盗まれたネックレス問題に対するカードベースゼロ知識証明

池田 昇太[1,a)]　品川 和雅[2,3,b)]

概要：二人の泥棒があるネックレスを盗んだとする。そのネックレスには複数種類の宝石が使われており、彼らはこれらの宝石を種類ごとに均等に分配したい。決められた回数の切断で、各種類の宝石を等しく分配することはできるだろうか。この問題は盗まれたネックレス問題と呼ばれる。本研究では、盗まれたネックレス問題に対してカードを用いたゼロ知識証明プロトコルを構成する。提案プロトコルのシャッフル回数は 3 回であり、これはネックレスのサイズに依らない定数である。

キーワード：カードベース暗号、ゼロ知識証明、盗まれたネックレス問題

Card-Based Zero-Knowledge Proofs for Necklace Splitting Problem

Shota Ikeda[1,a)]　Kazumasa Shinagawa[2,3,b)]

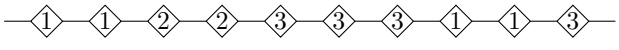
Abstract: Suppose that two thieves have stolen a necklace. The necklace contains several different types of jewels, and the thieves want to divide them equally among themselves. Can each type of jewel be divided equally with a fixed number of cuts? This problem is known as the necklace splitting problem. In this study, we present a zero-knowledge proof protocol using cards to solve the necklace splitting problem. The proposed protocol requires three shuffles, a constant that is independent of the size of the necklace.

Keywords: Card-based cryptography, Zero-Knowledge Proofs, Necklace splitting problem

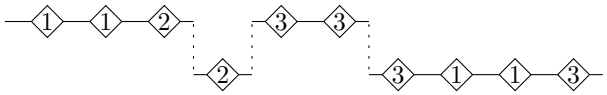
1. はじめに

1.1 盗まれたネックレス問題

3 種類の宝石（ルビー、エメラルド、サファイア）が無作為に並べられた以下のようなネックレスがあったとする。



ただし、上図で $1, 2, 3$ はそれぞれルビー、エメラルド、サファイアに対応する。このネックレスでは各種類の宝石の個数はすべて偶数個であり、ネックレスのワイヤーを何箇所か切断することにより 2 人で宝石を均等に分配したい。もちろん、宝石間のすべてのワイヤーを切断すれば均等に分配できるが、できるだけ少ない切断回数にしたい。このネックレスの場合、以下のような 3 回の切断で十分である。



この場合、上側を 1 人目が、下側を 2 人目が受け取れば、それが均等な分配である。

この問題は盗まれたネックレス問題あるいはネックレス分割問題と呼ばれており、n 種類の宝石のネックレスの場合は高々 n 回の切断で均等な分割が可能であることが証明されている [2, 7]。しかし、これらの証明は構成的証明ではなく、具体的な切断を求めるための効率的なアルゴリズムは知られていない。Ratsikas–Goldberg [6] はネックレス分割問題において n 回以下の具体的な切断を求めることは PPA（Polynomial Parity Argument）完全であることを証明した。また、Alon–West による証明 [2] に関しては YouTube チャンネル「3Blue1BrownJapan」の動画『Borsuk-Ulam の定理とネックレス問題 〜トポロジーでパ

[1] 茨城大学 Ibaraki University
[2] 筑波大学 University of Tsukuba
[3] 産業技術総合研究所 National Institute of Advanced Industrial Science and Technology
[a)] 25nm709x@vc.ibaraki.ac.jp
[b)] shinagawa@cs.tsukuba.ac.jp

ズルを解く〜』[1] を参考にされたい。PPA 完全な問題は多項式時間で解くことはできないことが予想されており、したがって構成的なネックレス分割問題も多項式時間で解くことはできないことが予想される。

1.2 カードベースゼロ知識証明

ゼロ知識証明 [8] とは、証明者 P が検証者 V に対し、ある主張が正しいことを納得させるためのプロトコルである。その最大の特徴は、証明の過程で『主張が正しい』という事実以外のいかなる情報も検証者 V に与えない点にある。通常のゼロ知識証明は計算機上での実行を想定するが、カードや封筒などの身近な物理的道具を用いてゼロ知識証明を構成する研究も行われており、カードベースゼロ知識証明と呼ばれている。カードベースゼロ知識証明は、身近な道具を用いるため非専門家にとっても直感的に理解しやすく、暗号技術への入り口としての教育的な価値を有している。

カードベースゼロ知識証明の研究では、パズルに対するゼロ知識証明が中心的に研究されている。これは、パズルの答えを知っている証明者 P が、答えを知らない検証者 V に対して、答えそのものを明かすことなく、P がパズルの答えを知っていることだけを納得させるプロトコルである。特に、数独 [9,10,17,18,20,21] やカックロ [4,13] やスリザーリンク [12] などのペンシルパズルについての研究が盛んである。

パズルに対するゼロ知識証明は、題材が親しみやすいため、ゼロ知識証明の入門的説明に適していると考えられている。その際に、国・地域・個人によってそれぞれ好きなパズルや知っているパズルが異なるため、さまざまなパズルに対するプロトコルが存在することによって、暗号教育の間口を広げることができる。また、パズルごとに実行容易さや説明のしやすさが異なるため、さまざまなパズルに対するプロトコルを研究することにより、教育応用に適したパズルが解明されることが期待される。また理論的な観点から見ると、パズルに対するゼロ知識証明プロトコルの効率性（カード枚数やシャッフル回数など計算複雑性に関わる指標）は、「そのパズルの検証条件のシンプルさ」を特徴づけている。一般に「解が与えられたときその正しさの検証は容易だが、解くのは困難なパズル」は美しいパズルの一つの条件とされる。この意味で、ゼロ知識証明プロトコルの効率性は、そうした観点における一つの尺度を提供していると解釈することもできる。

1.3 貢献

本稿では 2 人が宝石を分配する場合の「盗まれたネックレス問題」に対するカードベースゼロ知識証明プロトコルを二つ提案する。これは、ネックレスの正しい分割方法を知る証明者 P が、その具体的な切断位置を一切明かすことなく、検証者 V に対し「c 回の切断によって各種類の宝石を等しく分配できる」という事実を納得させるためのプロトコルである。

本稿が提案するプロトコルは、著者らが知る限り、探索問題の計算量クラスである PPA 完全に属する問題に対する初のカードベースゼロ知識証明である。本研究は、カードベースゼロ知識証明の対象を NP 完全問題から PPA 完全問題へと拡張するものであり、扱う計算量クラスの幅を広げる観点から教育的・学術的に高い価値を有するものと考えられる。

提案する二つのプロトコルはそれぞれ異なる利点を持つ。提案プロトコル 1 は、アルゴリズムの正当性が直感的に理解しやすい基本的な構成である。このプロトコルでは、2 色カード $4N$ 枚、数字カード N 枚を使用して、シャッフルはランダム二等分割カットを N 回、パイルスクランブルシャッフルを 2 回行う。一方、提案プロトコル 2 は、実用性に秀でた極めて効率的なプロトコルとして位置づけられる。このプロトコルは、2 色カード $2N$ 枚、数字カード $2N$ 枚を使用して、シャッフルはパイルスクランブルシャッフルを 3 回行う。このプロトコルの最大の特徴は、シャッフル回数が宝石の総数 N や種類数 n に依存しない定数回（3 回）である点だ。また、カード枚数も提案プロトコル 1 に比べて N 枚少なく、カード枚数に関しても効率的である。

2. 準備

本章ではプロトコルで使用するカードとシャッフル、サブプロトコルについて解説する。また、本稿で扱う盗まれたネックレス問題の問題設定について述べる。

2.1 使用するカード

本稿ではカードベース暗号 [3,5] で主に用いられている 2 色カードと数字カードを使用する。2 色カードは黒 $\boxed{\clubsuit}$ と赤 $\boxed{\heartsuit}$ の 2 種類のカードを用いて、1 ビットの情報を $\boxed{\clubsuit}\boxed{\heartsuit} = 0$、$\boxed{\heartsuit}\boxed{\clubsuit} = 1$ として符号化する。このとき、同じ色のカードは互いに区別できないものとし、特に裏向きに置かれたカード $\boxed{?}$ は表面の色に関わらず区別できない。この符号化に従って裏向きに置かれたカード組 $\boxed{?}\boxed{?}$ をコミットメントと呼ぶ。コミットメントは左右のカードを入れ替えるだけで、否定（NOT）をすぐに実現できるという特徴がある。このコミットメントを用いた符号化は提案プロトコル 1（3.1 節）で用いる。また、2 色カード 1 枚で行う符号化も存在し、1 枚符号化 [15,19] と呼ばれる。1 枚符号化では、1 ビットの情報を $\boxed{\clubsuit} = 0$、$\boxed{\heartsuit} = 1$ として符号化する。この符号化は提案プロトコル 2（3.2 節）で用いる。数字カードは $\boxed{1}\boxed{2}\cdots$ のように表面に数字が描かれ、裏面に $\boxed{?}$ が描かれているカードである。数字カードも 2 色カードと同様に裏向きに置かれたカードは表面の数字に関わらず区別できない。

表 1 盗まれたネックレス問題に対するカードベースゼロ知識証明プロトコル

プロトコル名	カード枚数（枚）		シャッフル回数（回）	
	2 色	数字	RBC	PSS
提案プロトコル 1（3.1 節）	$4N$	N	N	2
提案プロトコル 2（3.2 節）	$2N$	$2N$	0	3

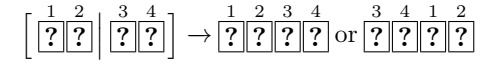
RBC はランダム二等分割カット、PSS はパイルスクランブルシャッフル、N は宝石の総数を表す。

2.2 シャッフル

本節ではプロトコルで用いるランダム二等分割カットとパイルスクランブルシャッフルの二つのシャッフルを解説する。

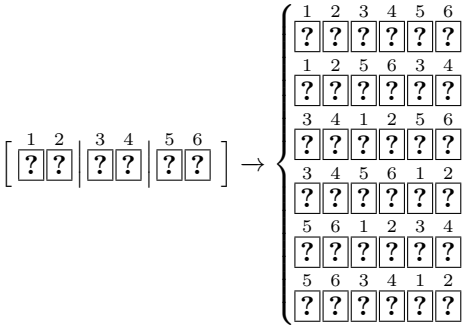
2.2.1 ランダム二等分割カット（RBC）

ランダム二等分割カット [14] は、カード列を半分に分け、その後 1/2 の確率でそのまま、残りの 1/2 の確率で左右を入れ替える操作である。また、シャッフルを適用した後、どちらになったかは誰にもわからない。カード列にランダム二等分割カットを適用することを $[\,\cdot\,|\,\cdot\,]$ と表記する。以下に例として、4 枚のカード列に対してランダム二等分割カットを適用した様子を示す。



2.2.2 パイルスクランブルシャッフル（PSS）

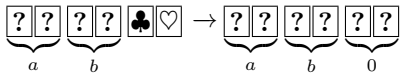
パイルスクランブルシャッフル [11] はカード列を同じ枚数からなる複数の束に分け、各束の中にあるカードの並びを保ったまま、束同士の順番を一様ランダムに並べ替える操作である。すなわち、i 番目のカード束を p_i、n 個のカード束 $(p_1, p_2, \ldots, p_n)$ とするとき、カード列に対してパイルスクランブルシャッフルを適用すると、一様ランダムな置換 $\pi \in S_n$ に対して $(p_{\pi^{-1}(1)}, p_{\pi^{-1}(2)}, \ldots, p_{\pi^{-1}(n)})$ が得られる。ここで、S_n は n 次対称群を表す。また、カード列にパイルスクランブルシャッフルを適用することを $[\,\cdot\,|\,\cdot\,|\cdots|\,\cdot\,]$ と表記する。以下に例として、各 2 枚からなる計 3 個の束に対してパイルスクランブルシャッフルを適用した様子を示す。



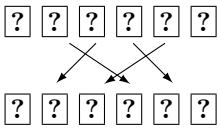
2.3 Mizuki–Sone の XOR プロトコル

Mizuki–Sone の XOR プロトコル [14] は 2 入力のコミット型 XOR プロトコルである。このプロトコルはシャッフルとしてランダム二等分割カット（2.2.1 節）を 1 回使う。さらに、このプロトコルは 0 のコミットメントを k 個追加することで、a のコミットメントのコピーを k 個作ることができる。XOR 計算をせず、コピーのみを目的とする場合は、コピープロトコルと呼ぶ。以下に 2 つの入力コミットメントと 0 のコミットメント 1 つから、$a \oplus b$ と a のコピーのコミットメントを出力する手順を示す。

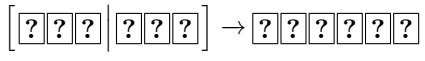
(1) 2 つのコミットメントを以下のように置く。



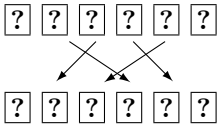
(2) カード列を以下のように並べ替える。



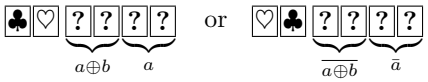
(3) ランダム二等分割カットを行う。



(4) カード列を以下のように並べ替える。



(5) 左端 2 枚のカードをめくる。その結果によって、$a \oplus b$ と a のコピーのコミットメントが以下のように得られる。



2.4 盗まれたネックレス問題の問題設定

本節では、盗まれたネックレス問題の問題設定を述べる。ネックレスには n 種類の宝石 $J_1, \ldots, J_n$ が含まれており、宝石 J_i はちょうど $2a_i$ 個ずつ存在すると仮定する $(1 \le i \le n, a_i \in \mathbb{N})$。したがって、ネックレス全体に含まれる宝石の総数は $N = 2\sum_{i=1}^{n} a_i$ 個である。宝石はあらかじめ無作為な順序で並べられており、ネックレスは輪状ではなく、一直線状のワイヤーに通された構造を持つものとする。このネックレスを切断し、切断された領域を 2 人が交互に取り合うことで各種類の宝石を正確に半分ずつに分割することが目標である。

証明者 P は、宝石と宝石の間の位置から選ばれる c 箇所を切断することで、各種類の宝石を等しく分割できること

を知っている。一方、検証者 V はその切断位置を知らない。証明者 P の目的は、切断位置そのものを明かすことなく、「c 回の切断によって各種類の宝石を正確に二等分できる」という主張が正しいことだけを、検証者 V に納得させることである。

3. 提案プロトコル

本章では二つの提案プロトコルについて実行手順と正当性について解説する。

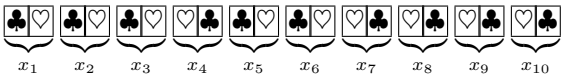
3.1 提案プロトコル 1

提案プロトコル 1 は二つの検証フェーズで構成されている。一つ目は**切断数検証フェーズ**である。切断数検証フェーズでは、証明者 P が「c 回の切断を行った」ことを検証者 V に切断位置に関する情報を一切明かさずに示す。二つ目は**等分配検証フェーズ**である。等分配検証フェーズでは、証明者 P が「切断が各種類の宝石を等分した」ことを検証者 V に切断位置に関する情報を一切明かさずに示す。提案プロトコル 1 は切断数検証フェーズを行ってから、切断検証フェーズを行うプロトコルである。

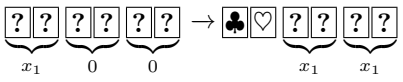
3.1.1 切断数検証フェーズの実行手順

本節では、提案プロトコル 1 における切断数検証フェーズの手順を解説する。

(1) 左から i 番目（$1 \le i \le N$）の宝石が Alice の宝石なら $x_i = 0$、Bob の宝石なら $x_i = 1$ と定め、証明者 P は $x_1, x_2, \cdots, x_N$ のコミットメントを入力する。ここで、0 と 1 の個数はそれぞれ $\frac{N}{2}$ 個であり、示すべきことは $x_i \neq x_{i-1}$ となる i が c 個であることである。例として 1.1 節のネックレスで証明者 P が正しく解を置いた場合は以下のようになる。



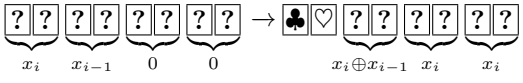
(2) x_1 のコミットメントに対して以下の Mizuki–Sone の XOR プロトコル（2.3 節）を実行する。



このとき、片方のコミットメントは保存し、もう一方のコミットメントは次の手順で使用する。

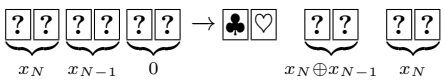
(3) 以下の手順を $i = 2, 3, \ldots, N-1$ の順に実行する。

(a) x_i のコミットメントに対して以下の Mizuki–Sone の XOR プロトコル（2.3 節）を実行する。

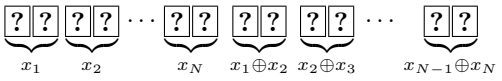


このとき、$x_i \oplus x_{i-1}$ と x_i のコミットメントは保存し、もう一方の x_i のコミットメントは次のループで使用する。

(4) x_N のコミットメントに対して以下の Mizuki–Sone の XOR プロトコル（2.3 節）を実行する。

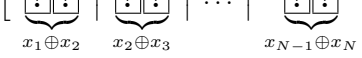


ここまでの手順で $x_1, x_2, \ldots, x_N$ の合計 N 個のコミットメントと $x_1 \oplus x_2, x_2 \oplus x_3, \ldots, x_{N-1} \oplus x_N$ の合計 $N-1$ 個のコミットメントが得られる。



また、$x_1 \oplus x_2, x_2 \oplus x_3, \ldots, x_{N-1} \oplus x_N$ のコミットメントで、値が 1 となるコミットメントの個数が切断数と一致する。

(5) $x_1 \oplus x_2, x_2 \oplus x_3, \ldots, x_{N-1} \oplus x_N$ の合計 $N-1$ 個のコミットメントに対してパイルスクランブルシャッフル（2.2.2 節）を適用する。

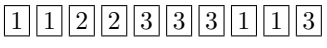


(6) シャッフルを適用したすべてのコミットメントをめくる。このとき、値が 1 のコミットメントが c 個あることを確認し、c 個あれば等分配検証フェーズに移り、c 個なければ、検証者 V は証明を棄却する。

3.1.2 等分配検証フェーズの実行手順

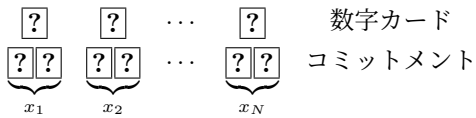
提案プロトコル 1 における等分配検証フェーズの手順を解説する。このフェーズを行うとき、切断数検証フェーズ（3.1.1 節）で得られた $x_1, x_2, \ldots, x_N$ の合計 N 個のコミットメントを用いる。

(1) 宝石の種類に対応させた数字カードを合計 N 枚置く。このとき、宝石 J_i に対して、整数 i が表面に描かれた数字カード $2a_i$ 枚を置く。1.1 節のネックレスの例では以下のように数字が置かれることになる。

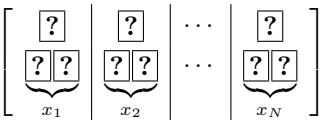


(2) 数字カードをすべて裏面にする。

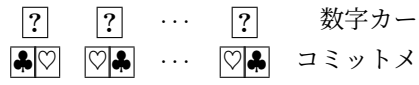
(3) 切断数検証フェーズ（3.1.1 節）で得られた $x_1, x_2, \ldots, x_N$ の合計 N 個のコミットメントを各数字カードの下に置く。



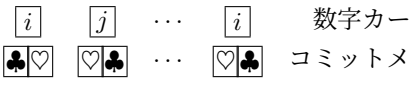
(4) 数字カードとコミットメントを一つの束として、N 個の束に対してパイルスクランブルシャッフル（2.2.2 節）適用する。



(5) コミットメントをすべてめくる。このとき、値が 0 のコミットメントの組（アリスの取り分）と、値が 1 のコミットメントの組（ボブの取り分）がどちらも $\frac{N}{2}$ 個あるかどうかを確認する。どちらも $\frac{N}{2}$ 個ずつあれば次の手順に移り、そうでなければ検証者 V は証明を棄却する。



(6) 数字カードをすべてめくる。このとき値が 0 のコミットメントの組（アリスの取り分）と、値が 1 のコミットメントの組（ボブの取り分）のそれぞれについて、宝石の分配が公平であるかを確認する。具体的には、双方の組において、各宝石の種類 J_i に対応する数字カード $\boxed{i}$ が a_i 枚あるかどうかを確認する。この条件がすべての種類の宝石について満たされていれば、検証者 V は証明を受理し、そうでなければ証明を棄却する。



3.1.3 提案プロトコル 1 の正当性

本節では、提案プロトコル 1 が完全性、健全性、ゼロ知識性を満たしていることを確認する。

完全性 証明者 P が c 回の切断によって全種類の宝石を等しく分配する正しい解を知っており、その解に従って忠実にプロトコルを実行すると仮定する。このとき、c 回の切断は隣り合う宝石の所属が切り替わる箇所、すなわち $x_i \neq x_{i-1}$ となる箇所が c 個あることを意味する。これは切断数検証フェーズ（3.1.1 節）の手順（4）を行った後、$x_i \oplus x_{i-1} = 1$ となるコミットメントが丁度 c 個作られていることと同値であるため 、手順（6）の検証を必ず通過する。また、等分配検証フェーズ（3.1.2 節）では正しい解が必ず持つ二つの性質を検証する。一つ目は宝石が $\frac{N}{2}$ 個ずつに等分されていることであり、これは手順（5）で確認される。二つ目は各種類 J_i の宝石が a_i 個ずつ等分されていることであり、これは手順（6）で確認される。証明者が持つ正しい解はこれらの条件を定義上満たすため、いずれの検証も必ず通過する。以上のように、証明者が正しい解に従う場合、両フェーズのすべての検証を通過するため、完全性が満たされていることがわかる。

健全性 証明者 P が c 回の切断によって全種類の宝石を等しく分配する正しい解を持たずに、検証者 V を欺こうとする場合を考える。このとき、証明のいずれかの段階で不正が必ず露見し、証明が棄却されることを示す。証明者 P の不正は、「宝石の分配が不公平である」場合と、「切断回数が c 回でない」場合のいずれか、あるいはその両方に分類される。まず、「宝石の分配が不公平である」場合は等分配検証フェーズ（3.1.2 節）で必ず検出される。具体的には、宝石を $\frac{N}{2}$ 個ずつに分けていない場合は手順（5）の検証で矛盾が発覚する。たとえ $\frac{N}{2}$ 個ずつに分けていたとしても、各種類ごとの宝石の分配が不公平（例えば、種類 J_i の宝石が a_i 個ずつでない）であれば、手順（6）でその不正が明らかになる。したがって、分配の公平性に関するいかなるごまかしも、このフェーズで必ず棄却される。また、「切断回数が c 回でない」場合は切断数検証フェーズ（3.1.1 節）で必ず検出される。切断回数が c 回でないということは、$x_i \neq x_{i-1}$ となる箇所の数が c ではないことを意味する。これは、$x_i \oplus x_{i-1} = 1$ となるコミットメントの総数が c 個でないことと同値である。よって、手順（6）でコミットメントの値をすべて確認した際に、その個数が c でないことが発覚し、証明は棄却される。以上より、証明者が正しい解を持たない場合、その不正が分配の公平性にあっても切断回数にあっても、プロトコルのいずれかの検証段階で必ず検出されるため、健全性が満たされていることがわかる。

ゼロ知識性 提案プロトコル 1 では正しい切断箇所に従ってカードが配置されていた場合、値が 0 のコミットメントと値が 1 のコミットメントがどちらもちょうど $\frac{N}{2}$ 個ある。そして、どちらの値でも各種類の宝石 J_i に対応する数字カード i がそれぞれ a_i 枚ずつある。このとき、切断数検証フェーズ（3.1.1 節）の手順（5）と等分配検証フェーズ（3.1.2 節）の手順（4）のパイルスクランブルシャッフル（2.2.2 節）により、その並び順は一様ランダムになるため、切断箇所に関する情報は一切漏れない。よって、ゼロ知識性が満たされていることがわかる。

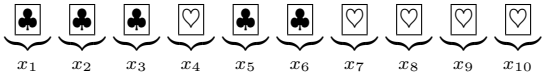
3.2 提案プロトコル 2

提案プロトコル 2 は提案プロトコル 1 よりシャッフル回数を少なくしたプロトコルである。

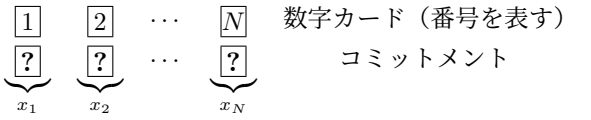
3.2.1 提案プロトコル 2 の実行手順

提案プロトコル 2 の手順は以下の通りである。

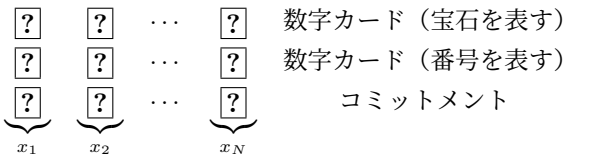
(1) 左から i 番目（$1 \leq i \leq N$）の宝石が Alice の宝石なら $x_i = 0$、Bob の宝石なら $x_i = 1$ と定め、証明者 P は $x_1, x_2, \ldots, x_N$ のコミットメントを入力する。例として 1.1 節のネックレスで証明者 P が正しく解を置いた場合は以下のようになる。



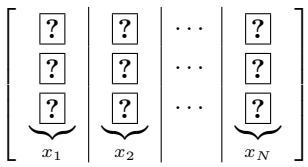
(2) N 個のコミットメントそれぞれで、x_i のコミットメントの上に整数 i が描かれた数字カード$\boxed{i}$を置く。その後、すべての数字カードを裏面にする。



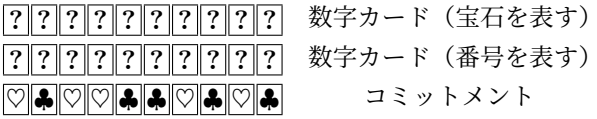
(3) 宝石の種類に対応させた数字カードを合計 N 枚置く。このとき、宝石 J_i に対して、数字カード$\boxed{i}$を $2a_i$ 枚置く。その後、すべての数字カードを裏面にする。



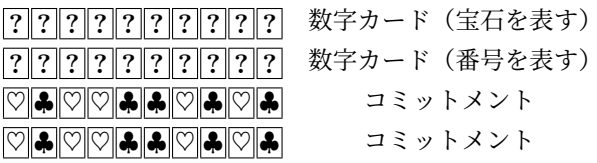
(4) 数字カードとコミットメントを一つの束として、N 個の束に対してパイルスクランブルシャッフル（2.2.2 節）を適用する。



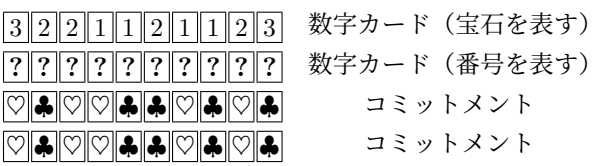
(5) 1 枚符号化のコミットメントをすべてめくり、ハートの枚数とクラブの枚数がそれぞれ $\frac{N}{2}$ 枚ずつであることを確認する。例として 1.1 節のネックレスで証明者 P が正しく解を置いた場合は以下のようになる。



どちらも $\frac{N}{2}$ 枚ずつであれば各カードの下に同じ 2 色カードを追加して次の手順に移り、そうでなければ検証者 V は証明を棄却する。

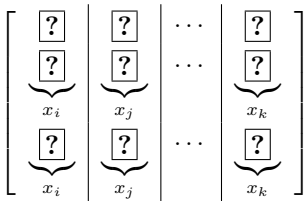


(6) 宝石を表す数字カードをすべてめくり、宝石の分配が公平であるかを確認する。具体的には、クラブを含む列の宝石たち（アリスの取り分）とハートを含む列の宝石たち（ボブの取り分）のそれぞれについて、各宝石の種類 J_i に対応する数字カード$\boxed{i}$が a_i 枚あるかどうかを確認する。例として 1.1 節のネックレスで証明者 P が正しく解を置いた場合は以下のようになる。

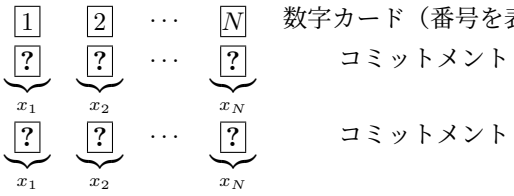


上記の条件がすべての種類の宝石について満たされていれば、検証者 V は宝石を表す数字カードを除去して次の手順に移り、そうでなければ証明を棄却する。

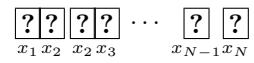
(7) 数字カードとコミットメントを一つの束として、N 個の束に対してパイルスクランブルシャッフル（2.2.2 節）を適用する。



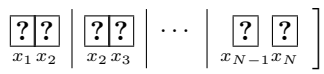
(8) 数字カードをすべてめくる。そして、数字カード i とコミットメント x_i 二つを一つの束にして、左から $1, 2, \ldots, N$ となるように並び替える。



(9) 1 枚符号化のコミットメントを x_i, x_{i+1} の 2 枚を一つの束にして、以下のような $N-1$ 個の 2 枚組を並べる。



(10)2 枚組を一つの束としてパイルスクランブルシャッフル（2.2.2 節）を適用する。



(11)シャッフルを適用したすべてのカードをめくる。このとき、二枚組の中に$\boxed{♣}\boxed{♡}$または$\boxed{♡}\boxed{♣}$となるカード組の個数の合計が c 個あるかどうかを確認する。c 個あれば検証者 V は証明を受理し、そうでなければ検証者 V は証明を棄却する。

3.2.2 提案プロトコル 2 の正当性

本節では、提案プロトコル 2 が完全性、健全性、ゼロ知識性を満たしていることを確認する。

完全性 証明者 P が c 回の切断によって全種類の宝石を等しく分配する正しい解を知っており、その解に従って忠実にプロトコルを実行すると仮定する。このプロトコルでは、正しい解が持つ性質をそれぞれの手順で検証している。宝石が $\frac{N}{2}$ 個ずつに等分されていることは手順 (5) で確認される。各種類 J_i の宝石が a_i 個ずつ等分されていることは手順 (6) で確認される。また、c 回の切断があることは、隣り合う宝石の所属が切り替わる箇所、すなわち $x_i \neq x_{i-1}$ となる箇所が c 個あることを意味する。これは手順 (9) を行った後、

$\boxed{\heartsuit}\boxed{\clubsuit}$ または $\boxed{\clubsuit}\boxed{\heartsuit}$ となる 2 枚組が丁度 c 個作られていることと同値であるため 、手順（11）の検証を必ず通過する。以上のように、証明者が正しい解に従う場合、すべての検証を通過するため、完全性が満たされていることがわかる。

健全性　証明者 P が c 回の切断によって全種類の宝石を等しく分配する正しい解を持たずに、検証者 V を欺こうとする場合を考える。このとき、証明のいずれかの段階で不正が必ず露見し、証明が棄却されることを示す。証明者 P の不正は、「宝石の分配が不公平である」場合と、「切断回数が c 回でない」場合のいずれか、あるいはその両方に分類される。まず、「宝石の分配が不公平である」場合、具体的には、宝石を $\frac{N}{2}$ 個ずつに分けていない場合は手順（5）の検証で矛盾が発覚する。たとえ $\frac{N}{2}$ 個ずつに分けていたとしても、各種類ごとの宝石の分配が不公平（例えば、種類 J_i の宝石が a_i 個ずつでない）であれば、手順（6）でその不正が明らかになる。したがって、分配の公平性に関するいかなるごまかしも、このフェーズで必ず棄却される。また、「切断回数が c 回でない」ということは、$x_i \neq x_{i-1}$ でない箇所の数が c ではないことを意味する。これは手順（9）を行った後、$\boxed{\heartsuit}\boxed{\clubsuit}$ または $\boxed{\clubsuit}\boxed{\heartsuit}$ となる 2 枚組がちょうど c 個作られていないことと同値である。よって、手順（6）でコミットメントの値をすべて確認した際に、その個数が c でないことが発覚し、証明は棄却される。以上より、証明者が正しい解を持たない場合、その不正が分配の公平性にあっても切断回数にあっても、プロトコルのいずれかの検証段階で必ず検出されるため、健全性が満たされていることがわかる。

ゼロ知識性　提案プロトコル 2 では正しい切断箇所に従ってカードが配置されていた場合、値が 0 のコミットメントと値が 1 のコミットメントがどちらもちょうど $\frac{N}{2}$ 個ある。そして、どちらの値でも各種類の宝石 J_i に対応する数字カード $\boxed{i}$ がそれぞれ a_i 枚ずつある。このとき、手順（4）のパイルスクランブルシャッフル（2.2.2 節）により、その並び順は一様ランダムになるため、切断箇所に関する情報は一切漏れない。よって、ゼロ知識性が満たされていることがわかる。

4. おわりに

本稿では盗まれたネックレス問題に対するカードベースゼロ知識証明プロトコルを二つ提案した。今後の研究課題としては、プロトコルのシャッフル回数やカード枚数のさらなる削減が挙げられる。また、2 人が宝石を分配する場合を一般化し、k 人で分割する場合のプロトコルや他の探索問題に対するカードベースゼロ知識証明プロトコルの構成なども考えられる。

謝辞　本研究は JSPS 科研費 JP23H00479、JP21K17702 と JST CREST JPMJCR22M1 の支援を受けた。

参考文献

[1] 3Blue1BrownJapan. Borsuk-ulam の定理とネックレス問題 ～トポロジーでパズルを解く～, 2022.

[2] N. Alon and D. B. West. The borsuk-ulam theorem and bisection of necklaces. *Proceedings of the American Mathematical Society*, 98(4):623–628, 1986.

[3] B. D. Boer. More efficient match-making and satisfiability the five card trick. In J.-J. Quisquater and J. Vandewalle eds., *EUROCRYPT 1989*, Vol. 434 of *LNCS*, pp. 208–217, Heidelberg, 1990. Springer.

[4] X. Bultel, J. Dreier, J. Dumas, and P. Lafourcade. Physical zero-knowledge proofs for akari, takuzu, kakuro and kenken. In *8th International Conference on Fun with Algorithms, FUN 2016*, Vol. 49 of *LIPIcs*, pp. 8:1–8:20, 2016.

[5] C. Crépeau and J. Kilian. Discreet solitary games. In D. R. Stinson ed., *Advances in Cryptology—CRYPTO' 93*, Vol. 773 of *LNCS*, pp. 319–330, Berlin, Heidelberg, 1994. Springer.

[6] A. Filos-Ratsikas and P. W. Goldberg. The complexity of splitting necklaces and bisecting ham sandwiches. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pp. 638–649, 2019.

[7] C. H. Goldberg and D. B. West. Bisection of circle colorings. *SIAM Journal on Algebraic Discrete Methods*, 6(1):93–106, 1985.

[8] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pp. 291–304, 1985.

[9] R. Gradwohl, M. Naor, B. Pinkas, and G. N. Rothblum. Cryptographic and physical zero-knowledge proof systems for solutions of Sudoku puzzles. In P. Crescenzi, G. Prencipe, and G. Pucci eds., *Fun with Algorithms*, Vol. 4475 of *LNCS*, pp. 166–182, Berlin, Heidelberg, 2007. Springer.

[10] R. Gradwohl, M. Naor, B. Pinkas, and G. N. Rothblum. Cryptographic and physical zero-knowledge proof systems for solutions of Sudoku puzzles. *Theory of Computing Systems*, 44(2):245–268, 2009.

[11] R. Ishikawa, E. Chida, and T. Mizuki. Efficient card-based protocols for generating a hidden random permutation without fixed points. In C. S. Calude and M. J. Dinneen eds., *Unconventional Computation and Natural Computation*, Vol. 9252 of *LNCS*, pp. 215–226, Cham, 2015. Springer.

[12] P. Lafourcade, D. Miyahara, T. Mizuki, L. Robert, T. Sasaki, and H. Sone. How to construct physical zero-knowledge proofs for puzzles with a "single loop" condition. *Theor. Comput. Sci.*, 888:41–55, 2021.

[13] D. Miyahara, T. Sasaki, T. Mizuki, and H. Sone. Card-based physical zero-knowledge proof for kakuro. *IEICE Transactions*, 102-A(9):1072–1078, 2019.

[14] T. Mizuki and H. Sone. Six-card secure AND and four-card secure XOR. In X. Deng, J. E. Hopcroft, and J. Xue eds., *Frontiers in Algorithmics*, Vol. 5598 of *LNCS*, pp. 358–369, Berlin, Heidelberg, 2009. Springer.

[15] V. Niemi and A. Renvall. Secure multiparty computations without computers. *Theor. Comput. Sci.*, 191(1–2):173–183, 1998.

[16] C. H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Jour-*

nal of Computer and System Sciences, 48(3):498–532, 1994.
[17] S. Ruangwises. Two standard decks of playing cards are sufficient for a ZKP for Sudoku. In C.-Y. Chen, W.-K. Hon, L.-J. Hung, and C.-W. Lee eds., *Computing and Combinatorics*, Vol. 13025 of *LNCS*, pp. 631–642, Cham, 2021. Springer.
[18] T. Sasaki, T. Mizuki, and H. Sone. Card-based zero-knowledge proof for Sudoku. In H. Ito, S. Leonardi, L. Pagli, and G. Prencipe eds., *Fun with Algorithms*, Vol. 100 of *LIPIcs*, pp. 29:1–29:10, Dagstuhl, Germany, 2018. Schloss Dagstuhl.
[19] K. Shinagawa. Card-based protocols with single-card encoding. In C. Anutariya and M. M. Bonsangue eds., *Theoretical Aspects of Computing*, Vol. 15373 of *LNCS*, pp. 182–194, Cham, 2025. Springer.
[20] K. Tanaka and T. Mizuki. Two uno decks efficiently perform zero-knowledge proof for Sudoku. In H. Fernau and K. Jansen eds., *Fundamentals of Computation Theory*, Vol. 14292 of *LNCS*, pp. 406–420, Cham, 2023. Springer.
[21] K. Tanaka, S. Sasaki, K. Shinagawa, and T. Mizuki. Only two shuffles perform card-based zero-knowledge proof for sudoku of any size. In *2025 Symposium on Simplicity in Algorithms (SOSA)*, pp. 94–107. SIAM, 2025.

付　　録

A.1 PPA 完全

本稿で扱う 2 人が宝石を分配する場合の盗まれたネックレス問題における計算困難性について解説するため、探索問題の計算量クラス TFNP、そのサブクラスであるクラス PPA について詳述する。この問題は、n 種類の宝石があれば高々 n 回の切断で公平な分割ができるため、解の存在が常に保証されている。そのためこの問題は、解の有無を問う決定問題のクラス（例えばクラス NP）ではなく、解を見つける探索問題のクラスに属する。

2 人が宝石を分配する場合の「盗まれたネックレス問題」は、Ratsikas–Goldberg [6] によって PPA 完全（Polynomial Parity Argument–complete）であることが証明されており、多項式時間アルゴリズムによる一般的な解法は知られていない。探索問題とは、入力 x に対してある解 y を見つけることを目的とし、その解と入力の組 (x, y) を非決定性チューリングマシンが多項式時間で検証できるような問題をクラス FNP と呼ぶ [16]。クラス FNP に属する問題のうち、任意の入力に対して必ず少なくとも一つの解が存在する関係 $R(x, y)$ を持つものはクラス TFNP（Total Function Nondeterministic Polynomial）と定義される。クラス TFNP は解の存在が数学的に保証された探索問題のクラスであり、存在証明の手法に基づいて複数のサブクラスに細分される。

クラス PPA はその一つで、無向グラフにおける握手補題[*1]が解の存在を保証することを根拠とし、標準問題 Leaf（Odd-Degree-Node）への多項式時間帰着により定義される。PPA 完全性とは、ある問題がクラス PPA に属し、かつクラス PPA の任意の問題からその問題への多項式時間帰着が存在することを指し、これらの問題には既知の多項式時間アルゴリズムが存在しない。このように、「盗まれたネックレス問題」の PPA 完全性は、その計算的困難性を厳密に位置づけるものである。

[*1] 任意の有限な無向グラフにおいて、奇数個の辺と接続している頂点（奇数次数頂点）の総数は、必ず偶数となる。