2021年度　卒業論文

ブロックチェーンを用いた出席管理システムの提案

関東学院大学

理工学部理工学科

情報ネット・メディアコース

218K6078　檜山　祐太

指導教員　塚田恭章

2022年　３月

**概要**

　本研究では、ブロックチェーンを用いて生徒の出席情報の改ざんを防止することを目指す。そこで生徒の出席を管理するシステムを提案、プロトタイプを実装することで生徒の出席情報の改ざんの防止することを確認する。

**目次**

第一章　はじめに.................................................................................................................1

第二章　ブロックチェーン..................................................................................................2

2-1ブロック構造………………………………………………………………………………3

　2-2ブロックチェーンの改ざん.........................................................................................4

第三章　出席管理システム......................................................................................................5

第四章　アーキテクチャ設計...............................................................................................6

　4-1要件定義......................................................................................................................6

　4-2システム概要...............................................................................................................7

第五章　システム詳細設計..............................................................................................8

　5-1開発環境.....................................................................................................................8

　5.2.プロトタイプ概要.......................................................................................................9

第六章　実行結果...................................................................................................................12

第七章　まとめ......................................................................................................................18

謝辞......................................................................................................................................19

参考文献...............................................................................................................................20

**第一章　はじめに**

　受講者の氏名を読み上げて出席を確認する「点呼」は、少人数の授業の場合には確実に本人確認できることから、最も手堅い方法だと言える。しかし、大人数の授業において点呼による出席確認をすると、点呼に要する時間が正規の授業時間を圧迫してしまう。また、紙媒体の出席カードの提出による出席確認の方法は、大人数の授業にも対応できるため多くの授業で採用されているが、出席カードの回収及び集計に時間がかかる[1]。このように、各種学校や塾などでの出席管理に関する作業は多くの労力を必要とし、その負担は小さくはない。しかし、学生の出席状況を把握することは必要不可欠であるため、近年では、ICT技術を用いてより効率よく出欠を確認することができる出席管理システムの導入が数多く進められている[2]。

暗号通貨をはじめとするデジタルで表現される資産の移転や取引などの履歴データを複数の利用者と管理者によって共有する仕組みのことをブロックチェーンや分散台帳と呼ぶ。ブロックチェーンの特徴として、取引情報をはじめとした様々な情報の記録を複数のコンピュータが協調動作することで実現するという点がある。情報を記録するという点では従来のデータベースと共通だが、管理者の異なる複数のコンピュータによって実現することを前提としていることが異なり、同じ台帳の複数を持ち合っている。また、ブロックチェーンの特徴として改ざんが困難という点がある。そこで、改ざんが困難であるブロックチェーンを用いて出席管理システムを作成したいと考えた。

本研究ではブロックチェーンを用いた出席管理システムを提案する。具体的には提案する出席管理システムの要件定義が、改ざん不可能、出席情報の登録、出席情報の公開であることを明らかにし、アーキテクチャを設計し、その一部をプロトタイプとして実装した。本プロトタイプの作成においては、ブロックチェーンを利用した非中央集権アプリケーションであるEthereumを使用した。ソースコードはSolidity言語を利用して記述し、Solidityコンパイラのバージョンは0.4.15を使用した。開発環境Remix上で実行し、正しい動作となることを確認した。また今後の課題として、限られた人しか関数を実行できないようにすることと、古くなり要らなくなった出席情報の削除する関数の作成が必要だと考えた。

　本研究の構成は以下のとおりである。第二章ではブロックチェーンについて説明する。第三章では出席管理システムについて説明する。第四章では本システムの要件定義とともにアーキテクチャ設計を示す。第五章では第四章のアーキテクチャの一部をプロトタイプとして実装する。第六章ではプロトタイプを実際に実行し、動作確認を行う。第七章では本研究を総括し、今後の課題について述べる。

**第二章　ブロックチェーン**

　暗号通貨をはじめとするデジタルで表現される資産の移転や取引などの履歴データを複数の利用者と管理者によって共有する仕組みのことをブロックチェーンや分散台帳と呼ぶ。ブロックチェーンの特徴として、取引情報をはじめとした様々な情報の記録を複数のコンピュータが協調動作することで実現するという点がある。情報を記録するという点では従来のデータベースと共通だが、管理者の異なる複数のコンピュータによって実現することを前提としていることが異なり、同じ台帳の複数を持ち合っている。ブロックチェーンの台帳複製のイメージを図１に示す。

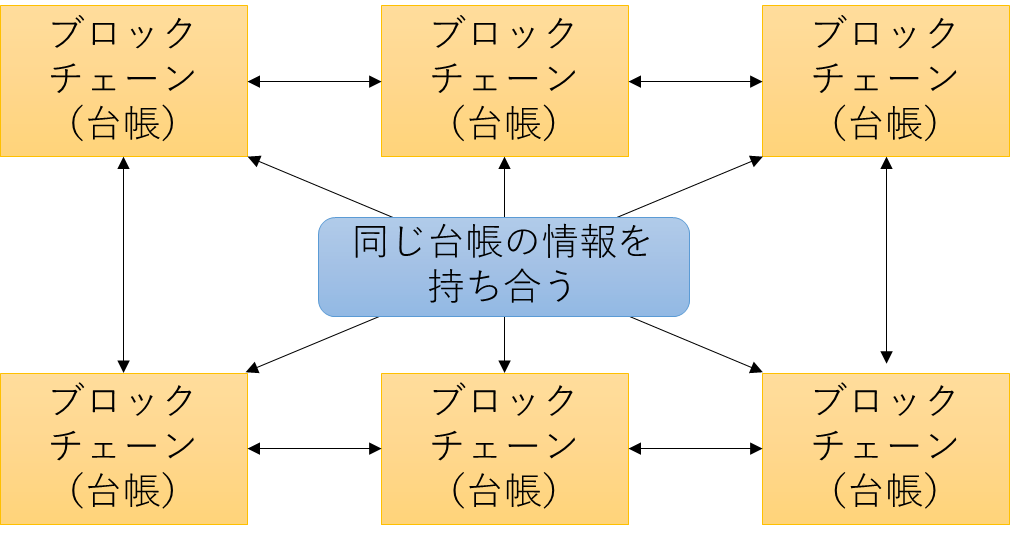


図１　ブロックチェーンの台帳複製のイメージ

ブロックチェーンはいくつかの取引情報をブロックとしてまとめることで、矛盾した取引情報がないようにしている。また、取引情報をブロックとしてまとめる際に、矛盾した取引情報が同じブロックに含まれないように検証が行われる。ブロックは生成される際、１つ前のブロックのハッシュ値を含むことで連鎖する形態をとる。このブロックの連鎖のことをブロックチェーンという。ブロックチェーンのモデル図を図２に示す。

グラフィカル ユーザー インターフェイス が含まれている画像

自動的に生成された説明

図２　ブロックチェーンのモデル図

**2-1　ブロック構造**

　ブロックチェーンの各ブロックは１つ前のブロックのハッシュ値と複数の取引情報、調整用のランダムな値であるnonceによって構成される。

**2-2　ブロックチェーンの改ざん**

　ブロックチェーンの改ざんについて図３に示す。

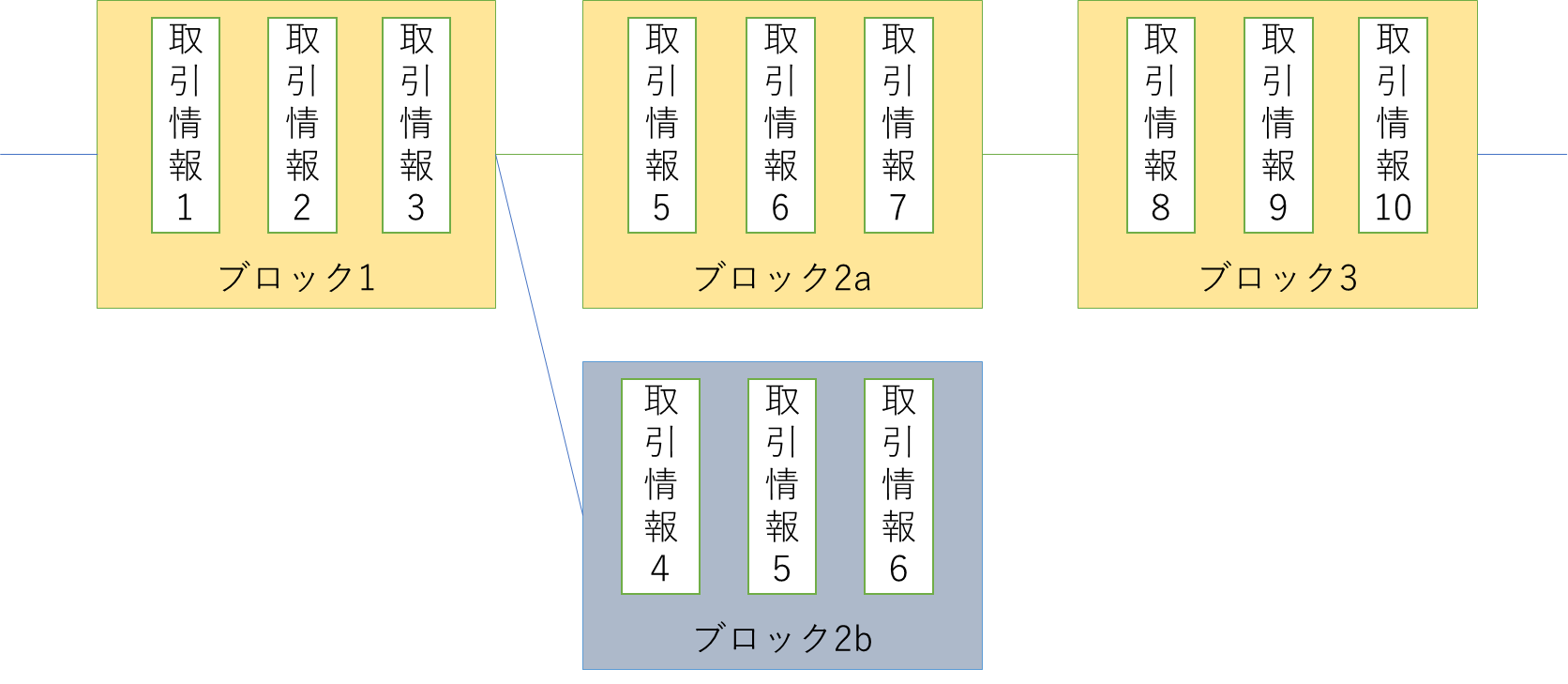


図３　ブロックチェーンの改ざん

　ブロックチェーンの改ざんは容易ではない。図３のブロック２aをブロック２bに攻撃者が書き換えようとした場合、ブロック２aをブロック２bに書き換えるだけでは不十分である。なぜなら、書き換えられたそのブロックのハッシュ値も変化するため、攻撃対象にブロック以降の全てのブロックを新たに作り直す必要があり、さらには、他のノードたちが行うブロック生成を追い越さなければならないからだ。そのため、ブロックチェーンの改ざんは容易ではない。

　このブロックチェーンの改ざんが困難な点を活かし、出席管理システムを作成したいと考えた。

**第三章　出席管理システム**

　出席管理システムとは、学校や塾などにおいて、生徒の出欠状況を管理するシステムのことをいう。出席管理システムには様々な種類があり、中でもICカードを読み取るカードリーダを用いた出欠システムの導入が広く進んでいる。また、各種デバイスを管理端末とする出席管理システムも増加していて、こうしたシステムでは学生が所持しているスマートフォンやパソコンを使用し、クラウドで提供されるものもあるため、専用機器を準備する必要がなく、手軽かつ低コストで出席管理が可能である [2]。

**第四章　アーキテクチャ設計**

　本研究で提案するシステムでは、生徒の出席情報はサーバに送信された際にサーバを介してブロックチェーンに登録される。学校はブロックチェーンに出席情報の確認申請を行うことで、部屋ごとの出席情報と生徒ごとの出席情報の確認を行うことができる。

**4-1　要件定義**

　本研究で提案するアーキテクチャには次の３つの要件を定義することにした。

　⑴　改ざん不可能：ブロックチェーンを用いて、出席者の情報を改ざん不可能にする。

　⑵　出席情報の登録：生徒の出席情報がサーバに送られた際、その時点での時刻とともに生徒の出席情報をブロックチェーンに登録する。

　⑶　出席情報の公開：学校はブロックチェーンに出席情報の確認申請を行うことで、部屋ごとの出席情報と生徒ごとの出席情報の確認を行うことができる。

**4-2　システム概要**

　図3に本研究で提案する出席管理システムのアーキテクチャを図４に示す。

グラフィカル ユーザー インターフェイス, アプリケーション, PowerPoint

自動的に生成された説明

図４ アーキテクチャ図

①サーバに送られてきた生徒の出席情報をブロックチェーンに登録する。

②学校はブロックチェーンに出席情報の確認申請を行うことができる。

③ブロックチェーンシステムに学校から出席情報の確認申請が行われた場合、ブロックチェーンシステムから出席情報を学校に送信する。

　今回の出席管理システムでは、生徒の出席情報をサーバに送信する部分以外の基本部分をプロトタイプとして実装する。

**第五章　システム詳細設計**

**5-1　開発環境**

　本プロトタイプの作成においては、ブロックチェーンを利用した非中央集権アプリケーションであるEthereumを使用した。

また、ソースコードはSolidity言語を利用して記述する。

Solidityコンパイラのバージョンは0.4.15を使用した。

開発環境はRemix[4]を用いて動作を確認する。図５に開発環境Remixの画面を示す。

グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明

図5 開発環境Remix

**5-2　プロトタイプ概要**

　今回の出席管理システムは図４で示したプロトタイプの部分である。

本プロトタイプは部屋名と生徒名は３つずつ登録しておくものし、必要になった際に新たに追加するものとする。また、時刻の登録と出力はunix時間を用いる。図６，７, ８に本プロトタイプのソースコードを示す。

テキスト

自動的に生成された説明

図６　プロトタイプのソースコード⑴

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション

自動的に生成された説明

図７　プロトタイプのソースコード⑵

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション

自動的に生成された説明

図８　プロトタイプのソースコード⑶

　図６の8行目で部屋名を３つ登録し、９行目で生徒名を３つずつ登録している。

図７の60-65行目にあるaddRoom関数が部屋名の追加の関数であり、この関数を実行することで部屋名の追加を行うことができる。

図７の68-73行目にあるaddStudent関数が生徒名の追加の関数であり、この関数を実行することで生徒名の追加を行うことができる。

図７の76-91行目にあるAttendance関数が出席情報を登録する関数であり、部屋名と生徒名を入力することで、その時点での時刻とともにそれらの情報を構造体に格納する。

図８の94-102行目にあるgetAttendanceDB関数が部屋名ごとの出席情報を出力する関数であり、部屋名を入力することで、その部屋名と出席した時刻、出席した生徒名が出力される。

図８の105-113行目にあるgetStudentDB関数が生徒名ごとの出席情報を出力する関数であり、生徒名を入力することで、その生徒名と出席した時刻、出席した部屋名が出力される。

**第六章　実行結果**

　開発環境Remixを利用してプロトタイプで実際に動作確認を行った。

　まず、部屋名の追加の関数であるaddRoom関数で「roomZ」という部屋名を追加した際の実行画面とログ画面を図９と１０に示す。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション

自動的に生成された説明

図９　addRoom関数の実行画面

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション

自動的に生成された説明

図１０　addRoom関数のログ画面

　図９の赤線で囲んだaddRoomの欄に、「”roomZ”」と入力し、実行することができる。

　図１０より、赤線で囲んだdecoded outputの欄に「ture」と出力されていることから、addRoom関数が正しく実行され、「roomZ」の追加に成功していることがわかる。

　次に、生徒名の追加の関数であるaddStudent関数で「studentX」という生徒名を追加した際の実行画面とログ画面を図１１と図１２に示す。

グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明

図１１　addStudent関数の実行画面

グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明

図１２ addStudent関数のログ画面

　図１１の赤線で囲んだaddStudentの欄に、「”studentX”」と入力し、実行することができる。

図１２より、赤線で囲んだdecoded outputの欄に「ture」と出力されていることから、addStudent関数が正しく実行され、「studentX」の追加に成功していることがわかる。

　次に、生徒の出席情報を登録するAttendance関数で、「roomA」に「studentA」が出席したことを登録した際の実行画面とログ画面を図１３と１４に示す。

グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明

図１３　Attendance関数の実行画面

グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明

図１４　Attendance関数のログ画面

　図１３の赤線で囲んだAttendanceの欄に、「”roomA”,”studentA”」と入力し、実行することができる。

　図１４より、赤線で囲んだlogsの欄に、「roomA」と「出席した時刻」、「studentA」と出力されていることから、Attendance関数が正しく実行され、「roomA」に「studentA」が出席したことがその時刻とともにそれらの情報が登録されたことがわかる。

　次に、部屋名ごとの出席情報を出力するgetAttendanceDB関数で、「roomA」の出席情報を出力した際の実行画面とログ画面を図１５と図１６に示す。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション

自動的に生成された説明

図１５　getAttendanceDB関数の実行画面

グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明

図１６　getAttendanceDB関数のログ画面

　図１５の赤線で囲んだgetAttendanceDBの欄に、「”roomA”」と入力し、実行することができる。

　図１６より、赤線で囲んだlogsの欄に、「roomA」と「出席した時刻」、「studentA」と出力されていることから、getAttendanceDB関数が正しく実行され、「roomA」の出席情報が出力されたことがわかる。

次に、部屋名ごとの出席情報を出力するgetStudentDB関数で、「StudentA」の出席情報を出力した際の実行画面とログ画面を図１７と図１８に示す。

グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明

図１７　getStudentDB関数の実行画面

グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明

図１８　getStudentDB関数のログ画面

　図１７の赤線で囲んだgetStudentDBの欄に、「”studentA”」と入力し、実行することができる。

　図１８より、赤線で囲んだlogsの欄に、「studentA」と「出席した時刻」、「roomA」と出力されていることから、getStudentDB関数が正しく実行され、「studentA」の出席情報が出力されたことがわかる。

以上のことから、本プロトタイプは「部屋名の追加」、「生徒名の追加」、「出席情報の登録」、「部屋名ごとの出席情報の出力」、「生徒名ごとの出席情報の出力」が正しく動作することが確認できた。

**第七章　まとめ**

　本研究ではブロックチェーンを用いた出席管理システムを提案し、そのアーキテクチャを示すとともに、プロトタイプを実装した。その結果、出席情報の登録や出席情報の確認が正しく動作することが確認できた。

　今回のシステムでは、管理者が存在せず誰もが全ての関数を実行できるようになっているため、関数を実行できる人に制限を設け、限られた人しか実行できないようにすることが今後の課題である。

また、出席情報を削除する関数を導入していないため、古くなり必要がなくなった出席情報の削除ができないので、出席情報を削除する関数を実装することも今後の課題だと考える。

　さらに、同じ時間に同じ人物が別々の部屋に出席した場合でも登録されてしまうため、一定時間内に同じ人物による複数の出席情報が登録されないようにする関数を実装することも今後の課題だと考える。

**謝辞**

　本研究を進めるにあたり、ご指導、ご助力頂いた塚田恭章教授に心より感謝いたします。また、多くのアドバイスを頂いたネットワークセキュリティ研究室の皆様に心より感謝いたします。ここで深謝の意を表し、謝辞とさせていただきます。

**参考文献**

[1] 檀 裕也：”出席管理システムAMUSEの設計と開発”, 松山大学論集, (2009)

[2] 出席管理システムの概要とおすすめのシステム11選、ミツカル

https://mitsu-karu.com/article/attendance-management/

[3] 佐藤雅史、長谷川佳祐、佐古和恵、並木悠太、梶ヶ谷圭佑、松尾真一郎：ブロックチェーン技術の教科書、C＆R研究所(2018)

[4] Remix – Solidity IDE

https://remix.ethereum.org