

2018 年度 卒業論文

ブロックチェーンを用いた電子投票システムの提案

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

215K6060 長井 龍一

指導教員 塚田恭章

2019 年 3 月

概要

本研究では、ブロックチェーン技術を用いた電子投票システムの提案とその評価およびプロトタイプの実装が行われている。既存の電子投票システムの弱点である、投票内容の改ざんおよび再投票に関する問題を解決した新たなシステムを提案することが目的となっている。今回提案したアーキテクチャを、暗号理論で挙げられている電子投票プロトコルの安全性の定義により評価をしている。また提案したアーキテクチャを基に、プロトタイプを実装し実行した結果、再投票に関する問題を解決したことが示されている。

目次

| | | |
|------|----------------|----|
| 第一章 | はじめに..... | 1 |
| 第二章 | ブロックチェーン..... | 2 |
| 第三章 | 解決すべき弱点..... | 4 |
| 第四章 | アーキテクチャ設計..... | 5 |
| 4.1. | 要件定義..... | 5 |
| 4.2. | システム概要..... | 7 |
| 4.3. | 評価..... | 9 |
| 第五章 | プロトタイプの実装..... | 10 |
| 5.1. | 開発環境..... | 10 |
| 5.2. | プロトタイプ概要..... | 12 |
| 5.3. | 実行結果..... | 14 |
| 第六章 | まとめ..... | 19 |
| | 謝辞..... | 20 |
| | 参考文献..... | 21 |

第一章 はじめに

近年、多くの国で電子投票が注目されており、多数の手法が試験的に導入・実施されている。中でもエストニアは電子投票に関する研究[1]が盛んであり、2005年以降実際の選挙でも電子投票が採用されている。エストニアとは、北ヨーロッパにあるバルト三国の一つである。国土の面積は日本の約9分の1であり、人口は約134万人という小国であるが、政府予算の約1パーセントをICTに投資しており世界におけるICT先進国としての地位を確保しつつある。他にも、アメリカの一部の州や韓国でも実際の選挙で電子投票が採用された例もある。エストニアの電子投票はICチップを内蔵したIDカードを用いて電子署名することで可能となっている。また、電子投票に関する法の整備も行われており、国を挙げて研究が進められている[2]。

しかし、既存の電子投票システムには投票内容などのデータが改ざんされるなど、様々な脆弱性が考えられるため広く普及していない。

そこで、本研究では改ざんが困難な分散ネットワークであるブロックチェーンに注目した。先行研究を調査した結果、ブロックチェーンを用いた電子投票システムに関する研究は行われているものの、解決しきれていない弱点が確認できた。

以上のことを踏まえて、本研究ではブロックチェーンを用いていくつかの弱点を解決した新しい電子投票システムのアーキテクチャを提案する。

本研究では、解決すべき弱点として「改ざん不可能」と「再投票可能」に注目した。これらの弱点を解決するためにブロックチェーン使用し、前回の票を-1して無効にすることで再投票を可能にした電子投票システムのアーキテクチャを考案した。本アーキテクチャは匿名性、公平性、適格性、個別検証可能性、二重投票の防止の性質を満たす。また、本研究で提案したアーキテクチャのうち投票および開票に関する一部をSolidity言語でプロトタイプとして実装し、正しい動作となることを確認した。

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。第三章では本研究で解決すべき弱点について説明する。第四章では第三章の弱点を解決したアーキテクチャを示す。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

ブロックチェーンは、いかなる中央機関を持たずともインターネットを通じた現金取引を可能にする P2P 決済システムとして 2008 年に Satoshi Nakamoto 名義の論文[3]の公表とともに発明されたビットコインを支える技術の 1 つである。ブロックチェーンは、その設計上極めて安全であり、ビザンチン障害耐性の高いシステムである。

ブロックには、トランザクションと 1 つ前のブロックから取得したハッシュ値が格納されている。このブロックが P2P で共有され、連なっていくデータ構造をブロックチェーンと呼ぶ。

ハッシュ値とは、SHA-256（セキュアハッシュアルゴリズム）を使用して生成される値である。SHA-256 は任意の長さの平文を入力とし、それを暗号化する一方向性関数である[4]。

以下の図は、SHA-256 で平文を暗号化する流れを示したものである。

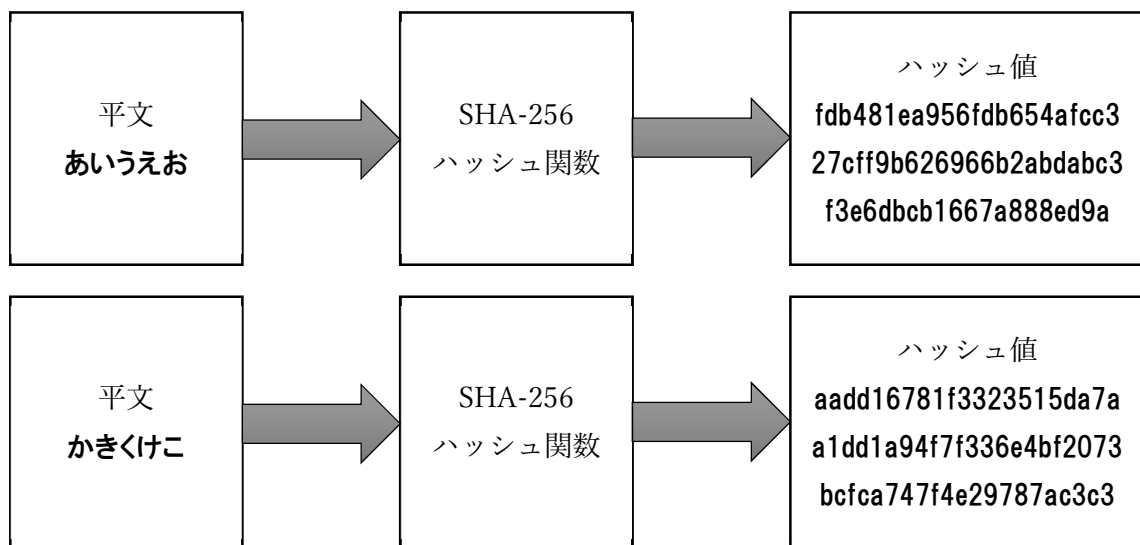


図 1. SHA-256 を用いたハッシュ化の比較

図 1 のように平文を入力すると、平文の長さにかかわらず同一サイズでほぼ内容の異なるハッシュ値が生成される。このハッシュ値をブロックチェーンに取り込むことで、それぞれのブロックが前のブロック情報を引き継ぐたった一つのブロックとして成り立つことになる。

図2にブロックチェーンに新たなブロックの追加を行う構造を示す。図2のようにブロックをチェーンに追加する際は、直前のブロックの内容をハッシュ化し、次のブロックにハッシュ値として取り込んでいる。

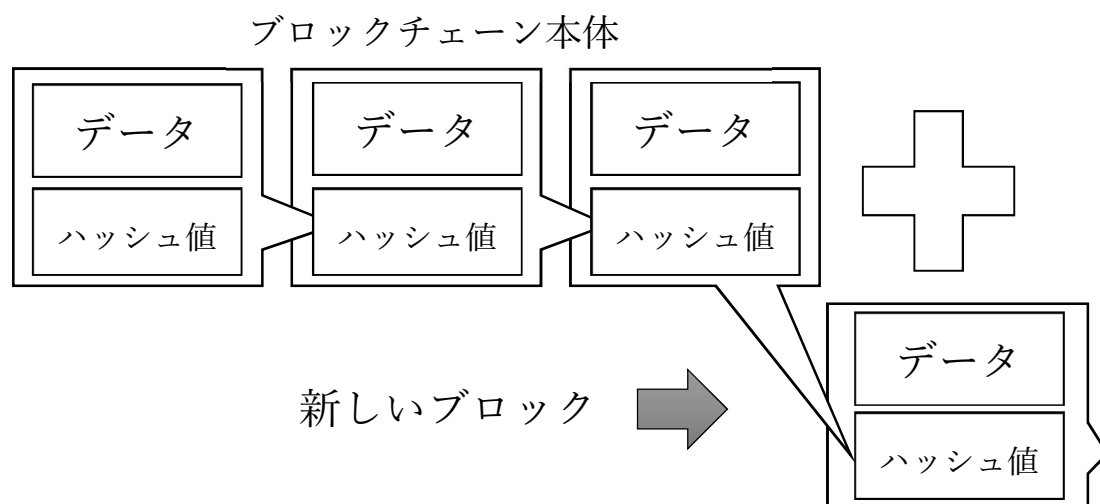


図2. ブロックチェーンの構造

図2のように全てのブロックにハッシュ値が連鎖的に組み込まれている。そのため、あるブロック内の情報を改ざんする場合はそのブロック以降全てのブロックのハッシュ値を書き換える必要がある。しかし、現在のコンピュータの性能ではブロックを追加するために必要な処理が遅く、正規のブロックチェーンネットワークに追いつくことができないため改ざんは困難であると言われている。

また、ブロックチェーンはP2Pネットワークを使用しており、瞬時にほぼすべてのノードがブロックを共有することで成立している。そのため、特定のノードで改ざんが行われてもノードの半数以上が同じブロックを共有しているため発見が容易である[5]。

これらのことから、ブロックチェーンは電子投票システムの安全性向上に寄与する可能性があると考えられる。

第三章 解決すべき弱点

電子投票に関する先行研究およびブロックチェーンの性質から次の弱点が考えられた。

- ① 既存のシステムでは投票内容が改ざんされる恐れがある。
- ② ブロックチェーンは改ざん困難なため、書き換えによる再投票ができない。

①に関しては、ほとんどの電子投票システムが抱える問題[6]であり、ブロックチェーンを用いた最大の理由である。②はブロックチェーンを用いた場合、改ざんが困難であるため、票の書き換えができず再投票が行えないことがわかる。再投票は票の買収問題を解決するために必要な要素であり、エストニアでの電子投票では再投票が可能となっている。本研究では、上記の弱点を解決した新しいアーキテクチャを提案する。

第四章 アーキテクチャ設計

本研究では第三章の弱点を解決したアーキテクチャを提案する。設計に必要な要件を定義し、設計および評価を行う。

4.1. 要件定義

先行研究および解決すべき弱点から、本研究で提案するアーキテクチャに次の4つの要件を定義することとした。

① 改ざん不可能

ブロックチェーンの性質を利用し、ブロック内に票などの情報を格納することで改ざん不可能な投票システムを設計する。

② 再投票可能

2回目以降の投票を行う場合、有権者の投票履歴を参照し前回投票を行った候補者の票を-1する情報を格納する。この処理を示した票を下記に示す。

表 1. 再投票の処理

| 候補者名 | 投票 1 回目 | 投票 2 回目 | 投票 3 回目 | 有効票 |
|------|---------|---------|---------|-----|
| A | +1 | -1 | +1 | ○ |
| B | | +1 | -1 | × |

表 1 では有権者が候補者 AB に対して A→B→A の順に 3 回投票を行った事を表している。最後に票を集計すると、候補者 A は+1-1+1 で+1 票、候補者 B は+1-1 で 0 票となる。よってこの場合有効票となるのは 3 回目に投票した候補者 A に対する票であり、正しい票の集計ができていることがわかる。

③ 票の集計・公開

票がブロックに追加されるタイミングで自動的に票を集計する。また、投票終了後にブロックチェーンを精査することでも票を集計することができる。ただし、票の公平性を保つため投票終了まで開票はできない。各有権者は投票後であればどのタイミングでも自分の票がブロックチェーンに存在することを確認できる。

④ 有権者の投票権失効

何らかの理由で投票権が失効し票が無効になる場合は、有権者データベースにその旨を記載する。投票終了後にオフライン上で有権者データベースを参照し、無効になった票の調整を行う。オフラインで作業することによって第三者による攻撃を避け安全に行うことができる考える。

4.2. システム概要

図3に本研究で提案する電子投票システムのアーキテクチャをブロック図で示す。

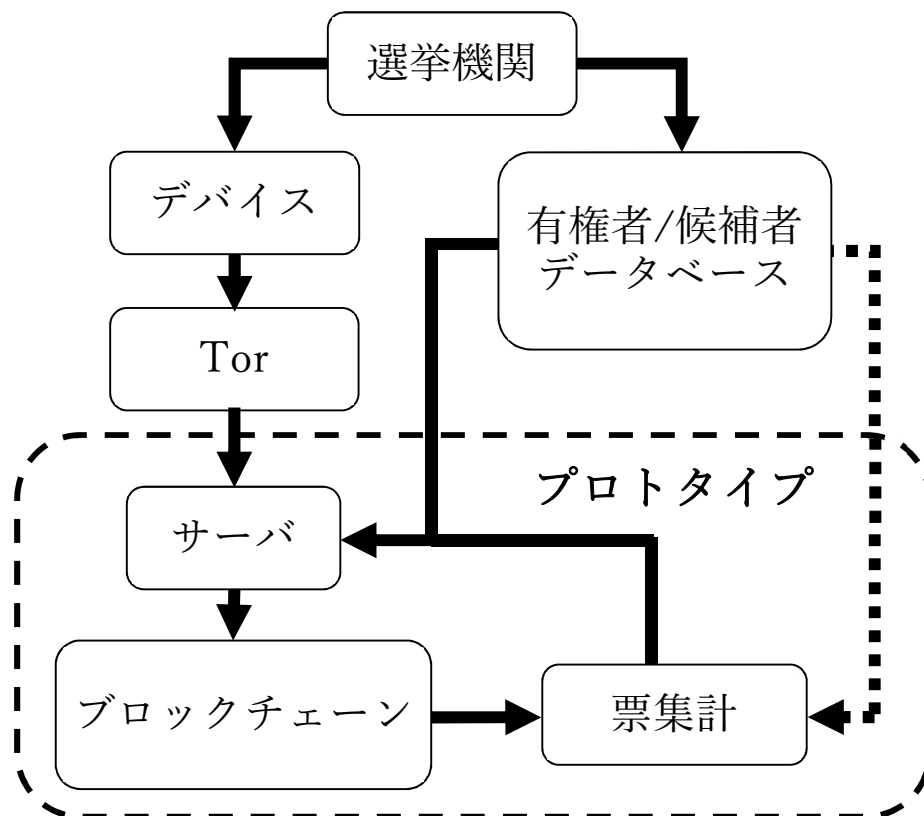


図3. 電子投票システムのブロック図

図3に示した図は、有権者がPCやスマートフォンなどの端末を用いてインターネットに接続しアプリまたはブラウザから投票を行うことを想定している。また、ブロックチェーンは選挙機関が所持するノードのみで構成されるプライベートなネットワークで運用することが望ましい。

はじめに、政府組織などの選挙機関から投票を行うために使用するアドレスを適切な有権者に配布する。ここでアドレスとは約30桁からなる英数字の乱数列であり、アドレス情報のみから個人を特定することは困難であるものとする。ここで、有権者に送ったアドレス情報を対応する有権者データベースに書き込む。サーバは有権者の認証を行う場合に有権者データベースを参照し、適切な有権者かどうかの確認を行う。

有権者が使用するデバイスとサーバ間の通信には匿名通信システムTorを利用し、投票内容と有権者を紐づけできないように秘匿を行う。TorとはThe Onion Routerの略で、自分が使用するIPアドレスを隠蔽し、本人を含めて全く知らない別のIPアドレスを用いて目的先ホストにアクセスする、すなわち別のIPアドレスに置き換えてアクセスするこ

とができる匿名化技術である[7]。しかし、Tor は IP アドレスの置き換えはできるが通信内容までは秘匿できない。つまり、どのアドレスの有権者がだれに投票したかという投票内容は悪意のある第三者に筒抜けになってしまう。ただ、投票内容はブロックチェーンに票が格納された時点で全ての人が確認できてしまうため注意すべき問題ではない。

投票意思を受け付けたサーバは、データベースを参照し適切な有権者であることを確認した後にその有権者に合った候補者一覧を出力する。データベースを参照するイメージを図 4 に示す。

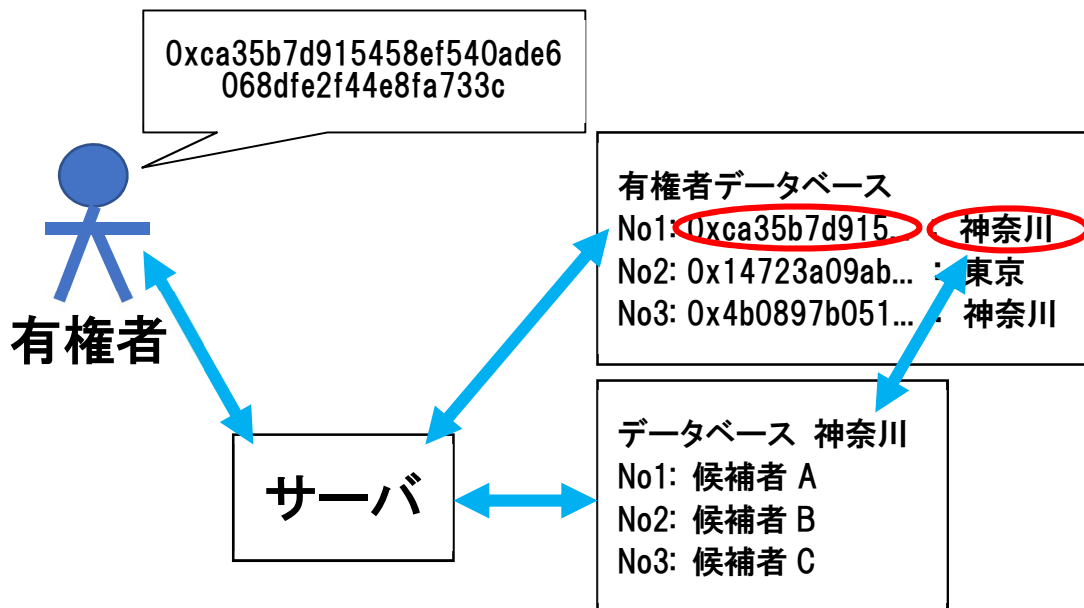


図 4. データベースを参照するイメージ

投票は有権者の「アドレス」と投票する「候補者名」をまとめたブロックをブロックチェーンに追加することで完了する。2 回目以降の再投票の場合は、前回の投票を無効にする -1 の情報も追加する。

ブロックチェーンに票が追加されたタイミングで同時に票の集計も行われる。また、投票終了後にもブロックチェーンを精査することで票を集計することができる。票の集計機能はブロックチェーンに付属するものであり、投票期間中はブロックチェーン以外からのアクセスは不可能となる。

有権者は各自のアドレスを使用して自分の票がブロックチェーンに格納されているかの確認をすることができる。

投票期間終了後にオフライン上で失効票の処理を行う。失効票とは、法律に則りなんらかの不正が起きた場合に失効する票のことである。

全ての処理が終了後、速やかに票を公開する。いかなる場合でも投票期間中に票数を公開することがあってはならない。

4.3. 評価

暗号理論において挙げられている電子投票プロトコルの安全性の定義[8]から、本研究で提案するアーキテクチャの評価を行う。

(1) 匿名性...自分の投票内容を他人に知られない。

匿名化技術 Tor を用いて通信を行うため、IP アドレスと投票内容の紐づけはできず、第三者に自分の投票内容を知られることはないため匿名性を満たす。

(2) 公平性...投票の結果は開票までわからない、投票の途中結果が露呈しない。

開票は投票終了後にしか行えないという制約を設けるため、投票期間中に途中結果が露呈することはないため、公平性を満たす。

(3) 適格性...有権者しか投票できない。

アドレスを用いてデータベースを参照し認証を行うため、アドレスを所持している適切な有権者のみが投票を行える。このため、適格性を満たす。

(4) 個別検証可能性...投票のデータが公開掲示板に記録され、有権者は自分の投票データが公開掲示板にあることを確認できる。

ブロックチェーンに記録された投票内容是有権者のアドレスを用いて確認することができる。これにより個別検証可能性を満たすと言える。

(5) 2重投票の防止

再投票を実現したことにより、何度投票を行っても1人1票しか適応されず、これを満たすと言える。

以上の性質から本アーキテクチャは匿名性、公平性、適格性、個別検証可能性、二重投票の防止を満たすと言える。

ただし、次の性質は本研究で満たすことを確認できていない。

(6) 頑健性...投票プロトコルの実行を妨害されない。

(7) 総合検証可能性...公開掲示板のデータから、開票と集計が正しく行われていることを誰もが確認できる。

上記の頑健性および総合検証可能性については今後の検討課題である。

第五章 プロトタイプの作成

第四章で提案したアーキテクチャの一部を Solidity 言語でプロトタイプを作成し，開発環境 Remix にて動作を確認する．

5.1. 開発環境

本研究ではイーサリアム（Ethereum）と呼ばれる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実行プラットフォームを使用する．イーサリアムは Ethereum Project によって 2013 年から開発が行われている．インターネット上にイーサリアムのネットワークが稼働しており，だれでも自由に参加できるため，パーミッションレス型のブロックチェーンに分類される[9]．

プロトタイプの作成には Solidity という，JavaScript に似た文法を持つ高級言語を使用する．またコンパイラバージョンは 0.4.15 を使用し，ブラウザ上の開発環境 Remix[10]にて動作を確認する．図 5 に開発環境 Remix の画面を示す．

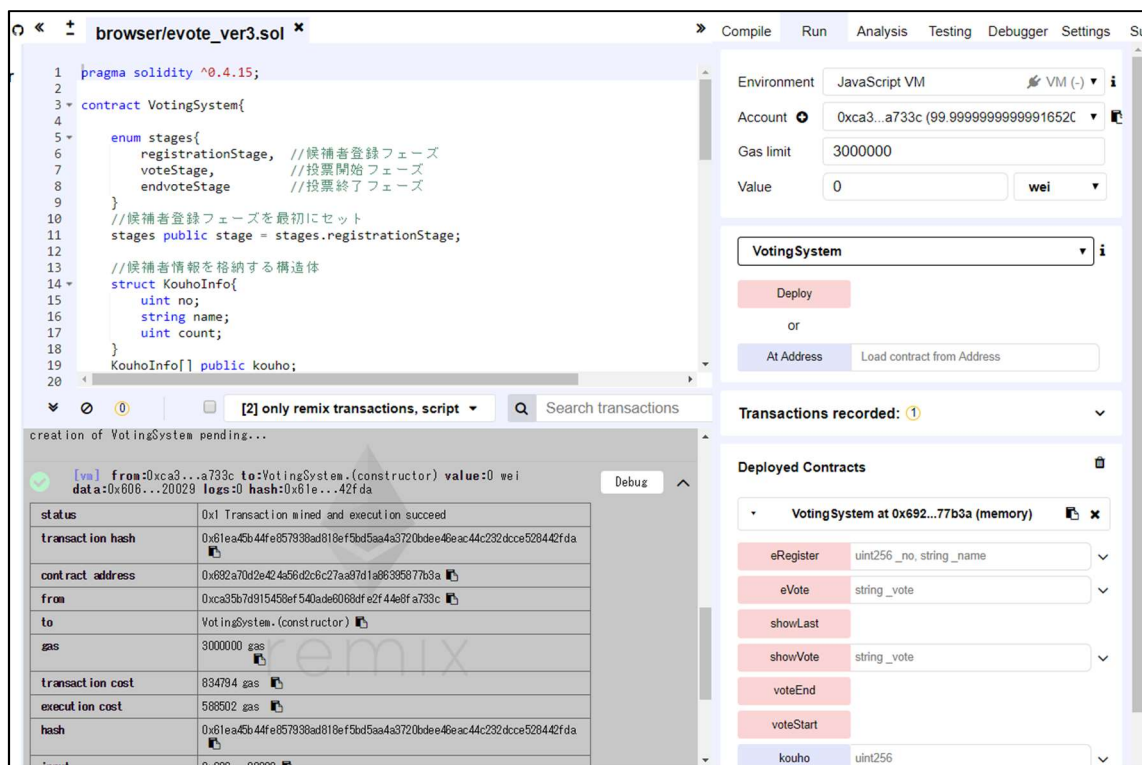


図 5. 開発環境 Remix

図5の左上にはソースコードが記載されており、左下には関数を実行した結果のログが灰色で表示されている。正常に関数が動作した場合は図6のように緑のチェック印が表示されるが、関数が動作しなかった場合は図7のような赤いバツ印が表示される。


| | |
|---|---|
|  [vm] from:0xca3...a733c to:VotingSystem.constructor value:0 wei data:0x606...20029 logs:0 hash:0x61e...42fda | |
| status | 0x1 Transaction mined and execution succeed |

図6. 関数が正常に動作した場合


| | |
|---|--|
|  [vm] from:0xca3...a733c to:VotingSystem.voteEnd() 0x692...77b3a value:0 wei data:0xddb...e8f09 logs:0 hash:0x9b5...1f8e8 | |
| status | 0x0 Transaction mined but execution failed |

図7. 関数の動作が失敗した場合

また、図5の右画面ではアカウントの切り替えや、関数の実行、引数の入力などが行える。

5.2. プロトタイプ概要

本研究で作成するプロトタイプは図3にある点線で囲った「プロトタイプ」の部分である。

本プロトタイプは候補者の登録、投票、開票の3つのフェーズからなるシステムである。図8, 9に本プロトタイプのソースコードを示す。また、本プロトタイプのソースコードはGitHub上に公開した[11]。

```
1  pragma solidity ^0.4.15;
2
3  contract VotingSystem{
4
5      enum stages{
6          registrationStage, //候補者登録フェーズ
7          voteStage,         //投票開始フェーズ
8          endvoteStage       //投票終了フェーズ
9      }
10     //候補者登録フェーズを最初にセット
11     stages public stage = stages.registrationStage;
12
13     //候補者情報を格納する構造体
14     struct KouhoInfo{
15         uint no;
16         string name;
17         uint count;
18     }
19     KouhoInfo[] public kouho;
20
21     //投票者の情報を格納する構造体
22     struct VoterInfo{
23         string lastVote;
24         uint vote_count;
25     }
26     mapping(string => KouhoInfo) kouhoData;
27     mapping(address => VoterInfo) voterData;
28
29     //フェーズ判定を行う
30     modifier atStage(stages _stage){
31         require(stage == _stage);
32         _;
33     }
34 }
```

図8. プロトタイプのソースコード(1)

```

35 //候補者を登録する関数
36 function eRegister(uint _no, string _name) public atStage(stages.registrationStage) {
37     kouho.push(KouhoInfo(_no, _name, 0));
38 }
39 //投票を行う関数
40 function eVote(string _vote)public atStage(stages.voteStage) {
41     if(voterData[msg.sender].vote_count >= 1){
42         kouhoData[voterData[msg.sender].lastVote].count--;
43     }
44     kouhoData[_vote].count++;
45     voterData[msg.sender].lastVote = _vote;
46     voterData[msg.sender].vote_count++;
47 }
48 //投票結果を表示する関数
49 function showVote(string _vote)public atStage(stages.endvoteStage) returns(uint){
50     return kouhoData[_vote].count;
51 }
52 //最後に投票した候補者を表示する関数
53 function showLast()public returns(string){
54     return voterData[msg.sender].lastVote;
55 }
56 //投票を開始する関数
57 function voteStart()public atStage(stages.registrationStage){
58     stage = stages.voteStage;
59 }
60 //投票を終了する関数
61 function voteEnd()public atStage(stages.voteStage) {
62     stage = stages.endvoteStage;
63 }
64
65 }

```

図9. プロトタイプのソースコード(2)

図8の5-9行目より、本プロトタイプでは3つのフェーズを定義した。

候補者登録フェーズ(registrationStage)では、図9の36-38行目にあるeRegister関数が実行可能であり、番号と候補者名を入力し構造体に格納する。すべての候補者が登録完了したら図9の57-59行目にあるvoteStart関数を実行し、投票開始フェーズ(voteStage)に移行する。

投票開始フェーズでは、図9の40-47行目にあるeVote関数が実行可能となっている。この関数では、はじめに再投票の判定を行っており、前回の票を無効にする処理を行う。その後、票を追加し投票回数の加算も行う。投票は何度も繰り返し行うことが可能で、アカウントを変えることで複数人の投票が可能になる。図9の61-63行目にあるvoteEnd関数を実行することで投票終了フェーズ(endvoteStage)に移行できる。

投票終了フェーズでは、図9にある49-51行目のshowVote関数が実行可能であり、候補者名を入力すると票の集計結果を出力することができる。

また、図9の53-55行目にあるshowLast関数は、有権者が最後に投票した票、つまり有効票として候補者名が出力される。この関数は投票後であればどのフェーズでも実行可能である。

5.3. 実行結果

開発環境 Remix を利用してプロトタイプで実際に投票を行った。投票を行うための条件を表 2, 3 に示す。

表 2. 投票の条件(1)

| 投票の条件 | |
|-------|---------------------|
| 有権者 | 5 名 (A,B,C,D,E) |
| 候補者 | 2 名 (kouho1,kouho2) |

表 3. 投票の条件(2)

| 有権者 | 投票回数 | | |
|-----|--------|--------|--------|
| | 1 | 2 | 3 |
| A | kouho1 | — | — |
| B | kouho1 | kouho2 | — |
| C | kouho2 | kouho2 | — |
| D | kouho1 | kouho2 | kouho1 |
| E | kouho1 | kouho1 | kouho2 |

まず、表 2 の通り有権者を 5 名、候補者を 2 名用意し、eRegister 関数で「kouho1」「kouho2」という名前で登録した。図 10 に候補者を登録した際のログ画面を示す。

| | |
|---|---|
| <pre>{ "uint256 _no": "0", "string _name": "kouho1" }</pre> | <pre>{ "uint256 _no": "1", "string _name": "kouho2" }</pre> |
|---|---|

図 10. 候補者登録のログ

図 10 の通り「kouho1」と「kouho2」の名前が登録されていることがわかる。

次に、表3の通りにA~Eの5つのアカウントで最大3回の投票を行う。例として図11、12に有権者Bが投票を行った2回のログを示す。

| | |
|--|---|
| [vm] from:0x147...c160c to:VotingSystem.eVote(string) 0x692...77b3a value:0 wei data:0x554...00000 logs:0 hash:0x525...19f18 | |
| status | 0x1 Transaction mined and execution succeed |
| transaction hash | 0x5256d16a4704301889f5bdb078f72e688753600273888ae5c8ab7b3504819f18 |
| from | 0x14723a09acf6d2a60dcdf7aa4aff308fddc160c |
| to | VotingSystem.eVote(string) 0x692a70d2e424a56d2c6c27aa97d1a86395877b3a |
| gas | 3000000 gas |
| transaction cost | 69676 gas |
| execution cost | 47508 gas |
| hash | 0x5256d16a4704301889f5bdb078f72e688753600273888ae5c8ab7b3504819f18 |
| input | 0x554...00000 |
| decoded input | <div> <div>"string_vote": "kouho1"</div> </div> |
| decoded output | { } |
| logs | [] |
| value | 0 wei |

図 11. 有権者 B の投票 1 回目のログ

| | |
|--|---|
| [vm] from:0x147...c160c to:VotingSystem.eVote(string) 0x692...77b3a value:0 wei data:0x554...00000 logs:0 hash:0xae6...f1c17 | |
| status | 0x1 Transaction mined and execution succeed |
| transaction hash | 0xae6bd45f1f4b2164db6f4b7c36d65947928691be4aa9d81d435dd22f987f1c17 |
| from | 0x14723a09acf6d2a60dcdf7aa4aff308fddc160c |
| to | VotingSystem.eVote(string) 0x692a70d2e424a56d2c6c27aa97d1a86395877b3a |
| gas | 3000000 gas |
| transaction cost | 65708 gas |
| execution cost | 43540 gas |
| hash | 0xae6bd45f1f4b2164db6f4b7c36d65947928691be4aa9d81d435dd22f987f1c17 |
| input | 0x554...00000 |
| decoded input | <div> <div>"string_vote": "kouho2"</div> </div> |
| decoded output | { } |
| logs | [] |
| value | 0 wei |

図 12. 有権者 B の投票 2 回目のログ

図 11、12 より、赤線で囲った decoded input の欄から、1 回目の投票は「kouho1」に行い、2 回目の投票を「kouho2」に行ったことがわかる。

次に showLast 関数を実行し、有権者 B の投票のうち有効票となっている票を確認する。showLast 関数の実行結果のログを図 13 に示す。

| | |
|--|---|
| [vm] from:0x147...c160c to:VotingSystem.showLast() 0x692...77b3a value:0 wei data:0xd79...9cbf1 logs:0 hash:0x80c...59ed7 | |
| status | 0x1 Transaction mined and execution succeed |
| transaction hash | 0x80cffffdac3471a881adb56700d541b605f38cc989bf44de41862b109b9759ed7 |
| from | 0x14723a09acff6d2a60dcdf7aa4aff308fddc160c |
| to | VotingSystem.showLast() 0x692a70d2e424a56d2c6c27aa97d1a86395877b3a |
| gas | 3000000 gas |
| transaction cost | 22937 gas |
| execution cost | 1665 gas |
| hash | 0x80cffffdac3471a881adb56700d541b605f38cc989bf44de41862b109b9759ed7 |
| input | 0xd79...9cbf1 |
| decoded input | {} |
| decoded output | { "0": "string: kouho2" } |
| logs | [] |
| value | 0 wei |

図 13. showLast 関数のログ

図 13 より、有権者 B の投票は「kouho2」に対して有効になっていることがわかる。このことから、投票を行う eVote 関数は正常に動作し、再投票も行えていることが確認できた。

最後に、開票の確認を行う。表 3 にある投票の条件より、本プロトタイプでは最後に投票した票が有効であるため、それぞれの有権者の有効票は表 4 に示した通りである。

表 4. 各有権者の有効票

| | 投票回数 | | | |
|-----|--------|--------|--------|--------|
| 有権者 | 1 | 2 | 3 | 有効票 |
| A | kouho1 | — | — | kouho1 |
| B | kouho1 | kouho2 | — | kouho2 |
| C | kouho2 | kouho2 | — | kouho2 |
| D | kouho1 | kouho2 | kouho1 | kouho1 |
| E | kouho1 | kouho1 | kouho2 | kouho2 |

このことから、「kouho1」は合計 2 票、「kouho2」は合計 3 票入ることがわかる。これを、showVote 関数を用いて確認する。showVote 関数の実行結果のログを図 14、15 に示す。図 14 は「kouho1」の票数、図 15 は「kouho2」の票数を出力している。















| | |
|--|---|
|  <pre>[vn] from:0x147...c160c to:VotingSystem.showVote(string) 0x692...77b3a value:0 wei data:0xe25...00000 logs:0 hash:0x549...4ad3a</pre> | |
| status | 0x1 Transaction mined and execution succeed |
| transaction hash | 0x54917145395f2230e2bd1a0f402442d9b63295a754e4c060d5a3ba03ad04ad3a  |
| from | 0x14723a09acff6d2a60dcdf7aa4aff308fddc160c  |
| to | VotingSystem.showVote(string) 0x692a70d2e424a56d2c6c27aa97d1a86395877b3a  |
| gas | 3000000 gas  |
| transaction cost | 23539 gas  |
| execution cost | 1371 gas  |
| hash | 0x54917145395f2230e2bd1a0f402442d9b63295a754e4c060d5a3ba03ad04ad3a  |
| input | 0xe25...00000  |
| decoded input | <pre>{ "string_vote": "kouho1" }</pre>  |
| decoded output | <pre>{ "0": "uint256: 2" }</pre>  |
| logs | []   |
| value | 0 wei  |

図 14. showVote 関数のログ(1)















| | |
|--|---|
|  [vm] from:0x147...c160c to:VotingSystem.showVote(string) 0x692...77b3a value:0 wei data:0xe25...00000 logs:0 hash:0xd4e...49f85 | |
| status | 0x1 Transaction mined and execution succeed |
| transaction hash | 0xd4e9ab8f840e3d01d4bda926c79575f2b286b0459292d6e16e23194267749f85  |
| from | 0x14723a09acf6d2a60dcdf7aa4aff308fddc160c  |
| to | VotingSystem.showVote(string) 0x692a70d2e424a56d2c6c27aa97d1a86395877b3a  |
| gas | 3000000 gas  |
| transaction cost | 23539 gas  |
| execution cost | 1371 gas  |
| hash | 0xd4e9ab8f840e3d01d4bda926c79575f2b286b0459292d6e16e23194267749f85  |
| input | 0xe25...00000  |
| decoded input | <div>  <pre>{ "string _vote": "kouho2" }</pre> </div> |
| decoded output | <div>  <pre>{ "0": "uint256: 3" }</pre> </div> |
| logs | []   |
| value | 0 wei  |

図 15. showVote 関数のログ(2)

図 14 から「kouho1」は票数が 2，図 15 から「kouho2」は票数が 3 ということが分かった。

以上のことから，本プロトタイプは投票および再投票に加え，票の集計が正しく動作することが確認できた。

しかし，現在の仕様では管理者が存在せず全ての関数が自由に操作できる．つまり，投票の最中に別の有権者が投票フェーズを終了させることも可能となっている．この課題は制御が必要な関数，特にフェーズを移行する関数に実行権限を持つ権利者を設定する必要があると考える．

第六章 まとめ

本研究ではブロックチェーンを用いた電子投票システムのアーキテクチャを提案した。また、その一部をプロトタイプとして、Solidity 言語で実装した。

本アーキテクチャは、改ざん不可能と再投票を実現した。また、暗号理論における安全性の定義から匿名性、公平性、適格性、個別検証可能性、二重投票の防止を満たすと言える。

プロトタイプを実行した結果、再投票や票の集計において正しい動作をすることが確認できた。

今後の課題としては、暗号理論における安全性の定義から頑健性および総合検証可能性についての再検討、プロトタイプの関数における実行権限について関数ごとに権限を設定する必要がある、といったことがあげられる。

謝辞

本研究を進めるにあたり，多大なるご指導，ご助力を頂いた指導教員である塚田恭章教授に深く感謝します．また，日々の議論を通じて多くの知識や意見を共有致しましたネットワークセキュリティ研究室の皆様に深く感謝いたします．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] Ivo Kubjas : “Using blockchain for enabling internet voting”, Tartu Univ,(2017).
- [2] 湯浅 壘道 : “エストニアの電子投票”, 社会文化研究所紀要,(2009).
- [3] Satoshi Nakamoto : “Bitcoin : A Peer-to-Peer Electronic Cash System”, (2008).
- [4] Ahmed Ben Ayed : “A conceptual secure blockchain-based electronic voting system”, IJNSA,(2017).
- [5] Andreas M. Antonopoulos : “ビットコインとブロックチェーン – 暗号通貨を支える技術 –”, NTT 出版,(2016).
- [6] 日経 XTECH : “高リスクの脅威が3つ—どうする日本の電子投票”,(2007),
<https://tech.nikkeibp.co.jp/it/article/OPINION/20071113/286974/>
- [7] 猪俣敦夫 : “サイバーセキュリティ入門 私たちを取り巻く光と闇”, 共立出版,(2016),
pp. 188-191.
- [8] 久保田 貴大 : “日本における安全なインターネット投票の導入に向けて”, 情報処理学会研究報告(2014).
- [9] 佐藤雅史,長谷川佳祐,佐古和恵,並木悠太,梶ヶ谷圭祐,松尾真一郎 : “ブロックチェーン技術の教科書”, シーアンドアール研究所,(2018).
- [10] Remix : “Solidity IDE”,
<http://remix.ethereum.org>
- [11] Nagai-r1 : “GitHub”,(2019),
https://github.com/Nagai-r1/Reserch_open/blob/master/evote_system.sol