღებს ახალ ტრანზაქციას, მის შენახვამდე, ის ახორციელებს გარკვეულ ვალიდაციებს ამ ტრანზაქციების მიმართ. ეს ვალიდაცია მოიცავს შემდეგ პუნქტებს:

- სინტაქსი და სტრუქტურა: მოწმდება, რომ ტრანზაქცია ექვემდებარება გარკვეულ სინტაქსსა და სტრუქტურას, რომელიც განსაზღვრულია ბლოკჩეინის პროტოკოლის მიერ. აქ მოწმდება ფორმატი, ასევე, გადარიცხვის შესასრულებლად აუცილებელი, ველების არსებობა.
- ხელმოწერის ვერიფიკაცია: კვანძები ამოწმებენ ციფრული ხელმოწერის ვალიდურობას, რისი საშუალებითაც დგინდება გადარიცხვა ნამდვილად შექმნილია მითითებული მომხმარებლის მიერ თუ არა. ასევე მოწმდება გადარიცხვის

უცვლელობა, რათა თავიდან იქნას არიდებული გადარიცხვის ქსელში მოდიფიკაცია და მონაცემების შეცვლა.

- ორმაგი ხარჯი: თითოეული კვანძი ამოწმებს ხომ არ ხდება ერთი და იმავე თანხის გადარიცხვა ორჯერ ან მეტჯერ ერთიდაიმავე მომხმარებლის მიერ. იგი ამოწმებს, რომ გადარიცხვაში აღწერილი მონაცემები აქამდე არ ყოფილა გამოყენებული რომელიმე სხვა ვალიდურ გადარიცხვაში. ამ შემოწმების გარეშე მომხმარებელს შეეძლება ერთი ბიტკოინი ორჯერ გადაურიცხოს სხვადასხვა პირებს.

თუ ტრანზაქცია რომელიმე შემოწმებას ვერ გაივლის, მაშინ კვანძი მიიჩნევს მას როგორც არავალიდურ ტრანზქციას და არ შეინახავს ლოკალურ მეხსიერებაში. ვალიდაციის წარმატებით გავლის შემთხვევაში კი ის შეინახავს ტრანზაქციას დროებით მეხსიერებაში, სანამ მისი ბლოკჩეინში ჩამატება არ მოხდება.

3.2 ორმაგი ხარჯის პრობლემა

ორმაგი ხარჯის პრობლემა წარმოიშვება ციფრული გადახდების დროს, როცა მომხმარებელი ერთსა და იმავე ვალუტას გზავნის ორ სხვადასხვა ადგილას. ტრანზაქციის დამუშავების დროს ერთ-ერთი პირობა რაც მოწმდება არის ის, რომ გადმომგზავნს აქვს საკმარისი ბალანსი ანგარიშზე. მხოლოდ ამ შემოწმების გავლის შემდეგ ხდება ტრანზაქციის დასრულება და თანხა ჯდება მიმღების ანგარიშზე. მაგრამ თუ გვაქვს ერთდროულად ორი გადარიცხვა, მაშინ ორივე შემოწმება მოხდება თითქმის ერთდროულად. ამ დროს შესაძლოა ორივემ ჩათვალოს, რომ ბალანსზე გადმომგზავნს აქვს საკმარისი თანხა და ტრანზაქცია ვალიდურად გამოაცხადოს.

საბანკო გადარიცხვების დროს ეს პრობლემა იმით გვარდება, რომ თითოეული ტრანზაქცია მუშავდება ცალკ-ცალკე ერთიდაიმავე ცენტრალური სისტემის მიერ. მეორე ტრანზაქციის დამუშავება დაიწყება მას შემდეგ, რაც პირველი დასრულდება. ამიტომ, პირველი ტრანზაქციის დასრულების შემდეგ გადმომგზავნს უკვე შეცვლილი ბალანსი ექნება და არა საწყისი. ბლოკჩეინში კი, რახან ცენტრალური სისტემა არ გვაქვს, ამიტომ ეს პრობლემა აქტუალურია და შესაბამის მოგვარებას საჭიროებს.

3.2.1 ორმაგი ხარჯის პრობლემა ბლოკჩეინში

ფულადი გადახდების დროს, როცა ანგარიშსწორება ხდება ნაღდი ფულით, ორჯერ გადახდის პრობლემა არ დგას, რადგან ფული, რომელსაც ახლა ვიხდით, შეუძლებელია რომ აქამდე გვქონოდა დახარჯული (წინააღმდეგ შემთხვევაში ეს კონკრეტული ფული, სხვა ადამიანის ხელში იქნებოდა). როცა საქმე ციფრულ გადახდებთან გვაქვს, ეს პრობლემა უმნიშვნელოვანეს დაბრკოლებას წარმოადგენს. რადგანაც ბლოკჩეინში ცენტრალიზებული სისტემა არ გვაქვს, ამიტომ ის მეთოდი, რომელსაც ბანკები იყენებენ ამ პრობლემის გადასაჭრელად, აქ გამოუსადეგარია და სხვაგვარად გადაწყვეტას საჭიროებს.

წარმოვიდგინოთ ასეთი სიტუაცია: ელისს თავის ანგარიშზე აქვს 1 მონეტა. იგი ამ ერთ მონეტას გადაურიცხავს ბობს და ასევე ევას. ამ ორი ოპერაციისთვის შეიქმნება ორი დამოუკიდებელი ტრანზაქცია, რომელთა დამუშავებაც ცალკ-ცალკე მოხდება. დავუშვათ პირველი ტრანზაქცია მიიღო A კვანძმა, ხოლო მეორე ტრანზაქცია B კვანძმა. მაშინ ორივე კვანძი ნახავს, რომ ელისის ანგარიშზე ნამდვილად არის ერთი მონეტა და ტრანზაქციას წარმატებულად ჩათვლის. შედეგად კი მივიღებთ, რომ ელისმა თავისი ერთი მონეტა გადაურიცხა ორ თავის მეგობარს და ახლა მათ ჯამში ერთის მაგივრად ორი მონეტა აქვთ. ეს პრობლემა ერთ-ერთი ფუნდამენტური საკითხია ციფრული ვალუტის არსებობისთვის. ამ პრობლემის მოგვარება და გადაჭრა შეძლო სატოში ნაკამოტომ, რომელმაც შექმნა ბიტკოინ ვალუტა და ასევე წარმოადგინა ნაშრომი double spending პრობლემის გადასაჭრელად.

3.2.2 ორმაგი დახარჯვის პრობლემის გადაჭრა ბლოკჩეინში

დავუშვათ, ელისმა თავისი მონეტა გადაურიცხა ორ სხვადასხვა მეგობარს. შესაბამისი ტრანზაქციები გაიგზავნა A და B კვანძებზე, რომლებმაც ჯერ არ იციან სხვა კონკურენტი ტრანზაქციის არსებობის შესახებ. ორივე ტრანზაქცია არის ვალიდური, რადგან ორივე კვანძი ამოწმებს და ადასტურებს, რომ გამომგზავნის ანგარიშზე არის საკმარისი თანხა, ტრანზაქციის შესასრულებლად. ამიტომ, ორივე ტრანზაქცისთვის შეიქმნება ახალი ბლოკი, ორივე ტრანზაქციის ბლოკისთვის მოხდება მათემატიკური ამოცანის გადაჭრა და ეს ბლოკები დაემატებიან როგორც არსებული ბლოკჩეინის ახალი გაგრძელებები. ამ შემთხვევაში მივიღებთ ბლოკჩეინის ორ სხვადასხვა გაგრძელებას. ორივე გაგრძელება თითქმის იდენტური იქნება, მაგრამ განსხვავებული იქნება ბოლოს დამატებული ბლოკით. ამ ეტაპზე შეუძლებელია იმის თქმა, რომელი გაგრძელებაა სწორი, ამიტომ ორივე გაგრძელება შეინახება ბლოკჩეინში. ასევე მოხდება ორივე ვერსიის გავრცელება ყველა კვანძთან, ანუ ყველა ბლოკჩეინში არსებულ კვანძს ახლა ექნება ბლოკჩეინის ორი ვერსია. ყველა კვანძი განაგრძობს ახალი ტრანზაქციების მიღებას და ბლოკების დამატებას ბლოკჩეინის თავის ვერსიაში. ასევე, ანახლებს სხვა კვანძებისგან მიღებულ ინფორმაციას ბლოკჩეინის მეორე ვერსიაზე. ახალი ბლოკების დამატებისას, თუ რომელიმე ვერსია აღმოჩნდა უფრო გრძელი ვიდრე მეორე, მაშინ კვანძები ამ ვერსიას აირჩევენ როგორც ნამდვილს, და მეორე ვერსიას გამოაცხადებენ როგორც არა ვალიდურს. ვერსიის არა ვალიდურად გამოცხადება ნიშნავს, რომ ყველა ის ბლოკი, რომელიც განსხვავებული იყო მეორე ვერსიაში, და ამ ბლოკებში შემავალი ყველა გადარიცხვები გამოცხადდება არავალიდურად. ამის შემდეგ დარჩება ბლოკჩეინის მხოლოდ ერთი ვერსია, რომელსაც გამოიყენებენ ახალი ბლოკების დასამატებლად. შესაბამისად, ელისის მიერ შექმნილი ორი გადარიცხვიდან, საბოლოოდ ერთ-ერთი გამოცხადდება ვალიდურად და ჩაითვლება წარმატებულად, ხოლო მეორე ტრანზაქცია გამოცხადდება არავალიდურად და არ შესრულდება.

ასეთ შემთხვევაში, გადამწყვეტი მნიშვნელობა აქვს იმას, თუ რამდენ კვანძს აქვს კონკრეტული ვერსიის ბლოკჩეინი. თუ ბლოკჩეინში სულ არსებობს 100 კვანძი და 51 მათგანი

ერთ-ერთ ვერსიას მიიჩნევს როგორც ვალიდურს, მაშინ კონსესუსის ალგორითმის გამოყენებით ეს ბლოკჩეინი ჩაითვლება ვალიდურად და დანარჩენი 49 კვანძიც განაახლებს თავიანთ ვერსიებს. ყველა დანარჩენი ვერსია გამოცხადდება არა ვალიდურად. შესაბამისად, გადამწყვეტი მნიშვნელობა აქვს იმას, რომელი ვერსიის ბლოკჩეინია ვალიდური უმრავლესობისთვის.

3.2.3 51 პროცენტიანი შეტევა

51 პროცენტიანი შეტევა, ბლოკჩეინში არავალიდური ტრანზაქციების ვალიდურად გასაღებისა და ამით სარგებლობის მიღების ერთ-ერთი გზა. მიუხედავად იმისა, რომ ამის განხორციელება ძალიან რთული და თითქმის შეუძლებლის ტოლფასია, ეს გზა მაინც მარტივი და ცალსახაა.

თითოეული ბლოკის ვალიდურობა წყდება დემოკრატიის პრინციპით: თუ უმრავლესობა გადაწყვეტს, რომ რაღაც ბლოკი არის ვალიდური, მაშინ ეს ბლოკი ჩაითვლება ვალიდურად. ამის გამო ბლოკჩეინ სისტემა შესაძლოა ჰაკერების სამიზნე გახდეს და კონსესუსის გამოყენებით არა ვალიდური ტრანზაქციები გამოაცხადონ ვალიდურად. ამის განსახორციელებლად კი საჭიროა, რომ ბლოკჩეინ სისტემაში არსებული გამოთვლითი რესურსის 51% აღმოჩნდეს ჰაკერების ხელში. ამის შემდეგ, მათ შეუძლიათ ნებისმიერი ტრანზაქცია ჩათვალონ ვალიდურად, თავის ბალანსზე არსებული ბალანსი შეცვალონ თავიანთი სურვილისამებრ, ნებისმიერ მომხმარებელს აუკრძალონ ტრანზაქციის განხორციელება და ასე შემდეგ.

მიუხედავად იმისა, რომ იდეა საკმაოდ მარტივია, მისი განხორციელება ძალიან რთულია, რადგან ბლოკჩეინ სისტემა შედგება დიდი რაოდენობით მაინერებისგან, კვაძებისა და მომხმარებლებისგან. ამ გამოთვლითი რესურსის 51%-ის კონტროლის ქვეშ მოქცევა, კი თითქმის შეუძლებელს უდრის.

თავი 4

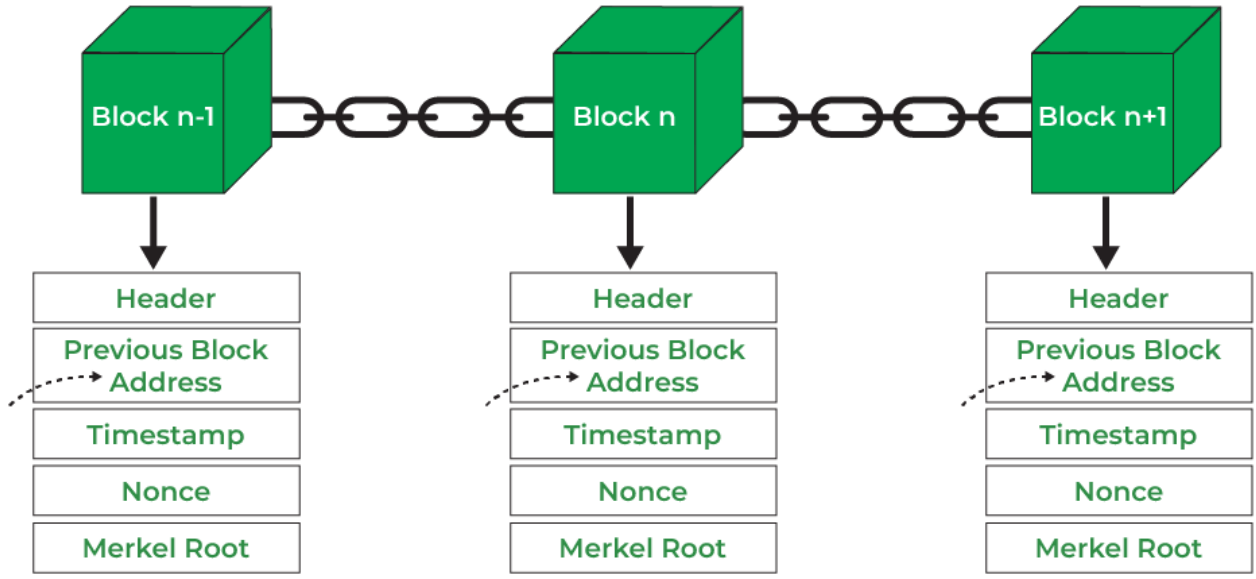
4.1 ახალი ბლოკის დამატება ბლოკჩეინში

მას შემდეგ, რაც კვანძები შეამოწმებენ გადარიცხვის ვალიდურობას, პროცესში მაინერები ერთვებიან. მაინერები ბლოკჩეინ სისტემის ერთ-ერთ ფუნდამენტურ რგოლს წარმოადგენენ და მონაწილეობენ ისეთ პროცესებში, როგორიცაა ტრანზაქციის ვალიდაცია, ახალი ბლოკის შექმნა, მათემატიკური ამოცანის გადაჭრა, ბლოკის ვალიდაცია და კონსესუსის მიღწევა. ამ შრომისათვის კი მაინერები ჯილდოვდებიან გარკვეული ანაზღაურებით.

მას შემდეგ რაც ტრანზაქციების ვალიდურობა დადასტურდება, მაინერები კვანძებში არსებული ტრანზაქციების პულიდან იღებენ ვალიდურ ტრანზაქციებს და ქმნიან ახალ ბლოკს.

თითოეული ბლოკი შედგება რამდენიმე ველისგან:

- წინა ბლოკის ჰეში
- სია იმ ტრანზაქციებისა, რომელიც ამ ბლოკშია განთავსებული
- ნონსი – სპეციალური დანიშნულების რიცხვი, რომელსაც ქვემოთ განვიხილავთ
- ამ ბლოკის ჰეში - ამ ბლოკში არსებული მონაცემების ჰეშის შედეგი.



ამ ველებისგან, მაინერი ტრანზაქციების სიას იღებს ტრანზაქციების პულიდან, წინა ბლოკის ჰეშს კითხულობს წინაბ ლოკიდან, ხოლო ნონსი და ამ ბლოკის ჰეშს მაინერი გამოთვლის თვითონ. ბლოკის ჰეში გამოითვლება SHA256 ალგორითმის გამოყენებით. ხოლო რაც შეეხება ნონსის გამოთვლას, აქ მაინერი ცდილობს რთული მათემატიკური პრობლემის გადაჭრას, რომელიც მდგომარეობს შემდეგში:
ბლოკის ჰეშის გამოსათვლელად, გაითვალისწინება მთლიანი ბლოკის მონაცემები, ნონსის ჩათვლით. ამიტომ, ნონსის ნებისმიერი მცირე ცვლილება, ჰეშის საბოლოო შედეგს რადიკალურად ცვლის. მათემატიკური ამოცანა კი იმაში მდგომარეობს, რომ მაინერმა უნდა შეარჩიოს ნონსის ისეთი მნიშვნელობა, რომ ამ ბლოკის SHA256-ის ჰეშის გამოთვლის შედეგი ნაკლები იყოს წინასწარ მოცემულ სამიზნე მნიშვნელობაზე.

ეს ამოცანა ერთი შეხედვით მარტივია, თუმცა გამოთვლისთვის საჭიროა დიდი გამოთვლითი რესურსის მობილიზება, რადგანაც წინასწარ შეუძლებელია ისეთი ნონსის გამოთვლა, რომლის მიხედვითაც ჰეშირებით მიღებული შედეგი იქნება გარკვეულ მნიშვნელობაზე ნაკლები. ამის გამოთვლისთვის საჭიროა უბრალოდ ნონსის ყველა

მნიშვნელობების გადარჩევა, მათზე ჰეშის გამოთვლა და შედარება სამიზნე მნიშვნელობასთან. ეს გამოთვლითი პროცესი საშუალოდ 10 წუთის განმავლობაში მიმდინარეობს და საბოლოოდ ახალი ბლოკის ფორმირებით მთავრდება. მაინერები ამ დროის განმავლობაში საშუალოდ ცდიან 20.6 კვადრილიონ ნონსის მნიშვნელობას და ითვლიან ჰეშს თითოეული მათგანისთვის.
ნონსის მნიშვნელობის გამოთვლას და ამის შედეგად ახალი ბლოკის ფორმირებას, ერთდროულად მრავალი მაინერი ცდილობს. თუმცა, ჯილდოს მხოლოდ ის მაინერი მიიღებს, რომელიც პირველი იპოვის ასეთ რიცხვით მნიშვნელობას.
მას შემდეგ რაც მაინერი ამოცანას გადაწყვეტს, იგი სხვა მაინერებს გაუგზავნის ახალ ბლოკს და ამოცანას გადაჭრილად გამოაცხადებს. დანარჩენი მაინერები ბლოკის მიღებისას მოახდენენ ამ ბლოკის ვალიდაციას, გამოთვლიან მის ჰეშს, შეადარებენ სამიზნე მნიშვნელობას და ნახავენ რომ ჰეშის შედეგი სამიზნე ჰეშზე ნაკლებია. მხოლოდ ამის შემდეგ შეინახავენ ამ ბლოკს როგორც ბლოკჩეინის ახალ ელემენტს. იმ შემთხვევაში, თუ ერთდროულად რამდენიმე მაინერი ამოხსნის ამოცანას და სხვებს გადაუგზავნის ბლოკებს სხვადასხვა ნონსით, მაშინ მოხდება ბლოკჩეინის გაყოფა და მაინერები მიიღებენ ბლოკჩეინის ორ ვერსიას. რადგან ამ ეტაპზე გაურკვეველია რომელი ვერსია გაგრძელდება, ამიტომ მაინერები ინახავენ ორივე ვერსიას და ელოდებიან შემდგომ განახლებებს. როცა ერთ-ერთი ვერსია გახდება უფრო გრძელი ვიდრე მეორე, მაშინ უფრო გრძელი ვერსია ჩაითვლება ბლოკჩეინის ახალ საბოლოო ვერსიად, ხოლო უფრო მოკლე ვერსია წაიშლება ყველა მაინერის და კვანძის მეხსიერებიდან.

ბლოკჩეინში ახალი ბლოკის დამატების შემდეგ, შესაძლოა ჰაკერმა ცადოს ბლოკში არსებული მონაცემების შეცვლა. შედეგად მივიღებთ, რომ ამ კონკრეტული ბლოკის ჰეში შეიცვლება, რის გამოც ჰეში აღარ იქნება მომდევნო ბლოკში არსებული ჰეშის ტოლი და ამ გზით სისტემა დაადგენს ბლოკჩეინში შეტანილ ცვლილებას.
იმისთვის, რომ სისტემამ ვერ დაადგინოს ბლოკის ჰეშებს შორის სხვაობა, ჰაკერს შეუძლია ყველა მომდევნო ბლოკებისთვის შეცვალოს ჰეშის მნიშვნელობა და მიიღოს ჰეშის მნიშვნელობების მიხედვით ვალიდური ბლოკჩეინი. თუმცა იგი მაინც ვერ შეძლებს თავისი ბლოკჩეინის ნამდვილად გასაღებას, რადგან ბლოკჩეინ სისტემა შექმნილია როგორც ბიზანტიელი გენერლების პრობლემისადმი მდგრადი. ჯერ განვიხილოთ თვითონ პრობლემა, და შემდეგ განვიხილოთ როგორ აგვარებს ამ პრობლემას ბლოკჩეინ სისტემა.

4.2 ბიზანტიელი გენერლების პრობლემა

ბიზანტიელი გენერლის პრობლემა დაკავშირებულია ერთ-ერთი ქალაქის აღების დროს წარმოშობილ პრობლემასთან. იმისათვის, რომ ბიზანტიელ გენერალს ქალაქი აეღო, საჭირო იყო ქალაქისთვის ყოველი მხრიდან ერთდროულად შეტევა, წინააღმდეგ შემთხვევაში შეტევა ჩაიშლებოდა. ბიზანტიელ ლეიტენანტებს შორის იყვნენ მოღალატეები, რომლებიც ყველაფერს ცდილობდნენ იმისათვის, რომ შეტევის წარმატებით

განხორციელებისთვის ხელი შეეშალათ. ლეიტენანტები განლაგებული იყვნენ სხვადასხვა მხარეს, გენერალს მათთან კომუნიკაცია მხოლოდ წერილების საშუალებით შეეძლო. გენერლის წინაშე წარმოიშვა შეტევის კოორდინაციის პრობლემა. თუ გენერალი წერილობითი ფორმით გასცემდა ბრძანებას, რომ შეტევა დაეწყოთ X საათზე, მოღალატე ლეიტენანტები ასევე გაგზავნიდნენ წერილს, რომ შეტევა დაეწყოთ Y საათზე. შედეგად კი ერთგული ლეიტენანტები მიიღებდნენ შეტევისთვის ორ განსხვავებულ ბრძანებას, რაც შეტევის წარმატებით განხორციელებას ხელს შეუშლიდა.

დეცენტრალიზებულ სისტემაში ზუსტად იგივე პრობლემა გვაქვს. ბლოკჩეინის შემთხვევაში, ჰაკერი აწყობს ახალ ბლოკჩეინს და უგზავნის ინტერნეტში არსებულ კვანძებში. ამ შემთხვევაში კვანძები იღებენ ორ სხვადასხვა მონაცემს და შეუძლებელია იმის გადაწყვეტა, რომელი მათგანია ჭეშმარიტი.

ბლოკჩეინ სისტემაში ამ პრობლემის მოგვარება ხდება შემდეგნაირად: მას შემდეგ, რაც კვანძი მიიღებს ახალ ბლოკჩეინზე ინფორმაციას, იგი გაუგზავნის ამავე ინფორმაციას სხვა კვანძებს. თითოეული კვანძი იქცევა ამგვარად და შედეგად მივიღებთ, რომ ყველა ყველას უგზავნის ბლოკჩეინზე ინფორმაციას. ამ პროცესის შემდეგ, თითოეულ კვანძს აქვს ერთი ჩანაწერი განსხვავებულ ბლოკჩეინზე, ხოლო ყველა დანარჩენი მიუთითებს ბლოკჩეინის საწყის მდგომარეობაზე. ამ შემთხვევაში, კვაძი ჩათვლის, რომ ის ბლოკჩეინის ვერსიაა სწორი, რომელიც უფრო მეტი კვანძისგან მიიღო. ამ დაცვის დასაძლევად, ჰაკერს დაჭირდება კვანძების 51%-ზე წვდომა განახორციელოს, რაც როგორც ზემოთ ავღნიშნეთ თითქმის შეუძლებელია.

4.3 კონსესუსი და ტრანზაქციის დასრულება

მას შემდეგ, რაც ახალი ბლოკის გავრცელება მოხდება და ყველა კვანძს ექნება ლოკალურად ახალი ბლოკი, კვანძებს შორის უნდა შედგეს კონსესუსი იმაზე, ეს ბლოკი ჩაემატოს თუ არა ბლოკჩეინში. თუ კვანძების უმეტესობა გადაწყვეტს, რომ ეს ბლოკი ვალიდურია და ბლოკჩეინში უნდა ჩაემატოს, მაშინ ბლოკი დაუკავშირდება ბლოკჩეინში არსებულ ბოლო ბლოკს ჰეშის გამოყენებით და დაემატება ბლოკჩეინს როგორც ბოლო ელემენტი. დამატების შემდეგ, ბლოკის შეცვლა აღარასდროს მოხდება და იგი სამუდამოდ შეინახება ბლოკჩეინში უცვლელი სახით. ბლოკის ჩამატების შემდეგ, მასში შემავალი ყველა ტრანზაქცია ცხადდება დასრულებულად.

4.4 მერკელის ხე

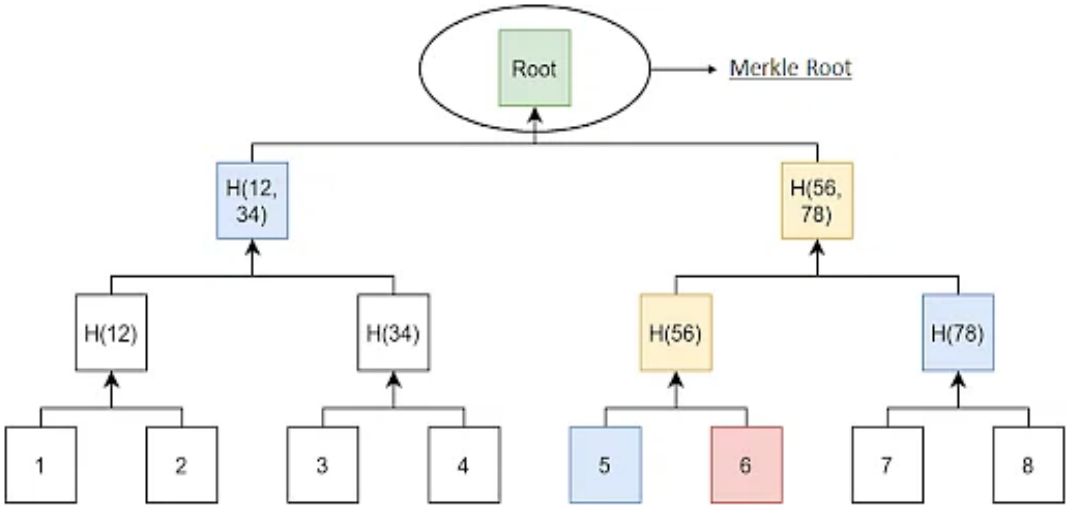
ბლოკჩეინ სისტემები გათვლილია დიდი რაოდენობით მონაცემების დამუშავებასა და შენახვაზე. მაგალითისთვის რომ განვიხილოთ, ბიტკოინის შემთხვევაში ერთი დღე-ღამის განმავლობაში ხორციელდება დაახლოებით 600,000 გადარიცხვა. თითოეული გადარიცხვა, დაწყებული ბიტკოინის პირველივე დღიდან, უცვლელად არის შენახული ბლოკჩეინში. თუ

დავთვლით დღემდე დაგროვებული გადარიცხვების მონაცემების რაოდენობას, ვნახავთ რომ ეს რიცხვი დაახლოებით 1,000,000,000-ის რიგის უნდა იყოს.

ბლოკჩეინის თითოეული ბლოკი დაახლოებით 2,000 ტრანზაქციას შეიცავს, ხშირად კი საჭირო ხდება ერთი ბლოკის შიგნით არსებული რომელიმე კონკრეტული ტრანზაქციის ვალიდურობის შემოწმება. იმისთვის, რომ შევამოწმოთ რომელიმე ტრანზაქცია ვალიდური არის თუ არა, უნდა დავთვალოთ ამ მთლიანი ბლოკის ჰეში, და შევადაროთ ბლოკში არსებულ ჰეშს. თუ ჰეში დაემთხვა, ე.ი. ყველა ტრანზაქცია უცვლელი ყოფილა, წინააღმდეგ შემთხვევაში, რაღაც ცვლილება მომხდარა ბლოკის შიგნით. ეს გამოთვლა მარტივია, თუმცა საკმაოდ გამოთვლილ რესურსს მოითხოვს. რეალურად გვჭირდება მთელი ბლოკი თავიდან წავიკითხოთ და ისე შევამოწმოთ ტრანზაქციის ვალიდურობა. მაშინ კი, როცა ტრანზაქციების რაოდენობა გაიზრდება, ალგორითმი ამ შემოწმებას საკმაო დროსაც მოანდომებს.

იმისათვის, რომ ტრანზაქციის ვალიდაცია, გაცილებით ოპტიმალურად, მოხდეს ამისთვის იყენებენ მერკელის ხეს. მერკელის ხე არის ორობითი ხე, რომლის ცალკეულ ფოთოლში ინახება ცალკეული ტრანზაქციის ჰეშის შედეგი. თუ ტრანზაქციების რაოდენობა 2-ის ხარისხი არაა, მაშინ ბოლო ტრანზაქციას რამდენჯერმე გაიმეორებენ მანამ სანამ ტრანზაქციების საერთო რაოდენობა არ გახდება 2-ის რაიმე ხარისხი.

მას შემდეგ რაც ტრანზაქციების რაოდენობა 2-ის ხარისხია, გამოვთვალოთ თითოეული მათგანის ჰეში. მივიღებთ 2-ს ხარისხის რაოდენობა ჰეშებს. თითოეული ეს ჰეშები დავაწყვილოთ და ორ-ორი ჰეშის დაჰეშვის შედეგად, მივიღოთ ისევ ახალი ჰეშები. ცხადია, ახალი ჰეშების რაოდენობა ორჯერ ნაკლები იქნება წინანდელი ჰეშების რაოდენობის. ასე გავაგრძელოთ მანამ, სანამ არ მივიღებთ ერთ ჰეშს. საბოლოოდ მიღებულ ჰეშს ეწოდება მერკელის ძირი, და ეს არის ის სიდიდე რომელსაც ბლოკებში ინახავენ ბლოკის ჰეშის აღსანიშნავად.



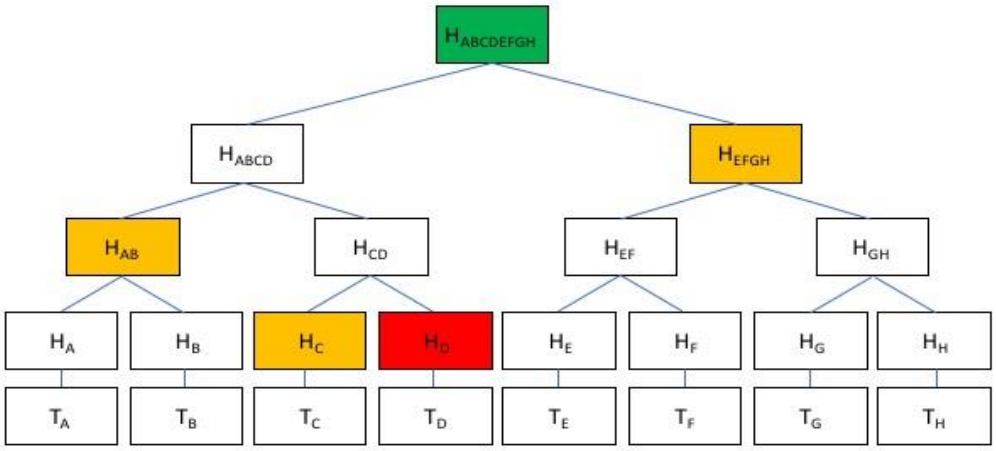
მერკელის ხის ძირის გამოყენებით, შესაძლებელია როგორც მთლიანი ბლოკის ვალიდაცია, ასევე ცალკეული ტრანზაქციის უცვლელობის ოპტიმალურად შემოწმება. განვიხილოთ მაგალითი: ვთქვათ მოცემული გვაქვს ტრანზაქციების სია:

T(A), T(B), … , T(H)

ვთქვათ გვინდა T(D) ტრანზაქციის უცვლელობის შემოწმება. მაშინ შეგვიძლია ავიღოთ შემდეგი ჰეშები

H(AB), H(C), H(EFGH)

მათი საშუალებით და T(D)-ს ჰეშის გამოყენებით შეგვიძლია გამოვთვალოთ რას უნდა უდრიდეს მერკელის ხის ძირი. თუ იგი ტოლია ბლოკში არსებული მერკელის ჰეშის, ე.ი. ტრანზაქცია ვალიდურია, წინააღმდეგ შემთხვევაში მივიღებთ, რომ ტრანზაქციაში რაღაც ცვლილება მოხდა.



ეს ალგორითმი ძალიან სწრაფია და ამასთან ნაკლებ გამოთვლას მოითხოვს. პირობითად, თუ გვაქვს n ცალი ტრანზაქცია, მაშინ რომელიმე ტრანზაქციის ვალიდურობის დასადგენად საკმარისი იქნება log(n) ცალი ჰეშის შემოწმება და მათი მეშვეობით ძირი ჰეშის გამოთვლა.

4.5 Proof of work

Proof of work არის ბლოკჩეინის სისტემებში არსებული კონსესუსის მიღწევის მექანიზმი, რათა მოხდეს ტრანზაქციების ვალიდაცია და დაცვა მოდიფიცირებისგან. ამ მექანიზმის მიხედვით, ბლოკჩეინში მონაწილე მაინერები ერთმანეთს ეჯიბრებიან რთული მათემატიკური პრობლემის გადაწყვეტაში, რათა შექმნან ბლოკჩეინის ახალი ბლოკები. ეს მექანიზმი სპეციალურად არის შექმნილი, რათა თითოეული ცვლილების შეტანა დიდ გამოთვლით რესურსთან იყოს დაკავშირებული. ამ სირთულის წყალობით კი ქსელში არსებულ მავნე მომხმარებლებს ძალიან გაურთულდებათ თავიანთი მიზნების სწრაფად და ნაკლები გამოთვლითი რესურსებით განხორციელება. მისი წყალობით, ბლოკჩეინში

ნებისმიერი ახალი ბლოკის დამატება მოითხოვს უდიდეს გამოთვლით რესურსს, ამის შედეგად კი ქსელი დაცული და უსაფრთხოა.

ახლა განვიხილოთ როგორ მუშაობს proof of work მექნიზმი:

- მაინინგი: მაინერები ერთმანეთს ეჯიბრებიან, რათა პირველმა გამოთვალონ მათემატიკური ამოცანა და ახალი ბლოკი დაამატონ ბლოკჩეინში.
- სირთულე: ამოცანის სირთულე განისაზღვრება სამიზნე მნიშვნელობით ყოველი ბლოკისთვის. ეს სიდიდე მთელი ბლოკჩეინისთვის ერთია და დროდადრო იცვლება იმისდა მიხედვით რამდენი მაინერი დაემატება მთლიან ბლოკჩეინ სისტემას. როცა ბევრი მაინერია დამატებული, მაშინ ამოცანის სირთულეც შესაბამისად იცვლება, რათა ბლოკების დამატება წინასწარ განსაზღვრული სიხშირით მოხდეს. ამ ეტაპზე სიხშირე ყოველ 10 წუთში ერთი ახალი ბლოკის დამატებაა.
- ამონახსნის ვერიფიკაცია: როცა რომელიმე მაინერი ამოცანას წარმატებით ამოხსნის, იგი დასადასტურებლად გადაუგზავნის ამ ბლოკს ყველა სხვა მაინერს. თითოეული მაინერი მარტივი გამოთვლებით დაადასტურებს ბლოკის ვალიდურობას და იმას, რომ ამოცანა სწორადაა ამოხსნილი.
- ჯილდო ახალი ბლოკის შექმნისთვის: ის მაინერი, რომელიც პირველი ამოხსნის ამოცანას, დაჯილდოვდება გარკვეული რაოდენობის კრიპტოვალუტით. ამ ეტაპზე ეს ჯილდო 6.5 ბიტკოინია. თუმცა ჯილდო ყოველ 4 წელიწადში ნახევრდება, რათა ბაზარზე ბიტკოინების რაოდენობა დაბალანსდეს და უსასრულოდ არ გაიზარდოს.

Proof of work-ის გარდა არსებობს კონსესუსის სხვა მექანიზმებიც. მაგალითად დღესდღეობით გამოიყენება ასევე Proof of Stake, რომელიც გარკვეულწილად განსხვავდება Proof of work-სგან. ამ მექანიზმში, მაინერები არ ეჯიბრებიან ერთმანეთს ვინ მოასწრებს პრობლემის გადაწყვეტას, არამედ მაინერი, რომელმაც უნდა ამოხსნას ეს პრობლემა, წინასწარვე ირჩევა. მაინერის შერჩევა არ ხდება შერჩევითად, აქ გადამწყვეტია რამდენი stake (ფსონი) გააჩნია თითოეულ მაინერს. რაც მეტი ფსონი აქვს, ე.ი. მით მეტი გამოთვლითი რესურსია. სწორედ ამიტომ, მაინერი გამოთვლითი რესურსის მიხედვით ირჩევა. შედეგად კი ვიღებთ გაცილებით ენერგო ეფექტურ კონსესუსის მექანიზმს, რადგან რამდენიმე მაინერის მაგივრად, მხოლოდ ერთი მაინერი ცდილობს ამოცანის გადაწყვეტას და გამარჯვებულის გარდა, დანარჩენების ენერგია ფუჭად არ იხარჯება. უსაფრთხოებას რაც შეეხება, აქ პირადი ინტერესები მოქმედებს. ის მაინერები, რომლებსაც დიდი ფსონები და გამოთვლითი რესურსები აქვთ, მოტივირებულები არიან რომ პირნათლად იმოქმედონ, სხვა შემთხვევაში მათ ეს ფსონები ჩამოერთმევათ და შესაბამისად დიდ ზარალს განიცდიან. მეორე მხრივ, ეს მექანიზმი ქსელში წარმოშობს ძლიერ მონაწილეებს, რომლებიც მუდმივად კონკურენტუნარიანები არიან ახალი ბლოკის დამუშავებაში და შესაბამისად მათ კიდევ უფრო მეტი შანსი აქვთ, რომ კიდევ უფრო გაიზრდებიან.

4.5.1 Proof of work-ის მათემატიკური ამოცანა

Proof of work-ის მექანიზმში, მაინერები ცდილობენ მათემატიკური პრობლემის გადაწყვეტას. მათი მიზანია ისეთი ნონს პარამეტრი შეარჩიონ, რომ მთლიანი ბლოკის SHA256 ჰეშირების ფუნქციის შედეგი იყოს წინასწარ ცნობილ target-ზე ნაკლები ან ტოლი. Target პარამეტრის მიხედვით ხდება ამოცანის სირთულის განსაზღვრა. რაც უფრო ნაკლებია სამიზნე პარამეტრი, მით უფრო რთულია ისეთი ნონსის შერჩევა, რომ ჰეშირების შედეგი მასზე ნაკლები ან ტოლი იყოს.

თავი 5

5.1 ბლოკჩეინ ტექნოლოგიის გამოყენებები

როგორც ვნახეთ, იმისათვის, რომ ბლოკჩეინ ტექნოლოგიამ სრულყოფილად იმუშაოს, ამისათვის საჭიროა ბევრი სხვადასხვა პრობლემის გადაწყვეტა, როგორიცაა უსაფრთხოება, მონაცემების დაცვა ცვლილებისგან, გამჭირვალეობა, ავთენტურობის დადგენა და ასე შემდეგ. იმის გამო, რომ ბლოკჩეინში ყველა ეს პრობლემა გადაწყვეტილია, იგი შეიძლება გამოყენებულ იქნას მრავალ სხვადასხვა ცხოვრებისეული საკითხების გადასაჭრელად. ბლოკჩეინის ერთ-ერთი ასეთი გამოყენების მაგალითია არჩევნების ჩატარება ბლოკჩეინ ტექნოლოგიების გამოყენებით. ამისთვის კი შესაძლებელია ბლოკჩეინი გამოყენებული იქნას არჩევნებზე ხმის მიცემისთვის.

5.1.1 ხმის მიცემის სისტემა ბლოკჩეინის გამოყენებით

ბლოკჩეინი შეიძლება გამოყენებული იქნას არჩევნებში რათა უზრუნველყოს გამჭირვალობა, დაცულობა და ნდობა საარჩევნო პროცესის მიმართ. განვიხილოთ როგორ შეუძლია ბლოკჩეინ ტექნოლოგიას არჩევნების ჩატარების უზრუნველყოფა:

- ამომრჩევლის რეგისტრაცია: ბლოკჩეინს შეუძლია უზრუნველყოს უსაფრთხო და დეცენტრალიზებული რეგისტრაციის პროცესი. უფლებამოსილ ამომრჩევლებს შეუძლიათ მათი ვინაობის ვერიფიცირება და ბლოკჩეინში რეგისტრაცია. ეს უზრუნველყოფს, რომ მხოლოდ და მხოლოდ უფლებამოსილი მომხმარემბლები მიიღებენ არცევნებში მონაწილებოას.
- გამჭირვალე და უცვლელი ჩანაწერი: ყოველი ხმა და არჩევანი შეინახება უცვლელ და გამჭირვალე ბლოკჩეინში. ამის საშუალებით კი შესაძლებელი ხდება თითოეული ხმის ვერიფიკაცია და ასევე უზრუნველყოფილი იქნება თითოეული ხმის უცვლელობა არჩევნების მთლიან პროცესში, მათ შორის ხმების დათვლის დროსაც.
- ვინაობის დაცულობა: ბლოკჩეინს შეუძლია ამომრჩევლის ვინაობის უსაფრთხოება. თითოეულ ამომრჩეველს მიენიჭება უნიკალური კრიპტოგრაფიული ან ციფრული იდენთიფიკატორი, რისი წყალობითაც შესაძლებელი გახდება უსაფრთხო

აუთენთიფიკაცია. შედეგად კი მივიღებთ, რომ არავის ეცოდინება, რომელმა ამომრჩეველმა რომელი კანდიდატის სასარგებლოდ მისცა ხმა.

- ხმისმიცემა: ამომრჩევლებს შეუძლიათ ხმა მისცენ ელექტრონულად ბლოკჩეინ პლატფორმაზე.
- ვერიფიკაცია და აუდიტი: იმის გამო, რომ ბლოკჩეინი დეცენტრალიზებული სისტემაა, შედეგები გადამოწმება შესაძლებელია ნებისმიერი მომხმარებლის მიერ, ცენტრალური მართვის სისტემისგან დამოუკიდებლად. ნებისმიერ ადამიანს შეუძლია, შეამოწმოს საარჩევნო შედეგის ვალიდურობა და ხმების გაუყალბებლობა.

როგორც ზემოთ ავღნიშნეთ, ბლოკჩეინი უზრუნველყოფს ამომრჩევლის ანონიმურობას, რაც იმაში მდგომარეობს, რომ მთლიანი ბლოკჩეინის ჩანაწერების ნახვის შემდეგად ვერავინ შეძლებს თითოეული ხმის მიმცემის ვინაობის დადგენას. ეს პრობლემა კი სხვადასხვანაირად წყდება. ერთ-ერთი ასეთი კრიპტოგრაფიული გადაწყვეტა არის zero knowledge proof.

5.1.2 Zero knowledge proof

Zero-knowledge proofs საშუალებას აძლევს მხარეს, რომ დაამტკიცოს რაიმე ინფორმაციის ცოდნა, ინფორმაციის გამჟღავნების გარეშე. მაგალითად: გვაქვს სეიფი, რომელიც ჩაკეტილია 10 ნიშნა კოდით. ელისი ცდილობს ბობს დაუმტკიცოს, რომ ამ სეიფის კოდი იცის ისე, რომ კოდი არ გაამჟღავნოს. ამისათვის საკმარისია ელისმა ბობის შეხედვის გარეშე გახსნას სეიფი და ბობს აჩვენოს გაღებული სეიფი. რადგან სეიფი ღიაა, ე.ი. ელისს ნამდვილად სცოდნია სეიფის კოდი, ოღონდ რა არის კოდი, ამის შესახებ ბობმა კვლავ არაფერი იცის. არჩევნების შემთხვევაში, ეს ტექნიკა გამოიყენება იმის დასამტკიცებლად, რომ მიცემული ხმა ვალიდურია, ოღონდ ყოველი დეტალი ამ მიცემული ხმის შესახებ (ამომრჩეველი, არჩეული კანდიდატი) კვლავ უცნობი დარჩება.

5.1.3 ინტერაქციული და არაინტერაქციული ალგორითმები

Zero knowledge proof ალგორითმები განირჩევა ორ სხვადასხვა კლასად: ინტერაქციული და არა ინტერაქციული.

ინტერაქციული ალგორითმის დროს, იმისთვის რომ ერთმა მხარემ მეორეს დაუმტკიცოს რაიმე ცოდნის ფლობა, საჭიროა გარკვეული ინტერაქციების ჩატარება ამ ორ მხარეს შორის. ეს ინტერაქციები განსაზღვრულია შემდეგნაირად, მათი წარმატებით დასრულების შემთხვევაში, მეორე მხარეს უტყუარად აქვს მტკიცება, რომ პირველმა მხარემ ნამდვილად იცის ესა თუ ის ინფორმაცია, თუმცა ეს ინფორმაცია მისთვის უცნობია. ინტერაქციული ალგორითმებისთვის დამახასიათებელია შემდეგი თვისებები:

- ინტერაქციები: როგორც წესი ამ შემთხვევაში საჭიროა ინტერაქციების რამდენიმე რაუნდი.
- ეფექტურობა: ინტერაქციული ალგორითმები ნაკლებად ეფექტურია, რადგან თითოეული დამატებითი რაუნდი დამატებითი დროსა და გამოთვლით რესურსს მოითხოვს.
- მოქნილობა: ინტერაქციული ალგორითმი მეტად მოქნილია და მარტივად შეიძლება მისი ადაპტირება იმისდა მიხედვით თუ რა ცოდნის დამტკიცებაა საჭირო. ამის მიხედვით შეიძლება შეიცვალოს რაუნდების რაოდენობა, სანდოობის კოეფიციენტის გაზრდა, თითოეული რაუნდის პროცესი და ასე შემდეგ.

არაინტერაქციული ალგორითმის დროს საჭირო არ არის ორ მხარეს შორის რაიმე ინტერაქცია და არ არის საჭირო რამდენიმე რაუნდის ჩატარება. არაინტერაქციული ალგორითმები ხასიათდება შემდეგი თვისებებით:

- არა-ინტერაქციულობა: ამ შემთხვევაში მტკიცება ცოდნის ფლობის შესახებ გენერირდება მხოლოდ ერთხელ, და ის ეგზავნება მეორე მხარეს. მეორე მხარე მხოლოდ ერთხელ ახდენს გადაგზავნილი პატამეტრების შემოწმებას, და მათი ვალიდურობის შემთხვევაში, მტკიცდება რომ გამომგზავნმა მხარემ იცის რაიმე ინფორმაცია.
- ეფექტურობა: არა-ინტერქაციული ალგორითმები როგორც წესი უფრო მეტად ეფექტურია კომუნიკაციის და გამომთვლელი რესურსების საჭიროების თვალსაზრისით. მთავარი მიზეზი კი ის არის, რომ ამ შემთხვევაში ინტერაქციები საჭირო არ არის. მტკიცება, ცოდნის ფლობის შესახებ იქმენა მხოლოდ ერთხელ, ხოლო ნებისმიერი მსურველი, რომელსაც სურს ინფორმაციის ფლობის დადასტურება, იყენებს ამ ერთხელ შექმნილ ინფორმაციას გადასამოწმებლად.
- სანდო წევრი: არა ინტერაქციული ალგორითმები ხშირად ეფუძნება სანდო წევრის არსებობას. სანდო წევრი აგენერირებს რაღაც ტექტს და უგზავნის უცვლელად ორივე მხარეს. მომავალში ეს საერთო ინფორმაცია გამოყენებული იქნება მტკიცებულების გენერირებისთვის და მტკიცებულების შემოწმებისთვისაც. ამიტომ ამ შემთხვევაში საჭიროა ასეთი სანდო წევრის არსებობა.
- სკალირება: არა ინტერაქციული ალგორითმები უფრო მეტად სკალირებადია, რადგან ისინი ნაკლებად მოიხმარენ გამოთვლით რესურსს.

5.1.4 Schnor-ის Zero Knowledge Proof ალგორითმი

შნორის ალგორითმი იყენებს ელიფსურ წირებს და მასზედ განსაზღვრულ არითმეტიკულ ოპერაციებს. განვიხილოთ მაგალითი, როცა ელისი ცდილობს დაუმტკიცოს ბობს რაიმე ინფორმაციის ფლობა. თავიდან ელისი ირჩევს შემთხვევით რიცხვს x, და გამოთვლის ელიფსურ წირზე განსაზღვრულ არითმეტიკულ გამოსახულებას:

$$P = x * G$$

სადაც G არის ელიფსური წირის გენერატორი წერტილი, რომელიც ცნობილია ყველასთვის და არის კონსტანტა. ხოლო P არის ღია გასაღები. ასევე ელისი ირჩევს მეორე შემთხვევით მნიშვნელობას r და გამოთვლის მის შესაბამის რიცხვს ასევე ელიფსურ წირზე

$$R = r * G,$$

ამის შემდეგ ელისი გამოითვლის მის ხელმოწერას შემდეგნაირად:

$$s = (r + x) \bmod N,$$

სადაც N დიდი კონსტანტა რიცხვია, რომელიც ასევე ყველასთვის ცნობილია. ამ გამოთვლების შემდეგ, ბობს ეგზავნება შემდეგი ორი პარამეტრი (R, s). ამ პარამეტრების მიღების შემდეგ, ბობმა უნდა დაადგინოს მათი ვალიდურობა. ამისათვის იგი ამოწმებს შემდეგი გამოსახულების ტოლობას:

$$s * G = P + R$$

თუ ეს გამოსახულება ტოლია, მაშინ ელისის მიერ გადმოგზავნილი პარამეტრები შეესაბამება სიმართლეს და ელისმა ნამდვილად იცის საიდუმლო გასაღები.

თუმცა ეს ალგორითმი არ არის დაცული მავნე ქმედებისგან. კერძოდ, ელისმა რომც არ იცოდეს საიდუმლო გასაღები, მას მაინც შეუძლია ისეთი პარამეტრები გაუგზავნოს ბობს, რომელიც გაივლის შემოწმებას და ბობი ჩათვლის, რომ ელისს აქვს საიდუმლო გასაღები. ამისთვის კი საკმარისია ელისმა გაუგზავნოს შემდეგი პარამეტრები:

$$R = s * G - P$$

ამ შემთხვევაში, ბობი შემოწმების დროს შეადარებს შემდეგ მნიშვნელობებს:

$$s * G = P + R$$

ხოლო რადგან

$$R = s * G - P$$

ე.ი. გვექნება

$$s * G = P + (s * G - P) = s * G$$

მივიღებთ იგივეობას, რაც ყოველთვის სწორი იქნება. ე.ი. არა აქვს მნიშვნელობა რა იქნება P, იგი ყოველთვის გაივლის შემოწმებას და ჩაითვლება, რომ ელისს აქვს საიდუმლო გასაღები.

ამ პრობლემის მოსაგვარებლად, ელისსა და ბობს შორის შემოიღეს რამდენიმე ინტერაქცია, რაც ცალსახად განსაზღვრავს როგორ წარიმართოს ამ ორ მხარეს შორის ურთიერთობა და ამ ინტერაქციის შედეგად, ელისს აღარ ექნება იმის საშუალება, რომ ცოდნა გააყალბოს. ინტერაქცია კი არის შემდეგი:

ელისი აგენერირებს x და r შემთხვევით მნიშვნელობებს და მათი საშუალებით გამოთვლის შემდეგ სიდიდეებს:

$$P = x * G, R = r * G$$

- ელისი ბობს უგზავნის R-ს. ბობი იღებს R-ს, აგენერირებს შემთხვევით c-ს, და უკან უგზავნის c-ს ელისს.
- ელისი იღებს c-ს, გამოთვლის და უგზავნის ბობს s-ს:

$$s = c * x + r$$

- ბობი s-ის მიღებისას ამოწმებს მონაცემების ვალიდურობას შემდეგი ტოლობის შემოწმებით:

$$s * G = c * P + R$$

თუ ეს ტოლობა სწორია, მაშინ ბობი დარწმუნდება, რომ ელისმა ნამდვილად იცის საიდუმლო გასაღები. ამ შემთხვევაში, ელისს არ შეუძლია წინასწარი მანიპულაცია, რადგან R-ის გაგზავნის შემდეგ, s-ის სიდიდეს ბობის მიერ გაგზავნილი c პარამეტრიც განსაზღვრავს.

ამ შემთხვევაში Zero knowledge proof ალგორითმი იმუშავებს უშეცდომოდ, თუმცა სახეზე გვაქვს ორ მხარეს შორის ინტერაქციის საჭიროება. თუ ელისს უნდა 10 სხვადასხვა ადამიანს დაუმტკიცოს საიდუმლო გასაღების ცოდნა, მაშინ მან ეს პროცესი ინდივიდუალურად უნდა აწარმოოს ათივე მათგანთან ცალკ-ცალკე რაც ნაკლებად ეფექტურია და მოითხოვს გამოთვლების გაათმაგებას.

ამ პრობლემის მოსაგვარებლად, მოიფიქრეს გამოსავალი, რის წყალობითაც ელისს შეუძლია გამოთვლა ჩაატაროს მხოლოდ ერთხელ, ხოლო შემდეგ ნებისმიერ მსურველს შეუძლია შეამოწმოს და დაადასტუროს ფაქტი, რომ ელისმა ნამდვილად იცის საიდუმლო გასაღები. ეს ალგორითმი უკვე არაინტერაქციულია და ნაკლებ გამოთვლებს საჭიროებს. რეალურად 10 ადამიანის შემთხვევაში, ელისი გამოთვლებს ატარებს მხოლოდ ერთხელ, ხოლო შემდეგ 10-ვე ადამიანი უბრალოდ შეამოწმებს ტოლობის მართებულობას, ელისის მიერ დაგენერირებული პარამეტრებით. ამით კი საერთო გამოთვლადობა საგრძნობლად მცირდება.

5.1.5 არაინტერაქციული ალგორითმი

რეალურად ინტერაქციული ალგორითმიდან არაინტერაქციული ალგორითმი ძალიან მარტივად მიიღება. შემოვიღოთ ჰეშირების ფუნქცია (რეალურ ცხოვრებაში გამოიყენება SHA256 ალგორითმი). როცა ელისი დააგენერირებს R მნიშვნელობას, იმის მაგივრად, რომ R გაუგზავნოს ბობს და შემდეგ მისგან მიიღოს c, ელისს შეუძლია R-სგან დააგენერიროს მისი ჰეში ჰეშირების ფუნქციის გამოყენებით და გამოიყენოს ჰეშირების შედეგი როგორც c. რახან ჰეშირების ფუნქცია აბრუნებს უნიკალურ შედეგებს, R-ის შეცვლა რომელიმე სხვა მნიშვნელობით, რომელზეც ჰეშირების ფუნქცია იგივე მნიშნველობას დააბრუნებს, შეუძლებელია. ამ შემთხვევაში გენერაციის და ვალიდაციის პროცესი გაიყოფა ორ დამოუკიდებელ ნაწილად და ექნება შემდეგი სახე:

- ელისი იღებს x და r შემთხვევით მნიშვნელობებს და მათი გამოყენებით იღებს P = x * G და R = r * G მნიშვნელობებს.
- ჰეშირების ფუნქციის გამოყენებით ელისი იღებს c-ს

$$c = hash(R, P)$$

- ელისი გამოთვლის s-ს შემდეგი ფორმულით:

$$s = c*x + r$$

- ელისი ასაჯაროებს (R, s) მნიშვნელობებს

ამის შემდეგ ნებისმიერ ადამიანს შეუძლია აიღოს ეს მნიშვნელობები და P ღია გასაღებთან კომბინაციით დაადასტუროს ამ პარამეტრების ვალიდურობა (შესაბამისად დაადასტუროს, რომ ელისს აქვს საიდუმლო გასაღები).

რაც შეეხება დადასტურების პროცესს, იგი იქნება შემდეგი:

- ბობი იღებს (R, s) მნიშნველობებს.
- იგი იყენებს იგივე ჰეშ ფუნქციას და ითვლის c = hash(R, P) მნიშვნელობას
- ამ პარამეტრების გამოყენებით ბობი შეამოწმებს შემდეგი ტოლობის სისწორეს:

$$s*G = c * P + R$$

თუ ეს ტოლობა სწორია, მაშინ ეს ადასტურებს იმას,რომ ელისმა საიდუმლო გასაღები იცის. რადგნა (R, s) უკვე ცნობილია, ნებისმიერ ადამიანს შეუძლია ამ მონაცემების ვალიდაცია ისე, რომ ელისისგან არანაირი დამატებითი მოქმედება არაა საჭირო.

5.1.6 რა საჭიროა r შემთხვევითი მნიშვნელობა?

როგორც ვნახეთ, ალგორითმში ხდება შემთხვევითი მნიშვნელობის დაგენერირება: x და r. ჩნდება კითხვა, რა საჭიროა r პარამეტრის დაგენერირება, როცა x არის შემთხვევითი დახურული გასაღები? განვიხილოთ შემთხვევა, როცა r საერთოდ არ გვაქვს. მაშინ (R, s) ის გამოქვეყნებისას გვაქვს შემდეგი სურათი:

$$c = hash(P), s = c * x$$

ანუ ბობს შეუძლია გამოთვალოს c და რადგან ასევე აქვს s, რომელიც სკალარია, მარტივი არითმეტიკული ოპერაციით შეუძლია ასევე გამოთვალოს x = s / c საიდუმლო გასაღები. R-ის დამატებით კი ბობმა უნდა იპოვოს r და შემდეგ გამოთვალოს x = (s - r) / c, რაც შეუძლებელია, რადგან r უცნობი სიდიდეა და მხოლოდ ელისმა იცის. შესაბამისად, r-ის გარეშე, საიდუმლო გასაღები დაცული აღარ არის.

თავი 6

6.1 ჭკვიანი კონტრაქტები

თანამედროვე ცხოვრებაში ადამიანებს ხშირად უწევთ თავიანთი შეთანხმების კონტრაქტით გაფორმება. ეს შეიძლება იყოს ქონების შეძენა, სესხის დამტკიცება და ასე შემდეგ. ასეთი სახის შეთანხმების დროს ბევრი სირთულე და გაურკვევლობა იჩენს თავს. ზოგჯერ საქმე სასამართლომდეც მიდის და შედეგად ყველაფერი ეს დროში ძალიან იწელება.

ამ გართულებებს თავიდან გვარიდებს ჭკვიანი კონტრაქტები. ფურცელზე დაწერილი კონტრაქტისგან განსხვავებით, ჭკვიანი კონტრაქტი არის ციფრული პროგრამა, რომელიც თავად უზრუნველყოფს შეთანხმებული პირობების შეუცდომლად განხორციელებას. ჭკვიანი კონტრაქტის მაგალითი შეიძლება იყოს ორ მხარეს დადებული შეთანხმება ბიტკოინის ყიდვის შესახებ. მაგალითად: ელისი გადაურიცხავს ბობს $10 000-ს, ხოლო ბობი გადაურიცხავს ელისს 1 ბიტკოინს. ჭკვიანი კონტრაქტი უზრუნველყოფს, რომ ეს ორივე ტრანზაქცია დასრულდეს წარმატებით, ან არ მოხდეს არცერთი გადარიცხვა. შესაბამისად ვერ მოხდება ისე, რომ ელისმა გადაურიცხოს ბობს $10 000 და ბობმა ბიტკოინი თავისთვის დაიტოვოს.

მაგალითის განხილვის შემდეგ, ახლა უფრო ფორმალურად განვსაზღვროთ, რა პრობლემებს გვიგვარებს ჭკვიანი კონტრაქტები:

- ავტომატიზაცია: ჭკვიანი კონტრაქტი ავტომატურად აღასრულებს შეთანხმებულ ნაბიჯებს, რის გამოც საჭირო არ არის ადამიანის ხელოვნური ჩარევა რაიმე პროცესში. ეს კი გამორიცხავს ადამიანური შეცდომის ფაქტორს შეთანხმების აღსრულების პროცესში.

- სანდოობა და უსაფრთხოება: ჭკვიანი კონტრაქტები დაშენებულია ბლოკჩეინ ტექნოლოგიებზე, რის შედეგადაც გვაქვს დეცენტრალიზებული და უცვლელი გარემო. ბლოკჩეინში არსებული კრიპტოგრაფიის პრინციპებისა და კონსესუსის მექანიზმების გამოყენებით ჭკვინაი კონტრაქტები იძლევა გარანტიას, რომ კონტრაქტში მოცემული პირობები სრულდება ისე, როგორც იყო გაწერილი ცენტრალური ავტორიტეტის საჭიროების გარეშე.
- შუამავლების არარსებობა: ტრადიციული კონტრაქტები როგორც წესი საჭიროებენ შუამავლებს, როგორებიცაა იურისტები, ბროკერები, ბანკირები, რათა უზრუნველყონ კონტრაქტის სწორად და სრულად აღსრულება. ჭკვიანი კონტრაქტების გამოყენებით კი ეს საჭიროება აღარ დგას, რადგან პროცესი მთლიანად ავტომატიზებულია ჭკვიანი კონტრაქტების პროგრამის მიერ. ამის შედეგად გვაქვს ნაკლები დანახარჯი კონტრაქტის აღსრულებაზე, გვაქვს გამჭირვალობა და ნაკლები რისკი, რომ კონტრაქტის პირობებს ვინმე შეცვლის მას შემდეგ რაც კონტრაქტი დაიდება.
- პროგრამულობა: ჭკვიანი კონტრაქტი დაწერილია როგორც პროგრამა, რის შედეგადაც გვაქვს მაღალი მობილურობა. კონტრაქტი შეიძლება შეიცავდეს კომპლექსურ ლოგიკას, გამოთვლებს და პირობით გამოსახულებებეს, რის შედეგადაც მიიღება სრულყოფილი შეთანხმება. ეს გარემოება კი საშუალებას გვაძლევს გვქონდეს ახალი ტიპის შეთანხმებები გარდა ტრადიციული ტიპის კონტრაქტებისა.

დასკვნა

ნაშრომში განხილულია ბლოკჩეინ ტექნოლოგიები და ბიტკოინი. წარმოდგენილია მათი მუშაობის პრინციპები და ის მათემატიკური პრობლემები, რომელთა გადაჭრა საჭიროა მათი გამართულად მუშაობისთვის. აქ აღწერილია კრიპტოგრაფიული ალგორითმები, რომლებიც განაპირობებს მომხმარებლების ანონიმურობას და გადარიცხვების დაცვას გაყალბებისგან. ახსნილია რამდენიმე ციფრული ხელმოწერის ალგორითმი და მოყვანილია მათი შედარება. ასევე მოყვანილია ბლოკჩეინ ტექნოლოგიის გამოყენების სხვადასხვა მაგალითები, როგორებიცაა არჩევნების ჩატარება და ჭკვიანი კონტრაქტები. ასევე მოყვანილია არჩევნების ჩატარებისთვის საჭირო Zero knowledge proof ალგორითმი და ახსნილია ერთ–ერთი ასეთი ალგორითმის მუშაობის პრინციპი.

ლიტერატურა

[1] G. Maxwell, A. Poelstra, Y. Seurin and P. Wuille, “Simple Schnorr Multi-signatures with Applications to Bitcoin”

[2] M. Drijvers, K. Edalatnejad, B. Ford, E. Kiltz, J. Loss, G. Neven and I. Stepanovs, “On the Security of Two-round Multi-signatures”, Cryptology ePrint Archive, Report 2018/417.

[3] W. Dai, "b-money," 1998.

[4] H. Massias, X.S. Avila, and J.-J. Quisquater, "Design of a secure timestamping service with minimal trust requirements," In 20th Symposium on Information Theory in the Benelux, May 1999.

[5] S. Haber, W.S. Stornetta, "How to time-stamp a digital document," In Journal of Cryptology, vol 3, no 2, pages 99-111, 1991.

[6] D. Bayer, S. Haber, W.S. Stornetta, "Improving the efficiency and reliability of digital time-stamping," In Sequences II: Methods in Communication, Security and Computer Science, pages 329-334, 1993.

[7] S. Haber, W.S. Stornetta, "Secure names for bit-strings," In Proceedings of the 4th ACM Conference on Computer and Communications Security, pages 28-35, April 1997.

[8] A. Back, "Hashcash - a denial of service counter-measure”

[9] R.C. Merkle, "Protocols for public key cryptosystems," In Proc. 1980 Symposium on Security and Privacy, IEEE Computer Society, pages 122-133, April 1980.

[10] W. Feller, "An introduction to probability theory and its applications," 1957

[11] Wikipedia: “RSA (Cryptosystem)”.

[12] Wikipedia: “Elliptic Curve Digital Signature Algorithm”.

[13] Github: “BOLT #8: Encrypted and Authenticated Transport, Lightning RFC”.

[14] Wikipedia: “Man in the Middle Attack”.

[15] StackOverflow: “How does a Cryptographically Secure Random Number Generator Work?”

[16] Wikipedia: “Cryptographically Secure Pseudorandom Number Generator.

[17] Wikipedia: “Schnorr Signature”.

[18] Blockstream: “Key Aggregation for Schnorr Signatures”.