2ZD-01

ビルディングパズルに対するゼロ知識証明

宮﨑 勇斗　　真鍋義文
工学院大学　情報学部　システム数理学科

１．はじめに

ビルディングパズル[1]は矢印の方向から数字の数だけのビルが見えるように高さの違うビルを並べるペンシルパズルである。各マスにはビルの高さを表す数字が入る。ビルディングパズルに対するゼロ知識証明を示す。ビルディングパズルのパズルインスタンスに対して、解答に関する情報を一切開示することなく、解答を知っていることを検証者に確信させることができるプロトコルを示す。本プロトコルはトランプのような物理的カードを用いて実現される。

２．定義

２．１ゼロ知識証明

ゼロ知識証明[5]は、証明者が検証者に自分が持っている命題が真であることを伝えるのに真であること以外の知識を与えずに証明する手法である。
ゼロ知識証明は次の３条件を満たさないといけない。

１．完全性
検証者は証明者が持つ命題が真である場合、真であることが必ずわかる。

２．健全性
検証者は、証明者が持っている命題が偽である場合、高い確率でそれが偽であることが見抜ける。

３．ゼロ知識性
証明者の持つ命題が真であるなら、検証者が証明者から知識を盗もうとしても「命題が真である」こと以外の何も知識を得ることができない。

２．２ビルディングパズル

本稿で取り扱うビルディングパズルの問題例を以下の図１の左に、答を図１の右に示す。問題はｎ×ｎの格子のマスから成り立っている。また、各行・列とも１からｎまでの数字が入り、数字はビルの高さを示すものである。数字と矢印の組は矢印の方向から見たときに見えるビルの数のことである。

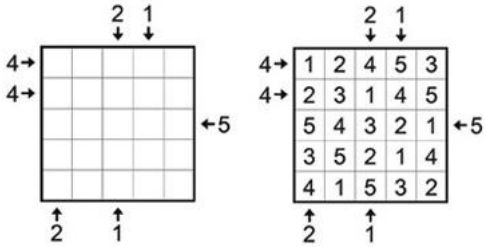

図１：ビルディングパズル

Zero knowledge proof for building puzzles

† Hayato Miyazaki・Faculty of Informatics, Kogakuin University
‡ Yoshifumi Manabe・Faculty of Informatics, Kogakuin University

３．証明プロトコル

３．１　すべてのマスに１からｎが含まれている証明

証明プロトコルではクローバーとハートのカードをおよび、1 からｎの数字のカードを用いる。ここでは♡,♣は全く同じカードが複数枚あり、トランプとは異なる。表が♡か♣のマークで、裏が同じ？であるカードを１ビットのデータを２枚で表現する。♣♡の組み合わせを０、♡♣の組み合わせを１とする。答えを知っている人を証明者 A、答えを知らない人検証者 B とする。A はすべてのマスに答えの数字を２進数に変換し、各ビットを２枚のカードで表現したものを伏せる。この証明手続きは数独のゼロ知識証明[3]と同じものである。A はある行に対して１からｎの数字カードを追加する。A はここである行に対してのヒントのチェックを行う。このヒントのチェックは３．２で説明される。ヒントのチェックが終わった後使われた行のカードをパイルスクランブルシャッフルし、２進数で表現された答えのカードのみを公開し、１からｎがあることを確認する。確認後、カードを伏せてもう１度パイルスクランブルシャッフルを適応し１からｎが書かれた数字カードを公開し１からｎの順になるようにカードを並べる。これをすべての行と列で行う。

このような方法で証明したとき、ゼロ知識証明の性質が満足されることを以下に示す。

１．完全性
A が答えにしたがって正しく数字をおくと B が作るすべてのパイルには１からｎまでのすべての数字が含まれていないといけない。それらを確認することで B はすべてのカードが解答通りにおかれていることが確認できる。

２．健全性
A がカードをある行または列の答えにおいて１からｎが１つずつでない場合、不正を必ず検知できる。

３．ゼロ知識性
B はカードの行と列にカードが１からｎ含まれていることは知っているがカードの正確な位置は知ることができない。

以上のことよりゼロ知識証明の条件を満たしている。

３．２　ヒントに従った配置のチェック

ここでは３．１での説明で必要になるヒントに従った配列のチェックを行う。実行するためには以下のプログラムを用いる。チェックする行（または列）でヒントの数字から近い方から X[1], X[2], …. X[n]とする。ｎは格子の大きさである。このプログラムはある矢印から見るビルの数がｋに正しく出力されるプログラムである。まずはじめにｋ=1 とセットし、Max の値を X[1]と置き、X[i]と比較を行い Max が大きければ、次に X[i+1]と比較をし、X[i]が大きければ Max の値に X[i]の値が入りｋの値が１増える。次に X[i+1]と比較をする。この手続きを i=2 からｎまで n-１回繰り返す。

```
int n
int k = 1;
int i;
int Max = X[1];
for(i=2 ; i<=n ; i++) {
            if(X[i] > Max) {
                        Max = X[i];
                       k++;
                      }
                    }
          printf("数は%d です。¥n",k);
```

次にこのプログラムで行う値の大小を比較するため金持ちプロコトルと呼ばれるプロコトルが必要になる。

金持ち比べは２つの値を見比べてお互いの値を知ることなくどちらの値が大きいかを調べる手法である。今回の研究では、既存のカードベースプロトコル[2]を用いる。

そしてここからはヒントとしてでている数字とその列の値を用いる。回路ベースの金持ち比べは以下のようなものである。

input : a = $(a_n,\dots,a_1)_2$,, b=$(b_n,\dots,b_1)_2$

$f_1=\overline{a_1}\cap b_1$;
for(i:2 to n){
　　$f_i=(\overline{a_i}\cap b_i)\cup((\overline{a_i}\cup b_i)\cap f_i\text{-}1)$;
}
output : f_n(=bool(a<b))

このときの結果がコミットされた１ビットデータで与えられる。 (♣♡のとき a<b,♡♣のとき b>a となる)

X[i]の配置が別列のチェックにも必要になるので配置を元に戻すための１からｎの数字のカードを置く。そして Max と X[i]の値を COPY プロトコル[4]を用いてコピーする。そしてコピーした値を回路ベースプロトコルを用いて比較を行う。そして出力は♣♡または♡♣となる。今回は♡♣のとき Max の方が大きく、♣♡のときは X[i]の方が大きくなるように計算を行う。出力されたカードを裏向きに置き、COPY プロコトルでコピーする。結果の左側のカードを Max と組にする。右側のカードを X[i]と組にする。そしてこの２組にパイルスクランブルシャッフルを行う。行った後結果のカードのみオープンする。♡のカードがある方を新しい Max とおく。次に新しい Max と X[i+1]に対して同じ手順を行う。この流れをｎ－１回繰り返す。

こうして出力されたｎ－１回の結果のｎ－１組のカードをパイルスクランブルシャッフルし結果のカードを公開する。このときの♣♡の組み合わせの個数を数えてその数に基のカウント１を足した結果がヒントとなる数字と同じになればこの行は正しい、そうでなければ正しくない。

そして実行が終わった後、パイルスクランブルカットを用いたあと数字が書かれたカードを公開して正しい配置に戻すことを行う。

この手続きは以下のゼロ知識証明を満たす。

１．完全性

上記のアルゴリズムで矢印の方向から見えるビルの数を求めることができることは明らかである。したがってＡが正しく答えを置いている場合、計算結果はヒントの数字に一致する。

２．健全性

Ａが置いたカードに対して、矢印の方向から見えるビルの数がヒントと異なっている場合、上記アルゴリズムによって不正を必ず検知できる。

３．ゼロ知識性

B はカードの行と列に対する矢印から見たときのｋの値を知ることができるがそれぞれの大小の関係値をしることはできない。

以上のことよりゼロ知識証明の条件を満たしている。

4. 結論

本論文ではカードを用いたビルディングパズルに対するゼロ知識証明のプロトコルを示した。

その結果ゼロ知識証明における条件の完全性、健全性、ゼロ知識性を満たすことが示せた。

参考文献

[1] ビルディングパズルのルールと解き方 (2021 年 12 月 1 日参照)
https://jp.calcblocks.com/
[2] Miyahara, D., Hayashi, Y.i., Mizuki, T., Sone, H.: "Practical card-based implementations of yao's millionaire protocol," Theoretical Computer Science Vol. 803, pp.207-221 (2020).
[3] Sasaki, T., Miyahara, D., Mizuki, T., Sone, H.: "Efficient card-based zero-knowledge proof for sudoku," Theoretical Computer Science Vol. 839, pp.135-142 (2020).
[4] Mizuki, T., Sone, H.: "Six-card secure and and four-card secure xor," Proc. of 3rd International Workshop on Frontiers in Algorithms(FAW 2009), LNCS Vol. 5598, pp.358-369 (2009).
[5]暗号解説シリーズ 「ゼロ知識証明」について解説！！ (2021 年 12 月 1 日参照)
https://qiita.com/kenmaro/items/d968375793fe754575fe