Computer Security Symposium 2021
26 - 29 October 2021

ブロックチェーンを用いたメンタルポーカープロトコルの提案

土井 貴仁[1,a)]　廣友 雅徳[2,b)]　福田 洋治[3]　毛利 公美[4]　白石 善明[5]

概要：メンタルポーカープロトコルはインターネットのような公開通信路を用いて遠隔の 2 者が公平にカードを配布するプロトコルである．メンタルポーカープロトコルとして様々な状況を対象としたプロトコルが提案されている．本稿では，ブロックチェーンを用いたメンタルポーカープロトコルを提案する．

キーワード：メンタルポーカー, ブロックチェーン, スマートコントラクト

Mental Poker Protocol Using Blockchain

Takanori Doi[1,a)]　Masanori Hirotomo[2,b)]　Youji Fukuta[3]　Masami Mohri[4]　Yoshiaki Shiraishi[5]

Abstract: The Mental Poker Protocol is a protocol in which two remote parties distribute cards fairly using a public communication channel such as the Internet. Protocols for various situations have been proposed as mental poker protocols. In this paper, we propose a mental poker protocol using blockchain.

Keywords: Mental Poker,Blockchain,Smart contract

1. はじめに

クラウドサービスが普及し，様々なサービスがネットワークを介して利用できるようになってきた．ネットワークサービスを提供する際，プライバシー保護の観点から，ユーザを特定することなく匿名のままサービスを提供するための技術がいくつか提案されている．匿名認証通信技術 [1] は，暗号化通信と匿名認証を同時に実現する技術である．匿名性を持たせて認証を行い，かつ通信の暗号化を行うことで，ユーザのプライバシー保護と通信の安全性をともに実現できる．しかし，暗号化通信には鍵生成センターが必要になったり，匿名認証には認証局が必要になり，これらに信頼を持たせなければならないといった課題もある．仮に，ユーザを認証さえるグループに所属させ，そのグループに所属する全ユーザに匿名性を持たせて ID を配布することができれば，ID ベース暗号を用いて匿名認証通信が実現できる．この ID 配布は，ID 配布サーバをディーラー，ID 取得クライアントをプレイヤーと見なすと，メンタルポーカープロトコル [2] が利用できる．

メンタルポーカープロトコルは，ディーラー・プレイヤー間でネットワークを介して公平にカードを配布するプロトコルである．公平とは，ディーラーはどのプレイヤーにどのカードが配布されたか分からない，また，プレイヤーは配布されるカードを選べないということである．これまでにいくつかのメンタルポーカーの拡張プロトコルが提案されている [3], [4], [5], [6]．文献 [5] では，P2P 環境におけるメンタルポーカープロトコルが提案されている．文献 [6]

1 佐賀大学大学院理工学研究科
Graduate of Science and Engineering, Saga University
2 佐賀大学理工学部
Faculty of Science and Engineering, Saga University
3 近畿大学理工学部
Faculty of Science and Engineering, Kindai University
4 岐阜大学工学部
Faculty of Engineering, Gifu University
5 神戸大学大学院工学研究科
Graduate School of Engineering, Kobe University
a) 21726013@edu.cc.saga-u.ac.jp
b) hirotomo@cc.saga-u.ac.jp

では，ビットコインの分散台帳技術を利用したメンタルポーカープロトコルが提案されている．これらは，サーバに信頼性を置かず，メンタルポーカープロトコルを実行できる特徴がある．

本稿では，ブロックチェーンを用いたメンタルポーカープロトコルを提案する．提案プロトコルでは，ディーラー・プレイヤー間のカードの受け渡しをブロックチェーンを用いて行う．プロトコル実行過程のカードの値をブロックチェーンに記録することで，プロトコルが正しく実行されたことをあとから確認することができる．また，プロトコルをスマートコントラクトによって実装することにより，プロトコルの処理の正当性を保証することができる．

2. ブロックチェーン

2.1 ブロックチェーンの概要

ブロックチェーン [7] とはビットコインなどの仮想通貨に用いられている基盤技術である．ブロックチェーンはネットワーク上にある端末同士を直接接続し，仮想通貨の取引記録を暗号技術を用いて分散的に処理・記録する．ブロックチェーンはデータベースを機能強化する仕組みである．ブロックチェーン上での取引記録のことをトランザクションという．取引をおこなうときにトランザクションが生成され，生成された複数のトランザクションを１つにまとめたものをブロックと呼ぶ．ブロックは自身のヘッダに直前のブロックヘッダのハッシュ値をもつため，複数のブロックが鎖で繋がったような構造である．また，ブロックチェーンには取引を行う際に使用するお金の宛先のアドレスをウォレットアドレスという．このウォレットアドレスは公開鍵暗号に基づいて生成され，秘密鍵と公開鍵のペアから求める．

ブロックチェーンは P2P(Peer to Peer) 方式での通信を行っている．少なくとも 2 台のノードが互いに 1 対 1 で接続して通信する方法を P2P 方式と呼び，P2P 方式で接続を行うノードがたくさん集まることで P2P 分散ネットワークが形成される．

P2P 分散ネットワークに参加するノードは原則としてすべて対等な機能を持つので，一部のノードが故障や停止しただけで全体が影響を受けることは少ない．よって，ネットワークの規模が大きくなるほど耐障害性に優れる．また，一度ブロックチェーンに書き込まれた情報は記録内容の変更や削除ができず，情報の改ざんを防ぐことが可能である．また，データベースを複数端末に保存するので，ある端末のデータが破損した場合は別の端末に保存されているデータで自動修復される．ブロックチェーンは仮想通貨の取引記録を保存する仕組みであるが，Ethereum などのプラットフォームではスマートコントラクトを作成して，ブロックチェーン上で実行可能.

2.2 スマートコントラクト

スマートコントラクトとは，あらかじめ設定されたルールが満たされたとき，自動的に実行されるプログラムのことである．スマートコントラクトを利用することによってトランザクション作成の複雑な処理を自動的に処理が可能になる．スマートコントラクトの一例として自動販売機があげられる．

ブロックチェーンでのスマートコントラクトにはメリットがある．まずは**信頼性**があげられる．スマートコントラクトは第三者を介する必要がなく，事前に設定したルールが満たされれば必ず自動的にプログラムが実行されるからだ.

次に**透明性**があげられる．プログラムされたスマートコントラクトの内容やそれにより実行された取引の記録はブロックチェーン上で公開される．したがって不正が行われた場合，記録をさかのぼることで不正を検知する。

3. 暗号プロトコル

3.1 暗号プロトコルの概要

暗号や認証などの技術を複合的に利用してより高度な機能を実現するものや暗号化技術を応用し，安全に通信を行うための手順を暗号プロトコルと呼ぶ．暗号プロトコルにはビットコミットメントやメンタルポーカー，金持ち比べプロトコルなどのマルチパーティプロトコルがある．次節では，メンタルポーカープロトコルについて述べる．

3.2 メンタルポーカープロトコル

メンタルポーカー [2] は「電話でポーカーをするための方法」であり，安全にカードを配布する方法を実現している．ディーラー A が 52 枚のカードをプレーヤー B に 5 枚配布することを考える．配布には以下の条件を満たす必要がある.

(1) A はどのカードを配布したかを知ることができない.
(2) B は配布されるカードを選ぶことができない.

本稿では暗号化と復号に用いる鍵を以下のように定義する.

$$K = (E, D)$$

ここで鍵 K は暗号化関数 E と復号関数 D の組である.

また，メンタルポーカーでは以下のような特徴を持つ特殊な暗号を使用する.

$$E_1(E_2(x)) = E_2(E_1(x))$$

これは複数の鍵で複数回暗号化する場合，暗号化の順序を変えても得られる結果が等しいことや，複数回暗号化された暗号文を復号する場合，どのような順序でも等しい結果が得られることを表している．この暗号系を使用することでメンタルポーカーを実現する.

4. 提案プロトコル

本稿で提案するプロトコルはメンタルポーカーのやり取りを行う途中で値の変更をした際に，ブロックチェーンに値を保存する．プロトコルで使用する値はユーザーの入力値以外はブロックチェーンから取得する．

4.1 プロトコル

提案プロトコルについて説明する．提案プロトコルでは以下をブロックチェーンに保存する．

- ディーラーのアドレス
- プレイヤーのアドレス
- 山札
- プレイヤーの手札

上記の情報は状態変数であり，常にブロックチェーン上に保存される．ブロックチェーン上の情報はすべて公開されているため，山札の中身とプレイヤーの手札は暗号化してブロックチェーン上に保存する．3.2 節で述べた特殊な暗号系を用いる．以下にプロトコルの手順について述べる．ディーラーとプレイヤーはそれぞれ暗号鍵をもつ．

1. ディーラーが山札を作成し，暗号化とシャッフルを行う．暗号化とシャッフルをした山札をブロックチェーンに保存する．
2. プレイヤーが山札を取得し，暗号化とシャッフルを行う．続いて山札から 5 枚手札を選び，手札と手札以外の山札をブロックチェーンに保存する．
3. ディーラーが全プレイヤーの手札を取得し，ディーラーが手札を復号する．その手札をブロックチェーンに保存し，手札を更新する．
4. プレイヤーが自分以外のプレイヤーの手札を復号する．その手札をブロックチェーンに保存し，手札を更新する．
5. プレイヤーは自分の手札を取得し，プレイヤー自身が復号して手札の中身を確認する．

本稿で提案しているプロトコルは通常のメンタルポーカーにブロックチェーンを組み合わせることで通信内容を保存する．ブロックチェーンに通信内容を保存することで以下のメリットがあげられる．

- 通信内容の改ざんを防ぐ．
- ブロックチェーンに保存された通信履歴を確認することで通信内容の検証が可能．

ブロックチェーンによって通信内容が保証されるため，メンタルポーカーの安全性が上がることに繋がっている．また，通信内容がブロックチェーンにすべて保存されているので，ブロックチェーンを確認することでどのような通信をしたのかを容易に確認可能．

このプロトコルではブロックチェーンを使用するため，スマートコントラクトを作成する．スマートコントラクトの内容はすべてブロックチェーンに保存されるため，暗号化とシャッフルの機能をスマートコントラクトに記述できない．暗号化とシャッフルはスマートコントラクトの外部で行い，その結果をスマートコントラクトを使ってブロックチェーンに保存する．

以下にプロトコルの詳細を述べる．ここではプレイヤーは n 人参加する．そこで本稿ではプレイヤーをそれぞれ $P_i(i = 0, 1, ..., n-1)$ とする．

4.1.1 ゲーム作成

1〜52 までの 52 枚のカードを山札として作成する．ここで山札を S，ディーラーの鍵を k_d とする．

1. ディーラーは 1〜52 の配列 S を生成しシャッフル．
2. ディーラーは自身の鍵 k_d で山札を暗号化．
3. ディーラーが暗号化した山札 $E_{k_d}(S)$ をブロックチェーンに保存．

以上の手順を行うことでブロックチェーンにディーラーの鍵で暗号化された山札とディーラーのアドレスが保存される．

4.2 プレイヤー参加

プレイヤーは山札がブロックチェーンに保存されたことを確認してから参加する．ここでプレイヤー P_i の手札を H_i，手札を引いた後の山札を S_{h_i}，プレイヤー P_i の鍵を k_{p_i} とする．

1. プレイヤー P_i は $E_{k_d}(S)$ をブロックチェーンから取得．
2. プレイヤー P_i は $E_{k_d}(S)$ をシャッフル．
3. プレイヤー P_i は鍵 k_{p_i} で $E_{k_d}(S)$ を暗号化．
4. プレイヤー P_i はシャッフルした $E_{k_{p_i}}(E_{k_{p_{i-1}}}(...(E_{k_d}(S))...))$ から任意の 5 枚の手札 $E_{k_{p_i}}(E_{k_{p_{i-1}}}(...(E_{k_d}(H_i))...))$ を選ぶ．
5. プレイヤー P_i は $E_{k_{p_i}}(E_{k_{p_{i-1}}}(...(E_{k_d}(H_i))...))$ と $E_{k_{p_i}}(E_{k_{p_{i-1}}}(...(E_{k_d}(S))...))$ をブロックチェーンに保存．

以上の手順を行うことでブロックチェーンに暗号化された山札とプレイヤー P_i の手札，プレイヤー P_i のアドレスが保存される．この手順をプレイヤー全員が実行する．

4.3 ディーラーによる手札の復号

プレイヤーが保存した手札はディーラーとプレイヤーが暗号化している．プレイヤーが手札の中身を確認するためにはディーラーとプレイヤー自身で復号しなければならない．まずはディーラーが手札を復号し，ブロックチェーンに保存する．

1. ディーラーは $E_{k_{p_i}}(E_{k_{p_{i-1}}}(...(E_{k_d}(H_i))...))$ をブロックチェーンから取得．
2. ディーラーは $E_{k_{p_i}}(E_{k_{p_{i-1}}}(...(E_{k_d}(H_i))...))$ をディーラーの鍵で復号

3. ディーラーは復号した手札 $E_{k_{p_i}}(E_{k_{p_{i-1}}}(...(E_{k_{p_1}}(H_i))...))$ をブロックチェーンに保存.

以上の手順をすべてのプレイヤー対して実行する．この手順を行うことでブロックチェーン上にあるプレイヤーの手札が更新される．

4.4 プレイヤーによる手札の復号

ディーラーがプレイヤーの手札を復号しているので，プレイヤーは手札を確認するには自身以外のプレイヤーに復号してもらう必要がある．ブロックチェーンに保存されている手札を取得し，復号を行う．

1. 以下の手順をプレイヤー P_i はプレイヤー P_{i+1} からプレイヤー P_{n-1} の手札に対して実行する.

1.1 プレイヤー P_i は $E_{k_{p_{i+1}}}(E_{k_{p_i}}(H_{i+1}))$ をブロックチェーンから取得.

1.2. プレイヤー P_i は $E_{k_{p_{i+1}}}(E_{k_{p_i}}(H_{i+1}))$ をプレイヤーの鍵で復号.

1.3. プレイヤー P_i は復号した手札 $E_{k_{p_{i+1}}}(H_{i+1})$ をブロックチェーンに保存.

以上の手順をプレイヤー P_1 からプレイヤー P_{n-2} が行うことで，ブロックチェーンに保存されている手札は自分自身の鍵だけで暗号化されている状態になる．

4.5 プレイヤーによる自身の手札を復号

ディーラーと自分以外のプレイヤーが自分の手札を復号しているので，プレイヤーは自分の手札を復号することで手札の中身を確認する．ブロックチェーンに保存されている手札を取得し，復号を行う．

1. プレイヤー P_i は $E_{k_{p_i}}(H_i)$ をブロックチェーンから取得.

2. プレイヤー P_i は $E_{k_{p_i}}(H_i)$ をプレイヤーの鍵で復号.

3. プレイヤー P_i は復号した手札 H_i を確認する.

以上の手順を行いプレイヤーは自分が選んだ手札を確認する．

5. 提案プロトコルの実装

5.1 環境

本稿で提案しているプロトコルは暗号化とシャッフルをスマートコントラクトの外部で行う．実装では外部で行う処理は JavaScript を使用する．よって，環境はブロックチェーンに必要な環境とブロックチェーンのスマートコントラクトを JavaScript で実行するためのライブラリ，JavaScript アプリケーション開発の環境が必要になる．以下でプロトコルの実装と実行に使用した環境を述べる．

Ethereum Ethereum とは分散型アプリケーションやスマートコントラクトを構築するためのプラットフォーム．また，スマートコントラクトを記述する solidity などのプログラミング言語を持ち，ネットワーク上のブロックチェーンに任意の分散型アプリケーション（Dapps）やスマートコントラクトを記述しそれを実行することが可能である．Ethereum には本番環境である Main ネットワークのほか，Ropsten や Kovan などのテストネットワークがある．実装したアプリケーションは Ropsten ネットワークを使用する．

solidity Ethereum 上で実行可能なプログラム言語であり，可読性と生産性が高くコントラクトを記述することに特化した高水準言語

Infura Ethereum ブロックチェーンのノードなどを構築することなく，開発者が Infura API を利用するだけで独自サービスや Dapps を開発できるようにしたホスティングサービス.

turffle truffle とは Ethereum アプリケーション開発におけるデファクトスタンダードのフレームワーク．スマートコントラクトのコンパイル・テスト・マイグレーション機能を実装している.

web3.js web3.js は HTTP や IPC，WebSocket を使用してローカルまたはリモートのイーサリアムノードと対話するためのライブラリのコレクション．web3.js のライブラリを使用することで geth などのブロックチェーンクライアントを毎回起動せずにスマートコントラクトを実行することが可能になる.

MetaMask MetaMask は，Web ブラウザの拡張機能やスマホアプリとして利用する仮想通貨専用の Web ウォレット．Ethereum の通貨である ETH や，Ethereum をベースに発行された ERC20 トークンを保管することが可能．MetaMask ではイーサリアムのブロックチェーンを基盤として開発された分散型アプリケーションや，ブロックチェーンゲームなどと連携させることも可能．Dapps の利用料金やブロックチェーンゲームのアイテム購入代金は，MetaMask を通じて決済する.

React React は，Facebook とコミュニティによって開発されているユーザインタフェース構築のための JavaScript ライブラリ.

以上を使うことにより提案プロトコルの実装が可能になる.

5.2 実装

実装では使用するブロックチェーンは Ethereum テストネットワークの一つである Ropsten ネットワークを使用する．Ethereum アプリケーション開発は truffle を使用し，コントラクトを作成する．また，コントラクトをブロックチェーンにデプロイする際は Infura を利用し，Ropsten ネットワークにデプロイする．外部で行う処理を JavaScript で行い，その結果をブロックチェーンに保存す

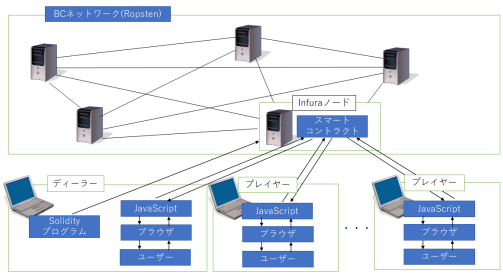

図 1 システム構成

るので，web3.js のライブラリを使用して JavaSript でコントラクトを実行するようにする．JavaScript で暗号化やシャッフル，コントラクトを実行するので JavaScript アプリケーションが必要になる．そこで，React を使用して SPA（シングルページアプリケーション）としてブラウザアプリケーションを作成する．Dapps を実行する際に仮想通貨を支払わなければならないので仮想通貨の支払いに MetaMask を利用する．

本稿で提案しているプロトコルを実行するためにディーラーがブロックチェーンにコントラクトをデプロイする．デプロイした結果を受けて Dapps のプログラムを修正し，プレイヤーに配布することで提案しているプロトコルをすべてのユーザーが実行可能になる．また，web3.js のライブラリを使用する際は JavaScript で非同期処理を行う必要がある．

6. まとめ

本稿では，ブロックチェーンを用いたメンタルポーカープロトコルの提案をした．本稿で提案しているプロトコルではメンタルポーカーにブロックチェーンを組み合わせることでメンタルポーカーの信頼性が上がる．また，匿名認証通信のセットアップに使用することが可能である．

参考文献

[1] 江村恵太，高橋健志，“匿名認証通信技術，” 情報通信研究機構研究報告，vol.62，no.2，2016.
[2] A. Shamir, R.L. Rivest, and L.M. Adleman, “Mental poker,” Mathematical Gardner, ed. D.E. Klarner, Wadsworth International, pp.37–41, 1981.
[3] C. Crepéau, “A zero-knowledge poker protocol that achives confidentiality of players’ strategy or how to achive an electronic poker face,” Proc. CRYPTO’86, Lecture Notes in Compute Science, vol.263, pp.239–247, 1986.
[4] G. Goldwasser and S. Micali, “Probabilistic encryption and how to play mental poker keeping secret all partial information,” Proc. 14th Annual ACM Symposium on Theory of Computing, pp.365–377, 1982.
[5] 畠山貴行，宮崎修一，“マジックプロトコルを用いた P2P 環境での公正なネットワークブラックジャック，” 2014 年度情報処理学会関西支部支部大会予稿集，2014.
[6] I. Bentov, R. Kumaresan and A. Miller, “Instantaneous Decentralized Poker,” Proc. ASIACRYPT 2017 pp.410–44, 2017.
[7] 総務省，ブロックチェーンの概要，2018. https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h30/html/nd133310.html