B-04

麻雀を用いた安全な計算のプロトコル設計

Secure Multiparty Computations Using the Mah-jong

上嶋 章宏[†]　　安藤 立紀[‡]

Akihiro UEJIMA[†]　　Tatsuki ANDO[‡]

1　はじめに

暗号プロトコルは，古くより軍事的・外交的な目的で研究が進められてきた．暗号プロトコルを用いて様々な目的を実現することを，**マルチパーティプロトコル**（multi-party protocol）と呼ぶ．例えば，電話で公平にポーカーを行うこと（メンタルポーカープロトコル）や，お互いの財産を秘密にして，どちらが金持ちかを知ること（金持ちの財産比べプロトコル）や，電話で同時に情報を交換すること（同時文書交換プロトコル）などがその一例として挙げられる．マルチパーティプロトコルの一種として，入力情報の漏洩なしに論理関数を計算するための**安全な計算**と呼ばれるものが存在する．本研究では，この安全な計算を研究対象とする．

本研究は，コンピュータ非依存暗号（cryptographic protocol without computer），レクリエーション暗号（recreational cryptography），あるいは人間暗号（human-centric cryptography）と呼ばれる研究分野に属す．その名称からも分かるように，楽しみながら計算を実現することを目的として含んでおり，根源的には暗号プロトコルの分野に属することから，セキュリティ関係の分野に研究成果の応用が期待できる．近年，個人へのなりすましを防止する目的で，個人認証の研究が盛んに行われている．指静脈認証システムに代表される生体認証は，その主たる研究成果である．本研究で扱う安全な計算は，複数人のプレイヤーで 1 つの計算を行うことから，個人認証ではなくグループ認証といったことに応用できると考えている．

当該分野での既存研究として，カード組（トランプ）[3]，PEZ ディスペンサー [1]，15 パズル [2] などの身近なものを用いて安全な計算を行うプロトコルが，既に示されている．

本研究の目的は，一般的によく知られたゲームである**麻雀**で使用する麻雀牌（以下，単に **牌** と呼ぶ）を用いて，安全な計算を実現するプロトコルの設計である．その際，任意の対称関数と一般の関数について，より多くの変数まで対応できることと，できるだけ実際の麻雀に似た動作・状況で安全な計算が行えることを，プロトコル設計の目標とする．

[†]大阪電気通信大学，Osaka Electro-Communication University

[‡]東芝デジタルメディアエンジニアリング，Toshiba Digital Media Engineering Corporation

2　準備

n 人のプレイヤー $P_1, P_2, \cdots, P_n$ が存在し，各プレイヤー P_i $(i = 1, 2, \cdots, n)$ は各々1 ビットの情報 $x_1, x_2, \cdots, x_n \in \{$ 真, 偽 $\}$ を秘密に持っている．このとき，全てのプレイヤーが，各々の持つ情報を秘密にしたまま，ある論理関数 $f : \{$ 真, 偽 $\}^n \to \{$ 真, 偽 $\}$ について，$f(x_1, x_2, \cdots, x_n)$ を計算することを**安全な計算**と呼ぶ．

プロトコル設計においては，安全性・正確性に注意が必要である．**安全性**とは，設計したプロトコルによる出力を各プレイヤー P_i が見たとき，他のどのプレイヤーが持つ情報についても，一切の知識を得られないということである．**正確性**とは，与えられた入力 $x_1, x_2, \cdots, x_n$ に対する関数 f の出力と，設計したプロトコルによる出力が，常に一致することである．

本研究に関わる麻雀のルールについて説明する．麻雀では，同種の牌は各 4 枚ずつ存在する．4 人のプレイヤーがそれぞれ 13 枚の手牌を持ち，牌の山（**ツモ山**）から 1 牌を引き入れ，その 14 枚の手牌で**役**を作る．役ができていないときは，手牌から任意の 1 牌を捨てなければならない．これを 4 人で順番に，ツモ山がなくなるか，誰かが役を完成させるまで続ける．引けば役が完成する牌のことを**待ち牌**と呼ぶ．13 枚の手牌で，特定の種類の牌（待ち牌）を 1 枚ツモれば役が完成する状態を**聴牌**（テンパイ）と呼ぶ．

今回のプロトコルでは，**国士無双**と呼ばれる役を使用する．これは，指定の 13 種の牌を各 1 枚ずつと，それらのうちどれかをもう 1 枚加えた 14 枚で完成される役である．以下では，指定の 13 種の牌をアルファベットの A〜M（あるいは a〜m）として表現する．但し，表現の都合上，アルファベットの大文字・小文字を区別して用いるが，両者は同種の牌を表す．

本稿では，国士無双を用いることで，51 変数までの任意の対称関数と，5 変数までの一般の関数について，安

全な計算を実現するプロトコルが存在することを示す．

3　対称関数の安全な計算

対称関数とは，関数 f の変数を任意に置換しても，f の出力の結果が変化しないような関数であり，n 変数の基本対称関数 S_i^n $(i = 0, 1, \cdots, n)$ は，n 個の入力のうち，ちょうど i 個の入力が真であるときのみ，出力が真となる関数である．n 変数の対称関数は，n 変数の基本対称関数の部分和集合で表せることから，対称関数の計算は，基本対称関数の計算に置き換えることができる．

以下に示すプロトコルでは，51 変数までの対称関数の安全な計算が行える．まず，その初期配置を図 1 に示す．初期配置において全ての牌は伏せられており，出力を得る時点まで表返されることはない．

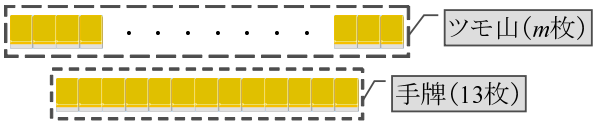

図 1: 牌の初期配置

各プレイヤー P_i は他の全てのプレイヤーに見えない状態で，1 人ずつ順番に以下のいずれかの操作を行う．

操作 A　プレイヤー P_i の持つ情報が偽ならば，プレイヤーはなにも操作を行わない．

操作 B　プレイヤー P_i の持つ情報が真ならば，プレイヤーはツモ山の右端の 1 牌を手牌の右端に加え，手牌の左端の 1 牌をツモ山の左端に加える．

全てのプレイヤーが各 1 回ずついずれかの操作をし終えたら，手牌をランダムに並び替える．その後手牌のみを全て表返し，手牌が A〜M 各 1 枚ずつで構成されていれば，得られる出力は真とする．A〜M のうち欠けているものがあれば，得られる出力は偽とする．このプロトコルを実現するための手牌とツモ山の具体的な牌の並び順の基本形を図 2 に示す．

図 2: 牌の並び順

図 2 の上段はツモ山を表し，下段は手牌を表す．図 2 は並び順の基本形であり，計算したい基本対称関数によって適時調整される．図 2 に対し操作 B を繰り返し実行して並び順をループさせていったとき，手牌が大文字のアルファベット A〜M のみで構成される状態ならば，その手牌に 13 種の牌の内どの 1 牌を加えても国士無双が完成する **13 面待ち** と呼ばれる特殊な聴牌の状態である．手牌中に小文字のアルファベット a〜m が 1 つ以上含まれている状態ならば，特定の 1 牌（指定の 13 種牌のうち，手牌に存在しない 1 種のみ）をその手牌に加えれば国士無双が完成する聴牌の状態である．本プロトコルでは，最後に手牌を表返したときに得られた牌の状態が，13 面待ちであれば得られる出力は真，13 面待ちでない聴牌であれば偽とする．

このプロトコルの安全性と正確性の略証を以下に示す．プレイヤーがいずれの操作を行ったとしても，その前後でツモ山と手牌の枚数は変わらない．従って，このプロトコルは操作の途中の段階では安全性を満たしていると言える．最後に手牌を見たとき，出力が真の場合，手牌の状態は 1 パターンしかない．A〜M と指定の 13 種の牌との対応関係を毎回ランダムに決定し，更に手牌を表返す直前に並び順をランダムに並び替えることによって，出力が偽の場合でも，手牌の状態は 1 パターンしかないといえる（詳細は省略する）．従ってここでも安全性は保たれ，プロトコル全体でも安全性を満たしていると言える．

基本対称関数 S_i^n を計算するとき，図 2 に対し，**操作 B** の逆を i 回あらかじめ行ったものを，S_i^n を計算する実際の初期配置とする（S_5^{51} を計算する牌の並び順の具体的な一例を図 3 に示す．同図では，赤枠で囲われた連続する 13 枚の牌が，唯一真の出力を得られる部分である．）．そのようにすれば，出力が真となるのは **操作 B** が i 回実行されたときのみであり，これは基本対称関数の性質と一致する．従って正確性にも問題はない．

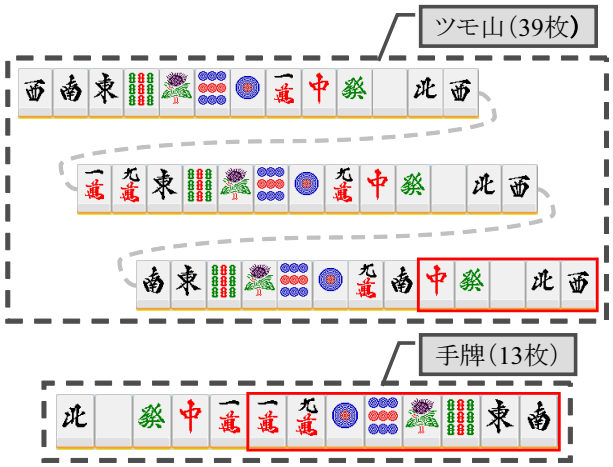

図 3: 基本対称関数 S_5^{51} を計算する牌の並び順

図 2 に示した並び順では，国士無双に使用できる牌 52 枚（13 種各 4 枚）を全て使用している．従ってこのプロトコルは 51 変数までの計算しかできない．

4　一般の関数の安全な計算

以下に示すプロトコルでは，5 変数までの一般の関数の安全な計算が行える．その初期配置は図 1 と同様にツモ山と手牌を用意するが，ツモ山は 1 列 17 枚で，2 段重ねになって伏せられており，手牌は A〜M が各 1 枚ずつの形で固定され，表向けられている．

各プレイヤー P_i は他の全てのプレイヤーに見えない状態で，1 人ずつ順番に以下のいずれかの操作を行う．

操作 A　プレイヤー P_j の持つ情報が偽であるならば，プレイヤー P_j はツモ山に対し，なにも操作を行わない．

操作 B　プレイヤー P_j の持つ情報が真であるならば，プレイヤー P_j は自分の j の値に従い，以下の操作を行う．

- ***j*=*n*-4：**　プレイヤー P_{n-4} は，2 段重ねになっているツモ山の上段と下段の牌をそのまま全て入れ替える（図 4 参照）．
- ***j*=*n*-3：**　プレイヤー P_{n-3} は，ツモ山の右端から 4 個の牌をそのままツモ山の左端に移す（図 5 参照）．この操作を 4 回繰り返し行う
- ***j*=*n*-2：**　プレイヤー P_{n-2} は，ツモ山の右端から 4 個の牌をそのままツモ山の左端に移す（図 5 参照）．この操作を 2 回繰り返し行う．
- ***j*=*n*-1：**　プレイヤー P_{n-1} は，ツモ山の右端から 4 個の牌をそのままツモ山の左端に移す（図 5 参照）．この操作を 1 回行う．
- ***j*=*n*：**　プレイヤー P_n は，ツモ山の上段右端の 1 牌をツモ山の上段左端に移す（図 6 参照）．

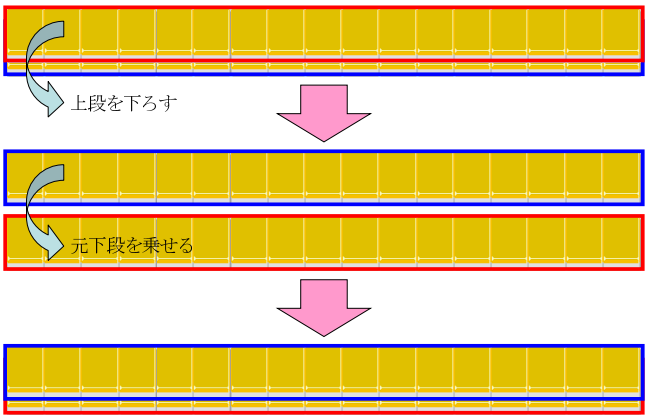

図 4: プレイヤー P_{n-4} が行う操作 B

全てのプレイヤーが各 1 回ずつ，いずれかの操作をし終えたら，ツモ山の上段右端の 1 牌を手牌に加える．これにより国士無双が完成すれば得られる出力は真，完成しなければ得られる出力は偽とする．

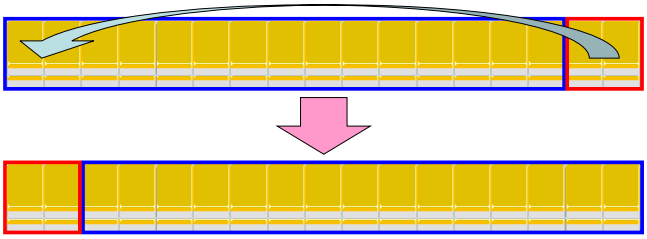

図 5: プレイヤー P_{n-3}, P_{n-2}, P_{n-1} が行う操作 B（の一部）

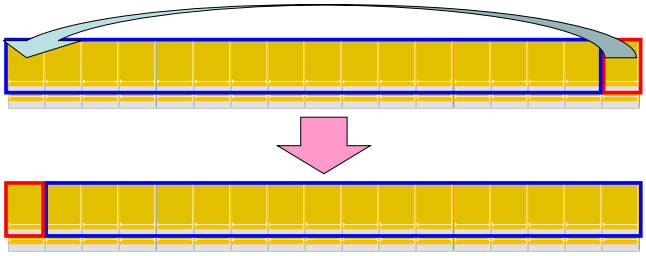

図 6: プレイヤー P_n が行う操作 B

プレイヤーがいずれの操作を行ったとしても，その前後でツモ山の枚数は変わらない．手牌は常に固定である．最後に引く牌は，待ち牌であるか，そうでないかのどちらかでしかない．従って安全性に問題はない（詳細は省略する）．

このプロトコルでは，入力パターン別に，最後に引くことになる牌が，個別に決まっている．その対応関係を図 7 に示す．但し，図中では真の入力を 1，偽の入力を 0 として 2 進数に変換した入力値を，更に 10 進数に変換して表している．ここで，左端の 2 牌は，実際にはツモられることのない，冗長な牌である．そのような牌がある理由は，プロトコル設計の方針に従い，実際の麻雀で用意される山の枚数に合わせたためである．

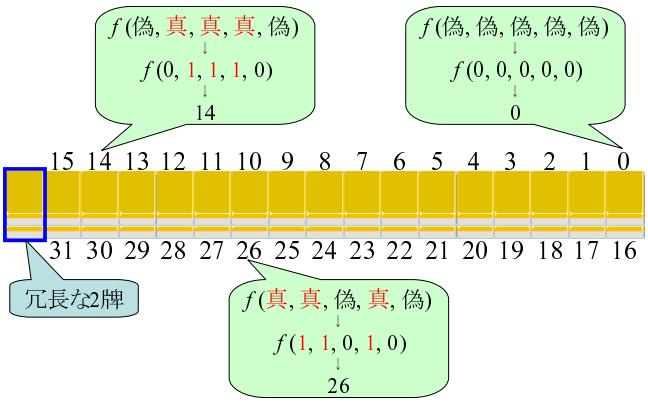

図 7: 入力パターンとツモ山の位置との対応関係

図 7 に示された対応関係に従い，計算したい関数で，出力が真となる入力パターンに対応するツモ山の牌の位置に待ち牌を置き，それ以外の全ての部分には，待ち牌でない牌を置いておく．(具体的な一例を式 (1)，表 1，図 8

に示す.）

$$f(x_1,x_2,x_3,x_4,x_5) = x_1(x_2+\overline{x_3})(\overline{x_4}+\overline{x_5}) + \overline{x_1}(x_3+x_4)(\overline{x_2}+x_5) \quad (1)$$

表 1: 5 変数の関数例（式 (1)）の真理値表

	x_1	x_2	x_3	x_4	x_5	f
0	0	0	0	0	0	0
1	0	0	0	0	1	1
2	0	0	0	1	0	0
3	0	0	0	1	1	1
4	0	0	1	0	0	1
5	0	0	1	0	1	1
6	0	0	1	1	0	1
7	0	0	1	1	1	1
8	0	1	0	0	0	0
9	0	1	0	0	1	0
10	0	1	0	1	0	0
11	0	1	0	1	1	1
12	0	1	1	0	0	0
13	0	1	1	0	1	0
14	0	1	1	1	0	1
15	0	1	1	1	1	1

	x_1	x_2	x_3	x_4	x_5	f
16	1	0	0	0	0	1
17	1	0	0	0	1	1
18	1	0	0	1	0	1
19	1	0	0	1	1	0
20	1	0	1	0	0	0
21	1	0	1	0	1	0
22	1	0	1	1	0	0
23	1	0	1	1	1	0
24	1	1	0	0	0	1
25	1	1	0	0	1	1
26	1	1	0	1	0	1
27	1	1	0	1	1	0
28	1	1	1	0	0	1
29	1	1	1	0	1	1
30	1	1	1	1	0	1
31	1	1	1	1	1	0

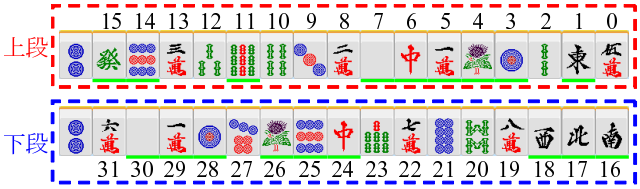

図 8: 5 変数の関数例（式 (1)）を計算する初期配置の一例

このようにすれば，関数の出力とプロトコルによって得られる出力は，常に一致する．従って正確性にも問題はない．麻雀における待ち牌の最多数は 39 枚であり，国士無双 13 面待ちの状態がこれに該当する．5 変数関数の入力パターンは 32 種類であり，39 枚の待ち牌のうち，最大でも 32 枚の牌を用いれば安全な計算が実現できる．6 変数関数の入力パターンは 64 種類であるため，このプロトコルでは 5 変数までの計算しかできない．

5 まとめ

本研究では，国士無双を用いて 51 変数までの対称関数と，5 変数までの一般の関数の安全な計算を行うプロトコルを示した．任意の論理関数に対応可能なプロトコル設計は重要であり，一方では安全な計算を行う場合，各プレイヤーが対等な立場であるという仮定は自然であるため，計算対象とする関数を対称関数と限定して議論することは妥当あると言える．

今回のプロトコル設計で注目した役は国士無双であった．この役は，指定の 13 種の牌を各 1 枚ずつと，それらのうちどれかをもう 1 枚加えた 14 枚で完成され，麻雀における役の中でもかなり特殊な形をしている．国士無双の聴牌の形は，麻雀における一般的な役の中でかなり多く，157 種類存在する．さらに，同役の特殊な聴牌形（13 面待ち状態）が全ての役の中で最大の待ち牌数を有する．これら 2 つの性質が，プロトコル実現のために使用する役として国士無双を選んだ重要な理由である．

例えば，**七対子**と呼ばれる役は，任意の 7 種の牌を各 2 枚ずつ揃えた 14 枚の牌で完成される役であり，国士無双についで特殊な形をしている．プロトコル設計において，安全性を保つためには，このようなある種の“ 特殊性 ”が重要な役割りを担う．使用する役の変更も含め，同種の牌は 4 枚までという制限を守りつつ，より多くの変数まで対応できるプロトコルの設計や，より麻雀らしい動作・状況で安全な計算を行えるプロトコルの設計などが今後の課題である．

謝辞

本稿の図中で使用した牌画像は「競技麻雀サークル・一向聴」[1] のフリー画像を使用した．この場を借りて謝意を表したい．

参考文献

[1] J. Balogh, J. A. Csirik, Y. Ishai, and E. Kushilevitz, “Private Computation Using a PEZ Dispenser,” Theoretical Computer Science, Vol. 306, pp. 69-84, 2003.

[2] T. Mizuki, Y. Kugimoto, and H. Sone, “Secure Multiparty Computations Using the 15 Puzzle (Extended Abstract),” LNCS 4616, pp. 255-266, 2007.

[3] A. Stiglic, “Computations with a deck of cards,” Theoretical Computer Science, Vol. 259, pp. 671-678, 2001.

[1] http://www.mahjong.to/