カード枚数を削減した秘匿バイナリ数拳ビンゴ

猪狩 紫雲[1]　駒野 雄一[1,a)]　水木 敬明[2]

概要：我々は DICOMO2025 において、手遊びの数拳とビンゴを組み合わせることで、ビンゴマシンを用いずともビンゴを楽しむことができる、数拳ビンゴを提案した。さらに、我々は、数拳ビンゴにカードベース暗号の技術を組み合わせることで、心理的駆け引きを楽しむことができる秘匿数拳ビンゴも提案した。しかし、秘匿数拳ビンゴではゲームに多くのカードが必要となることが課題であった。本稿では、秘匿数拳ビンゴでビンゴカードに配置される数の表記にバイナリ整数コミットメントを利用した秘匿バイナリ数拳ビンゴを提案する。さらに、バイナリ整数コミットメントを用いて数を効率よく表記する一方でゲームの面白さが保たれるように、配置される数の範囲とビンゴカードのサイズを秘匿数拳ビンゴから変更する。これらの変更により単純に比較することはできないが、秘匿バイナリ数拳ビンゴは秘匿数拳ビンゴに比べて、必要となるカード枚数やシャッフル数を削減することができる。

キーワード：ビンゴ、数拳、カードベース暗号、秘匿数拳ビンゴ

Secure Binary Suken BINGO

Ren Igari[1]　Yuichi Komano[1,a)]　Takaaki Mizuki[2]

Abstract: At DICOMO2025, we proposed Suken BINGO, by combining the hand game Suken with BINGO, that allows players to enjoy BINGO without using a BINGO machine. Furthermore, we proposed secret Suken BINGO, which combines Suken BINGO with card-based cryptography to allow players to enjoy psychological tactics. However, the secret Suken BINGO requires the large number of cards to play the game. In this paper, we propose secret binary Suken BINGO, which uses binary integer commitments to represent the numbers placed on BINGO cards in secret Suken BINGO. Furthermore, to efficiently represent numbers using the binary integer commitments while keeping the fun of the game, we modify the range of integers and the size of BINGO cards from secret Suken BINGO. Although the range of integers and the size of the bing cards are different, secret binary Suken BINGO requires fewer cards and fewer shuffles than secret Suken BINGO.

Keywords: BINGO, Suken, card-based cryptography, secure Suken BINGO

1. はじめに

ビンゴとは、ビンゴカードとランダムな数字を発生させるビンゴマシンを利用して、2 人以上のプレーヤーで遊ぶゲームである。ビンゴカードは 5×5 の 25 マスを持ち、中央を除く 24 マスに数字が書かれている。ビンゴカードに書かれている数字には規則性があり、左端の列から 1 ～ 15、16 ～ 30、31 ～ 45、46 ～ 60、61 ～ 75 の範囲の数字がランダムに書かれており、プレーヤーごとにビンゴカードの数字の並びは異なる。中央のマスは初期状態で穴があいている。ゲームでは、ビンゴマシンで乱数を一つずつ発生させ、手持ちのカードにその乱数が記載されていたら、プレーヤーはそのマスに穴をあける。縦横斜めのいずれか一列の 4 マスに穴があいたらプレーヤーは「リーチ」を宣言し、5 マスに穴があいたら「ビンゴ」を宣言する。ゲームでは、「ビンゴ」を宣言する早さを競う。

数拳とは、中国発祥のゲームである。ゲームでは二人のプレーヤーが同時に片手の指で数を示すとともに、それら

[1] 千葉工業大学
Chiba Institute of Technology
[2] 東北大学
Tohoku University
[a)] yuichi.komano@p.chibakoudai.jp

の合計数の予測値を言い合い、合計数を先に言い当てたほうが勝ちとなる。現代でも指を用いた遊びがいくつかあり、地域により呼び方は異なるが指スマなどがその例である。

我々は、文献 [14] において、ビンゴと数拳を組み合わせた新たな心理戦ゲームである**数拳ビンゴ**を提案した。数拳ビンゴでは、ビンゴマシンを用いず、プレーヤーが宣言した数の合計数を利用してビンゴを実行する。さらに、我々 [14] は、数拳ビンゴとカードベース暗号を組み合わせ、ビンゴの盤面を秘匿することで心理的な駆け引きをより楽しむことができる**秘匿数拳ビンゴ**を提案した。しかし、秘匿数拳ビンゴを実行するためには多くのカードが必要となることが課題であった。

本稿は、秘匿数拳ビンゴのカードの表記方法を変更し、カード枚数を削減した**秘匿バイナリ数拳ビンゴ**を提案する。

1.1 関連研究

本節では、カードベース暗号の技術をゲームに応用した例を紹介する。

池田ら [2] は、ヒットアンドブローを出題者なしで遊ぶ手法を与えた。また、Ruangwises ら [8] は、UNO において仮想的なプレーヤーを作るプロトコルを提案した。

他には、人狼に応用できる秘密のグループ分けプロトコル [1]、ババ抜きゲームの仮想プレーヤープロトコル [10]、秘密計算を活用した新しいカードゲーム「ガムロ」[13] が提案されている。

また、カードゲーム以外のゲームに対するカードベース暗号の応用例としては、15 パズルやルービックキューブの問題を一様ランダムに生成できるプロトコル [9] や、将棋やチェスにおいて先手後手を希望を踏まえて決定するプロトコル [12] などがある。

1.2 本稿の貢献

本稿の貢献は以下のとおり。

- 本稿では、秘匿数拳ビンゴ [14] でビンゴカードに配置される数の表記にバイナリ整数コミットメントを利用した “秘匿バイナリ数拳ビンゴ” を提案する。
- 数の表記にバイナリ整数コミットメントを利用することを踏まえて、無駄なく表記できるような数の範囲とゲームの面白さをが保たれるようなビンゴカードのサイズを考察し、ゲームの変数として設定する。
- 提案する秘匿バイナリ数拳ビンゴの安全性と効率を考察する。ビンゴカードのサイズや配置される数の範囲は異なるが、秘匿バイナリ数拳ビンゴは秘匿数拳ビンゴに比べて、ゲームに必要となるカードの枚数やシャッフルの回数を削減することができる。

本稿の構成は以下のとおり。2 節では、準備として、本稿で用いるカード組と整数の表記法、カードの基本操作について説明したのち、文献 [14] の数拳ビンゴと秘匿数拳ビンゴを復習する。3 節では、本稿で提案する “秘匿バイナリ数拳ビンゴ” について、構成のアイデアと具体的な構成、および、その変形例についてを説明する。そして、4 節で提案した秘匿バイナリ数拳ビンゴの安全性と効率を考察し、秘匿数拳ビンゴとの比較を議論する。最後に、5 節でまとめを述べる。

表 1 整数と整数コミットメントの対応表

整数	整数コミットメント
0	♡♣♣♣♣♣♣♣♣♣
1	♣♡♣♣♣♣♣♣♣♣
2	♣♣♡♣♣♣♣♣♣♣
3	♣♣♣♡♣♣♣♣♣♣
4	♣♣♣♣♡♣♣♣♣♣
5	♣♣♣♣♣♡♣♣♣♣
6	♣♣♣♣♣♣♡♣♣♣
7	♣♣♣♣♣♣♣♡♣♣
8	♣♣♣♣♣♣♣♣♡♣
9	♣♣♣♣♣♣♣♣♣♡

2. 準備

本節では、準備として、本稿で扱うカードとカードの操作、および、秘匿数拳ビンゴについて説明する。

2.1 カード

黒赤カード $\boxed{\clubsuit}\boxed{\heartsuit}$ のように、それぞれのカードの表面には ♣ か ♡ の記号が書かれており、どのカードの裏面も同じ模様 $\boxed{?}$ である。

数字カード $\boxed{1}\boxed{2}\boxed{3}\cdots\boxed{m}$ のように、それぞれのカードの表面には 1 から n までの番号が書かれており、どのカードの裏面も同じ模様 $\boxed{?}$ である。

$1 \le i \le n$ なる数字カード $\boxed{i}$ が裏返しで置かれている状態を

$$\underset{i}{\boxed{?}}$$

であらわす。

2.2 カードを用いた整数の表記

本節では、カードを用いた整数の表記法として、整数コミットメント、桁表記された整数コミットメント、バイナリ整数コミットメント [4] について説明する。

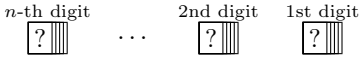
2.2.1 整数コミットメント

整数 $i \in \{0, 1, 2, \cdots, k-1\}$ を、$k-1$ 枚の黒カードと 1 枚の赤カードを用いてあらわす。これらのカードを裏返した状態を**整数コミットメント**とよぶ。整数コミットメントのカードの状態を $\boxed{?}\boxed{?}\cdots\boxed{?}$ あるいは $\boxed{?\!\!\|\!\!|}$ であらわし、$E_k(i)$ と表記する。表 1 に、整数表記の対応をまとめる。

2.2.2 桁表記された整数コミットメント

整数コミットメントを用いて、整数 k を $n(> \log_{10} k)$ 桁の 10 進数として桁表記する手法を**桁表記された整数コミッ**

トメントとよぶ。

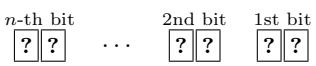


桁表記された整数コミットメントを用いることで、整数 k のコミットメントを $10n$ 枚（$O(\log_{10} k)$ 枚）のカードであらわすことができる。

秘匿数拳ビンゴ [14] では、ビンゴカードの盤面に 0 から 74 までの整数に対応するカード束（桁表記された整数コミットメント）を配置し、進行者と妨害者が宣言する数の和として 0 から 79 までの整数に対応するカード束（桁表記された整数コミットメント）を利用した。この場合には、n は 2 であり、10 の位に対応するカードは 8 枚のカードを利用すれば十分である。

2.2.3 バイナリ整数コミットメント

0 を♣♡を裏返したカード列、1 を♡♣を裏返したカード列であらわす手法を**バイナリコミットメント**とよぶ。整数 k を $n(>\log_2 k)$ 桁の 2 進数として桁表記する手法を**バイナリ整数コミットメント**とよぶ。



本稿は、バイナリ整数コミットメントを利用してカード枚数を削減した秘匿バイナリ数拳ビンゴを提案する。

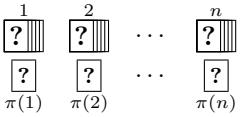
2.3 本稿で用いるカードの基本操作

次に、本稿で用いるカードの操作について説明する。

2.3.1 Pile-scramble シャッフル

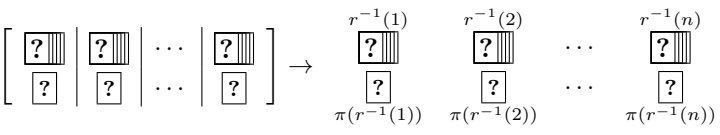
pile-scramble **シャッフル** [3] は、同じサイズのカード束がいくつかあるとき、それらの束を崩さずに（例えば輪ゴムなどで固定し）シャッフルする操作である。

例として、n 個のコミットメントと、置換 $\pi \in S_n$ に対応したカードが裏向きで次のように並んでいるとしよう。



ただし、コミットメントの上の数字は、便宜上のインデックスを表しており、S_n は n 次の対称群をあらわす。

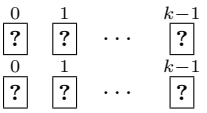
各コミットメントとその下のカードを 1 つの束として考え、pile-scramble シャッフルを適用すると、一様ランダムな置換 $r \in S_n$ が生じ、次のように遷移する。



2.3.2 Pile-shifting シャッフル

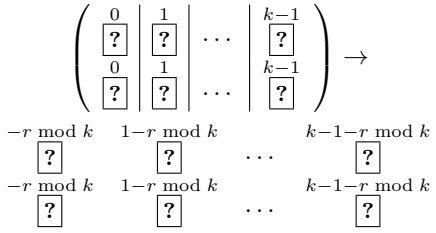
pile-shifting **シャッフル** [6], [11] は、同じサイズのカード束がいくつかあるとき、それらの束を崩さずにランダムに巡回シフトする操作である。

例として、k 枚のカードからなる 2 つのコミットメントが裏向きで次のように並んでいるとしよう。



ただし、カードの上の数字は、カードの位置を示す便宜上のインデックスとして用いている。

上下のカードを 1 つの束として考え、pile-shifting シャッフルを適用すると、ランダムな整数 r が生じてカードが右側に r 列だけ巡回シフトし、次のように遷移する。

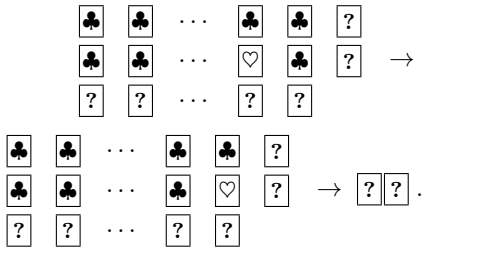


2.3.3 桁表記された数のカード束の加算プロトコル

二つの整数 x, y が $x = \sum_{j=0}^{s-1} k^j d_j^{(x)}$ と $y = \sum_{j=0}^{s-1} k^j d_j^{(y)}$ のように、それぞれ s 桁の k 進数としてあらわされ、それぞれの各桁が整数コミットメントで符号化されているものとする。このとき、それらの和を計算する加算プロトコル [14], [15] を説明する。

(1) $j = 0, 1, 2, \cdots, s-1$ として以下の操作を繰り返す。
- (a) それぞれの $(j+1)$ 桁目のカード列 $E_k(d_j^{(x)})$, $E_k(d_j^{(y)})$ の右に k 枚の♣を並べて $E_{2k}(d_j^{(x)})$, $E_{2k}(d_j^{(y)})$ とする。$(j-1)$ 回目の繰り返し処理で得られた繰り上がり（キャリー）のカード列 $E_2(c_{j-1})$ の右に $(k-2)$ 枚の♣を並べて $E_{2k}(c_{j-1})$ とする（c_{j-1} は 0 または 1 であり、$j=0$ のときの $E_{2k}(c_{-1})$ は $E_{2k}(0)$ とする）。
- (b) ステップ (1a) で得たカード列 $E_{2k}(d_j^{(x)})$, $E_{2k}(d_j^{(y)})$, $E_{2k}(c_{j-1})$ をすべて加算する。そして、加算結果のうち、k 未満に対応するカードを 1 行目に置き、k 以上に対応するカードを 2 行目に置く。その後、それらのカード列を左右反転する。
- (c) 1 行目の右に♡、2 行目の右に♣を並べて裏返す。
- (d) 行でまとめて pile-scramble シャッフルする。
- (e) 3 行目に $E_k(0)$ を並べて裏返す。
- (f) 左 k 列を pile-shifting シャッフルする。
- (g) 上から 2 行のカード列の左 k 枚を公開して♡が右端に来るように左 k 列を巡回シフトする。このとき、3 行目が $x+y$ の結果を k 進数表記した際の $(j+1)$ 桁目のコミットメントであるため、加算結果として出力する。また、1 行目と 2 行目のうち、♡を含む行の右端のカードと含まない行の右端のカードを左から順番に並べると、$(j+1)$ 桁目の加算の繰り上がり（キャリー）の計算結果 $E_2(c_j)$

となる。下図の例では、2 行目の右端のカードと 1 行目の右端のカードを、それぞれ $E_2(c_j)$ の左から 1 枚目と 2 枚目とする。公開したカードは以降の手順で再利用することができる。



(2) $(s-1)$ 回目の繰り上がり（キャリー）の計算結果 $E_2(c_{s-1})$ の右に $k-2$ 枚の ♣ を並べ、$x+y$ の結果を 10 進数表記した際の $(s+1)$ 桁目のコミットメントとして出力する。

2.3.4 バイナリ整数コミットメントの加算プロトコル

Mizuki ら [4] は、二つのバイナリコミットメント（ビット）の和を効率よく計算するための半加算器と、その半加算器を利用した全加算器を提案した。この全加算器を利用することで、二つのバイナリ整数コミットメントの和を計算することができる。

♣♡ を 0、♡♣ を 1 とするとき、二つのビット a,b のビットコミットメントを入力とする半加算器は以下のように構成される。

(1) b のビットコミットメントを、文献 [5] のコピープロトコルで二つに複製する。コピーには、2 枚ずつの黒赤のカードと、1 回のシャッフルが必要となる。
(2) $a,b,b,0$ に対応する四つのビットコミットメントを一列に並べる。
(3) 2 枚目から 6 枚目の 4 枚のカードを、1 枚分左に巡回シフトする。
(4) 8 枚のカードを左右 4 枚ずつに分けて、pile-shifting シャッフルする。
(5) 2 枚目から 6 枚目の 4 枚のカードを、1 枚分右に巡回シフトする。
(6) 左 2 枚のカードを表向きにする。
(7) 表向きのカードが ♣♡ ならば、3 枚目と 4 枚目のカード組を $a \oplus b$（和）、7 枚目と 8 枚目のカードを $a \wedge b$（繰り上がり）として出力する。あるいは、表向きのカードが ♡♣ ならば、3 枚目と 4 枚目のカードの順番を入れ替えて $a \oplus b$（和）、5 枚目と 6 枚目のカードを $a \wedge b$（繰り上がり）として出力する。

上述の半加算器は、4 枚の追加カードと 2 回のシャッフル操作が必要となる。

また、下位からの繰り上がり c のビットコミットメントと、二つのビット a,b のビットコミットメントを入力とする全加算器は、以下のように構成される。ここで、$s' = (a \oplus b) \oplus c$ は下位からの繰り上がりを含めたそのビットの和であり、$c' = (a \wedge b) \vee ((a \oplus b) \wedge c)$ は上位への繰り上がりをあらわす。

(1) 上述の半加算器を利用して、$a \wedge b, a \oplus b$ のビットコミットメントを計算する。残っているめくられていない 2 枚のカードをシャッフルし、フリーカードにする。
(2) $a \oplus b$ と c のビットコミットメントを上述の半加算器に入力し、$(a \oplus b) \wedge c$ と $s' = (a \oplus b) \oplus c$ のビットコミットメントをそれぞれ求める。
(3) $a \wedge b$ のビットコミットメントと $(a \oplus b) \wedge c$ のビットコミットメントを、文献 [5] の OR プロトコル（AND プロトコルの入力と出力をそれぞれ反転させたプロトコル）でビット和を計算し、c' のビットコミットメントを得る。

上述の全加算器は、4 枚の追加カードと 6 回のシャッフル操作が必要となる[*1]。

2.3.5 二組のカード束の一致判定

本節では、秘匿数拳ビンゴや秘匿バイナリ数拳ビンゴで用いる、二組のカード束の一致判定を行うための手順 [14] を示す[*2]。秘匿数拳ビンゴや秘匿バイナリ数拳ビンゴでは、マス目ごとにマスに対応する数のカード束が、あるカード束と一致するかを判定するとともに、一致しない場合にはマスに対応する数のカード束と別のカード束との一致判定も行う。そこで、本節では一致判定を行ったうえで、判定後に元のカード束を復元する手順も示す。

(1) 一致判定を行う二組のカードのそれぞれを、2 行のカード列となるようにカードを伏せた状態のまま並べる。
(2) 2 行のカード列の上に、第 0 行目のカードとして左から順番に数字カード 1 2 3 ··· を伏せて並べる。
(3) 3 行のカード列に対して、列方向の 3 枚のカードを一つの束とみなして、pile-scramble シャッフルを適用する。
(4) 下の 2 行のカード列（一致判定を行う二組のカード束に対応するカード）を表向きにし、カード列が一致しているか否かを判定する。
(5) 一致判定を行った後、表向きになっている下の 2 行のカード列のカードをそれぞれ伏せる。
(6) 3 行のカード列に対して、pile-scramble シャッフルを適用する。
(7) 第 0 行の数字カードを表向きにし、列方向の 3 枚のカードを一つの束とみなして、第 0 行の数字カードが 1 2 3 ··· と順番どおり並ぶように束の順番を入れ替える。
(8) 下の 2 行のカード列のそれぞれを並べ直し、一致判定を行った二つのカード束を復元する。

[*1] 文献 [16] の手法を用いると、シャッフル回数を 3 回にできる。
[*2] Ruangwises ら [7] の overwriting プロトコルを使えば実現できるが、ここではより直接的でシンプルな手順を示す。

上述のプロトコルの数字カードの代わりに、整数コミットメントや桁表記された整数コミットメント、バイナリ整数コミットメントなどのように、数に対応する黒赤カードの列を利用しても良い。

2.4 数拳ビンゴ

本節では、[14] で発表した数拳ビンゴについて説明したのち、秘匿数拳ビンゴについて紹介する。本稿では、秘匿数拳ビンゴを改良したものを提案する。

通常のビンゴカードには 1 から 75 の数字が書かれているが、[14] で説明した数拳ビンゴでは 1 を減じた 0 から 74 までの数字が書かれたビンゴカードを用いるものとする[*3]。このとき、ビンゴカードに書かれる数字は左端の列から順にそれぞれ $0 \sim 14$、$15 \sim 29$、$30 \sim 44$、$45 \sim 59$、$60 \sim 74$ の範囲となる。また、$\mathcal{D}$ を数の集合とし、本稿では $\mathcal{D} = \{0, 1, 2, 3, 4\}$ とする。

数拳ビンゴには、進行者と妨害者の二種類のプレーヤーがいる。進行者 P_i は、自身のビンゴカードに書かれた $0 \sim 74$ の数字のマスに穴があくように、数字 d_i を宣言するプレーヤーである。妨害者 P_j は、進行者のビンゴカードに穴があくことを妨害をするプレーヤー（2 名以上いてもよいが、議論を簡単にするために本節では 1 名と仮定する）であり、$\mathcal{D}$ に含まれる数から選んだ d_j を宣言する。そして、それぞれのプレーヤーは自身のビンゴカードに $d_i + d_j \pmod{75}$ が書かれていたら、そのマスに穴をあける。このような手続きを、進行者と妨害者の役割を入れ替えながら実行し、通常のビンゴと同様、ビンゴカードの縦横斜めのいずれか一列に穴がそろう早さを競う。

2.5 秘匿数拳ビンゴ

秘匿数拳ビンゴは、数拳ビンゴのビンゴカードの各マスに 0 から 74 までの桁表記された整数コミットメントを配置する。そして、進行者と妨害者は、桁表記された整数コミットメントを用いて数を宣言し、それらを加算・一致判定してビンゴカードに穴をあけてゆく。

秘匿数拳ビンゴでは宣言される数が秘匿されるために相手のビンゴカードに配置された数を知ることはできず、数拳ビンゴよりも心理的要素を高めることができる。

(1) プレーヤー P_1 と P_2 のそれぞれが、以下の手順でビンゴカードを作成する。ビンゴカードのマスに対応する数字は自身のビンゴカードに関してのみ見ることができ、相手のビンゴカードのマスに対応する数字は知ることができない。
   (a) プレーヤー P_1 は、0 から 74 までの数に対応した 75 個の桁表記された整数コミットメント（2.2.2 節）を一列に並べて pile-scramble シャッフルを適用し、左端から 25 個のカード束を 5×5 の行列状に配置する。このとき、プレーヤー P_i は自身のビンゴカードの盤面の数字（対応するカード束）を、他のプレーヤーには見せることなく確認する。残った 50 個のカード束は一枚ずつ並べて pile-scramble シャッフルを適用することで、後に再利用する。
   (b) プレーヤー P_2 は、ステップ 1(a) で再利用するカードに 50 枚の $\boxed{\heartsuit}$ と 16×25 枚の $\boxed{\clubsuit}$ を追加し、ステップ 1(a) と同様の処理を実行する。
   (c) プレーヤー P_i は、5×5 の行列上に配置されたそれぞれのカード束の隣に、そのマスに穴があいていないことを示す $\boxed{\clubsuit}$ を表向きに配置する。
(2) P_1 を進行者、P_2 を妨害者とする。
(3) いずれかがビンゴを宣言するまで、以下を繰り返す。
   (a) 進行者と妨害者は、それぞれ $0 \sim 74$ と $0 \sim 4$ のうちから数 d_1 と d_2 を選び、それぞれに対応する桁表記された整数コミットメントを机の上に並べる。
   (b) プレーヤーは、2.3.3 節の加算プロトコルを利用して、$d = d_1 + d_2 \in \{0, 1, 2, \cdots, 79\}$ に対応する桁表記された整数コミットメントを入手する。そして、75 を除数とする剰余算 [14] により、$d \pmod{75}$ の桁表記された整数コミットメントを得る。
   (c) プレーヤーはそれぞれ、自身のビンゴカードで $\boxed{\clubsuit}$ が置かれたそれぞれのマスに関して以下を実行する。
      (i) マスに置かれたカード束と、ステップ (3b) で得られた $d \pmod{75}$ に対応するカード束を、2.3.5 節の手順で一致判定する。このとき、2.3.5 節のプロトコルの性質から、一致判定を行った後にも元の二つのカード束が復元されて手元に残ることに注意しよう。
      (ii) 二つのカード束が一致している場合は、そのマスに置かれた $\boxed{\clubsuit}$ を取り除き、マスに対応する復元されたカード束と、穴があいたことを示す $\boxed{\heartsuit}$ を配置して、次のステップに移る。一致しない場合は、穴があいていないことを示す $\boxed{\clubsuit}$ のカードの隣に、マスに対応する復元されたカード束を配置し、前述のステップに戻って次のマスに対して一致判定を行う。
   (d) 二人のプレーヤーは、縦横斜めのいずれかの列で四つのマスに $\boxed{\heartsuit}$ が並んだら、リーチを宣言する。
   (e) 二人のプレーヤーは、縦横斜めのいずれかの列で五つのマスに $\boxed{\heartsuit}$ が並んだら、ビンゴを宣言する。このとき、ビンゴを宣言するプレーヤーが一人である場合にはそのプレーヤーの勝ちとなり、二人

[*3] 以降で説明する数拳ビンゴにおいて、プレーヤーが宣言し合った数の合計値（の 75 での剰余）に 1 を加えた数を合計値とみなすことで、1 から 75 が書かれた通常のビンゴカードを用いても数拳ビンゴを実行することができる。

のプレーヤーがともにビンゴを宣言する場合にはゲームは引き分けとなる。

3. 秘匿バイナリ数拳ビンゴ

本節では、カード枚数を削減した秘匿数拳ビンゴを構成するためのアイデアと、提案方式の処理手順を説明する。

3.1 アイデア

秘匿数拳ビンゴ [14] では、各マスに 0 から 74 の桁表記された整数コミットメントを配置していた。本稿で提案する秘匿バイナリ数拳ビンゴでは、ビンゴカードの各マスにバイナリ整数コミットメントを配置する。

このとき、マスに配置される整数の最大値を 2 冪の数より 1 少ない値とすると、カードを効率よく利用できる。整数の最大値の候補としては、$15 = 2^4 - 1, 31 = 2^5 - 1, 63 = 2^6 - 1, 127 = 2^7 - 1$ が考えられる。これらのうち、15 と 31 は数が小さく、ビンゴがすぐに終わってしまってゲームを楽しむことができないと考えられる。一方、最大値を 127 とすると、数が大きすぎてなかなかビンゴが成立しないと考えられる。そこで、本稿ではカードの最大値を 63 としてゲームを考える。

通常のビンゴや秘匿数拳ビンゴでは、5×5 のマスからなるビンゴカードを利用した。5×5 のビンゴカードに 0 から 63 の数を配置すると、数の配置と宣言がランダムに行われる場合[*4]、$\frac{25}{64}$ の確率でビンゴカードに穴があき、通常のビンゴや秘匿数拳ビンゴ（確率 $\frac{1}{3}$）よりもビンゴカードの穴があきやすい。そのため、穴があいたときの満足感が薄れてしまうと考えられる。そこで、本稿ではビンゴカードのサイズを 4×4 とする。

すなわち、秘匿バイナリ数拳ビンゴは 4×4 のビンゴカードを使用し、0 から 63 のバイナリ整数コミットメントを配置する。このとき、数の配置と宣言がランダムに行われる場合には、ビンゴカードに穴があく確率は $\frac{1}{4}$ となる。通常のビンゴや秘匿数拳ビンゴよりも穴があきにくくなるがゲームとして許容できる範囲であると考えられ、穴があいたときの満足感も保たれる。

秘匿バイナリ数拳ビンゴでは、進行者と妨害者が宣言した数の和の 64 での剰余を計算する。剰余は、二つのバイナリ整数コミットメントをビットごとに加算し、7 ビット目への桁上がりを無視することで効率的に計算できる。

[*4] 秘匿数拳ビンゴや秘匿バイナリ数拳ビンゴでは、進行者が自身のビンゴカードに穴があくように数を宣言し、妨害者はそれを妨害するための小さな数を宣言する。そのため、ビンゴマシンが出力する乱数に依存してビンゴカードに穴を開ける通常のビンゴとは異なり、進行者にとっては数の宣言はランダムとはならない。一方、妨害者にとっては、進行者が宣言した数に依存して自身のビンゴカードに意図せずに穴があき、偶然に数が一致する満足感を得ることができる場合がある。

3.2 秘匿バイナリ数拳ビンゴ

秘匿バイナリ数拳ビンゴは、二人のプレーヤー P_1, P_2 により、以下の手順で実行される。

(1) プレーヤー P_1 と P_2 のそれぞれが、以下の手順でビンゴカードを作成する。ビンゴカードのマスに対応する数字は自身のビンゴカードに関してのみ見ることができ、相手のビンゴカードのマスに対応する数字は知ることができない。
  (a) プレーヤー P_1 は、0 から 63 までの数に対応した 64 個のバイナリ整数コミットメント（2.2.3 節）を 1 列に並べて pile-scramble シャッフルを適用し、左端から 16 個のカード束を 4×4 の行列状に配置する。このとき、プレーヤー P_i は自身のビンゴカードの盤面の数字（対応するカード束）を、他のプレーヤーには見せることなく確認する。残った 48 個のカード束は一枚ずつ並べて pile-scramble シャッフルを適用することで、後に再利用する。
  (b) プレーヤー P_2 は、ステップ 1(a) で再利用するカードに $\boxed{\clubsuit}$ と $\boxed{\heartsuit}$ を 16 枚ずつ追加し、ステップ 1(a) と同様の処理を実行する。
  (c) プレーヤー P_i は、4×4 の行列上に配置されたそれぞれのカード束の隣に、そのマスに穴があいていないことを示す $\boxed{\clubsuit}$ を表向きに配置する。
(2) P_1 を進行者、P_2 を妨害者とする。
(3) いずれかがビンゴを宣言するまで、以下を繰り返す。
  (a) 進行者と妨害者は、それぞれ $0 \sim 63$ と $0 \sim 4$ のうちから数 d_1 と d_2 を選び、それぞれに対応するバイナリ整数コミットメントを机の上に並べる。
  (b) プレーヤーは、2.3.4 節の加算プロトコルを利用し、7 ビット目への桁上がりを無視することで $d = (d_1 + d_2 \bmod 64) \in \{0, 1, 2, \cdots, 63\}$ に対応するバイナリ整数コミットメントを入手する。
  (c) プレーヤーはそれぞれ、自身のビンゴカードで $\boxed{\clubsuit}$ が置かれたそれぞれのマスに関して以下を実行する。
    (i) マスに置かれたカード束と、ステップ (3b) で得られた d に対応するカード束を、2.3.5 節の手順で一致判定する。このとき、2.3.5 節のプロトコルの性質から、一致判定を行った後にも元の二つのカード束が復元されて手元に残ることに注意しよう。
    (ii) 二つのカード束が一致している場合は、そのマスに置かれた $\boxed{\clubsuit}$ を取り除き、マスに対応する復元されたカード束と、穴があいたことを示す $\boxed{\heartsuit}$ を配置して、次のステップに移る。一致しない場合は、穴があいていないことを示す $\boxed{\clubsuit}$ のカードの隣に、マスに対応する復元されたカード束を配置し、前述のステップに

戻って次のマスに対して一致判定を行う。

(d) 二人のプレーヤーは、縦横斜めのいずれかの列で三つのマスに$\boxed{\heartsuit}$が並んだら、リーチを宣言する。

(e) 二人のプレーヤーは、縦横斜めのいずれかの列で四つのマスに$\boxed{\heartsuit}$が並んだら、ビンゴを宣言する。このとき、ビンゴを宣言するプレーヤーが一人である場合にはそのプレーヤーの勝ちとなり、二人のプレーヤーがともにビンゴを宣言する場合にはゲームは引き分けとなる。

3.3 変形例

3.2 節では、4×4 のビンゴカードを使用し、0 から 63 のバイナリ整数コミットメントを配置した秘匿バイナリ数拳ビンゴを提案した。これに対し、文献 [14] で述べたゲーム変数に応じた変形例を考えることができる。本節では、特に二つの点に着目した変形例について説明する。

3.3.1 カードのサイズと配置する数の最大値の変更

3.1 節で述べたように、ビンゴカードに配置する整数の最大値の候補として $127=2^7-1$ も考えられる。ここで、5×5 のビンゴカードを用いる場合には、数の配置と宣言がランダムに行われるとすると、ビンゴカードに穴があく確率は $\frac{25}{128}\approx\frac{1}{5}$ となる。

数の最大値は 127 として、6×6 のビンゴカードを用いる場合には、ビンゴカードに穴があく確率は $\frac{36}{128}\approx\frac{1}{3.6}$ となり、通常のビンゴ（および秘匿数拳ビンゴ、確率 $\frac{1}{3}$）と 3.2 節の秘匿バイナリ数拳ビンゴ（確率 $\frac{1}{4}$）の中間程度となる。

3.3.2 ビンゴの判定条件の変更

通常のビンゴや秘匿数拳ビンゴでは、5×5 のビンゴカードを用いた。そのため、中央のマスをあけることでビンゴの可能性を高めることができた。

一方、秘匿バイナリ数拳ビンゴでは行と列の数が偶数であり、縦・横・斜めの要となるマスが存在しない。そこで、通常の斜めの列に加えて（あるいは、それに代えて）、ある指定したマスを通る縦・横・斜めのいずれかの列に穴が揃ったらビンゴが成立したとすることもできる。

例えば、1 行目の 4 マスを (1,1),(1,2),(1,3),(1,4)、2 行目の 4 マスを (2,1),(2,2),(2,3),(2,4) のようにあらわすとき、(1,2) が指定されたなら、縦：{(1,2),(2,2),(3,2),(4,2)}、横：{(1,1),(1,2),(1,3),(1,4)}、斜め：{(2,1),(1,2),(4,3),(3,4)}、斜め：{(4,1),(1,2),(2,3),(3,4)} のいずれかに穴が揃ったらビンゴが成立するとしてもよい。

4. 考察

3.2 節で述べた秘匿バイナリ数拳ビンゴの安全性と効率を評価し、秘匿数拳ビンゴと比較する。

4.1 安全性

本節では、秘匿バイナリ数拳ビンゴを実行すれば正しくゲームの勝者を決定することができること（完全性）、ゲームを実行する過程で穴があいたマス以外のマスに関して[*5]ビンゴカードのマスに対応する数の情報が漏れないこと（機密性）がなりたつことを示す。

完全性：秘匿バイナリ数拳ビンゴは、ステップ (1a) のカード束の並べ方から、各マスには異なる数に対応したカード束が配置される（同じ数字のカード束を複数のマスに不正に配置することはできない）。このようにカード束を配置したビンゴカードに対して、カードベース暗号の加算や一致判定プロトコルなどの特性から、プレーヤーが宣言した数に基づいてカードに正しく穴をあけることができ、ゲームを正しく実行できる。また、一致判定の結果に基づいて穴のあいた状態を示すカードを公開の状態で操作することで、ゲームの勝者も正しく判定することができる。

機密性：秘匿バイナリ数拳ビンゴでは、宣言した数の加算と、ビンゴカードの盤面に記載された数の一致判定をカードベース暗号の技術を利用して実行する。使用するカードベース暗号の技術（加算や一致判定など）の安全性から、宣言した数やビンゴカードに書かれた数の情報は漏れないことが保証できるため、秘匿バイナリ数拳ビンゴの機密性も保証される。

4.2 効率

秘匿バイナリ数拳ビンゴの基本方式の効率を評価する。

秘匿バイナリ数拳ビンゴに必要となるカードは、ステップ (1) で $12\times64+12\times16$ 枚のカードが必要となる[*6]。そのため、参加人数を 2 とするとき、合計 960 枚のカードが必要となる。

プロトコルで必要なシャッフル数は、ステップ (1) で 2× (参加人数) 回、ステップ (3b) で 36 回、ステップ 3(c-i) で 2× (チェックするマス目数) × (参加人数) 回である。ステップ (3) はゲームが終了するまで繰り返され、チェックするマス目の数は高々 16 であるため、シャッフルの回数の合計は高々 $4+100\times$ (繰り返し回数) となる。

4.3 秘匿数拳ビンゴとの比較

秘匿バイナリ数拳ビンゴと秘匿数拳ビンゴ [14] を比較した結果を、表 2 に示す[*7]。

表 2 の「マス目」はビンゴカードのサイズ、「数字」はビンゴカードに配置される数字の範囲、「数字の表記」はビンゴカードに配置する数字の表記方法をあらわす。「カード枚数」は、ゲームを実行するために必要となるカードの枚

[*5] 秘匿バイナリ数拳ビンゴを実行する際に、妨害者のビンゴカードのマスに穴があいた場合は、進行者はそのマスに対応する数が自ら宣言した数の近辺の値である（宣言した数 +4 の範囲の数である）という情報を入手することができる。

[*6] ステップ (3) の二組のカード束の一致判定では、数字カードの代わりにステップ (1a) で破棄したカードを再利用する。

[*7] 本稿では秘匿数拳ビンゴの効率も再評価した。

表 2 秘匿数拳ビンゴと秘匿バイナリ数拳ビンゴの比較

方式	マス目	数字	数字の表記	カード枚数	シャッフル数
秘匿数拳ビンゴ	5×5	$0 \sim 74$	桁表記整数コミットメント	1800	$10 + 110\ell$
秘匿バイナリ数拳ビンゴ	4×4	$0 \sim 63$	バイナリ整数コミットメント	960	$4 + 100\ell'$

数をあらわす。「シャッフル数」は、ゲーム中に実行されるシャッフルの回数をあらわす、ℓ と ℓ' は秘匿数拳ビンゴと秘匿バイナリ数拳ビンゴで数字を宣言する繰り返し処理の回数である。

ビンゴカードのサイズや配置される数字の範囲が異なるために単純に比較することはできないが、秘匿バイナリ数拳ビンゴは秘匿数拳ビンゴに対してカード枚数とシャッフル数を削減することができる。一方、ステップ 1(a) で（後に数を宣言する際に参考とするために）自身のビンゴカードの数字を確認するとき、桁表記された整数コミットメントを用いる秘匿数拳ビンゴに比べて、バイナリ整数コミットメントから 10 進数への変換が必要となる。

5. まとめ

本稿は、秘匿数拳ビンゴ [14] の数の表記にバイナリ整数コミットメントを利用し、ビンゴカードのサイズや配置される数字の範囲を調整した秘匿バイナリ数拳ビンゴを提案し、安全性と効率を考察した。ビンゴカードのサイズや配置される数の範囲は異なるが、秘匿バイナリ数拳ビンゴは秘匿数拳ビンゴに比べてゲームに必要となるカードの枚数やシャッフルの回数を削減することができる。

謝辞: 本研究は JSPS 科研費（JP24K14951, JP24K02938, JP23H00479）の助成を受けたものです。文献 [14] の発表に対して有益なコメントをいただきました須賀祐治氏に感謝いたします。

参考文献

[1] Yuji Hashimoto, Kazumasa Shinagawa, Koji Nuida, Masaki Inamura, and Goichiro Hanaoka. Secure grouping protocol using a deck of cards. *IEICE Trans. Fundam.*, E101.A(9):1512–1524, 2018.

[2] Shota Ikeda and Kazumasa Shinagawa. How to play Mastermind without game master. In *Theory and Applications of Models of Computation*, LNCS, Cham, 2025, to appear. Springer.

[3] Rie Ishikawa, Eikoh Chida, and Takaaki Mizuki. Efficient card-based protocols for generating a hidden random permutation without fixed points. In Cristian S. Calude and Michael J. Dinneen, editors, *Unconventional Computation and Natural Computation*, volume 9252 of *LNCS*, pages 215–226, Cham, 2015. Springer.

[4] Takaaki Mizuki, Isaac Kobina Asiedu, and Hideaki Sone. Voting with a logarithmic number of cards. In Giancarlo Mauri, Alberto Dennunzio, Luca Manzoni, and Antonio E. Porreca, editors, *Unconventional Computation and Natural Computation*, volume 7956 of *LNCS*, pages 162–173, Berlin, Heidelberg, 2013. Springer.

[5] Takaaki Mizuki and Hideaki Sone. Six-card secure AND and four-card secure XOR. In Xiaotie Deng, John E. Hopcroft, and Jinyun Xue, editors, *Frontiers in Algorithmics*, volume 5598 of *LNCS*, pages 358–369, Berlin, Heidelberg, 2009. Springer.

[6] Akihiro Nishimura, Yu-ichi Hayashi, Takaaki Mizuki, and Hideaki Sone. Pile-shifting scramble for card-based protocols. *IEICE Trans. Fundam.*, 101(9):1494–1502, 2018.

[7] Suthee Ruangwises, Tomoki Ono, Yoshiki Abe, Kyosuke Hatsugai, and Mitsugu Iwamoto. Card-based overwriting protocol for equality function and applications. In Da-Jung Cho and Jongmin Kim, editors, *Unconventional Computation and Natural Computation*, volume 14776 of *LNCS*, pages 18–27, Cham, 2024. Springer.

[8] Suthee Ruangwises and Kazumasa Shinagawa. Simulating virtual players for UNO without computers. In *Unconventional Computation and Natural Computation*, LNCS, Cham, 2025, to appear. Springer.

[9] Kazumasa Shinagawa, Kazuki Kanai, Kengo Miyamoto, and Koji Nuida. How to covertly and uniformly scramble the 15 puzzle and rubik's cube. In Andrei Z. Broder and Tami Tamir, editors, *Fun with Algorithms*, volume 291 of *LIPIcs*, pages 30:1–30:15, Dagstuhl, Germany, 2024. Schloss Dagstuhl.

[10] Kazumasa Shinagawa, Daiki Miyahara, and Takaaki Mizuki. How to play old maid with virtual players. In Bo Li, Minming Li, and Xiaoming Sun, editors, *Frontiers of Algorithmics*, volume 14752 of *LNCS*, pages 53–65, Singapore, 2025. Springer.

[11] Kazumasa Shinagawa, Takaaki Mizuki, Jacob Schuldt, Koji Nuida, Naoki Kanayama, Takashi Nishide, Goichiro Hanaoka, and Eiji Okamoto. Card-based protocols using regular polygon cards. *IEICE Trans. Fundam.*, E100.A(9):1900–1909, 2017.

[12] Yuto Shinoda, Daiki Miyahara, Kazumasa Shinagawa, Takaaki Mizuki, and Hideaki Sone. Card-based covert lottery. In Diana Maimut, Andrei-George Oprina, and Damien Sauveron, editors, *Innovative Security Solutions for Information Technology and Communications*, volume 12596 of *LNCS*, pages 257–270, Cham, 2021. Springer.

[13] Mizuki Takaaki, Kuzuma Tomoki, Hirano Tomoya, Oshima Ririn, and Yasuda Momofuku. Gakmoro: An application of physical secure computation to card game. In ???, editor, *Unconventional Computation and Natural Computation*, volume ??? of *LNCS*, pages ???–???, Cham, 2025. Springer.

[14] 猪狩 紫雲, 駒野 雄一, 水木 敬明. 数拳ビンゴ. **マルチメディア、分散、協調とモバイル** *DICOMO2025* **シンポジウム**,*6H-3*, 2025.

[15] 猪狩 紫雲, 小高 駿, 駒野 雄一, 水木 敬明. カードを用いる桁表記された整数コミットメントと加算プロトコル. *2025* **年暗号と情報セキュリティシンポジウム** *(SCIS 2025), 3D2-3*, 2025.

[16] 島津大, 品川和雅. 効率的な全加算プロトコルとその加算プロトコルへの応用. *2025* **年暗号と情報セキュリティシンポジウム** *(SCIS 2025),3D1-2*, 2025.