Vol.2022-AL-189 No.5
2022/9/15

UNOを用いた数独に対するゼロ知識証明について*

田中 滉大[1,a)]　水木 敬明[1]

概要：数独に対するゼロ知識証明とは，数独の解を知る証明者がその解についての情報を一切明かすことなく，「解が存在し，証明者が解を知っていること」を検証者に納得させるというものである．2009 年に Gradwhol らは数独に対する初の物理的なゼロ知識証明プロトコルをカード組を用いて構築した．しかしこのプロトコルは健全性エラーが発生する問題がある．その後 2020 年に Sasaki らは健全性エラーがなく，36 回のシャッフルで実現できる改良型のゼロ知識証明プロトコルを提案した．このプロトコルでは 1 から 9 の数字が書かれた同一カードが 18 セット必要であり，より日常的に用意できる道具を用いることが望ましい．そこで，2022 年に Ruangwises は，標準的なトランプ 2 セットを用いたプロトコルを提案した．しかし，このプロトコルは 322 回のシャッフルが必要であり，実用的に利用するのは難しい．そこで本稿では，日常的に用意可能である UNO を 2 セット用い，新しいプロトコルを構築し，シャッフル回数の削減を実現する．具体的には，16 回のシャッフルで数独に対するゼロ知識証明を実現する．従って，提案プロトコルは前述の Sasaki らのプロトコルよりもシャッフル回数が半分以下で済む，非常に効率的なものである．

1. はじめに

数独 (Sudoku) とは，代表的な論理パズルの 1 つである．標準的な数独は 3×3 の 9 つのブロックに区切られた，9×9 の格子で構成されている．一部のセルには既に 1 から 9 の数字が埋まっており，残りの空いているセルに各行，各列，各ブロックに 1 から 9 の数字がちょうど 1 つずつ現れるように数字を置いていくことを目的とする．図 1 にパズル例題とその解を示す．

この数独を一般化したものは $\sqrt{n}\times\sqrt{n}$ の n 個ブロックに区切られた $n\times n$ の格子で構成されており，空いているセルに各行，各列，各ブロックに 1 から n の数字がちょうど 1 つずつ現れるように数字を置いていくことを目的とする．ただし，n は完全平方数である．上述の標準的な数独は $n=9$ に対応する．

1.1 数独に対するゼロ知識証明

本稿では，数独に対するゼロ知識証明を扱う [5,6,17,18,23,24]．数独に対するゼロ知識証明とは，ある数独パズルに対してその解を知る証明者 P が解を知らない検証者 V に対して，「解が存在し，証明者 P が解を知っていること」を解についての情報を一切明かすことなく検証者 V に納得させるというものである．ゼロ知識証明は以下の 3 つの性質を満たす必要がある．

完全性　証明者 P が解を知っているならば，検証者 V は常に納得する．

健全性　証明者 P が解を知らないならば，検証者 V は常に拒否する．

ゼロ知識性　証明者 P の持つ命題が真であるならば，検証者 V はその命題が真であること以外の何の知識も得られない．すなわち，検証者 V は解に関する情報を一切得ない．

また，本稿で扱うゼロ知識証明はコンピュータ等の電子デバイスを一切用いず，物理的なカード組を用い，人間の手でプロトコルを実行する物理的なゼロ知識証明である．物理的なプロトコルの利点として，プロトコルが正しく実行されていることを確認しやすいこと，ゼロ知識証明の概念を知らない非専門家も理解しやすいという教育的価値などが挙げられる．

1.2 既存プロトコル

2009 年に Gradwhol ら [6] は数独に対する物理的なゼロ知識証明プロトコルを初めて開発した．しかし，このプロトコルでは健全性エラーが発生し，健全性エラー確率を無視できるほど小さくするためには何度もプロトコルを繰り返さなければならない．また，健全性エラー確率を 0 にするための別のプロトコルを提案しているが，スクラッチカードという消耗品を使用しなければならない．

*数独 (Sudoku) と UNO は，それぞれ Nikoli Co., Ltd. と Mattel, Inc. の商標あるいは登録商標である．

[1] 東北大学
Tohoku Uniersity
[a)] kodai.tanaka.r2@dc.tohoku.ac.jp

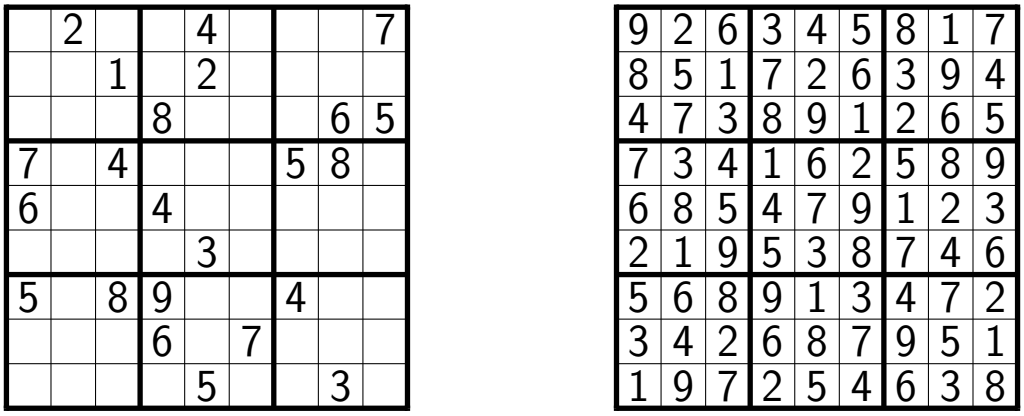


図 1: 標準的な数独の例題とその解

その後 2020 年に Sasaki ら [23] は健全性エラーのないプロトコルを 3 種類提案した．中でもプロトコルの途中で証明者の知識を必要とせずシャッフル回数が少ないのは，プロトコル B と名付けられたものであり，表 1 の通り，標準的な数独に対して 36 回のシャッフルでゼロ知識証明を実現できる．このプロトコルでは，1 から 9 の数字が書かれたカード 1 2 ··· 9 をそれぞれ 18 枚に加え，任意の相異なる 9 枚のカードが必要となっている．仮にこのプロトコルをトランプで実装するには，同一のトランプが 18 セット必要である．もし UNO を用いる場合には，9 セット必要である．

2022 年には Ruangwises [18] が標準的なトランプを用いたプロトコルを 2 種類提案した．これらはプロトコルの途中で証明者の知識を必要とする．表 1 の通り，提案プロトコルの 1 つである Method B ではトランプ 2 セットという日常的に用意可能な道具で標準的な数独に対するゼロ知識証明が実行可能である．すなわち，$13 \times 4 = 52$ 枚のカード

♣A ♢A ♡A ♠A ♣2 ♢2 ♡2 ♠2 ··· ♣K ♢K ♡K ♠K

とジョーカー 2 枚（異なる模様の 2 枚）を 1 セットとして，このようなトランプを 2 セット用意すれば，プロトコルを実行することができる．しかし，必要なシャッフル回数が 322 回であり多い．またシャッフル回数が 108 回である Method A もあるが，トランプ 2 セットでは実装できず，シャッフル回数も 100 回以上であり多い．

1.3 本稿の貢献

本稿では，日常的に用意可能な UNO を 2 セット用意することで実行可能であり，かつシャッフル回数が少なく，健全性エラーを持たないゼロ知識証明プロトコルを提案する．具体的には，黄色の数字カード Y1 Y2 ··· Y9 をそれぞれ 4 枚，赤色の数字カード R1 R2 ··· R9 をそれぞれ 3 枚，青色の数字カード B1 B2 ··· B9 をそれぞれ 3 枚，加えて任意の相異なる 27 枚のカードを用い，必要なシャッフル回数は 16 回であり，プロトコルの途中で証明者の知識は必要としない．既存プロトコルとの比較を表 1 に示す．

また，本提案プロトコルは一般化された $n \times n$ の数独にも適用可能であり，使用カード枚数とシャッフル回数を表 2 に示す．

1.4 関連研究

数独以外のパズルや問題に対してもこれまで数多く物理的ゼロ知識証明プロトコルが構築されている．例を挙げると次の通りである．美術館 (Akari) [1]，覆面算 (Cryptarithmetic) [8]，橋をかけろ (Hashiwokakero) [22]，ひとりにしてくれ (Hitori) [15]，縦横さん (Juosan) [12]，カックロ (Kakuro) [1, 13]，賢くなるパズル (KenKen) [1]，マカロ (Makaro) [2]，ましゅ (Masyu) [9]，ののぐらむ (Nonogram) [3, 16]，のりのり (Norinori) [4]，ナンバーリンク (Numberlink) [19, 20]，ぬりかべ (Nurikabe) [15]，波及効果 (Ripple Effect) [21]，スリザーリンク (Slitherlink) [9, 10]，バイナリーパズル (Takuzu) [1, 12]．

2. 準備

本節では，提案プロトコルで使用するカードやシャッフル操作について説明する．

2.1 使用するカード組

本稿の提案プロトコルで使用するカードについて述べる．以下では，9×9 の数独に対して使用するカードについて説明する（一般化数独に対するプロトコルで使用するカードについては 5 節で述べる）．

まず数独の解を表現するために使用するカードとして，表面が赤，青，黄色のカードでかつ数字 1 から 9 のいずれかが書かれており，裏面は全て同一であるものを用いる（これらをエンコーディングカードと呼ぶ）．カードの例を以下に示す．

表面: R1 Y6 B9　裏面: ? ? ?

具体的には前述の通り，黄色の数字カード Y1 Y2 ··· Y9 を

表 1: 提案プロトコルと既存プロトコルとの比較

プロトコル	カード枚数	シャッフル回数	実装可能なカード種
Sasaki ら（B）[23]	171	36	トランプ 18 セット UNO 9 セット
Ruangwises（Method B）[18]	108	322	トランプ 2 セット
提案プロトコル	117	16	UNO 2 セット

表 2: 一般化された数独でのカード枚数とシャッフル回数

プロトコル	カード枚数	シャッフル回数
Sasaki ら（B）[23]	$2n^2+n$	$4n$
Ruangwises（Method B）[18]	$n^2+2n+3\sqrt{n}$	$2n^2(\sqrt{n}-1)$ (for even n) $2n^2(\sqrt{n}-1)+2$ (for odd $n>9$)
提案プロトコル	$n^2+n\sqrt{n}$ (for even n) $n^2+n+n\sqrt{n}$ (for odd n)	$7\sqrt{n}-5$

それぞれ 4 枚，赤色の数字カード R1 R2 ⋯ R9 をそれぞれ 3 枚，青色の数字カード B1 B2 ⋯ B9 をそれぞれ 3 枚用いる．

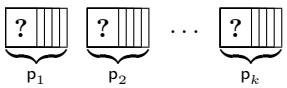
またエンコーディングカードとは別に，表面が異なり，裏面が同一であるカードを 27 枚用いる（これらを追加カードと呼ぶ）．

UNO は 1 セットにつき赤，青，黄色が 2 セットずつ含まれているので，UNO を 2 セット用意することで上述の必要カードを満たす．なぜなら，エンコーディングカードで必要な分を取り除いても，それ以外に表面の異なるカードが 27 種類含まれており，追加カードとして使えるからである．

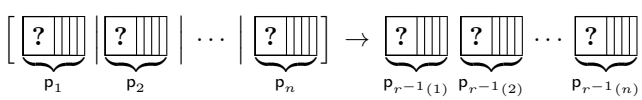
2.2 パイルスクランブルシャッフル

本稿の提案プロトコルで使用するシャッフル操作について説明する．

複数枚のカードを重ねた束が k 個存在し，各束の枚数は全て同じであるとし，各束を $\mathsf{p}_i\,(1 \le i \le k)$ で表す．



パイルスクランブルシャッフル [7] は，カードの束の列にランダムな置換 $r \in S_k$ を適用する操作であり，$[\cdot|\cdots|\cdot]$ で表す．



ここで，S_k は k 次対称群を表す．

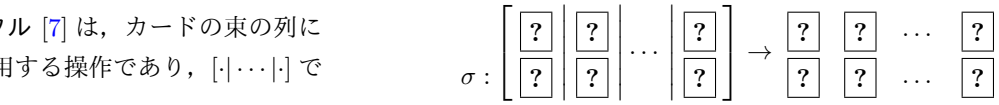
パイルスクランブルシャッフルを実際に実現するには，各束のカードをスリーブや封筒などに入れて，それらを（元の並びが分からなくなるまで）かき混ぜることで実装可能である．

2.3 一意性検証プロトコル

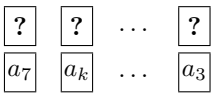
2.2 節で述べたパイルスクランブルシャッフルを使用した**一意性検証プロトコル**について説明する．このプロトコルは Sasaki ら [23, 24] によって提案されている．

このプロトコルを用いると，ある異なる k 枚からなるカード列 $\sigma = (a_1, a_2, \ldots, a_k)$ が裏向きの状態で与えられたとき，σ が $\{a_1, a_2, \ldots, a_k\}$ で構成されていることを，その順序を明かすことなく検証することができる．$(a_1, a_2, \ldots, a_k)$ とは別に異なる k 枚のカード $(x_1, x_2, \ldots, x_k)$ を用意し，以下の手順を実行する（公衆の面前で任意の誰かが実行する）．

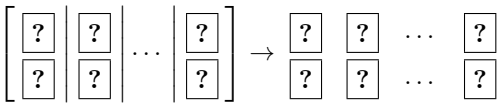
(1) カード列 $\sigma = (a_1, a_2, \ldots, a_k)$ の上に，k 枚のカード $(x_1, x_2, \ldots, x_k)$ を裏向きにして順番通りに並べる．次のように各列を束として，2.2 節で述べたパイルスクランブルシャッフルを適用する．



(2) 下段のカード列を表にし，$\{a_1, a_2, \ldots, a_k\}$ で構成されていることを確認する．



(3) 下段のカードを裏向きにし，パイルスクランブルシャッフルを適用する．



上段のカードを表にし，左から $x_1, x_2, \ldots, x_k$ という順番になるようにカード束を並べ替える．これにより，カード列 σ が元の順番のまま復元される．

このプロトコルにより，ある裏向きに置かれたカード列がどのカードで構成されているかを順序を明かすことなく，かつ順序を崩さずに検証することができる．本稿のプロトコルでも，この一意性検証プロトコルをサブプロトコルとして活用する．

3. サブプロトコル

本節では，提案プロトコルで用いるいくつかのサブプロトコルを構築する．

以下では，9×9 の数独を用いて説明するが，一般的な $n \times n$ の数独にも適用可能である（ただし，操作は同様に行えるが，n の値によっては使用カードが一般的な UNO では実現できない）．また，9×9 の数独の 3×3 ブロックを図 2 のように $A, B, \ldots, I$ と呼ぶことにする．

3.1 証明者によるカードの配置

ある 9×9 の数独パズルに対して，その解を知っている証明者 P はまず以下の操作を行う．

(1) P はすでに数字が書かれているセルに対応する数字のカードを裏向きに置く．ただし，ブロック A, B, C には黄色のカード，ブロック D, E, F には赤のカード，ブロック G, H, I には青のカードを置く（本稿ではこの配色で置いたが，同色のカードが横 3 ブロックごとに置かれていれば，任意の配色でもプロトコルは正しく実行できる）．

(2) P は自身の解に従って，残りの空いているセルにもカードを前項と同じ配色で裏向きに置く．

3.2 色検証サブプロトコル

このサブプロトコルでは，横 3 ブロックが同色であることを検証する．以下の操作を実行する．

(1) 追加カード 27 枚を用意し，横 3 ブロックに対して 2.3 節で述べた一意性検証プロトコルを応用し，横 3 ブロックのカード 27 枚が全て同色であることと，1 から 9 の番号がそれぞれ 3 枚ずつ出現することを確認する（例えば，ブロック A, B, C に対して適用すると，ブロック A, B, C のカード 27 枚が黄色の 1〜9 番それぞれ 3 枚で構成されていることを確認する）．そうでない場合には棄却される．

3.3 3 行（列）同時検証サブプロトコル

ブロック A, B, C には黄色のカード，ブロック D, E, F には赤のカード，ブロック G, H, I には青のカードが裏向きに置かれているとする．このサブプロトコルでは，（異なる色の）3 行（列）がそれぞれ 1 から 9 で構成されていることを検証する．以下では，1, 4, 7 行目の 3 行に対して検証を行う場合を例に取り，操作を説明する．

(1) 1 行目，4 行目及び 7 行目のカードをすべて取り出し，その 27 枚のカード列に対して 2.3 節で述べた一意性検証プロトコルを適用し，27 枚が黄色，赤，青の 1 から 9 の各 1 枚ずつで構成されていることを確認する．（それぞれの行が異なる色である前提なので，各行は 1 から 9 で構成されていることになる．）そうでない場合には棄却される．

他の（異なる色の）3 行や，3 列にも同様に適用できる．

3.4 色変更サブプロトコル

このサブプロトコルでは 3×3 ブロックの内のカードの数字を変えることなく，色のみを変える．以下では，黄色のカードで構成されたプロックと赤色のカードで構成されたプロックがあるとして，操作を説明する．以下の手順により，各ブロックが 1 から 9 の数字で構成されているならば，色だけが交換される（そうでない場合には棄却される）．

(1) 18 枚の追加カードの列 $(x_1, x_2, \ldots, x_{18})$ を用意し，2 つのブロックのカードの上に裏向きに並べる．これらにパイルスクランブルシャッフルを適用する．

(2) エンコーディングカードを表にする．1 から 9 の黄色カードと赤色カードが漏れなく出現していることを確認し（そうでない場合には棄却する），同じ数字同士のカードを入れ替える（例えば，黄色の 3 と赤の 3 の位置を交換する）．

(3) エンコーディングカードを裏向きにし，パイルスクランブルシャッフルを適用する．追加カードを表にし，左から $x_1, x_2, \ldots, x_{18}$ の順番となるように並べ替える．エンコーディングカードを元の位置に戻す．

この手順を拡張すると，黄色のブロック，赤色のブロック，青色のブロックが与えられたとき，追加カード 27 枚とフリーの Y1 Y2 ... Y9 を用いて，黄色と赤色のブロックの色を交換し，青色のブロックを黄色に変更することがパイルスクランブルシャッフル 2 回により可能である．またこのとき同時に，各ブロックが 1 から 9 の番号の数字で構成されていることが確認できる．

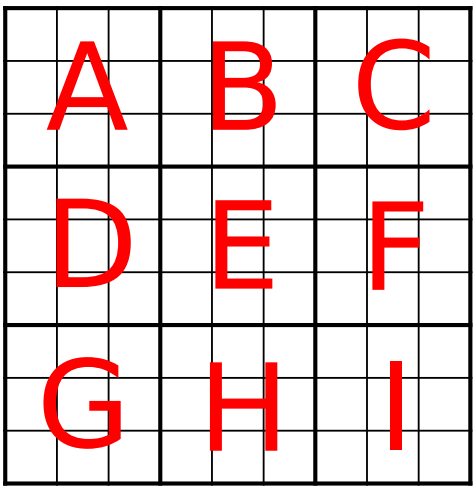


図 2: 分割した 3 × 3 ブロック

4. 9 × 9 数独に対する提案プロトコル

本節では，9 × 9 数独に対するゼロ知識証明を実現するプロトコルを構築する．4.1 節でプロトコルの手順を示し，4.2 節で必要なカード枚数やシャッフル回数について言及し，4.3 節で正当性を述べる．

4.1 プロトコルの手順

ここでは，9 × 9 数独に対する提案プロトコルの手順を述べる．解を知っている証明者 P は，3.1 節で述べたようにカードを裏向きに配置し，これをプロトコルへの入力とする．入力が与えられたあとは，証明者 P の知識は必要とせず，以降は誰が実行してもよい，いわゆる**非対話型**物理的ゼロ知識証明 [11] となっている．

追加カード 27 枚とフリーの Y1 Y2 ··· Y9 を用意し（UNO 2 セットの中で用意可能である），以下の手順により，検証者 V に各行，各列，各ブロックに 1 から 9 の数字がちょうど 1 回ずつ現れることを納得させる．

(1) 3.2 節で与えた色検証サブプロトコルをブロック A, B, C，ブロック D, E, F，ブロック G, H, I に適用する．棄却されない場合，ブロック A, B, C の 27 枚は Y1 Y2 ··· Y9 がそれぞれ 3 枚で構成され，ブロック D, E, F は R1 R2 ··· R9 がそれぞれ 3 枚で構成され，ブロック G, H, I は B1 B2 ··· B9 がそれぞれ 3 枚で構成されていることになる．

(2) 3.3 節で与えた 3 行同時検証サブプロトコルを 1, 4, 7 行目の 3 行に適用する．また，2, 5, 8 行目の 3 行に対しても 3 行同時検証サブプロトコルを適用する．棄却されない場合，1, 2, 4, 5, 7, 8 行目はそれぞれ 1 から 9 番の数字で構成されていることになる．加えて，前ステップの検証と合わせると，残りの 3, 6, 9 行目も自動的にそれぞれ 1 から 9 番の数字で構成されていることになる．

(3) 列の検証を行えるようにするため，図 3 のように，ブロック A, D, G が黄色，ブロック B, E, H が赤色，ブロック C, F, I が青色となるように変更したい．そこで 3.4 節で述べた色変更サブプロトコルをブロック B, D, G に対して適用する．（このときフリーの B1 B2 ··· B9 が生じる．）同様にして，ブロック C, F, H に対しても色変更サブプロトコルを適用する．これらの色変換の後は，ブロック B, C, D, F, G, H それぞれについて，1 から 9 番の数字で構成されていることが検証される．加えて，ステップ 1 の検証結果を踏まえると，残りのブロック A, E, I についても自動的に 1 から 9 番の数字で構成されていることになる．

(4) 3.3 節で述べた 3 列同時検証サブプロトコルを 1, 4, 7 列目の 3 列に適用する．なお，このとき，取り出した 3 列を復元する必要はないので，2.3 節で述べた一意性検証プロトコルのカードを戻す操作は行わない（すなわち，シャッフル回数は 1 回となる）．同様の操作を 2, 5, 8 列目の 3 列に対しても行う．行の場合と同様に，残りの 3, 6, 9 列目も自動的に検証される．

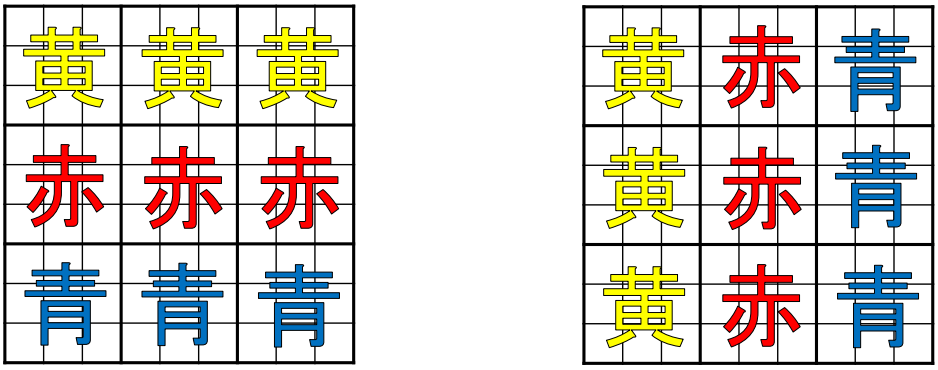
以上が提案プロトコルの手順である．

4.2 使用カード枚数とシャッフル回数

ここでは，4.1 節で提案したプロトコルおいて使用するカード枚数とシャッフル回数を述べる．

使用カード枚数は 117 枚であり，その内訳は次の通りである．

- エンコーディングカード： 81 + 9 = 90 枚（+9 枚はス



(a) 色変更プロトコル適用前　(b) 色変更プロトコル適用後

図 3: 色変更前と変更後のブロックに対する配色

テップ 3 におけるブロック G の分の黄色カード）

- 追加カード： 27 枚

シャッフル回数は 16 回であり，その内訳は次の通りである．

- ステップ 1：色検証サブプロトコル（シャッフル 2 回必要）を 3 回実行するので，合計 6 回
- ステップ 2：行検証サブプロトコル（シャッフル 2 回必要）を 2 回実行するので，合計 4 回
- ステップ 3：色変更サブプロトコル（シャッフル 2 回必要）を 2 回実行するので，合計 4 回
- ステップ 4：列検証サブプロトコル（シャッフル 1 回必要）を 2 回実行するので，合計 2 回

4.3 プロトコルの正当性

ここでは，提案プロトコルが完全性，健全性，ゼロ知識性を満たすことを述べる．

完全性：入力として正しい数独の解に対応したカードが置かれた場合，プロトコルの構成から明らかなように，途中で棄却されることなく，プロトコルが終了する．従って，完全性を満たす．

ゼロ知識性：提案プロトコルは非対話型の物理的ゼロ知識証明プロトコルであり，入力が証明者によって置かれたあとは，カードベース暗号の計算モデル [11, 14] に従い動作し，カード列をめくる前にはパイルスクランブルシャッフルが適用されているため入力に関する情報は一切漏れず，情報理論的に安全である．従って，ゼロ知識性を満たす．

健全性：数独の解に正しく対応せずに入力が置かれた場合（横 3 ブロックが同色ではない，各行，各列，各 3×3 ブロックに同じ数字が少なくとも 2 回以上現れている）について以下の通りとなり，健全性を満たす．

- 横 3 ブロックが同色ではない場合，ステップ 1 により棄却される．
- 各行に同じ数字が少なくとも 2 回以上現れる場合，ステップ 2 により棄却される．
- 各 3×3 ブロックに同じ数字が少なくとも 2 回以上現れる場合，ステップ 3 により棄却される．
- 各列に同じ数字が少なくとも 2 回以上現れる場合，ステップ 4 により棄却される．

5. 一般化数独に対する提案プロトコル

前節で提案したプロトコルは，一般化された $n \times n$ の数独に対応できるように容易に拡張できる．本節では，一般化された数独にプロトコルを拡張する場合の使用カードとシャッフル回数について説明する．

5.1 使用カード

一般化数独の場合，以下のカードを用意する必要がある．

- エンコーディングカード：$n^2 + n$ 枚（$+n$ は n が奇数のときのみ色変更サブプロトコル時に使用）．
- 追加カード：$n\sqrt{n}$ 枚

エンコーディングカードにおける $+n$ 枚はエンコーディングカードとして用意した $\sqrt{n}$ 色の中の 1 色のみさらに 1 から n までそれぞれ 1 枚，計 n 枚用意するということである．また，n が奇数のときのみであるのは，4 節のステップ 3 のようにエンコーディングカード同士で交換できないブロックがあるからである．n が偶数のときは全てのブロックがエンコーディングカード同士で交換可能であるため，必要としない．

5.2 シャッフル回数

シャッフル回数は $7\sqrt{n} - 5$ 回であり，その内訳は以下の通りである．

- ステップ 1：色検証サブプロトコル（シャッフル 2 回必要）を $\sqrt{n}$ 回実行するので， 合計 $2\sqrt{n}$ 回
- ステップ 2：行検証サブプロトコル（シャッフル 2 回必要）を $(\sqrt{n} - 1)$ 回実行するので，合計 $2(\sqrt{n} - 1)$ 回
- ステップ 3：色変更サブプロトコル（シャッフル 2 回必要）を $(\sqrt{n} - 1)$ 回実行するので，合計 $2(\sqrt{n} - 1)$ 回

- ステップ 4：列検証サブプロトコル（シャッフル 1 回必要）を $(\sqrt{n}-1)$ 回実行するので，合計 $\sqrt{n}-1$ 回

6. おわりに

本稿では，数独に対する新たな物理的ゼロ知識証明プロトコルを提案した．提案プロトコルでは，9×9 の数独に対して，日常的に用意可能である UNO を用い，カードの色という特徴を利用することで既存プロトコルよりも少ないシャッフル回数での実現を可能とした．具体的には，UNO を 2 セットとシャッフルを 16 回で実行できる．

また，一般的な $n\times n$ の数独に対しても同様の手法を用いることで少ないシャッフル回数での実現が可能である．完全平方数 n に対する $n\times n$ の数独におけるシャッフル回数を図 4 に示す．図の通り一般的な数独に対しても提案プロトコルはシャッフル回数を抑えている．一方で，提案プロトコルでは色の特徴を用いているので，使用カードを $\sqrt{n}$ 色用意しなければならず，n が大きくなるにつれ，実用が難しくなる．

今後の課題として，番号付きスリーブを用いたさらなるシャッフル回数とカード枚数の削減，また，他の論理パズルに対するより効率的な物理的ゼロ知識証明プロトコルの検討に取り組む．

謝辞 本研究は，JSPS 科研費 JP21K11881 の助成を受けたものである．

参考文献

[1] Bultel, X., Dreier, J., Dumas, J.-G. and Lafourcade, P.: Physical Zero-Knowledge Proofs for Akari, Takuzu, Kakuro and KenKen, *Fun with Algorithms* (Demaine, E. D. and Grandoni, F., eds.), LIPIcs, Vol. 49, Dagstuhl, Germany, Schloss Dagstuhl, pp. 8:1–8:20 (online), available from ⟨https://doi.org/10.4230/LIPIcs.FUN.2016.8⟩ (2016).

[2] Bultel, X., Dreier, J., Dumas, J., Lafourcade, P., Miyahara, D., Mizuki, T., Nagao, A., Sasaki, T., Shinagawa, K. and Sone, H.: Physical Zero-Knowledge Proof for Makaro, *Stabilization, Safety, and Security of Distributed Systems*, LNCS, Vol. 11201, pp. 111–125 (online), available from ⟨https://doi.org/10.1007/978-3-030-03232-6_8⟩ (2018).

[3] Chien, Y.-F. and Hon, W.-K.: Cryptographic and Physical Zero-Knowledge Proof: From Sudoku to Nonogram, *Fun with Algorithms* (Boldi, P. and Gargano, L., eds.), LNCS, Vol. 6099, Berlin, Heidelberg, Springer, pp. 102–112 (online), available from ⟨https://doi.org/10.1007/978-3-642-13122-6_12⟩ (2010).

[4] Dumas, J.-G., Lafourcade, P., Miyahara, D., Mizuki, T., Sasaki, T. and Sone, H.: Interactive Physical Zero-Knowledge Proof for Norinori, *Computing and Combinatorics* (Du, D.-Z., Duan, Z. and Tian, C., eds.), LNCS, Vol. 11653, Cham, Springer, pp. 166–177 (online), available from ⟨https://doi.org/10.1007/978-3-030-26176-4_14⟩ (2019).

[5] Gradwohl, R., Naor, M., Pinkas, B. and Rothblum, G. N.: Cryptographic and Physical Zero-Knowledge Proof Systems for Solutions of Sudoku Puzzles, *Fun with Algorithms* (Crescenzi, P., Prencipe, G. and Pucci, G., eds.), LNCS, Vol. 4475, Berlin, Heidelberg, Springer, pp. 166–182 (online), available from ⟨https://doi.org/10.1007/978-3-540-72914-3_16⟩ (2007).

[6] Gradwohl, R., Naor, M., Pinkas, B. and Rothblum, G. N.: Cryptographic and Physical Zero-Knowledge Proof Systems for Solutions of Sudoku Puzzles, *Theory of Computing Systems*, Vol. 44, No. 2, pp. 245–268 (online), available from ⟨https://doi.org/10.1007/s00224-008-9119-9⟩ (2009).

[7] Ishikawa, R., Chida, E. and Mizuki, T.: Efficient Card-Based Protocols for Generating a Hidden Random Permutation Without Fixed Points, *Unconventional Computation and Natural Computation* (Calude, C. S. and Dinneen, M. J., eds.), LNCS, Vol. 9252, Cham, Springer, pp. 215–226 (online), available from ⟨https://doi.org/10.1007/978-3-319-21819-9_16⟩ (2015).

[8] Isuzugawa, R., Miyahara, D. and Mizuki, T.: Zero-Knowledge Proof Protocol for Cryptarithmetic Using Dihedral Cards, *Unconventional Computation and Natural Computation* (Kostitsyna, I. and Orponen, P., eds.), LNCS, Vol. 12984, Cham, Springer, pp. 51–67 (online), available from ⟨https://doi.org/10.1007/978-3-030-87993-8_4⟩ (2021).

[9] Lafourcade, P., Miyahara, D., Mizuki, T., Robert, L., Sasaki, T. and Sone, H.: How to Construct Physical Zero-Knowledge Proofs for Puzzles with a "Single Loop" Condition, *Theor. Comput. Sci.*, Vol. 888, pp. 41–55 (online), available from ⟨https://doi.org/10.1016/j.tcs.2021.07.019⟩ (2021).

[10] Lafourcade, P., Miyahara, D., Mizuki, T., Sasaki, T. and Sone, H.: A Physical ZKP for Slitherlink: How to Perform Physical Topology-Preserving Computation, *Information Security Practice and Experience* (Heng, S.-H. and Lopez, J., eds.), LNCS, Vol. 11879, Cham, Springer, pp. 135–151 (online), available from ⟨https://doi.org/10.1007/978-3-030-34339-2_8⟩ (2019).

[11] Miyahara, D., Haneda, H. and Mizuki, T.: Card-Based Zero-Knowledge Proof Protocols for Graph Problems and Their Computational Model, *Provable and Practical Security* (Huang, Q. and Yu, Y., eds.), LNCS, Vol. 13059, Cham, Springer, pp. 136–152 (online), available from ⟨https://doi.org/10.1007/978-3-030-90402-9_8⟩ (2021).

[12] Miyahara, D., Robert, L., Lafourcade, P., Takeshige, S., Mizuki, T., Shinagawa, K., Nagao, A. and Sone, H.: Card-Based ZKP Protocols for Takuzu and Juosan, *Fun with Algorithms* (Farach-Colton, M., Prencipe, G. and Uehara, R., eds.), LIPIcs, Vol. 157, Dagstuhl, Germany, Schloss Dagstuhl, pp. 20:1–20:21 (online), available from ⟨https://doi.org/10.4230/LIPIcs.FUN.2021.20⟩ (2020).

[13] Miyahara, D., Sasaki, T., Mizuki, T. and Sone, H.: Card-based Physical Zero-knowledge Proof for Kakuro, *IEICE Trans. Fundam.*, Vol. 102, No. 9, pp. 1072–1078 (online), available from ⟨https://doi.org/10.1587/transfun.E102.A.1072⟩ (2019).

[14] Mizuki, T. and Shizuya, H.: A formalization of card-based cryptographic protocols via abstract machine, *Int. J. Inf. Secur.*, Vol. 13, No. 1, pp. 15–23 (online), available from ⟨https://doi.org/10.1007/s10207-013-0219-4⟩ (2014).

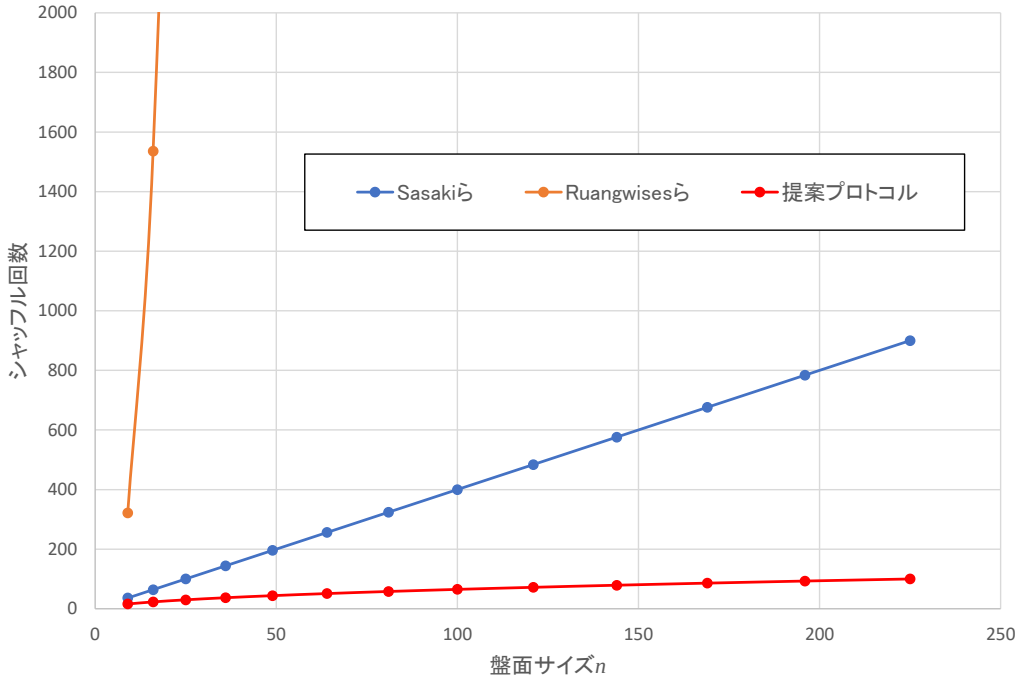


図 4: $n \times n$ 数独に対する既存プロトコルと提案プロトコルのシャッフル回数

[15] Robert, L., Miyahara, D., Lafourcade, P. and Mizuki, T.: Interactive Physical ZKP for Connectivity: Applications to Nurikabe and Hitori, *Connecting with Computability* (De Mol, L., Weiermann, A., Manea, F. and Fernández-Duque, D., eds.), LNCS, Vol. 12813, Cham, Springer, pp. 373–384 (online), available from ⟨https://doi.org/10.1007/978-3-030-80049-9_37⟩ (2021).

[16] Ruangwises, S.: An Improved Physical ZKP for Nonogram, *Combinatorial Optimization and Applications* (Du, D.-Z., Du, D., Wu, C. and Xu, D., eds.), LNCS, Vol. 13135, Cham, Springer, pp. 262–272 (online), available from ⟨https://doi.org/10.1007/978-3-030-92681-6_22⟩ (2021).

[17] Ruangwises, S.: Two Standard Decks of Playing Cards Are Sufficient for a ZKP for Sudoku, *Computing and Combinatorics* (Chen, C.-Y., Hon, W.-K., Hung, L.-J. and Lee, C.-W., eds.), LNCS, Vol. 13025, Cham, Springer, pp. 631–642 (online), available from ⟨https://doi.org/10.1007/978-3-030-89543-3_52⟩ (2021).

[18] Ruangwises, S.: Two Standard Decks of Playing Cards are Sufficient for a ZKP for Sudoku, *New Gener. Comput.*, Vol. 40, pp. 49–65 (online), available from ⟨https://doi.org/10.1007/s00354-021-00146-y⟩ (2022).

[19] Ruangwises, S. and Itoh, T.: Physical Zero-Knowledge Proof for Numberlink, *Fun with Algorithms* (Farach-Colton, M., Prencipe, G. and Uehara, R., eds.), LIPIcs, Vol. 157, Dagstuhl, Germany, Schloss Dagstuhl, pp. 22:1–22:11 (online), available from ⟨https://doi.org/10.4230/LIPIcs.FUN.2021.22⟩ (2020).

[20] Ruangwises, S. and Itoh, T.: Physical Zero-Knowledge Proof for Numberlink Puzzle and k Vertex-Disjoint Paths Problem, *New Gener. Comput.*, Vol. 39, No. 1, pp. 3–17 (online), available from ⟨https://doi.org/10.1007/s00354-020-00114-y⟩ (2021).

[21] Ruangwises, S. and Itoh, T.: Physical Zero-Knowledge Proof for Ripple Effect, *WALCOM: Algorithms and Computation* (Hong, S., Nandy, S. and Uehara, R., eds.), LNCS, Vol. 11737, Cham, Springer, pp. 296–307 (online), available from ⟨https://doi.org/10.1007/978-3-030-68211-8_24⟩ (2021).

[22] Ruangwises, S. and Itoh, T.: Physical ZKP for Connected Spanning Subgraph: Applications to Bridges Puzzle and Other Problems, *Unconventional Computation and Natural Computation* (Kostitsyna, I. and Orponen, P., eds.), Cham, Springer, pp. 149–163 (2021).

[23] Sasaki, T., Miyahara, D., Mizuki, T. and Sone, H.: Efficient card-based zero-knowledge proof for Sudoku, *Theor. Comput. Sci.*, Vol. 839, pp. 135–142 (online), available from ⟨https://doi.org/10.1016/j.tcs.2020.05.036⟩ (2020).

[24] Sasaki, T., Mizuki, T. and Sone, H.: Card-Based Zero-Knowledge Proof for Sudoku, *Fun with Algorithms* (Ito, H., Leonardi, S., Pagli, L. and Prencipe, G., eds.), LIPIcs, Vol. 100, Dagstuhl, Germany, Schloss Dagstuhl, pp. 29:1–29:10 (online), available from ⟨https://doi.org/10.4230/LIPIcs.FUN.2018.29⟩ (2018).