Computer Security Symposium 2020
26 - 29 October 2020

希望に基づく秘匿先手決定プロトコルとその拡張

篠田 悠斗[1,a)]　宮原 大輝[1,2]　品川 和雅[3,2]　水木 敬明[4]　曽根 秀昭[4]

概要：将棋などの 2 人で行うボードゲームでは，ゲームを始める手番（先手・後手）によって取りやすい戦略が異なり，プレーヤーごとに得意な手番があるため，プレーヤーの希望を踏まえた上で公平に手番を決めるのが良い．しかし，取りたい手番をそのまま相手に伝えてしまうと，自身の戦略が相手に漏れてしまう．したがって，お互いに取りたい手番の情報は秘匿したまま，自分が持ちたい手番はできるだけ取れるように手番を決定できれば嬉しい．この問題は，現在一般に行われている振り駒やじゃんけんなどでは解決できない．本稿では，この問題の定式化を行い，この問題を解決する「希望に基づく秘匿先手決定プロトコル」を物理的なカード組を用いて構成する．これに加えて，問題を多人数に拡張し，希望者のみから公平かつ秘密に抽選する「隠密抽選プロトコル」も提案する．

キーワード：物理的暗号技術，カードベース暗号，秘密計算

A Secure Protocol for Deciding the First Turn and Its Extension

Yuto Shinoda[1,a)]　Daiki Miyahara[1,2]　Kazumasa Shinagawa[3,2]　Takaaki Mizuki[4]　Hideaki Sone[4]

Abstract: We have to decide Sente/Gote (who will be the first player) when we play shogi. It should be decided based on players' requests because strategies of shogi rely on the first move or not and most of players have their preference in regard to Sente/Gote. However, if they simply tell their requests to each other, they will notice the other's strategy. Therefore, it is preferable that they can decide the first turn according to their requests while hiding any information about them. Note that this problem cannot be solved by Furigoma and rock paper scissors. In this paper, we formalize this problem and propose a "secure protocol for deciding the first turn" that solves this problem using a deck of physical cards. Moreover, we extend this problem to the multi-player setting and propose a "covert lottery protocol" that securely chooses a single player according to their requests.

Keywords: Physical cryptography, Card-based protocols, Secure multiparty computation

1. はじめに

2 人のプレーヤーが将棋の手番（先手・後手）を決める状況を考えよう．一般的には，じゃんけんや振り駒によってくじ引きのように決めることが通常であるが，ここではプレーヤーの希望を踏まえた，インテリジェントな方法を考える．将棋では，先手番と後手番で取りやすい戦法や，プレーヤーによって得意な手番があることから，（作戦を漏らさないために）自分が持ちたい手番の情報は相手に教えたくないが，自分が持ちたい手番はできるだけ取れるように決定できると嬉しい．すなわち，入力情報を互いに秘匿しながら，両者の希望する手番が異なるときにはその希望をそのまま出力し，一致するときにはランダムに手番を出力したい．本稿では，このようなある種の秘密計算を行

[1] 東北大学大学院情報科学研究科
Graduate School of Information Sciences, Tohoku University
[2] 産業技術総合研究所（AIST）
[3] 電気通信大学
The University of Electro-Communications
[4] 東北大学サイバーサイエンスセンター
Cyberscience Center, Tohoku University
[a)] yuto.shinoda.q7@dc.tohoku.ac.jp

うプロトコルを “希望に基づく秘匿先手決定プロトコル” と呼ぶことにする．

じゃんけんをして勝った人が自身の希望する手番を宣言することでは問題を解決できないことに注意しよう．なぜなら，負けた人の希望手番は漏れないものの，勝った人の希望手番は必ず漏れてしまうからである．また，将棋で一般的に用いられる振り駒の場合は，手番に関する希望情報は受け付けないまま，ランダムに手番を決めていることに注意しよう．

1.1 問題の定式化

希望に基づく秘匿先手決定プロトコルの機能 $\mathcal{F}$ の形式的な記述を与える．2 人のプレーヤー P_1 と P_2 がそれぞれ希望を表す秘密の入力ビット $x_1 \in \{0,1\}$ と $x_2 \in \{0,1\}$ を持っている．すなわち，各プレーヤー P_i に対して，$x_i = 1$ は先手番を希望し，$x_i = 0$ は後手番を希望していることを表す．また，機能 $\mathcal{F}$ はプレーヤー P_1 が持つ手番を出力するものとする．すなわち，$\mathcal{F} = 1$ は P_1 が先手番を持ち，$\mathcal{F} = 0$ は P_1 が後手番を持つことを表す．以上のことを踏まえ，機能 $\mathcal{F}$ は次のように自然に定義できる．

$$\mathcal{F}(x_1, x_2) \coloneqq \begin{cases} x_1 & \text{if } x_1 \oplus x_2 = 1 \\ i \xleftarrow{\$} \{0,1\} & \text{otherwise} \end{cases} \quad (1)$$

ここで，$\xleftarrow{\$}$ は集合からランダムに要素を 1 つ取り出すことを表している．すなわち，$x_1 \neq x_2$ のときは P_1 が持ちたい手番 x_1 を出力し，$x_1 = x_2$ のときはランダムに手番を出力する．

この問題は，$x_1 \neq \mathcal{F}(x_1, x_2)$ のとき，$x_1 = x_2$ であることが P_1 に漏れてしまい，$x_2 = \mathcal{F}(x_1, x_2)$ のときには $x_1 = x_2$ が P_2 に漏れてしまうことに注意しなければならない．すなわち，希望した手番を取ることができなかったプレーヤーは，相手が自分と同じ手番を希望していたことが分かってしまう．例えば，両者ともに先手番を希望して，P_1 が先手番を取ったとき，P_2 は希望したにもかかわらず先手番を取れなかったことから，相手も先手番を希望していて手番がランダムに決められたことが分かってしまう．しかし，これは $\mathcal{F}$ の性質上漏れざるを得ない情報である．[i]

1.2 問題の拡張

本稿では，さらにプレーヤー数を一般の $n\,(\geq 2)$ に拡張した場合を考える．具体的には，n 人から 1 人だけを抽選する場合を考えよう．この n 人の中に希望者が 1 人でもいればそのような希望者の中から当選者をランダムに決め，希望者がいなければ全員の中からランダムに当選者を決める計算となる．本稿では，このような計算を行うプロトコルを “隠密抽選プロトコル” と呼ぶことにする．応用先としては，例えば学校のクラスで代表者を 1 人決める場面において活用できると考えられる．

隠密抽選プロトコルの機能 $\mathcal{G}_n$ の形式的な記述は次の通りである．n 人のプレーヤーを考え，各プレーヤー P_i が希望を表す秘密の入力ビット $x_i \in \{0,1\}$ を持っている．すなわち，$x_i = 0$ は当選者となることを希望しないことを表し，$x_i = 1$ は当選者となることを希望することを表す．最初に，関数 $\mathsf{True} : \{0,1\}^n \to 2^{\{1,2,\ldots,n\}}$（$2^{\{1,2,\ldots,n\}}$ は $\{1,2,\ldots,n\}$ のベキ集合）を以下のように定義する．

$$\mathsf{True}(x_1, x_2, \ldots, x_n) \coloneqq \{i \mid x_i = 1, 1 \leq i \leq n\} \quad (2)$$

関数 True を用い，$\mathcal{G}_n$ は次のように記述できる．

$$\mathcal{G}_n(x_1, \ldots, x_n) \coloneqq \begin{cases} i \xleftarrow{\$} \mathsf{True}(x_1, \ldots, x_n) & \text{if } \mathsf{True}(x_1, \ldots, x_n) \neq \emptyset \\ i \xleftarrow{\$} \{1, 2, \ldots, n\} & \text{otherwise} \end{cases} \quad (3)$$

繰り返しになるが，$x_i = 1$ なるプレーヤー P_i の中から抽選を行うことを原則とし，そのようなプレーヤーがいない場合には全員を対象として 1 人をランダムに選んでいる．

この問題は，$x_i = 1$ かつ $\mathcal{G}_n \neq i$ のとき，$x_{\mathcal{G}_n} = 1$ であることがプレーヤー P_i に漏れてしまう．また，$x_i = 0$ かつ $\mathcal{G}_n = i$ のとき，プレーヤー P_i に $j \neq i$ である全てのプレーヤー P_j について $x_j = 0$ であることが漏れてしまう．しかし，これらのことも $\mathcal{G}_n$ の性質上漏れざるを得ないものである．

ここで $\mathcal{G}_n$ について $n = 2$ である場合を考える．このとき，$\mathcal{G}_2(0,0) \xleftarrow{\$} \{1,2\}$, $\mathcal{G}_2(1,1) \xleftarrow{\$} \mathsf{True}(1,1) = \{1,2\}$ であり，$\mathcal{G}_2(1,0) \xleftarrow{\$} \{1\}$，すなわち $\mathcal{G}_2(1,0) = 1$ であり，$\mathcal{G}_2(0,1) = 2$ であるので，$\mathcal{G}_2$ と式 (1) で定義した機能 $\mathcal{F}$ は，出力の形式は異なるが，本質的には同じである．したがって，$\mathcal{G}_n$ は $\mathcal{F}$ の自然な拡張である．

1.3 貢献

本稿では，上述の機能 $\mathcal{F}$ を実現するカードベースプロトコルを提案する．すなわち，物理的なカード組を用い，秘匿先手決定プロトコルを効率的に構築できることを示す．具体的には，2 者間の入力に対する排他的論理和（XOR）の値に基づいて，入力された手番を表すカード列をそのまま出力するか，乱数を表すカード列を出力するプロトコルを構築する．

さらに，6 枚 2 入力 AND プロトコル [1] を用いることで上で定義した機能 $\mathcal{G}_n$ を実現する隠密抽選プロトコルを効率的に構築できることを示す．後で詳しく説明するが，こ

[i] $x_1 = x_2$ のとき，抽選に当たったプレーヤーは希望の手番を取れる一方，外れたプレーヤーは相手の手番選好を知ることができるので，ある意味公平と言えるかもしれない．また，外れたプレーヤーが相手の選好を観戦している第三者に漏らすことは，自分自身の手番選好も漏らしていることを意味する．プレーヤー P_1 と P_2 はこのような $\mathcal{F}$ の性質を理解した上で，秘密計算するインセンティブを双方が感じるなら実行すればよい．

の提案プロトコルは，6 枚 AND プロトコル [1] において本来出力されずに余るカード列も活用する．

1.4 関連研究

物理的なカード組を用いて実装される秘密計算は**カードベース暗号**と呼ばれ，den Boer が five-card trick [2] を 1989 年に提案した以降，様々なプロトコルが提案されている．例えば，3 人の参加者の賛成・反対を秘匿したまま，その賛否の多数決を出力する 3 入力多数決プロトコル [3–5]，2 者間の所持金情報を秘匿したまま，所持金の大小関係だけを出力する金持ち比べ [6–8]，複数人の所持金情報を秘匿したまま，それらのランキング情報だけを出力するランキング計算 [9]，任意の数のグループに参加者を秘密に分割する秘匿グルーピング [10] などが挙げられる．また，ある問題に解が存在することを，解に関する情報を漏らすことなく証明するゼロ知識証明への応用も存在する [11,12]．

2. 準備

本節では，提案プロトコルで用いるカード組の性質を定義し，その後既存プロトコルを紹介する．

2.1 カード組

カードは表面が黒 ♣ または赤 ♡ のいずれかで，裏面は同一の ? であるものを用いる．カードの大きさは全て同一とし，表面の絵柄以外では完全に区別がつかないものとする．4 節で提案する隠密抽選プロトコルでは，**ナンバーカード**も用いる．ナンバーカードは，表面には $\boxed{1}\boxed{2}\cdots\boxed{m}$ のように数字が書いてあり，裏面は黒赤のカードと同様に ? である．

カードベース暗号では大抵，黒と赤のカードを以下のように組にすることで，0 と 1 の符号化を行う．

$$\boxed{\clubsuit}\boxed{\heartsuit} = 0, \quad \boxed{\heartsuit}\boxed{\clubsuit} = 1$$

この符号化ルールに従って 2 枚の裏になったカード列がビット $x \in \{0,1\}$ を表すとき，そのカード列を x の**コミットメント**と呼び，次のように表記する．

$$\underbrace{\boxed{?}\boxed{?}}_{x}$$

2.2 ランダム二等分割カット

ランダム二等分割カットは Mizuki と Sone によって考案されたシャッフル方法 [1] である．m を自然数とし，$2m$ 枚のカード列が存在するとき，カード列を半分に分割し，2 つのカード束を左右にランダムに入れ替えるシャッフルをする．2 つのコミットメント $a,\ b \in \{0,1\}$ にランダム二等分割カットを適用する例を以下に示す．最初に，このカード束を 2 等分する．この例では，a と b に分けることになる．

$$\underbrace{\boxed{?}\boxed{?}}_{a}\ \underbrace{\boxed{?}\boxed{?}}_{b}$$

次に，2 つに分けたカード束の左右を，順序が分からなくなるまで入れ替える．

$$[\ \boxed{?}\boxed{?}\ |\ \boxed{?}\boxed{?}\ ] \rightarrow \boxed{?}\boxed{?}\boxed{?}\boxed{?}$$

ここで，$[\cdot|\cdot]$ はランダム二等分割カットを示すものとし，以降もこの記号を用いる．この結果，a, b または b, a の並びのカード束が，それぞれ 1/2 の確率で得られる．

ランダム二等分割カットをそのまま手で実行した場合，シャッフル後のカード列の順序が漏れてしまうという懸念が存在するが，裏面の上下非対称性を利用する安全な実装方法が示されている [13]．

2.3 Pile-scramble シャッフル

Pile-scramble シャッフルは Ishikawa らによって導入されたシャッフル方法 [14] である．具体的には，カード枚数がすべて等しい m 個のカード束を $(\mathsf{pile}_1, \mathsf{pile}_2, \ldots, \mathsf{pile}_m)$ としたとき，$(\mathsf{pile}_{\pi^{-1}(1)}, \mathsf{pile}_{\pi^{-1}(2)}, \ldots, \mathsf{pile}_{\pi^{-1}(m)})$ のようにランダムな束の並びが得られるシャッフルのことを指す．ここで，S_m を m 次の対称群とするとき，$\pi \xleftarrow{\$} S_m$ である．このようなランダムなカード束の並び替えは，封筒などの身近な道具を利用して実装できる．

2.4 6 枚 AND プロトコル

この既存プロトコルは，ランダム二等分割カットを用いて実現された AND プロトコル [1] であり，2 つの入力 a と b のコミットメントと 2 枚の追加カードから $a \wedge b$ のコミットメントを得る．

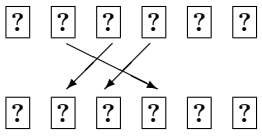
$$\underbrace{\boxed{?}\boxed{?}}_{a}\ \underbrace{\boxed{?}\boxed{?}}_{b}\ \boxed{\clubsuit}\boxed{\heartsuit} \rightarrow \underbrace{\boxed{?}\boxed{?}}_{a\wedge b}$$

プロトコルの流れを以下に示す．

(1) a と b のコミットメントを並べ，その間に 0 のコミットメントを並べる．

$$\underbrace{\boxed{?}\boxed{?}}_{a}\ \underbrace{\boxed{?}\boxed{?}}_{0}\ \underbrace{\boxed{?}\boxed{?}}_{b}$$

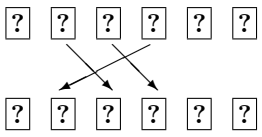
(2) 以下のように並べ替える．



(3) ランダム二等分割カットを適用する．

$$[\ \boxed{?}\boxed{?}\boxed{?}\ |\ \boxed{?}\boxed{?}\boxed{?}\ ] \rightarrow \boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}$$

(4) 以下のように並べ替える．



(5) 左の 2 枚を表向きにする．その絵柄によって，以下の 2 パターンのように $a \wedge b$ のコミットメントが得られ，さらに $\overline{a} \wedge b$ も得ることができる [15]．

(i) ♣♡ $\underbrace{??}_{a\wedge b}\ \underbrace{??}_{\overline{a}\wedge b}$ (ii) ♡♣ $\underbrace{??}_{\overline{a}\wedge b}\ \underbrace{??}_{a\wedge b}$

ここで，あるビット列 (x,y) が与えられたとき，get と shift の 2 つの動作を以下のように定義する．

$$\begin{aligned} &\mathsf{get}^0(x,y)=x, && \mathsf{get}^1(x,y)=y \\ &\mathsf{shift}^0(x,y)=(x,y), && \mathsf{shift}^1(x,y)=(y,x) \end{aligned} \tag{4}$$

これらを用いると，任意のビット $\ell\in\{0,1\}$ に対して AND 計算は以下のように書ける [3]．

$$a\wedge b=\mathsf{get}^a(\mathsf{shift}^0(0,b))=\mathsf{get}^{a\oplus\ell}(\mathsf{shift}^\ell(0,b)) \tag{5}$$

6 枚 AND プロトコルは，式 (5) に基づき，a の値を漏らさずに $a\wedge b$ のコミットメントを得ている．実際，ステップ 2 からステップ 4 の操作によって，ステップ 1 で置いたカード列は以下のように変形される．

$\underbrace{??}_{a}\ \underbrace{??}_{0}\ \underbrace{??}_{b} \to \underbrace{??}_{a\oplus r}\ \underbrace{????}_{\mathsf{shift}^r(0,b)}$

ここで，$r\xleftarrow{\$}\{0,1\}$ はステップ 3 で行ったランダム二等分割カットによって生成された乱数ビットであり，偶数回入れ替えを行っていれば $r=0$，奇数回入れ替えを行っていれば $r=1$ である．また，あるビット列 (x,y) に対し，左側の記法は右側を意味する．

$\underbrace{????}_{(x,y)}\ \underbrace{??}_{x}\ \underbrace{??}_{y}$

したがって，ステップ 5 で $a\oplus r$ の値を明らかにすることで，所望のコミットメントの位置が分かる．

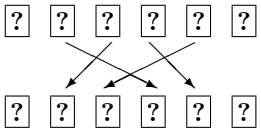
2.5 1 入力保存 XOR プロトコル

この既存プロトコルは，a, b の入力コミットメントに対し，入力のうち 1 つを保存しつつ $a\oplus b$ のコミットメントを出力するプロトコルである [3]．以下にプロトコルの流れを示す．

(1) 入力 a と b のコミットメントを並べ，さらにその右側に 0 のコミットメントを並べる．

$\underbrace{??}_{a}\ \underbrace{??}_{b}\ \underbrace{??}_{0}$

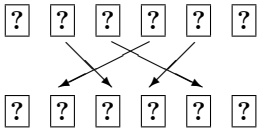
(2) 以下のように並べ替える．



(3) ランダム二等分割カットを適用する．

$[\,???\mid???\,] \to ??????$

(4) 以下のように並べ替える．



(5) 左 2 枚を表向きにする．その絵柄によって，以下の 2 パターンのように $a\oplus b$ のコミットメントと a のコミットメントが得られる．

(i) ♣♡ $\underbrace{??}_{a\oplus b}\ \underbrace{??}_{a}$ (ii) ♡♣ $\underbrace{??}_{\overline{a\oplus b}}\ \underbrace{??}_{\overline{a}}$

コミットメントの否定はそのカード 2 枚を入れ替えるだけで良いため，(ii) の場合も所望のコミットメントを得ることができる．

3. 希望に基づく秘匿先手決定プロトコル

本節では，$\mathcal{F}$ を実現する希望に基づく秘匿先手決定プロトコルについて，まずプロトコルの構築における考え方をまとめ，次に実際のプロトコルの記述を示す．提案プロトコルは，2 人の希望を表すコミットメント x_1 と x_2 を入力とし，2 枚の追加カードを用いて，P_1 が持つ手番のコミットメント $\mathcal{F}(x_1,x_2)$ を出力する．

$\underbrace{??}_{x_1}\ \underbrace{??}_{x_2}\ ♣♡ \to \underbrace{??}_{\mathcal{F}(x_1,x_2)}$

3.1 プロトコルの考え方

式 (1) をもう一度振り返ろう．2 つの入力コミットメント x_1 と x_2 の値が異なる際には $\mathcal{F}=x_1$ であり，一致する際には $\mathcal{F}=r\xleftarrow{\$}\{0,1\}$ である．すなわち，次の式が成り立つ．

$$\begin{aligned} &\mathcal{F}(x_1,x_2)= \\ &\mathsf{get}^{x_1\oplus x_2}(r,x_1)=\mathsf{get}^{x_1\oplus x_2}(\mathsf{shift}^0(r,x_1)) \end{aligned} \tag{6}$$

したがって，まず初めに次のカード列を用意する．

$\underbrace{??}_{x_1\oplus x_2}\ \underbrace{??}_{r}\ \underbrace{??}_{x_1}$

これは，2.5 節で紹介した 1 入力保存 XOR プロトコルを用いた後に r のコミットメントを作成すれば良い．式 (6) と式 (5) を見比べると，式 (6) は上のカード列に 6 枚 AND プロトコルをそのまま適用することで，値を漏らさずに計算できる．

3.2 プロトコルの実現

プロトコルの詳細な流れを以下に示す．

(1) P_1 と P_2 はそれぞれ，x_1 と x_2 のコミットメントを秘密に作る．

$\underbrace{??}_{x_1}\ \underbrace{??}_{x_2}$

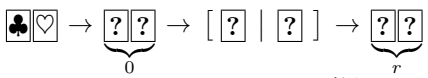
(2) x_1 と x_2 のコミットメントに対して 1 入力保存 XOR プロトコルを適用し，x_1 のコミットメントと $x_1\oplus x_2$ のコミットメントを得る．

$\underbrace{??}_{x_1}\ \underbrace{??}_{x_2}\ \underbrace{??}_{0} \to ♣♡\ \underbrace{??}_{x_1\oplus x_2}\ \underbrace{??}_{x_1}$

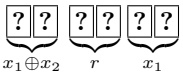
このプロトコルにおいて，2 枚の表になったカード ♣ と ♡ も得られる．

(3) 前ステップの 1 入力保存 XOR プロトコルで表になっ

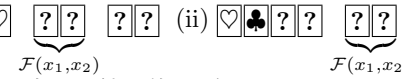
た 2 枚のカードを裏返し，ランダム二等分割カットを適用することで乱数 $r \xleftarrow{\$} \{0,1\}$ のコミットメントを得る．



以上で必要な 3 つのコミットメントが揃った．



(4) 上の 3 つのコミットメントに対し 6 枚 AND プロトコルを適用し，$\mathcal{F}(x_1, x_2)$ のコミットメントを得る．



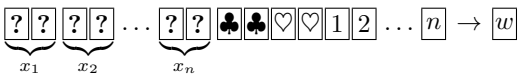
本プロトコルの流れは結果的に（r のコミットメントの作成を除くと）Nishida らが提案した 3 入力多数決関数プロトコル [3] と一致していた．$\mathcal{F}$ の定義から自明ではないが，本プロトコルは $(x_1, \overline{x_2}, r)$ の多数決を取っていると見なすこともできる．

3.3 安全性について

一般に，カードベースプロトコルではカードをめくるときのみ入力情報が漏れ得る．提案プロトコルは既に安全性が証明されている既存プロトコルを組み合わせて構成されているため，情報理論的に安全である．実際，6 枚 AND プロトコルと 1 入力保存 XOR プロトコルのステップ 5 でめくられるカード列は，♣♡ と ♡♣ である確率が 1/2 であり，入出力に依存せずランダムなため，入力情報は一切漏れていない．

4. 隠密抽選プロトコル

本節では，$\mathcal{F}$ の拡張である機能 $\mathcal{G}_n$ を実現する手法について，最初に本プロトコルの考え方をまとめ，次に具体的な実現方法を示す．提案プロトコルは，n 人の希望を表す n 個のコミットメント $x_1, x_2, \ldots, x_n$ を入力とし，4 枚の赤黒カードと n 枚のナンバーカードを追加して，当選者 $w = \mathcal{G}_n(x_1, x_2, \ldots, x_n)$ を表すナンバーカードを 1 枚出力する．

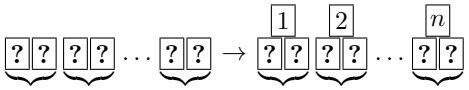


4.1 プロトコルの概要

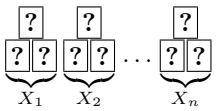
式 (3) をもう一度振り返ろう．$\mathcal{G}_n$ を実現するためには，希望者がいる場合にはその希望者の集合 $\mathsf{True}(x_1, x_2, \ldots, x_n)$ から抽選を行い，希望者がいない場合には全体の集合 $\{1, 2, \ldots, n\}$ から抽選を行えば良い．これを達成するために，最初に n 個の入力コミットメントに Pile-scramble シャッフルを適用し，誰が何番目の入力か分からなくさせる．このように並びを崩してもどの入力が誰のものか分かるようにするため，予め x_i のコミットメントにナンバーカード $\boxed{i}$ を付しておく．

表 1　y_i とトークン t の変化

X_i	t	$y_i = X_i \wedge t$	$t \coloneqq \overline{X_i} \wedge t$
♣♡	♡♣	♣♡	♡♣
♡♣	♡♣	♡♣	♣♡
♣♡	♣♡	♣♡	♣♡
♡♣	♣♡	♣♡	♣♡
♡♣	♣♡	♣♡	♣♡

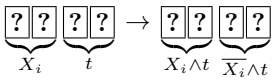


すなわち，いま一様ランダムに置換された次の列になっている．



このとき，コミットメント $X_1, X_2, \ldots, X_n$ を左から順にめくると，初めて 1 が判明した入力コミットメントは希望者の集合からランダムに選ばれたといえるため，それに付するナンバーカードを当選者として出力すれば良い．もし全ての入力値が 0 なら，一番右にあるナンバーカードを出力することで全体からランダムに当選者を選ぶことになる．この原理に基づきプロトコルを構成していくが，当然，そのまま入力コミットメントをめくってしまうと，当選者の入力値や 0 を入力した人数が漏れてしまう．例えば，$n = 5$ として，Pile-scramble シャッフルを適用した入力コミットメント列が $0, 0, 1, 0, 1$ とした時，3 人目までの入力コミットメントをめくっていくことになる．この場合，少なくとも 2 人が 0 を入力し，さらに当選者は 1 を入力していたことが全員に漏れてしまう．さらに，$0, 0, 0, 0, 0$ の場合には，当選者以外の全ての入力コミットメントが 0 であることが全員に漏れてしまう．以上のようなことを避けるため，入力値は秘匿したまま上の計算をしなければならない．

この計算のために，“トークン” コミットメントを導入する．トークンは，各々の入力コミットメントを書き換えていくために用いる．すなわち，初めて入力値が 1 となったときだけそのコミットメントを 1 とし，残りを全て 0 とすることで，当選者を決定する．具体的には，トークンの初期値を 1 として一番左の入力と AND 計算を行っていき，その AND の値を入力コミットメントに置き換える（入力が 1 かつトークンが 1 のときだけ 1 を出力する）．トークンは初めて 1 が出力されるまで 1 のままであり，1 が出力された後は 0 となれば良い．この計算は入力の否定とトークンの値の AND 計算で達成される．以上より，左から i 番目の入力値 X_i とトークン t が与えられたとき，次の計算を行い，i 番目のコミットメントを $y_i = X_i \wedge t$ で置き換え，トークンを $t \coloneqq \overline{X_i} \wedge t$ で置き換える $(1 \le i \le n-1)$．



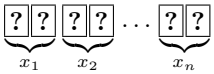
(7)

ただし，$t = 1$ で始める．具体的に，$X_1, X_2, \ldots, X_5 = 0, 1, 0, 1, 1$ の場合を考える．このとき，X_i と t から計算される y_i と t は表 1 のようになる．最初に，$X_1 = 0$ であるから $y_1 = 0 \wedge 1 = 0$ となり，$t := 1 \wedge 1 = 1$ となる．次に $X_2 = 1$ であるから，$y_2 = 1 \wedge 1 = 1$ となり，$t := 0 \wedge 1 = 0$ と計算される．$y_i = X_i \wedge t$，$t := \overline{X_i} \wedge t$ であることから，t が一度でも 0 になればいずれの AND 計算も以降の計算では 0 になる．式 (7) を実現するには，6 枚 AND プロトコルを用いれば良いため，隠密抽選プロトコルは 6 枚 AND プロトコルを $n-1$ 回用いることで構成できる．さらに，$n-1$ 番目の入力まで全て 0 であったときトークンは $t = 1$ であるため，$y_n = t$ とすることで希望者がいない場合の抽選が成立する．$n-1$ 番目までに 1 のコミットメントが 1 つでもあればそのときのトークンは $t = 0$ であり，この場合も $y_n = t$ とすれば良い．

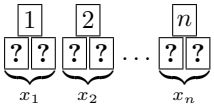
4.2 プロトコルの実現

プロトコルの詳細な流れを以下に示す．

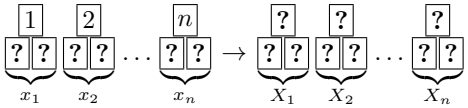
(1) 各プレーヤーは秘密に入力を作り，n 個の入力コミットメントを得る．



(2) x_i のコミットメントの上にナンバーカード $\boxed{i}$ を置き，3 枚のカードからなるカード束を n 個作る．



(3) 各ナンバーカードを裏向きにして Pile-scramble シャッフルを適用する．

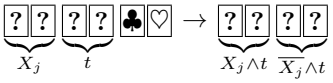


シャッフル後のコミットメントの値を $X_1, X_2, \ldots, X_n$ とする．

(4) 赤黒のフリーカードを一組使い，$\boxed{\heartsuit}\boxed{\clubsuit}$ と並べて裏返しトークン $t = 1$ のコミットメントを作る．

(5) $j = 1$ として次の計算を $n-1$ 回行う．

(a) X_j とトークン t のコミットメントを入力とし，残り 1 組のフリーカード $\boxed{\clubsuit}\boxed{\heartsuit}$ を用いて 6 枚 AND プロトコルを行うことで，次のコミットメントを得る．

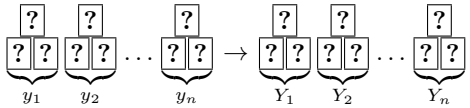


前者のコミットメントは，X_j のコミットメントと束になっていたナンバーカードと束にする．後者のコミットメントは次のトークン t とする．出力を決定する際にめくった 2 枚のカード $\boxed{\clubsuit}\boxed{\heartsuit}$ は，次の AND 計算に再利用する．

(b) $j := j + 1$ とする．

(6) トークン t のコミットメントを y_n とする．

(7) コミットメント y_i とナンバーカードから成る n 個のカード束に対し，再度 Pile-scramble シャッフルを適用する．



シャッフル後のコミットメントを $Y_1, Y_2, \ldots, Y_n$ とする．

(8) コミットメント $Y_1, Y_2, \ldots, Y_n$ をめくり，1 が出たカード束のナンバーカードのみを表向きにする．そのナンバーカードに該当するプレーヤーが当選者である．

4.3 安全性について

3.3 節で述べたように，カードをめくるときに情報が漏れ得るので，ステップ 5 とステップ 8 に着目する．ステップ 5 では安全な 6 枚 AND プロトコルを使用しているため，入力情報が漏れることはない．ステップ 8 では当選者を示すためにコミットメント $Y_1, Y_2, \ldots, Y_n$ をめくる．その結果，1 つのコミットメント Y_i のみが当選者を表す 1 であり，その他のコミットメントは 0 である．このとき，前ステップで適用した Pile-scramble シャッフルの特性から，i は他の値とは独立して $\{1, 2, \ldots, n\}$ に一様ランダムに分布している．したがって，本プロトコルでめくられるカードは入出力から独立して一様ランダムであるため，情報理論的に安全である．

5. 結論

本稿では，新たな問題として希望情報を秘匿したままそれに基づいて将棋の先手と後手を決定する問題を定式化し，これを解決する “希望に基づく秘匿先手決定プロトコル” を考案した．本プロトコルは 2 つの入力情報に対する XOR の値に基づいて，希望が一致しない場合には両者の希望通りに出力し，一致する場合にはランダムに手番を決定させることで実現した．さらに，問題の拡張として，プレーヤーの数を 2 人から n 人に拡張した問題についても定式化し，それを解決する “隠密抽選プロトコル” を考案した．このプロトコルは，入力順序をランダムにした各希望情報に対して，トークンと呼ばれるコミットメントを導入し 6 枚 AND プロトコルを適用していくことで，希望者がいる場合にはランダムな順序の中で 1 番目になった希望者を当選者とし，希望者がいない場合には一番最後のプレーヤーを当選者とすることで実現できた．隠密抽選プロトコルでは，各プレーヤーが当選者になったかどうかを表す情報をコミットメントのまま得られる．これはさらに次の計算を適用させたり，秘匿したまま出力させなければならな

い場面などに対して更なる応用ができると考えられる．

参考文献

[1] T. Mizuki and H. Sone, “Six-card secure AND and four-card secure XOR,” Frontiers in Algorithmics, eds. by X. Deng, J.E. Hopcroft, and J. Xue, vol.5598, pp.358–369, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2009.

[2] B. denBoer, “More efficient match-making and satisfiability the five card trick,” Advances in Cryptology — EUROCRYPT ’89, eds. by J.-J. Quisquater and J. Vandewalle, vol.434, pp.208–217, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 1990.

[3] T. Nishida, Y. Hayashi, T. Mizuki, and H. Sone, “Securely computing three-input functions with eight cards,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol.E98.A, no.6, pp.1145–1152, 2015.

[4] T. Nakai, S. Shirouchi, M. Iwamoto, and K. Ohta, “Four cards are sufficient for a card-based three-input voting protocol utilizing private permutations,” Information Theoretic Security, ed. by J. Shikata, vol.10681, pp.153–165, Lecture Notes in Computer Science, Springer, Cham, 2017.

[5] Y. Watanabe, Y. Kuroki, S. Suzuki, Y. Koga, M. Iwamoto, and K. Ohta, “Card-based majority voting protocols with three inputs using three cards,” 2018 International Symposium on Information Theory and Its Applications (ISITA), pp.218–222, 2018.

[6] T. Nakai, Y. Tokushige, Y. Misawa, M. Iwamoto, and K. Ohta, “Efficient card-based cryptographic protocols for millionaires’ problem utilizing private permutations,” Cryptology and Network Security, eds. by S. Foresti and G. Persiano, vol.10052, pp.500–517, Lecture Notes in Computer Science, Springer, Cham, 2016.

[7] H. Ono and Y. Manabe, “Efficient card-based cryptographic protocols for the millionaires’ problem using private input operations,” 2018 13th Asia Joint Conference on Information Security (AsiaJCIS), pp.23–28, 2018.

[8] D. Miyahara, Y. Hayashi, T. Mizuki, and H. Sone, “Practical card-based implementations of Yao’s millionaire protocol,” Theoretical Computer Science, vol.803, pp.207–221, 2020.

[9] K. Takashima, Y. Abe, T. Sasaki, D. Miyahara, K. Shinagawa, T. Mizuki, and H. Sone, “Card-based secure ranking computations,” Combinatorial Optimization and Applications, eds. by Y. Li, M. Cardei, and Y. Huang, pp.461–472, Springer International Publishing, Cham, 2019.

[10] Y. Hashimoto, K. Shinagawa, K. Nuida, M. Inamura, and G. Hanaoka, “Secure grouping protocol using a deck of cards,” Information Theoretic Security, ed. by J. Shikata, vol.10681, pp.135–152, Lecture Notes in Computer Science, Springer, Cham, 2017.

[11] T. Sasaki, D. Miyahara, T. Mizuki, and H. Sone, “Efficient card-based zero-knowledge proof for Sudoku,” Theoretical Computer Science, pp.1–8, to appear.

[12] P. Lafourcade, D. Miyahara, T. Mizuki, A. Nagao, H. Sone, L. Robert, K. Shinagawa, and S. Takeshige, “Card-based ZKP protocols for Takuzu and Juosan,” In 10th International Conference on Fun with Algorithms (FUN), pp.20:1–20:20, Island of Favignana, Trapani, Italy, 2020.

[13] I. Ueda, D. Miyahara, A. Nishimura, Y. Hayashi, T. Mizuki, and H. Sone, “Secure implementations of a random bisection cut,” International Journal of Information Security, pp.445–452, 2019.

[14] R. Ishikawa, E. Chida, and T. Mizuki, “Efficient card-based protocols for generating a hidden random permutation without fixed points,” Unconventional Computation and Natural Computation, eds. by C.S. Calude and M.J. Dinneen, vol.9252, pp.215–226, Lecture Notes in Computer Science, Springer, Cham, 2015.

[15] T. Nishida, Y. Hayashi, T. Mizuki, and H. Sone, “Card-based protocols for any boolean function,” Theory and Applications of Models of Computation, eds. by R. Jain, S. Jain, and F. Stephan, vol.9076, pp.110–121, Lecture Notes in Computer Science, Springer, Cham, 2015.