遷移問題とゼロ知識証明

田村 祐馬（東北大学 大学院情報科学研究科）

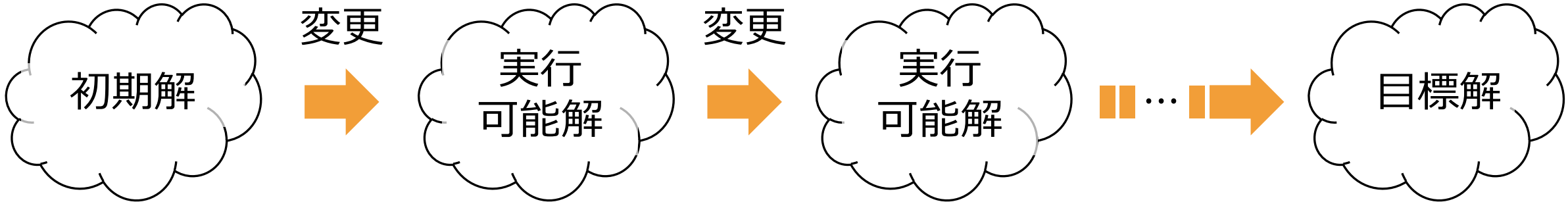
遷移問題とゼロ知識証明

田村 祐馬（東北大学 大学院情報科学研究科）

普段の専門分野
- グラフアルゴリズム
- パラメータ化計算量
- **遷移問題**

遷移問題

- 従来の探索問題（SATやペンシルパズル）では，解を1つ求めることが目的
- 遷移問題では初期解と目標解が与えられ，特定のルール下で初期解から目標解への**変更手順（遷移）**を見つけることが目的



- 遷移問題は2008年に東北大学の伊藤 健洋教授らによって提唱
- 理論計算機科学分野において近年の研究が盛ん
  - 科研費・学術変革領域研究（B）の採択課題（2020-2022年度）
  - 国際会議WALCOM 2025で独立したセッション

Hot topic　　Hot topic

- **遷移問題 × カードベース暗号 = 新たな研究領域？**

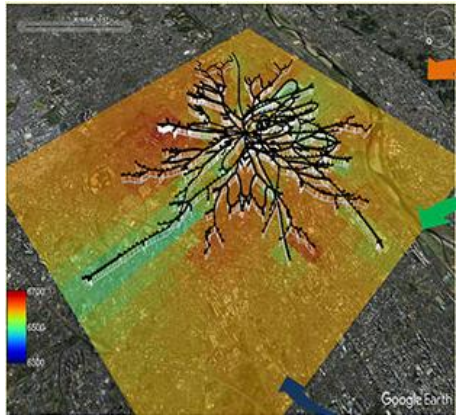
遷移問題

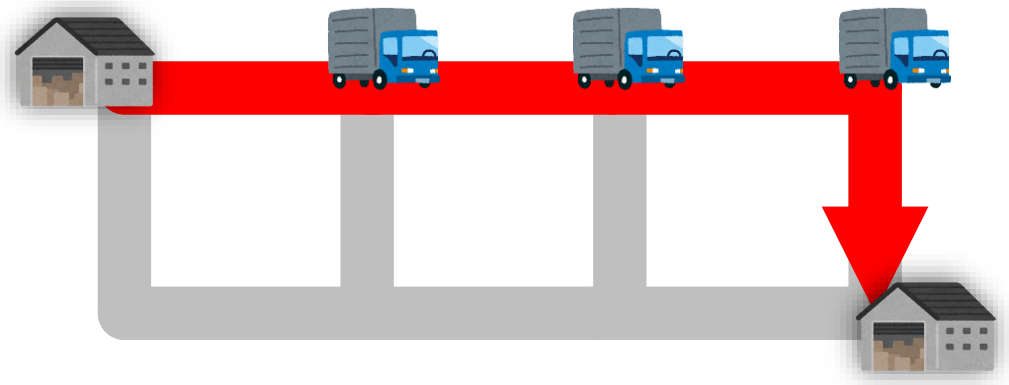
遷移問題の応用先：

- パズルやゲーム（15パズル・倉庫番・スーパーマリオブラザーズ）

1	2	3	4
5	6	11	7
9	10	8	
13	14	15	12

- 産業分野（配電網切替・宅配トラックの経路変更）





- 数学（Nash-Williams theoremの別証明）

15パズル

- サイズ4 × 4の盤面上に1~15が書かれた15個のタイルが存在
- 空白セルが1つだけ存在

ルール：空白セルに隣り合ったタイルのみ空白セルに動かせる

1	2	3	4
5	6	11	7
9	10	8	
13	14	15	12

15パズル

- サイズ4 × 4の盤面上に1~15が書かれた15個のタイルが存在
- 空白セルが1つだけ存在

ルール：空白セルに隣り合ったタイルのみ空白セルに動かせる

1	2	3	4
5	6	11	
9	10	8	7
13	14	15	12

15パズル

- サイズ4 × 4の盤面上に1~15が書かれた15個のタイルが存在
- 空白セルが1つだけ存在

ルール：空白セルに隣り合ったタイルのみ空白セルに動かせる

1	2	3	4
5	6	11	7
9	10	8	
13	14	15	12

初期状態

目標状態

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

目標: 初期状態から目標状態へ変更（遷移）させる.

15パズル

- サイズ4 × 4の盤面上に1~15が書かれた15個のタイルが存在
- 空白セルが1つだけ存在

ルール：空白セルに隣り合ったタイルのみ空白セルに動かせる

初期状態

1	2	3	4
5	6	11	7
9	10		8
13	14	15	12

目標状態

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

目標: 初期状態から目標状態へ変更（遷移）させる.

15パズル

- サイズ4 × 4の盤面上に1~15が書かれた15個のタイルが存在
- 空白セルが1つだけ存在

ルール：空白セルに隣り合ったタイルのみ空白セルに動かせる

初期状態

1	2	3	4
5	6		7
9	10	11	8
13	14	15	12

目標状態

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

目標: 初期状態から目標状態へ変更（遷移）させる.

15パズル

- サイズ4 × 4の盤面上に1~15が書かれた15個のタイルが存在
- 空白セルが1つだけ存在

ルール：空白セルに隣り合ったタイルのみ空白セルに動かせる

初期状態

1	2	3	4
5	6	7	
9	10	11	8
13	14	15	12

目標状態

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

目標: 初期状態から目標状態へ変更（遷移）させる.

15パズル

- サイズ4 × 4の盤面上に1~15が書かれた15個のタイルが存在
- 空白セルが1つだけ存在

ルール：空白セルに隣り合ったタイルのみ空白セルに動かせる

初期状態

1	2	3	4
5	6	7	8
9	10	11	
13	14	15	12

目標状態

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

目標: 初期状態から目標状態へ変更（遷移）させる.

15パズル

- サイズ4 × 4の盤面上に1~15が書かれた15個のタイルが存在
- 空白セルが1つだけ存在

ルール：空白セルに隣り合ったタイルのみ空白セルに動かせる

初期状態

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

遷移可能

?

目標状態

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

目標: 初期状態から目標状態へ変更（遷移）させる.

15パズル

- サイズ4 × 4の盤面上に1~15が書かれた15個のタイルが存在
- 空白セルが1つだけ存在

ルール：空白セルに隣り合ったタイルのみ空白セルに動かせる

初期状態

1	2	3	4
5	6	7	8
9	10	11	12
13	15	14	

目標状態

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

15パズル

- サイズ4 × 4の盤面上に1~15が書かれた15個のタイルが存在
- 空白セルが1つだけ存在

ルール：空白セルに隣り合ったタイルのみ空白セルに動かせる

初期状態

1	2	3	4
5	6	7	8
9	10	11	12
13	15	14	

遷移不可能

?

目標状態

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

- どのような変更を行っても，この初期状態から目標状態へ遷移することは**不可能**

15パズル

- サイズ4 × 4の盤面上に1~15が書かれた15個のタイルが存在
- 空白セルが1つだけ存在

ルール：空白セルに隣り合ったタイルのみ空白セルに動かせる

1	2	3	4
5	6	7	8
9	10	11	12
13	15	14	

初期状態

目標状態

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

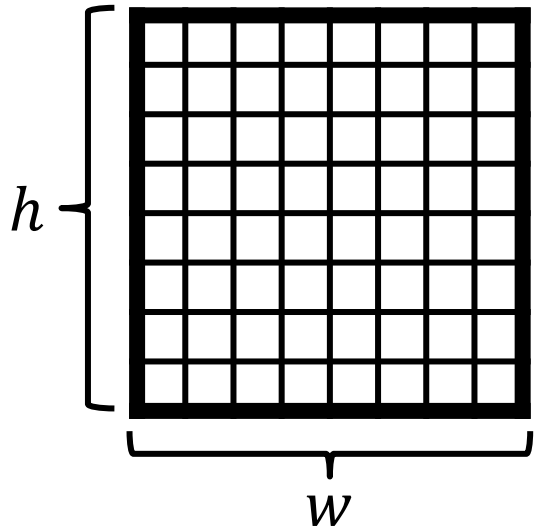
Q1. 与えられた初期状態から目標状態へ遷移可能か？ **到達可能性問題**

Q2. 目標状態へ遷移可能なら，最短の手順は何か？ **最短遷移問題**

$(w \times h)$パズル

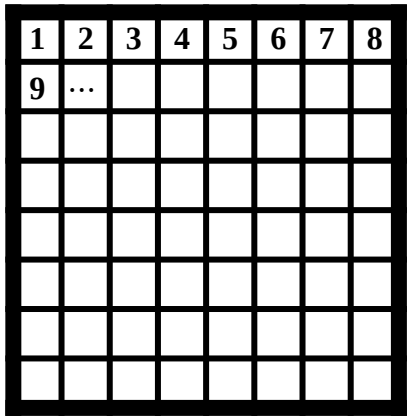
- サイズ$w \times h$の盤面上に1~$wh-1$が書かれた$wh-1$個のタイルが存在.
- $w = h = 4$のとき，15パズルに等しい.

ルール：空白セルに隣り合ったタイルのみ空白セルに動かせる



初期状態

目標状態



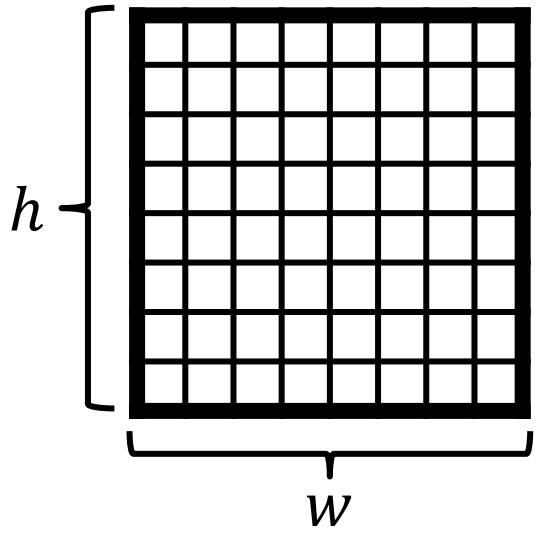
Q1. 与えられた初期配置から目標状態へ遷移可能か？ **到達可能性問題**

Q2. 目標状態へ遷移可能なら，最短の手順は何か？ **最短遷移問題**

$(w \times h)$パズル

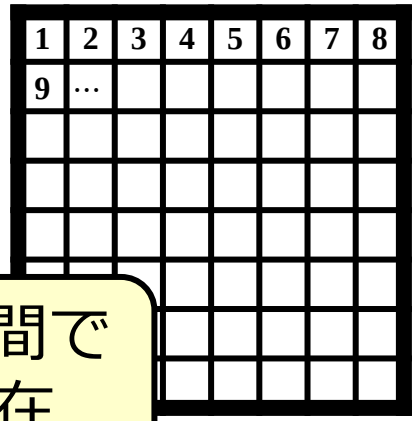
- サイズ$w \times h$の盤面上に1~$wh-1$が書かれた$wh-1$個のタイルが存在.
- $w = h = 4$のとき，15パズルに等しい.

ルール：空白セルに隣り合ったタイルのみ空白セルに動かせる



初期状態

目標状態



?

$(w \times h)$パズルなら多項式時間で
判定するアルゴリズムが存在

Q1. 与えられた初期配置から目標状態へ遷移可能か？ **到達可能性問題**

Q2. 目標状態へ遷移可能なら，最短の手順は何か？ **最短遷移問題** NP困難

$(w \times h)$パズル

Table 1: Values of $T(w, h)$.

$h \backslash w$	1	2	3	4	5	6	7	8	9
1	0	1	2	3	4	5	6	7	8
2	1	6	21	36	55	80	108	140	
3	2	21	31	53	84				
4	3	36	53	80					
5	4	55	84						
6	5	80							
7	6	108							
8	7	140							
9	8								

- $(w \times h)$パズルの初期状態から目標状態への最短ステップ数の最大値
- 4×5パズルすら未解決
- 最短手順を自分だけが知っていることには**価値がある** ➡ **ゼロ知識証明（ZKP）**

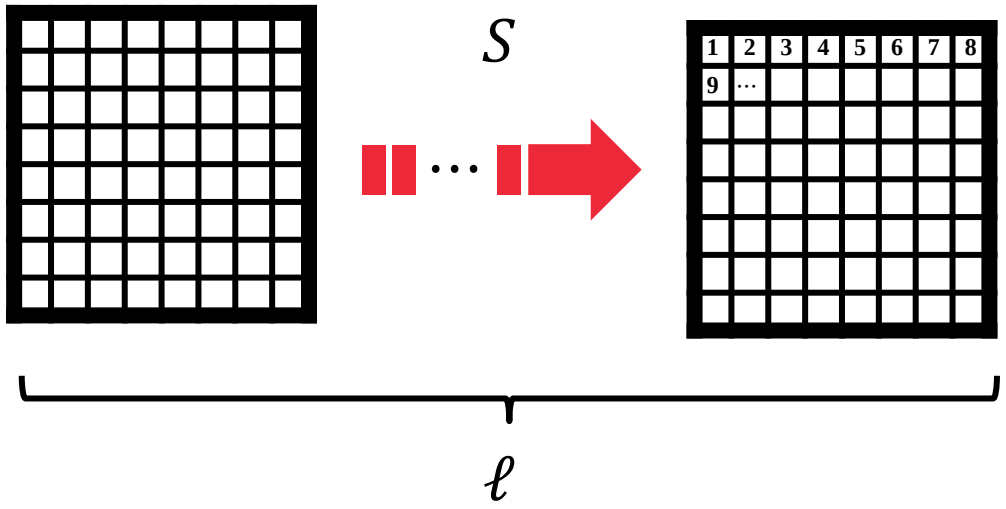
ゼロ知識証明

Peggy

A) Peggyはℓステップからなる遷移手順を得たことをVictorに伝えたい.

B) 一方で，遷移手順に関するいかなる情報もVictorに漏らしたくない.

Victor



A)とB)を満たすカードベースZKPプロトコルを構築する

ゼロ知識証明

Peggy

A) Peggyはℓステップからなる遷移手順を得たことをVictorに伝えたい.

B) 一方で，遷移手順に関するいかなる情報もVictorに漏らしたくない.

Victor

なぜ$(w \times h)$パズルのカードベースZKPを考えるのか？

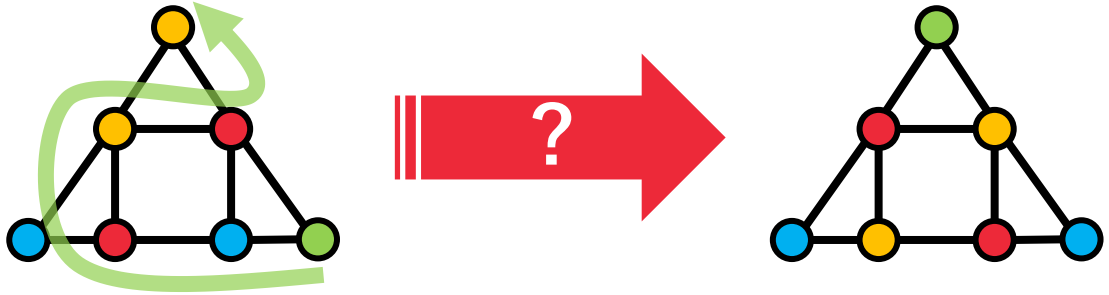
- 遷移問題とカードベース暗号のコラボレーションの第一歩として最適
- オブジェクトの移動は遷移問題で共通して現れる概念
  → 本研究で得られたテクニックは，他の遷移問題にも応用が期待
- 15パズルの知名度は高く，暗号理論の教育において有用

本研究の成果 [Tamura, Suzuki, Mizuki APKC2024]

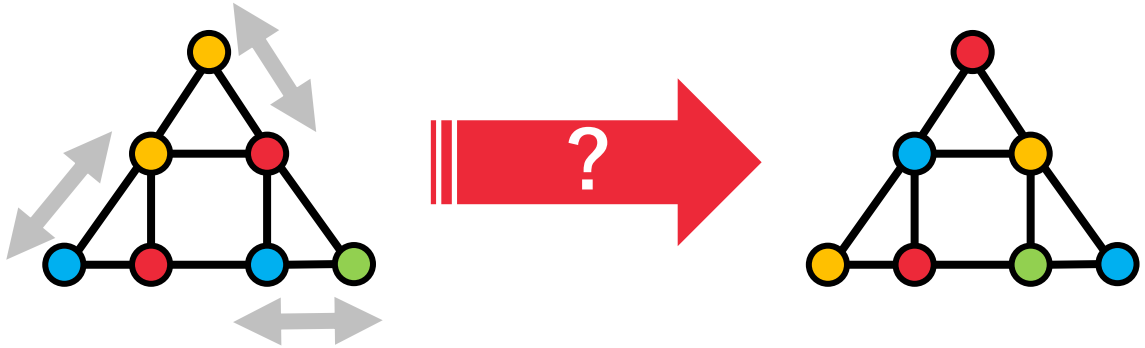
以下の問題に対するカードベースZKPを構築

- $(w \times h)$パズル

- シーケンシャルトークンスワッピング
  (1つのトークンを順にスワップ)



- トークンスワッピング
  (隣接したトークンをスワップ)



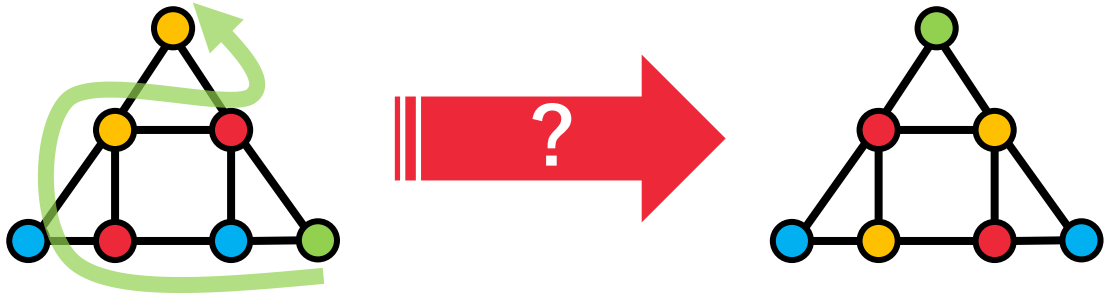
本研究の成果 [Tamura, Suzuki, Mizuki APKC2024]

以下の問題に対するカードベースZKPを構築

- $(w \times h)$パズル

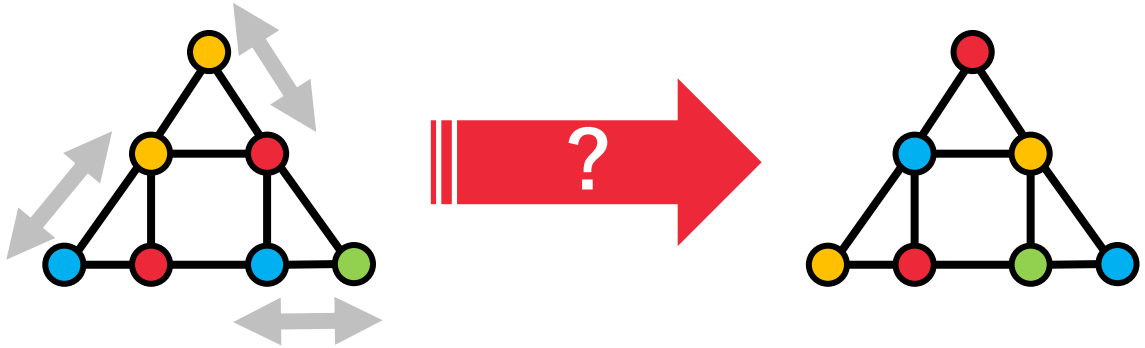
プロトコル修正

- シーケンシャルトークンスワッピング
  (1つのトークンを順にスワップ)



プロトコル修正

- トークンスワッピング
  (隣接したトークンをスワップ)



(2 × 2)パズルに対するカードベースZKP

初期

2	3
1	4

→

2	4
1	3

→

4	2
1	3

→

1	2
4	3

→

1	2
3	4

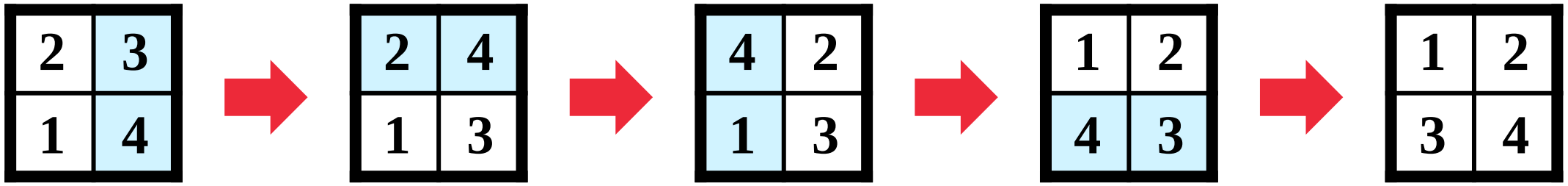
目標

Peggy

- Peggyのみ初期状態から目標状態への遷移手順を知っている
- 「4」を，空白を表す数字とする
- ゼロ知識証明を行うためにカードを並べていく

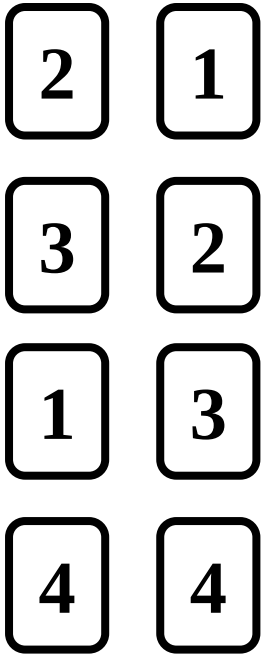
(2 × 2)パズルに対するカードベースZKP

初期



目標

Peggy

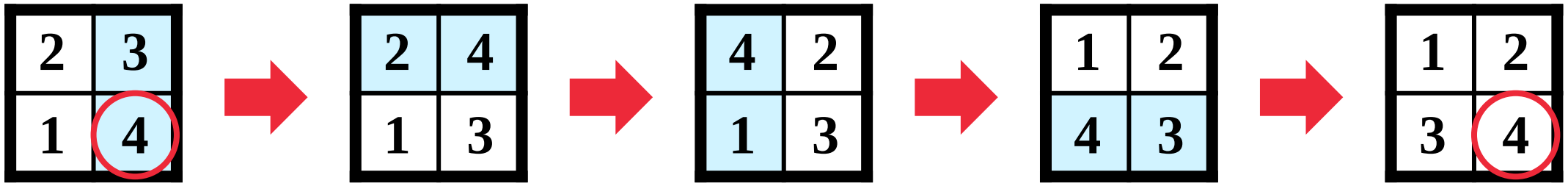


- **1列目**：
初期状態の左上から見て，同じ数字のカードを並べる．

- **2列目**：
目標状態の左上から見て，同じ数字のカードを並べる．
（1から4までの数字を昇順に並べる．）

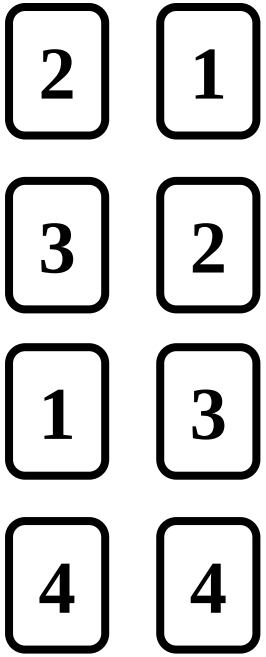
(2 × 2)パズルに対するカードベースZKP

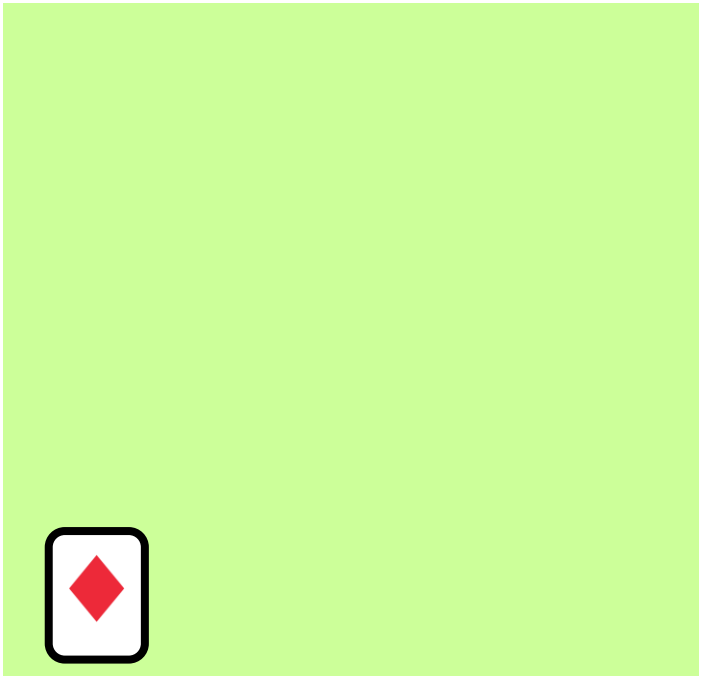
初期



目標

Peggy





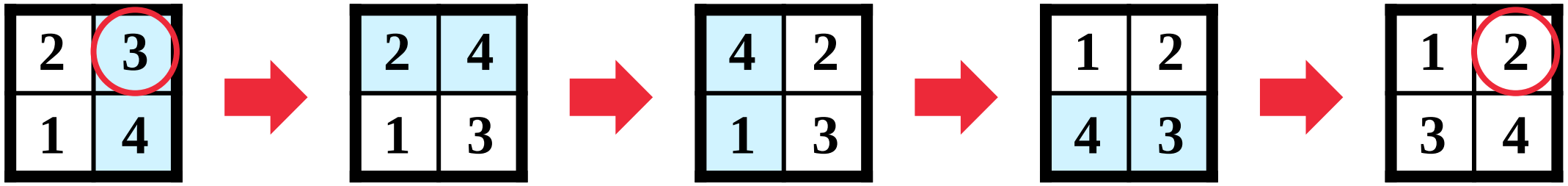
緑の領域に
- ♦は4（空白セル）の初期位置
- ♥は移動するタイルの位置

を表すように♦と♥を置く．
（その他は♣で埋める．）

➢ **緑1列目**： 初期状態の4は，目標状態では**4**が置いてある場所にある．**4行目**に♦．

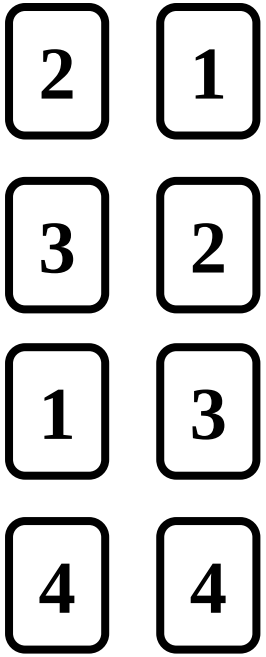
(2 × 2)パズルに対するカードベースZKP

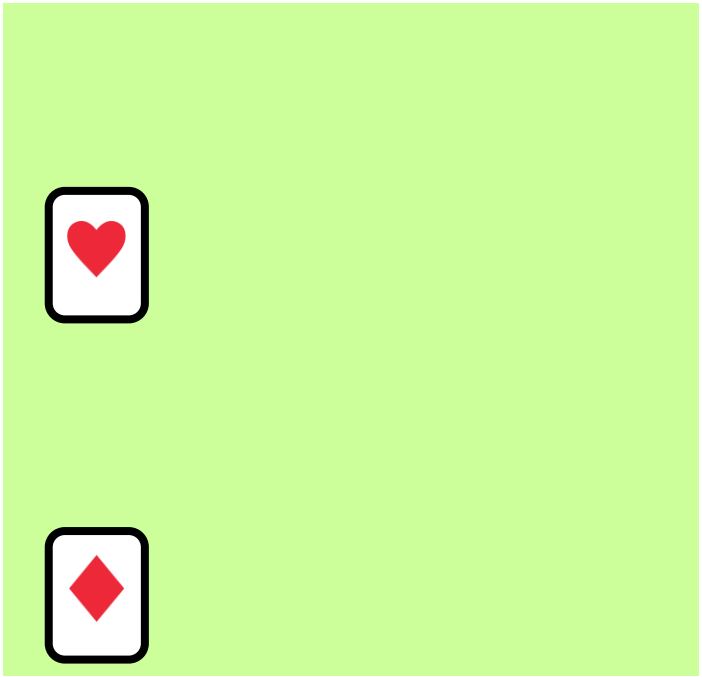
初期



目標

Peggy



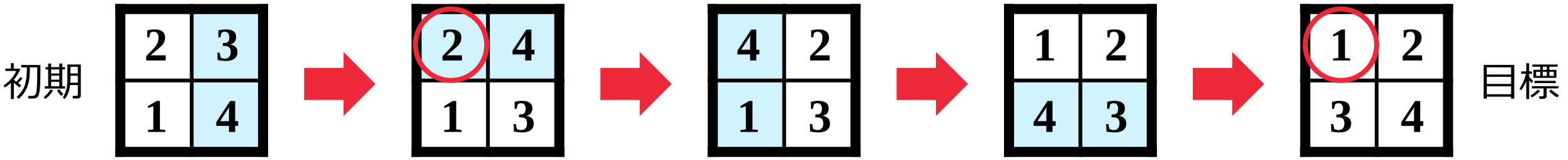


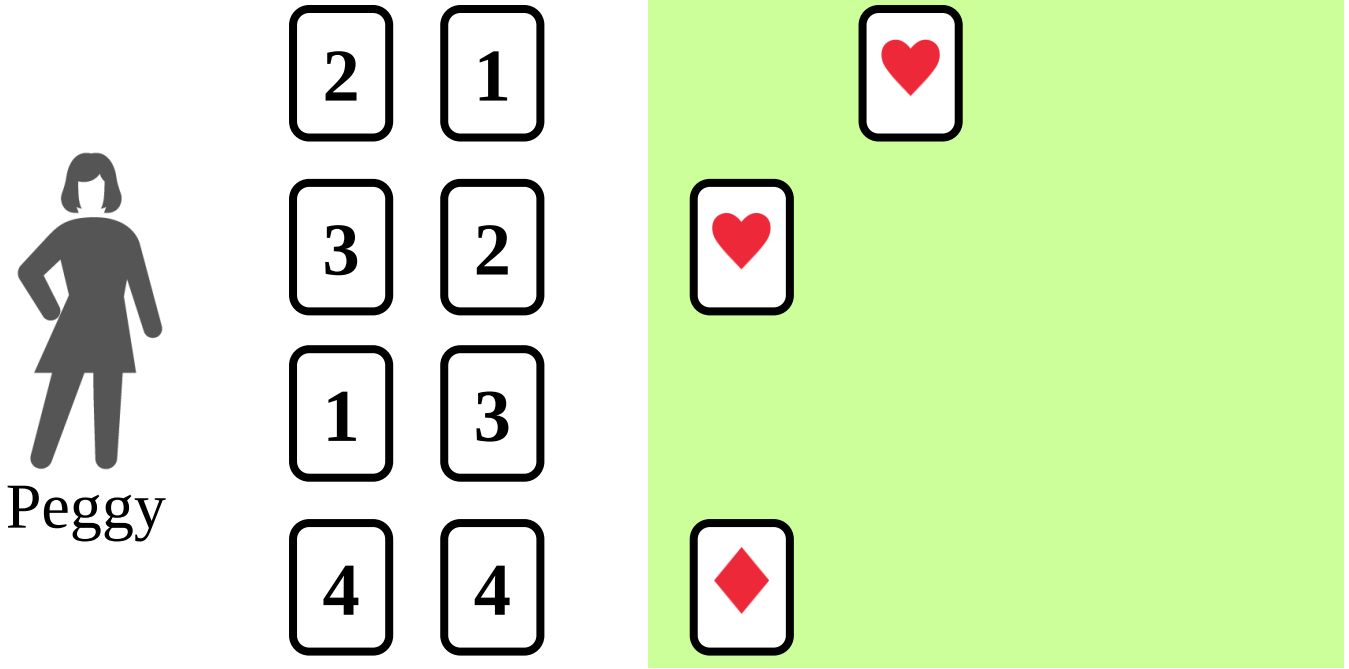
緑の領域に
- ♦は4（空白セル）の初期位置
- ♥は移動するタイルの位置

を表すように♦と♥を置く．
（その他は♣で埋める．）

➢ **緑1列目**：1ステップ目，目標では2が置いてある場所のタイルを移動．**2行目**に♥．

(2 × 2)パズルに対するカードベースZKP



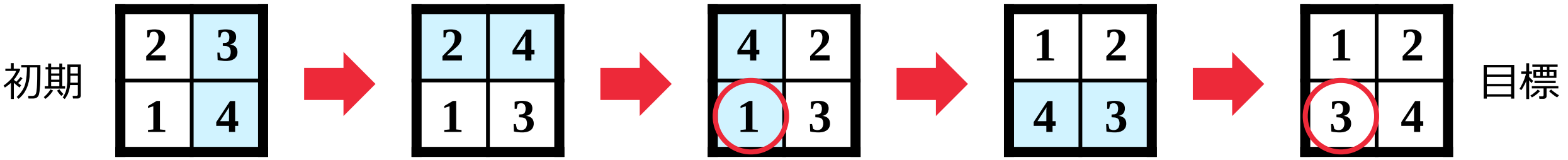


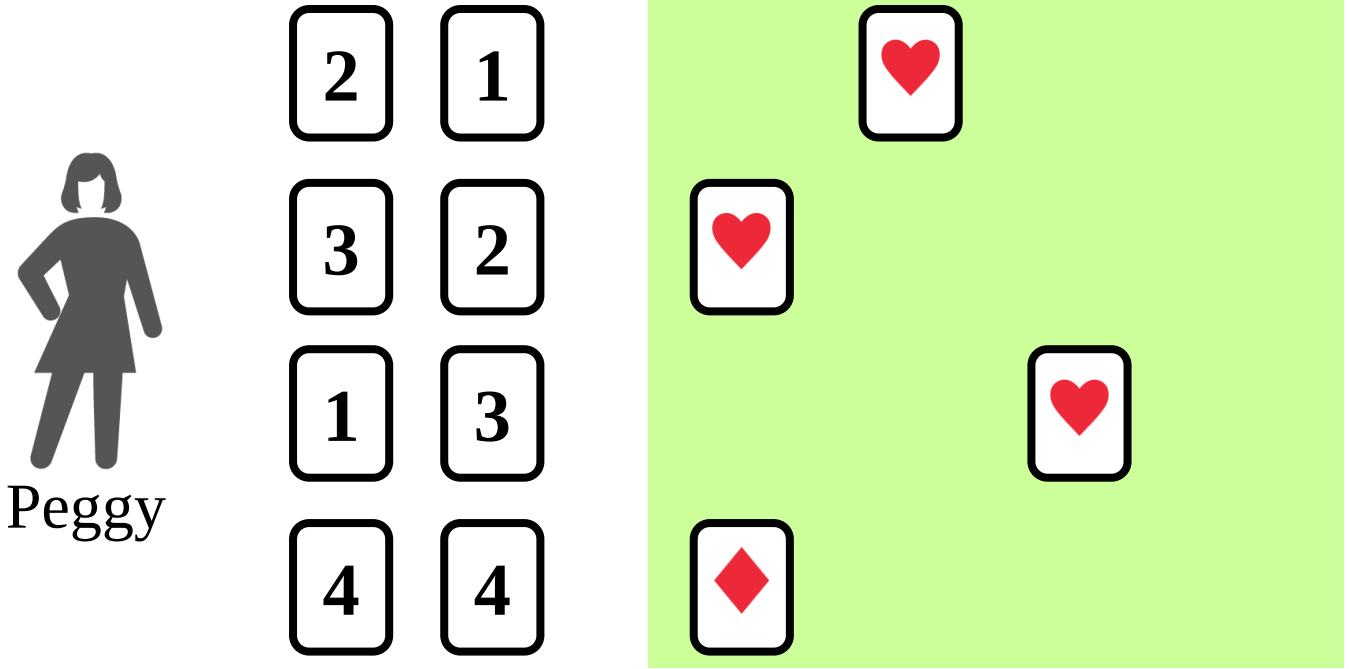
緑の領域に
- ♦は4（空白セル）の初期位置
- ♥は移動するタイルの位置

を表すように♦と♥を置く．
（その他は♣で埋める．）

➢ **緑2列目**：2ステップ目，目標では**1**が置いてある場所のタイルを移動．**1行目**に♥．

(2 × 2)パズルに対するカードベースZKP





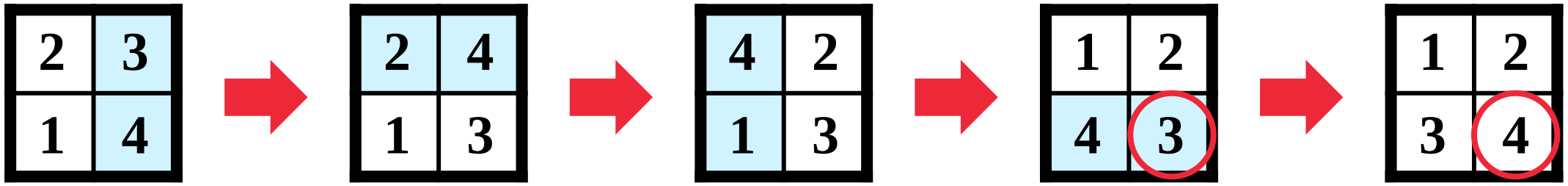
緑の領域に
- ♦は4（空白セル）の初期位置
- ♥は移動するタイルの位置

を表すように♦と♥を置く．
（その他は♣で埋める．）

➢ **緑3列目**：3ステップ目，目標では**3**が置いてある場所のタイルを移動．**3行目**に♥．

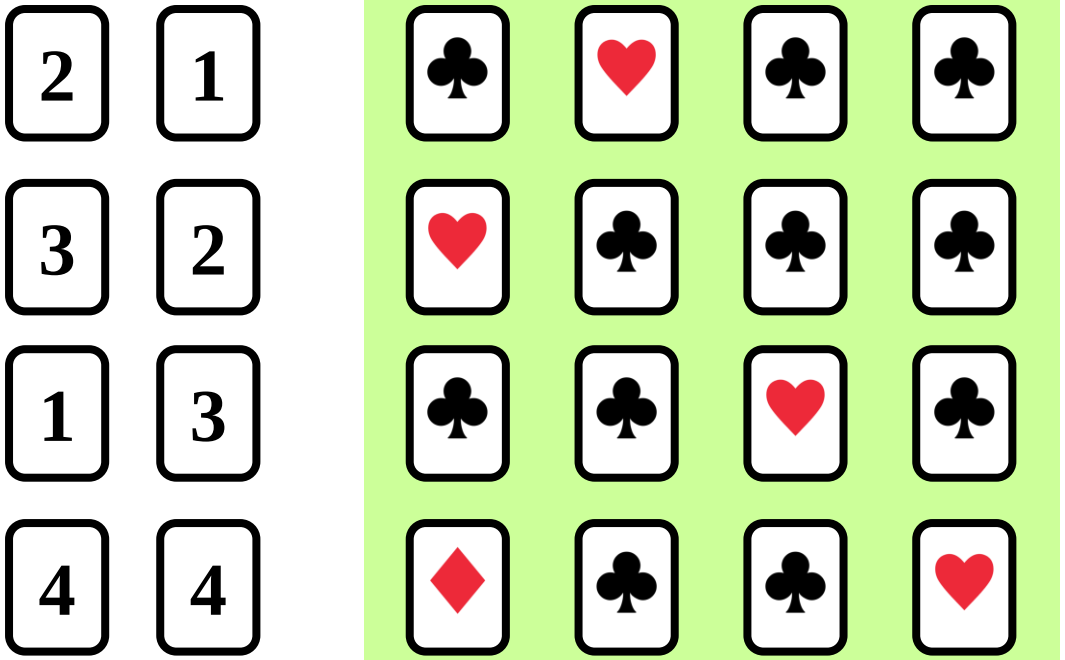
(2 × 2)パズルに対するカードベースZKP

初期



目標

Peggy



緑の領域に
- ♦は4（空白セル）の初期位置
- ♥は移動するタイルの位置

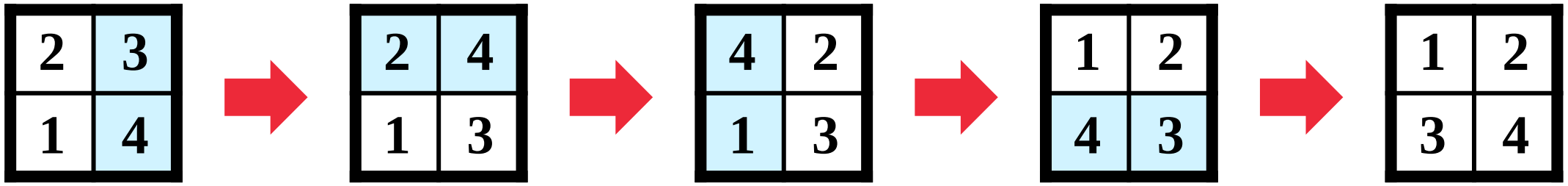
を表すように♦と♥を置く．
（その他は♣で埋める．）

遷移手順で移動するタイルの情報をカードに変換

➢ **緑4列目**：4ステップ目，目標では**4**が置いてある場所のタイルを移動．**4行目**に♥．

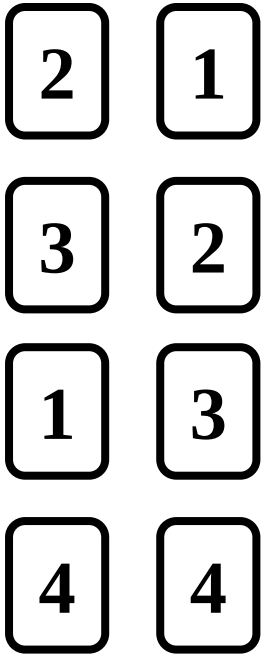
(2 × 2)パズルに対するカードベースZKP

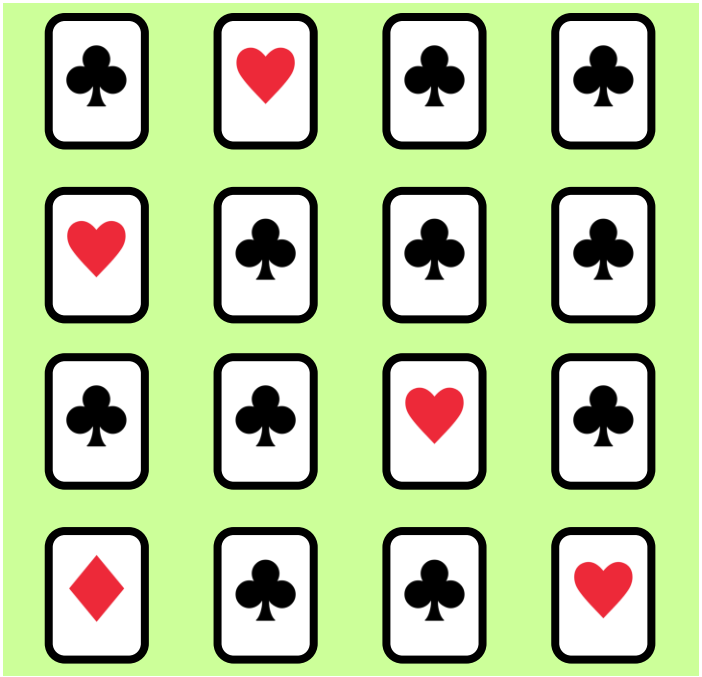
初期



目標

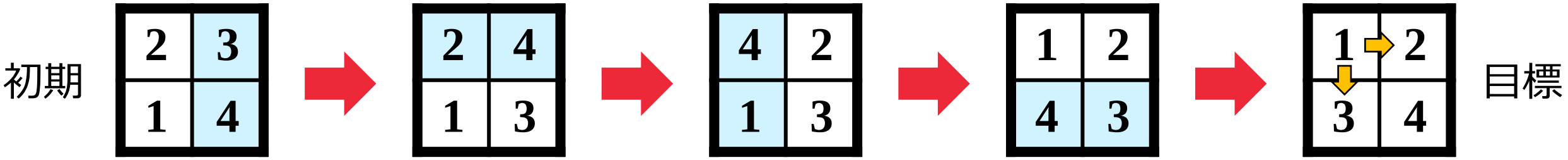
Peggy

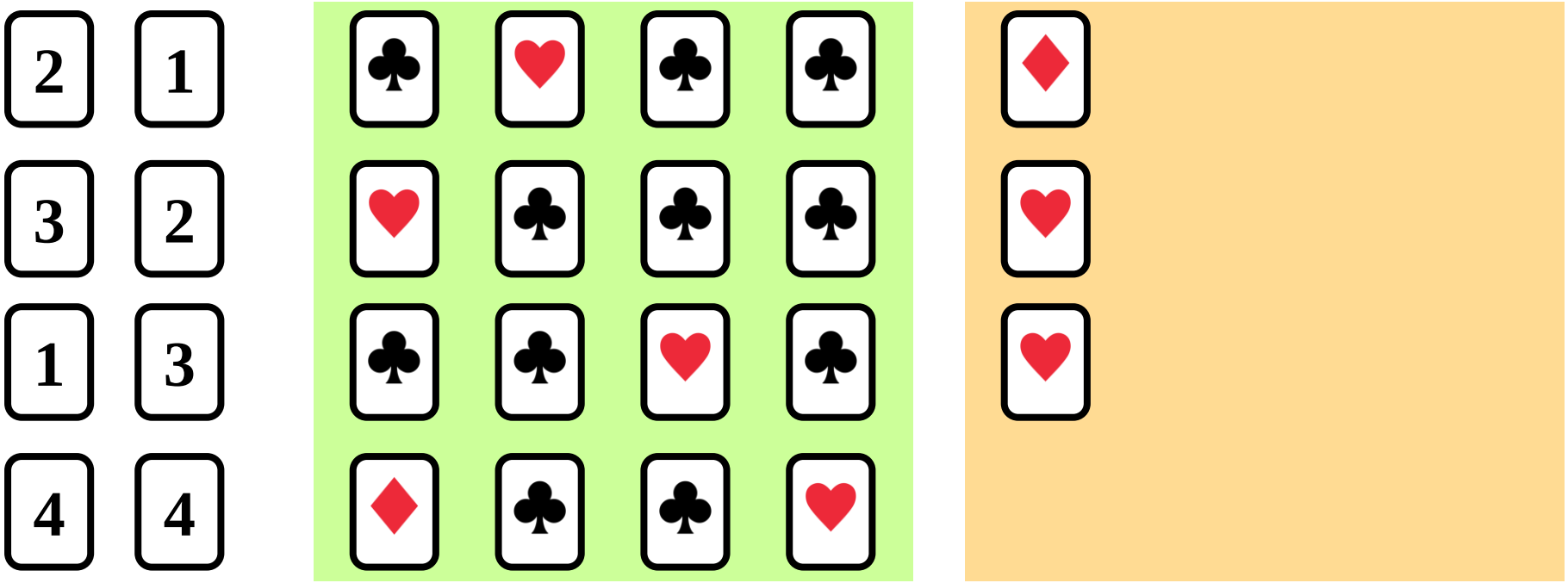




タイル位置の隣接関係
をカードに変換

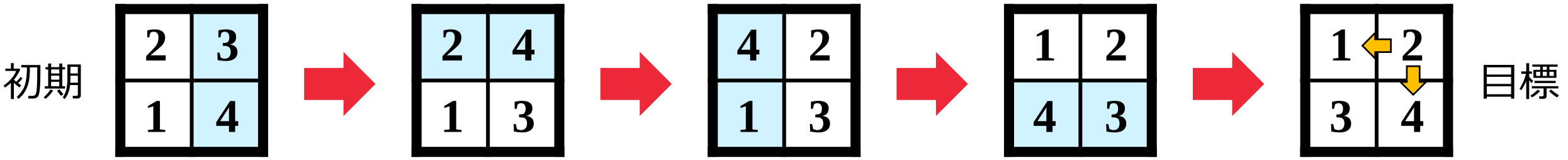
(2 × 2)パズルに対するカードベースZKP

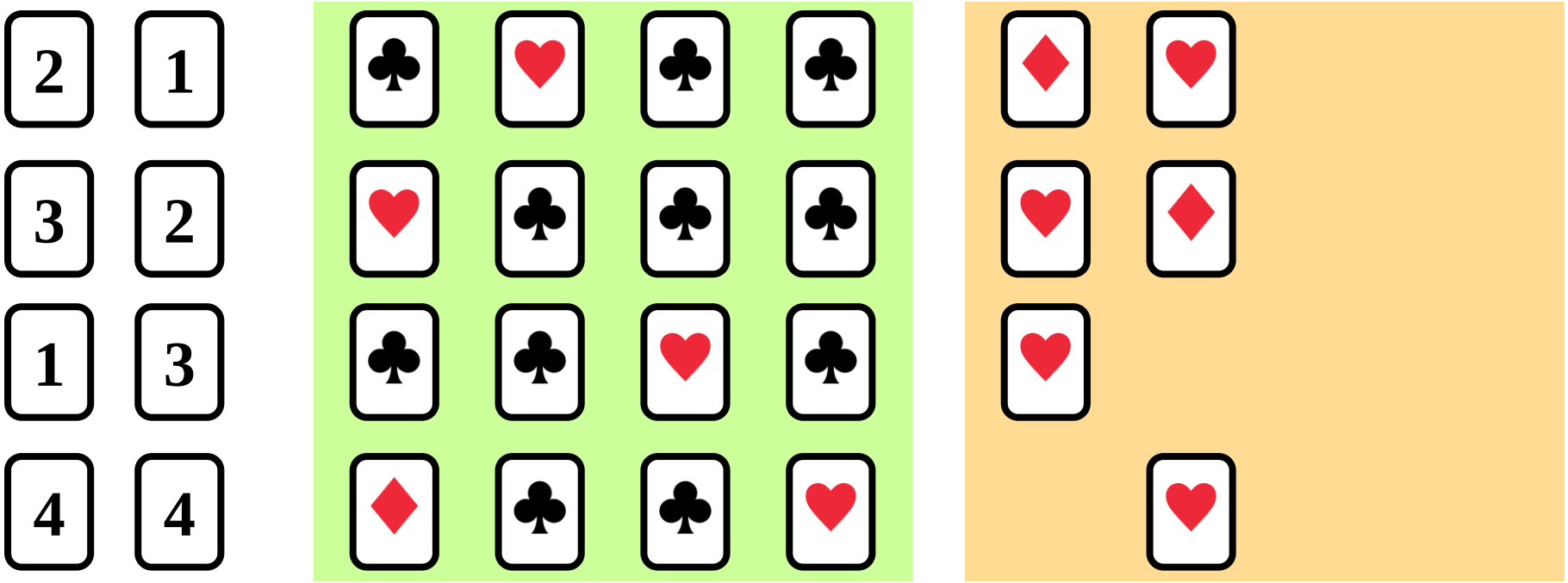




- **橙1列目**：目標において1は2/3と隣接．**1行目**に♦，**2/3行目**に♥．

(2 × 2)パズルに対するカードベースZKP

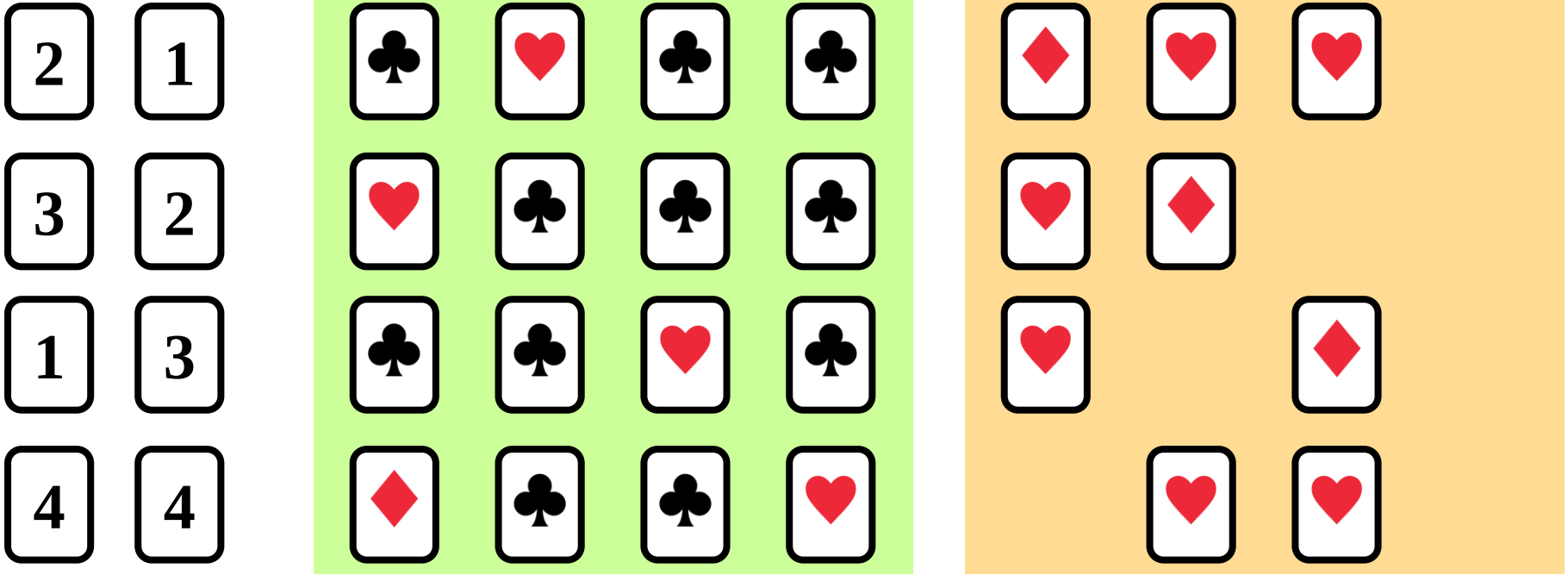




- **橙2列目**：目標において2は1/4と隣接．**2行目**に♦，**1/4行目**に♥．

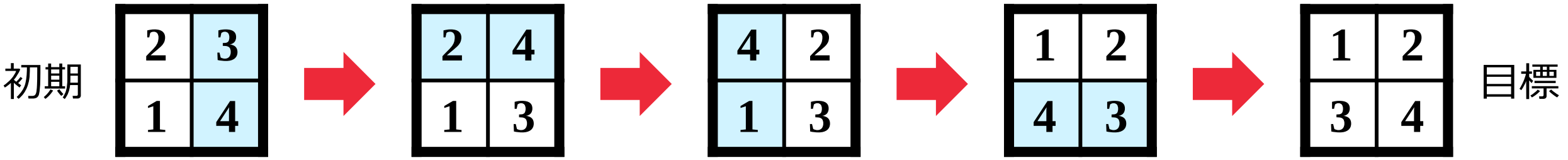
(2 × 2)パズルに対するカードベースZKP

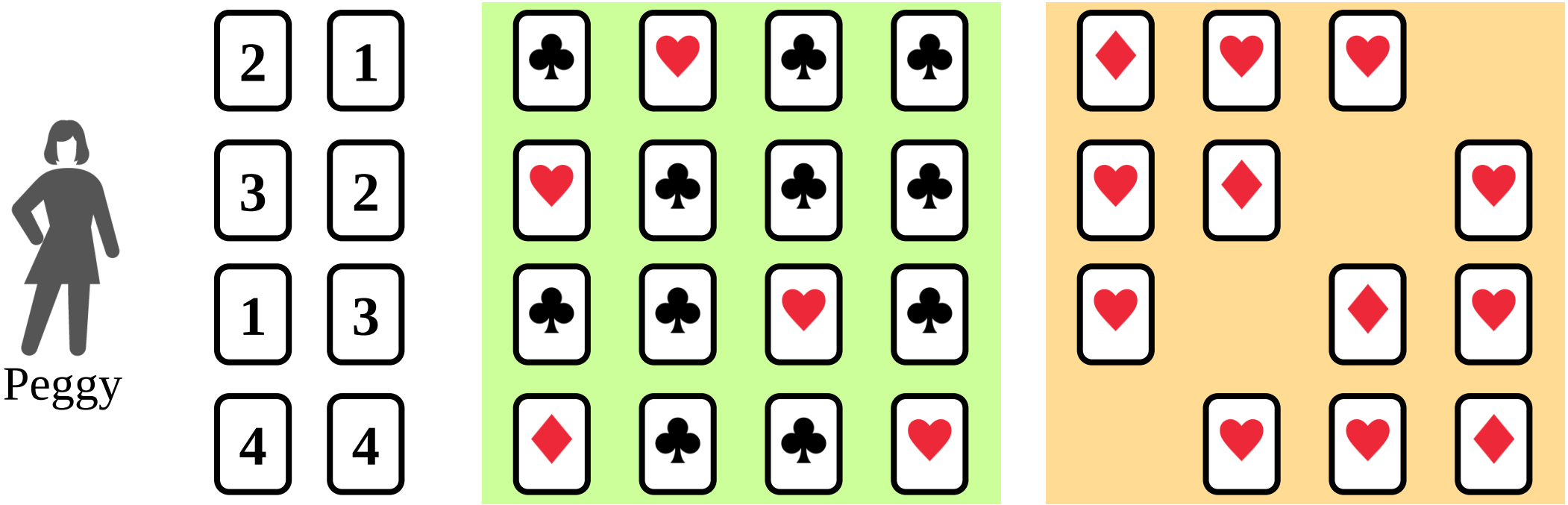
初期 → | 2 | 3 | → | 2 | 4 | → | 4 | 2 | → | 1 | 2 | → | 1 | 2 | 目標
| 1 | 4 | | 1 | 3 | | 1 | 3 | | 4 | 3 | | 3 | 4 |



➢ **橙3列目**：目標において3は1/4と隣接．**3行目**に♦，**1/4行目**に♥．

(2 × 2)パズルに対するカードベースZKP





- **橙4列目**：目標において4は2/3と隣接．**4行目**に♦，**2/3行目**に♥．

(2 × 2)パズルに対するカードベースZKP

初期

2	3
1	4

→

2	4
1	3

→

4	2
1	3

→

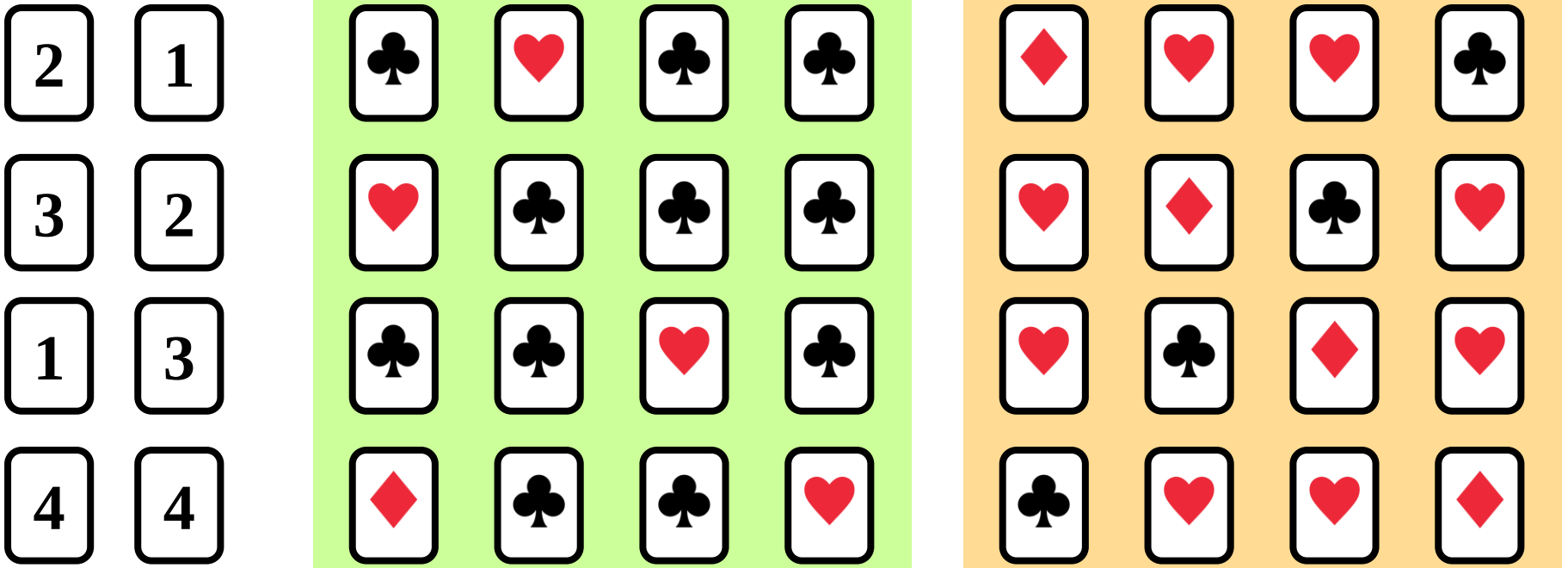
1	2
4	3

→

1	2
3	4

目標

Peggy



- 空いたところには♣のカードを配置

(2 × 2)パズルに対するカードベースZKP

初期

2	3
1	4

➡

2	4
1	3

➡

4	2
1	3

➡

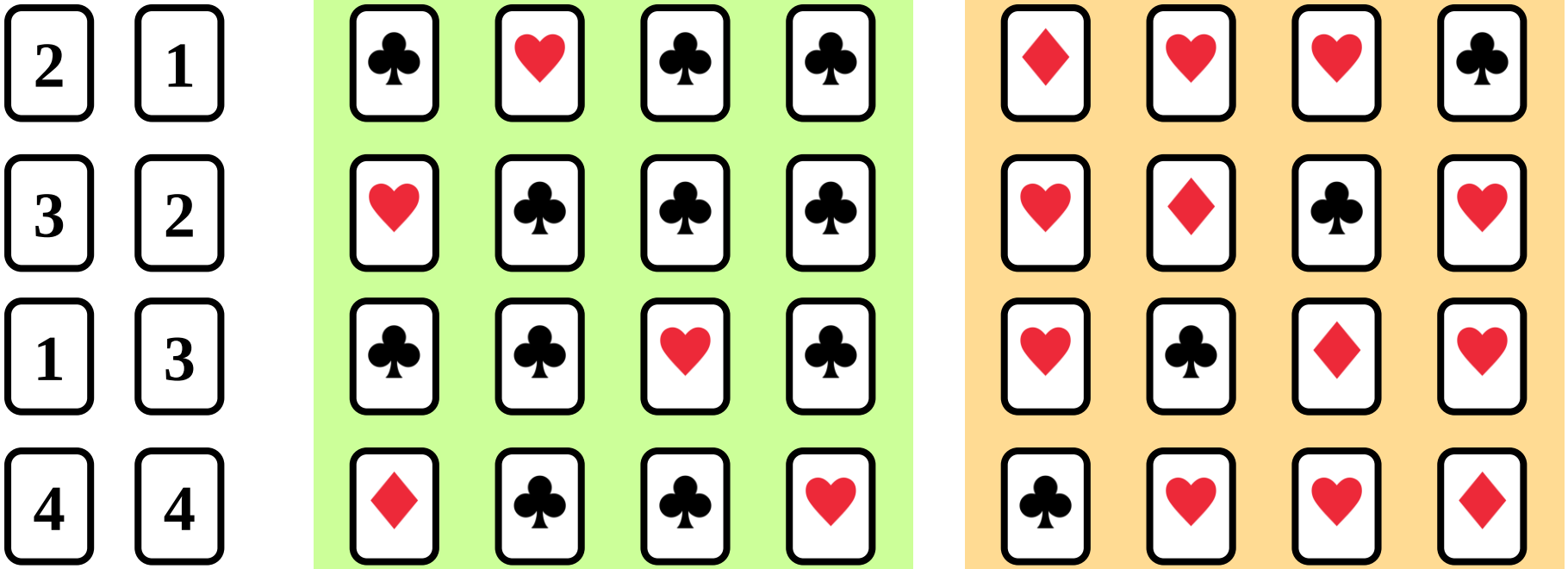
1	2
4	3

➡

1	2
3	4

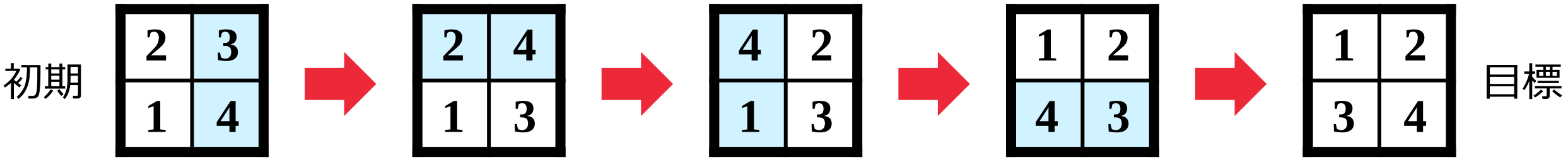
目標

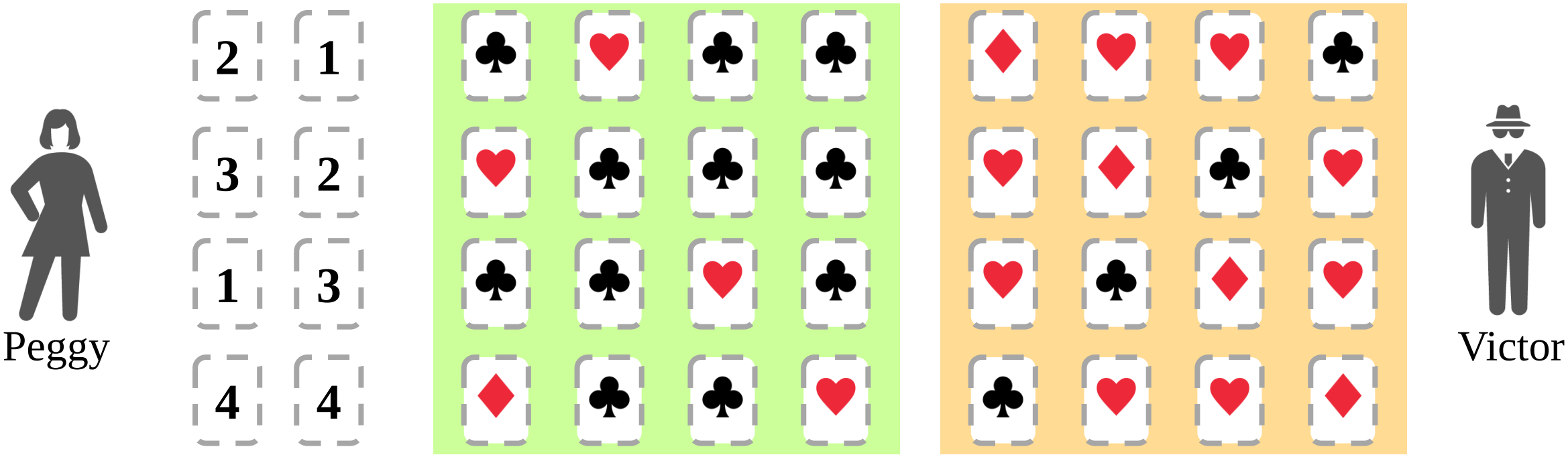
Peggy



- ♦と♥に対応するタイルは隣接 → **タイルの交換可能性を表現**

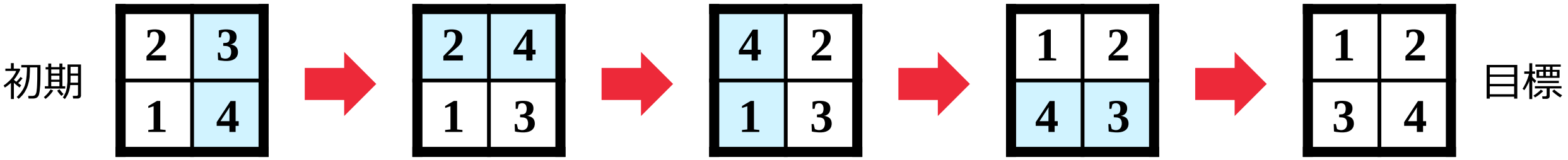
(2 × 2)パズルに対するカードベースZKP

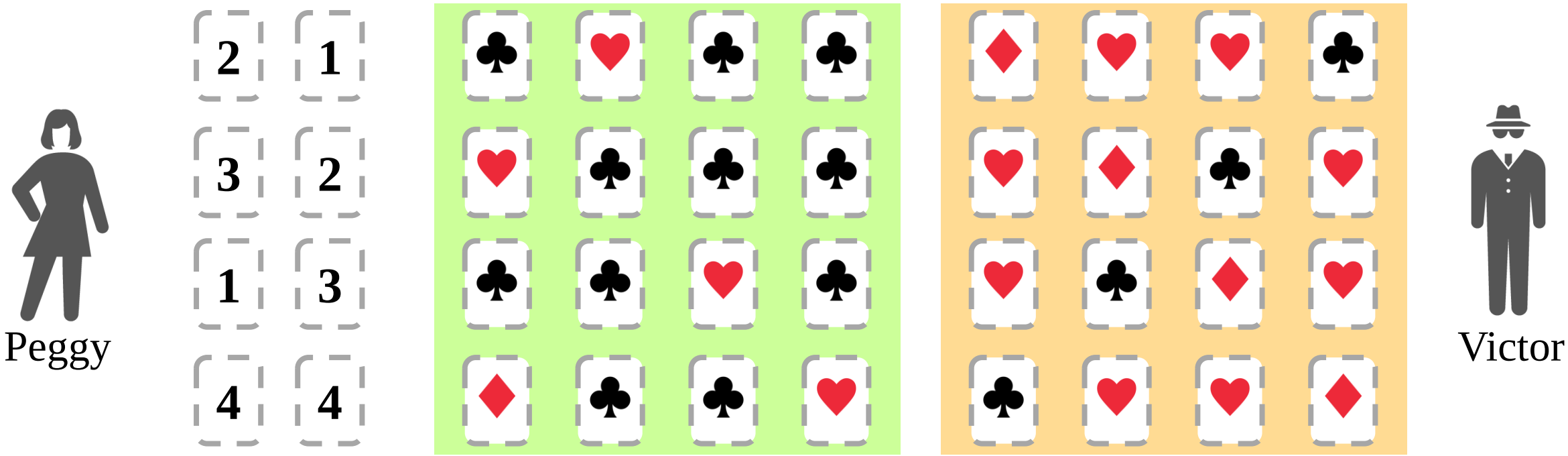




- 準備完了．全てのカードを裏返し（灰色点線）にした後，Victorを呼ぶ.

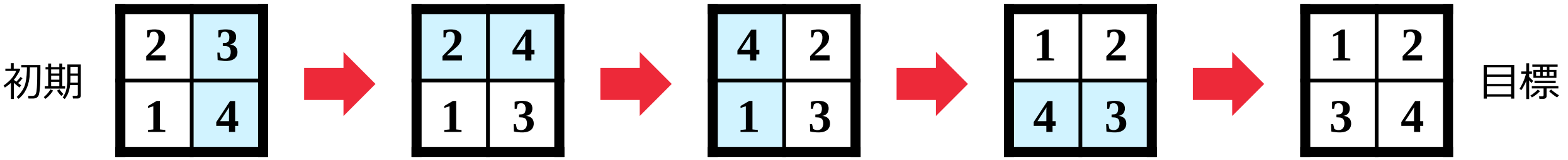
(2 × 2)パズルに対するカードベースZKP

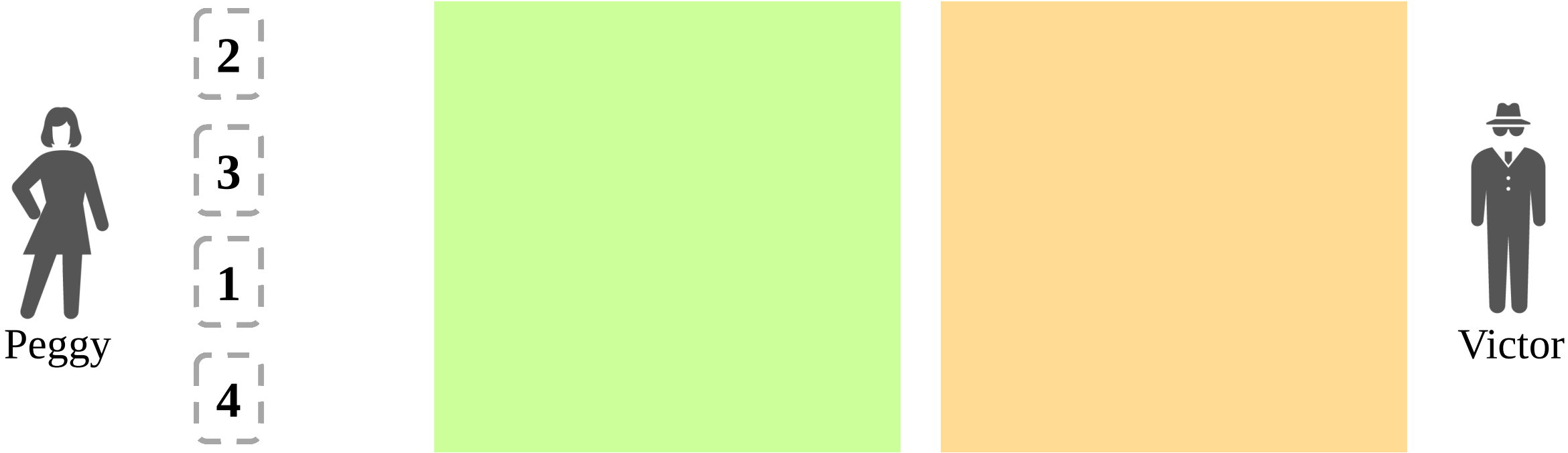




➢ 行に関するパイルシャッフルを実行．**移動するタイル情報を秘匿．**

(2 × 2)パズルに対するカードベースZKP

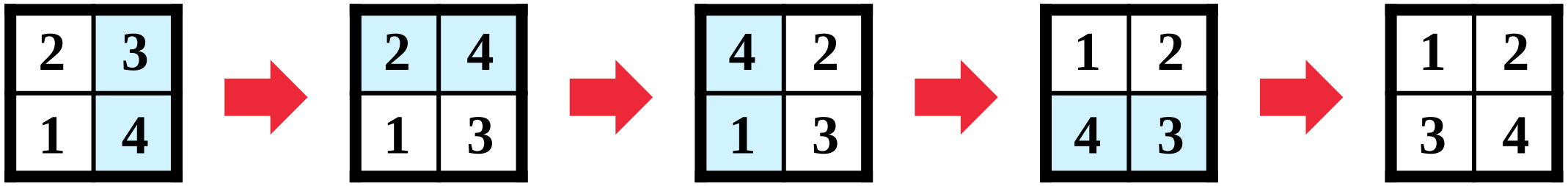




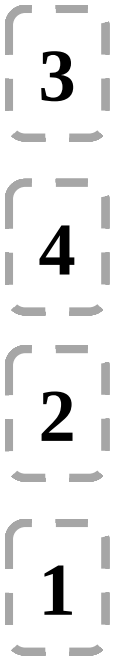
- 行に関するパイルシャッフルを実行．**移動するタイル情報を秘匿．**

(2 × 2)パズルに対するカードベースZKP

初期



目標

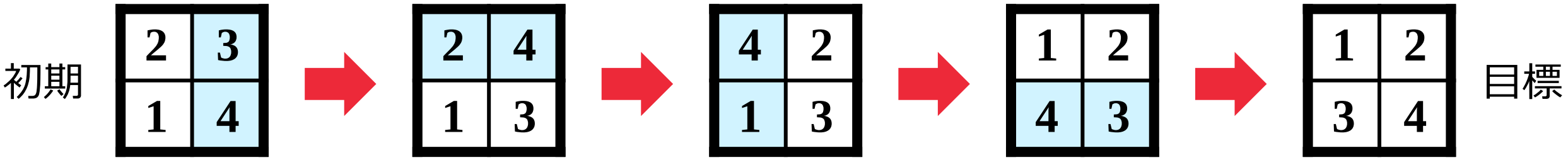


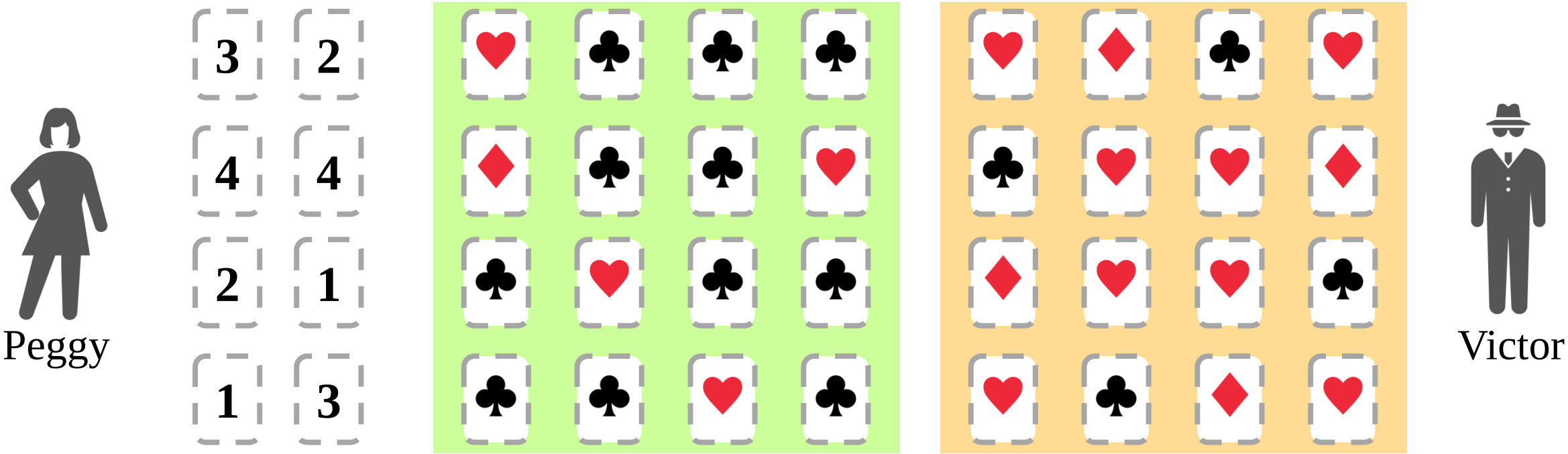
Peggy

Victor

- 行に関するパイルシャッフルを実行．**移動するタイル情報を秘匿．**

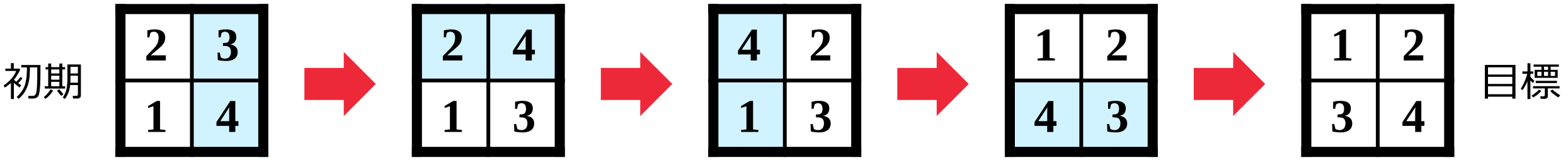
(2 × 2)パズルに対するカードベースZKP





- 行に関するパイルシャッフルを実行．**移動するタイル情報を秘匿．**

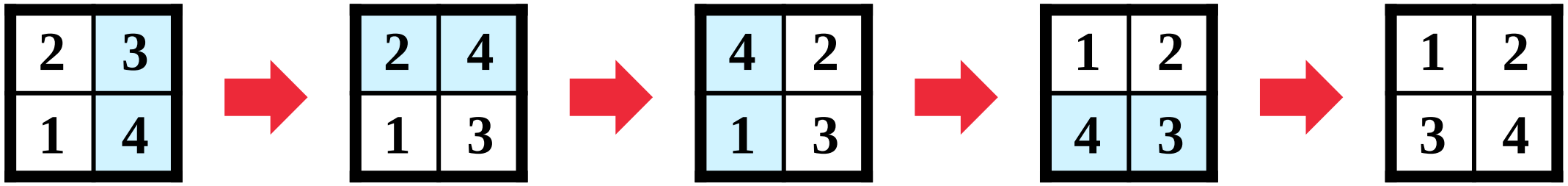
(2 × 2)パズルに対するカードベースZKP



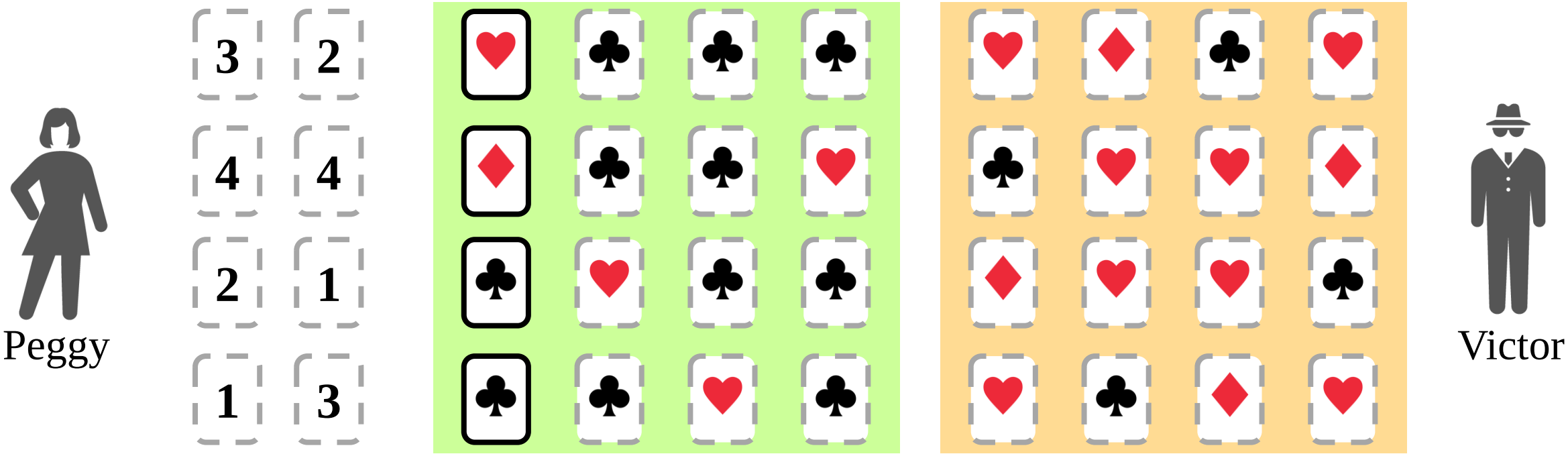
- シャッフル後，**緑領域の1列目**のカードを表にする．

(2 × 2)パズルに対するカードベースZKP

初期

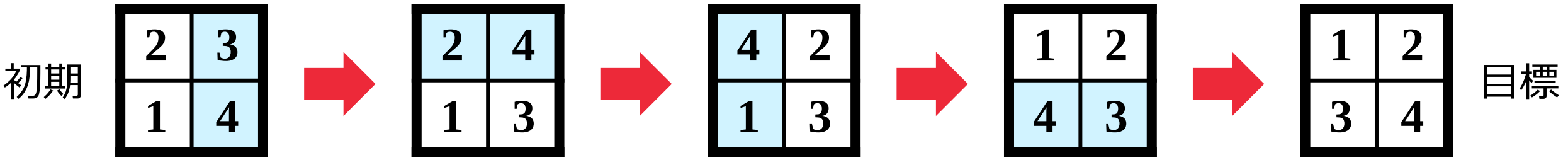


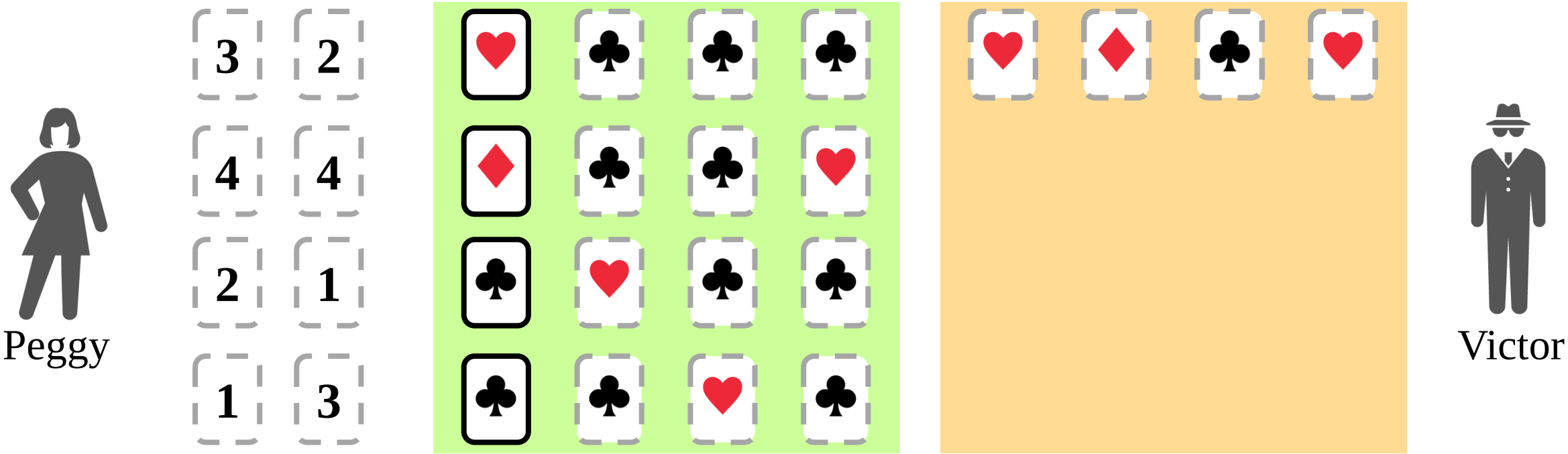
目標



➢ オレンジ領域のカードに対し，**列に関する**パイルシャッフルを実行.

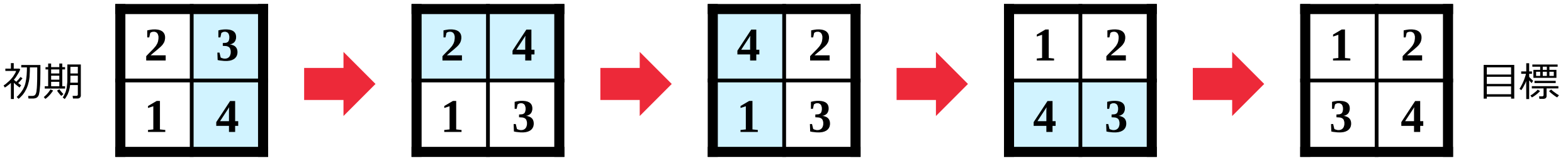
(2 × 2)パズルに対するカードベースZKP

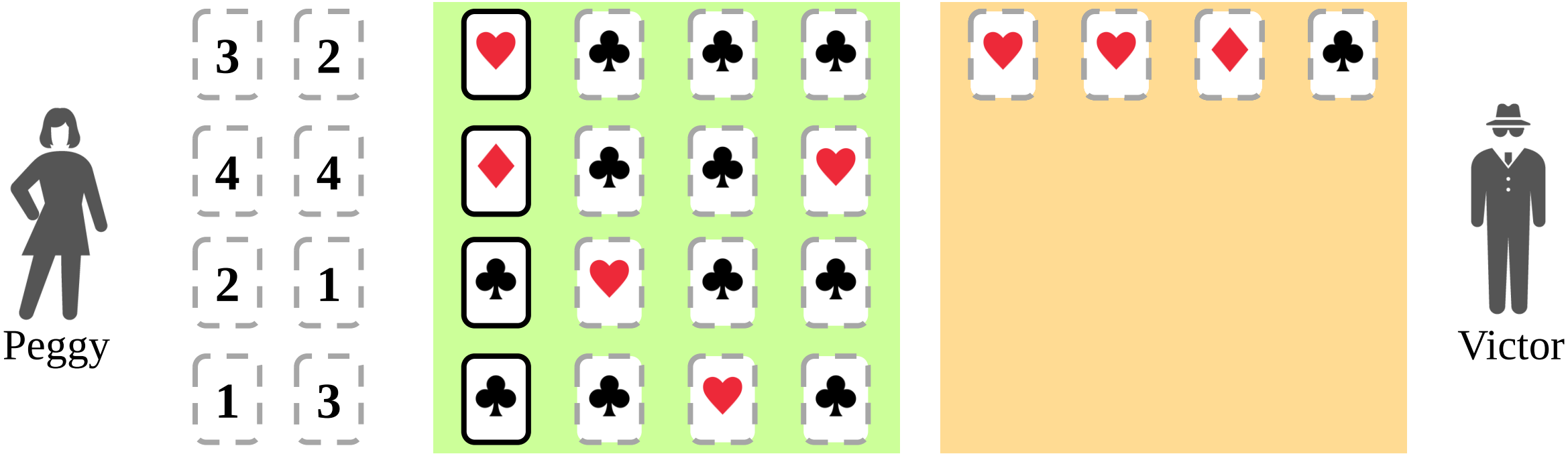




➢ オレンジ領域のカードに対し，**列に関する**パイルシャッフルを実行.

(2 × 2)パズルに対するカードベースZKP





➢ オレンジ領域のカードに対し，**列に関する**パイルシャッフルを実行.

(2 × 2)パズルに対するカードベースZKP

初期

2	3
1	4

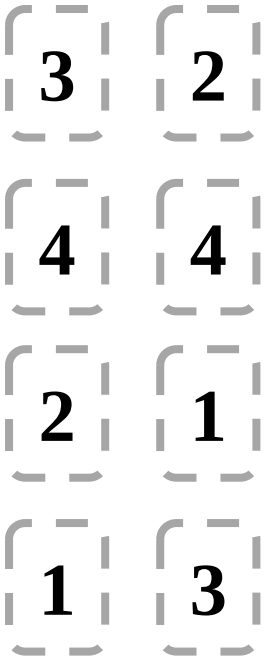
2	4
1	3

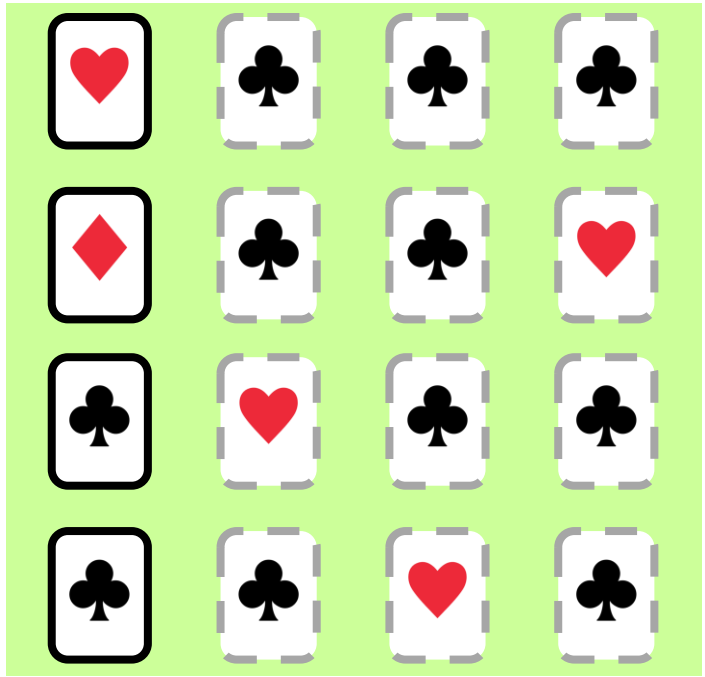
タイルの隣接情報を秘匿しつつ
列の情報は変えない

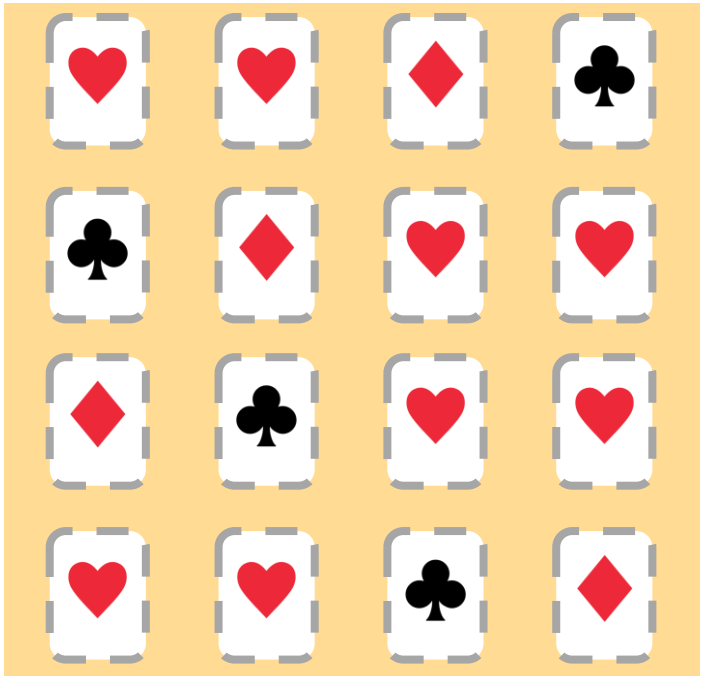
1	2
3	4

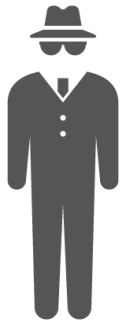
目標

Peggy





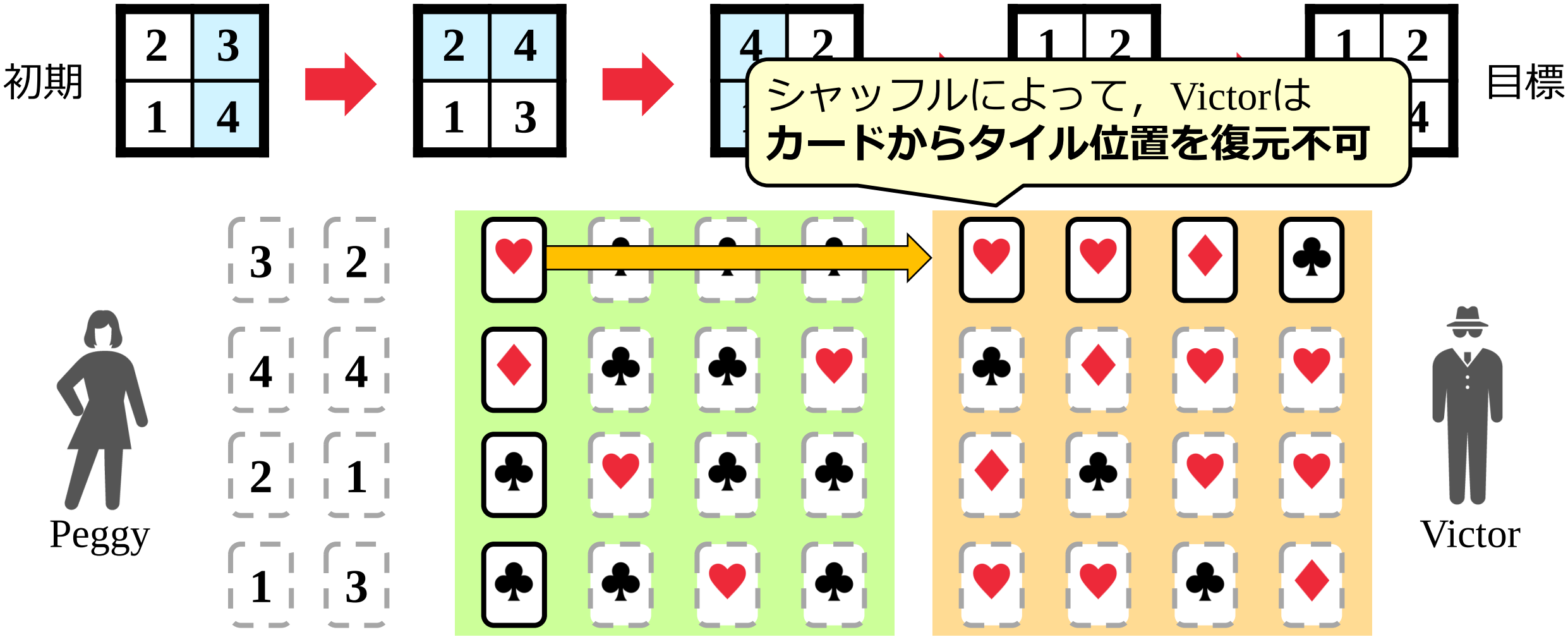




Victor

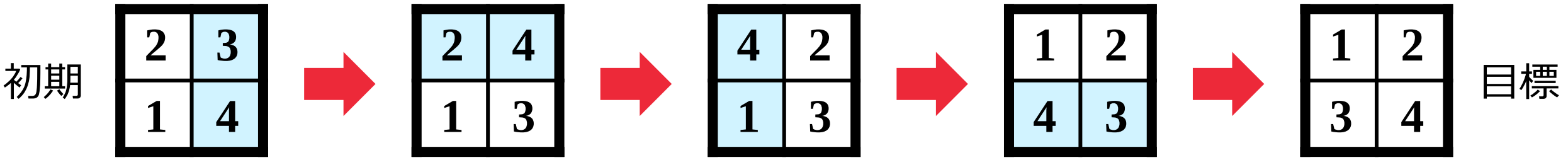
➢ オレンジ領域のカードに対し，**列に関する**パイルシャッフルを実行.

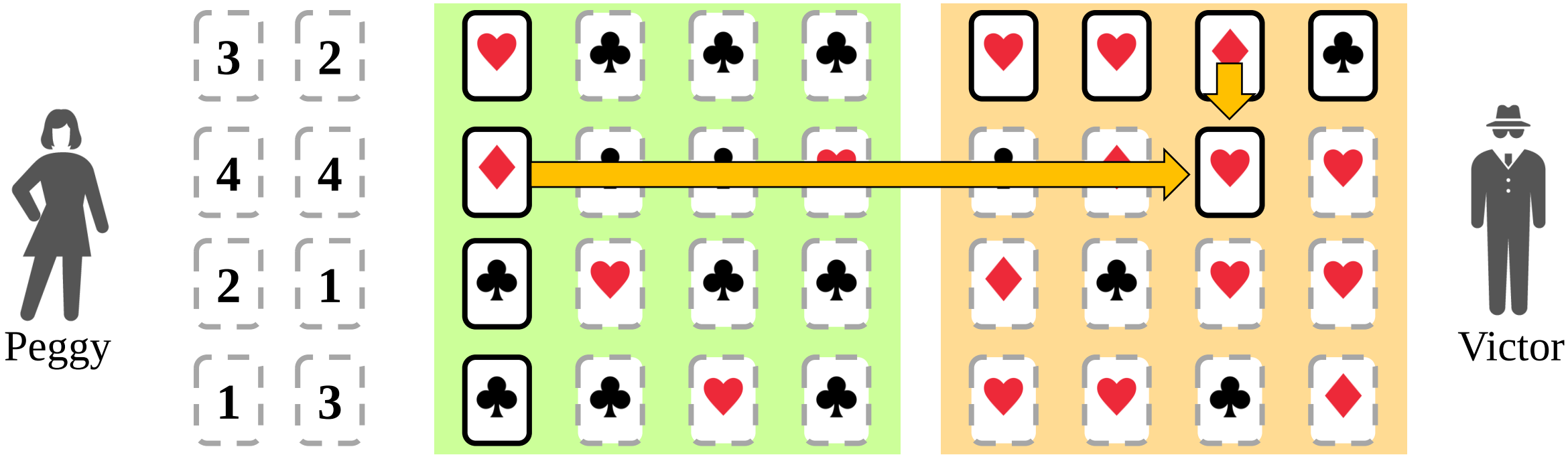
(2 × 2)パズルに対するカードベースZKP



➢ オレンジ領域において，緑領域で見えている♥と同じ行のカードを表にする．

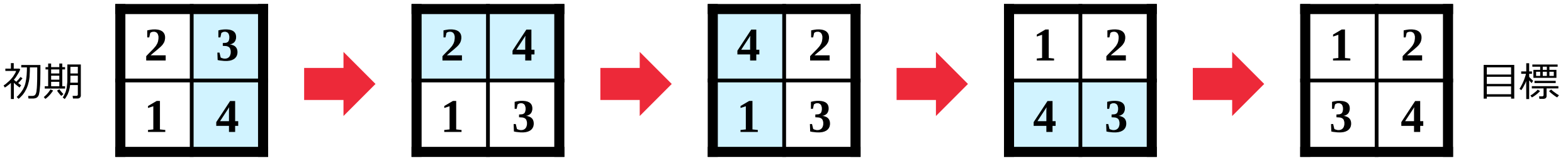
(2 × 2)パズルに対するカードベースZKP

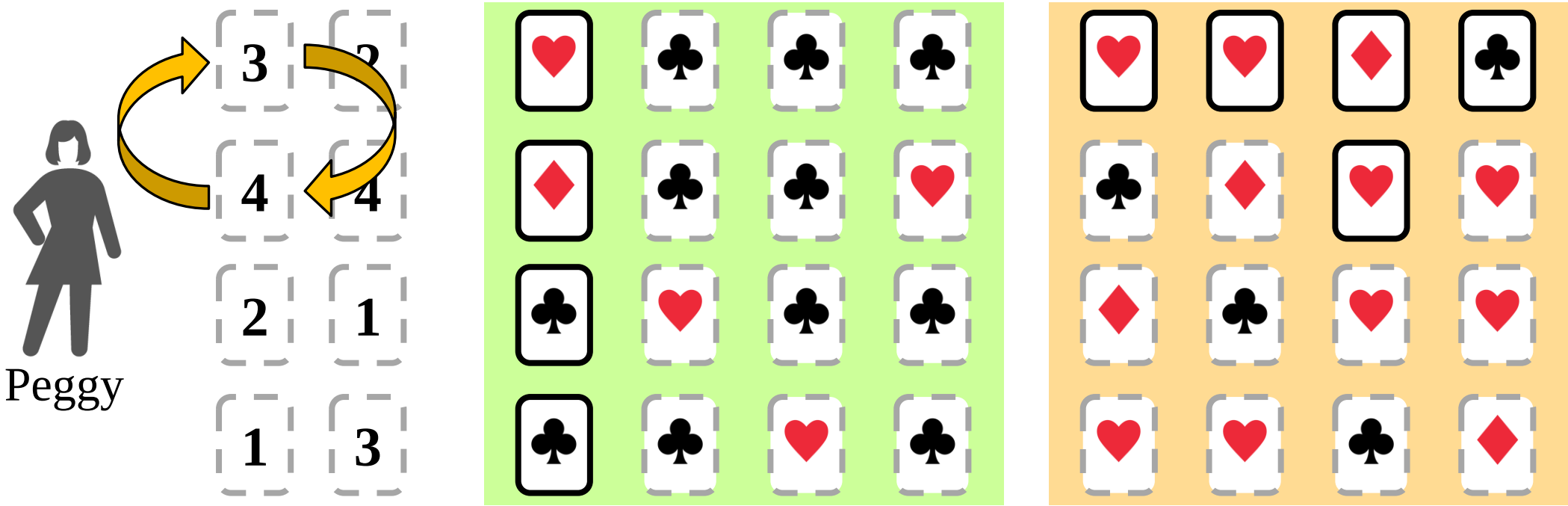




➢ 2つの♦の交点に位置するカードを表にする．　（**交換可能であることの確認**）

(2 × 2)パズルに対するカードベースZKP

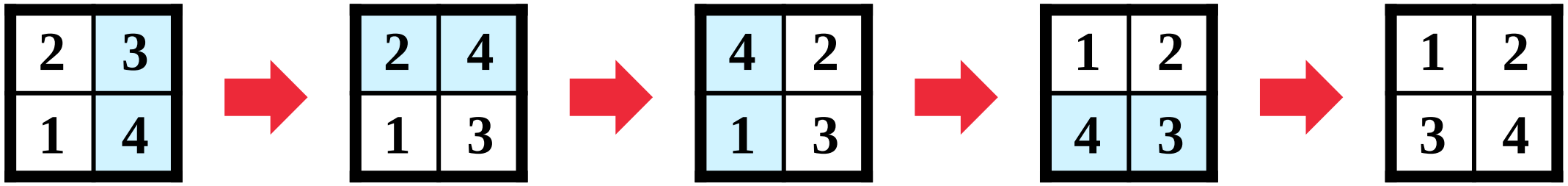




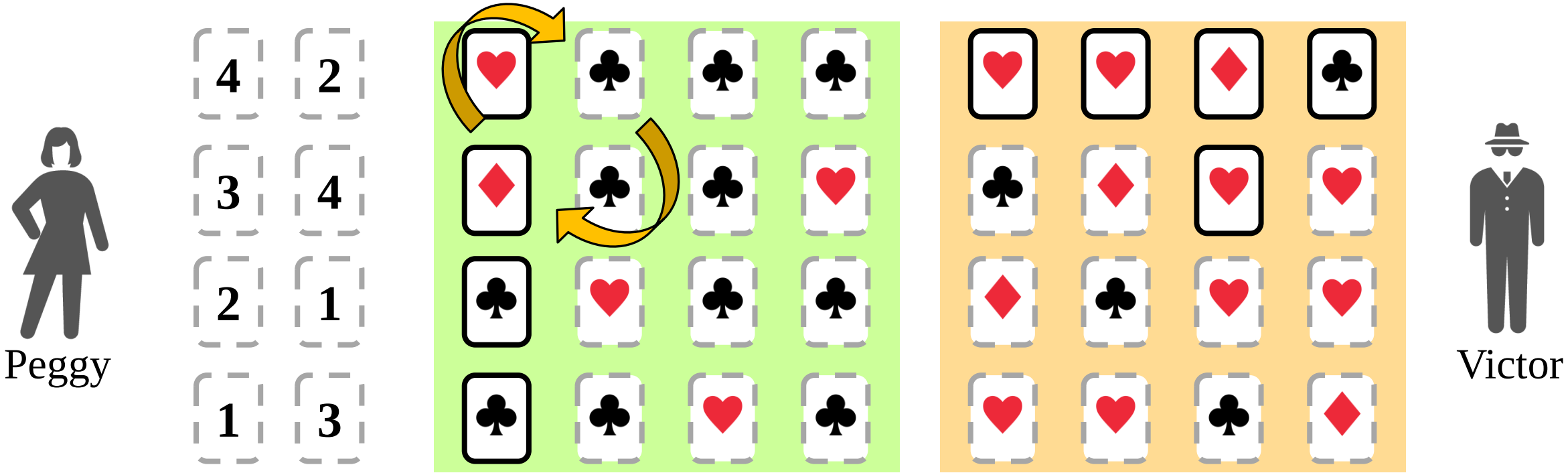
- **1列目**：♦が見えている行のカードを交換（＝**タイルの交換**）．

(2 × 2)パズルに対するカードベースZKP

初期

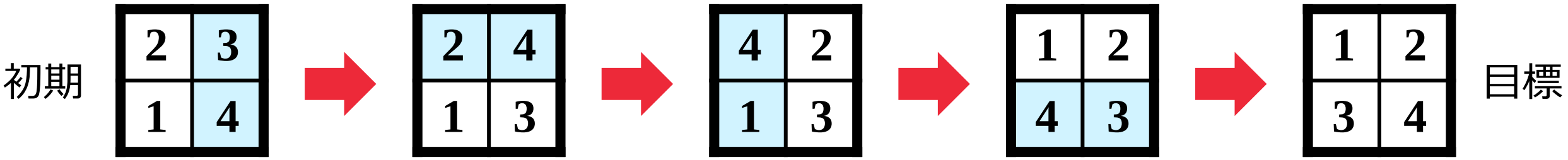


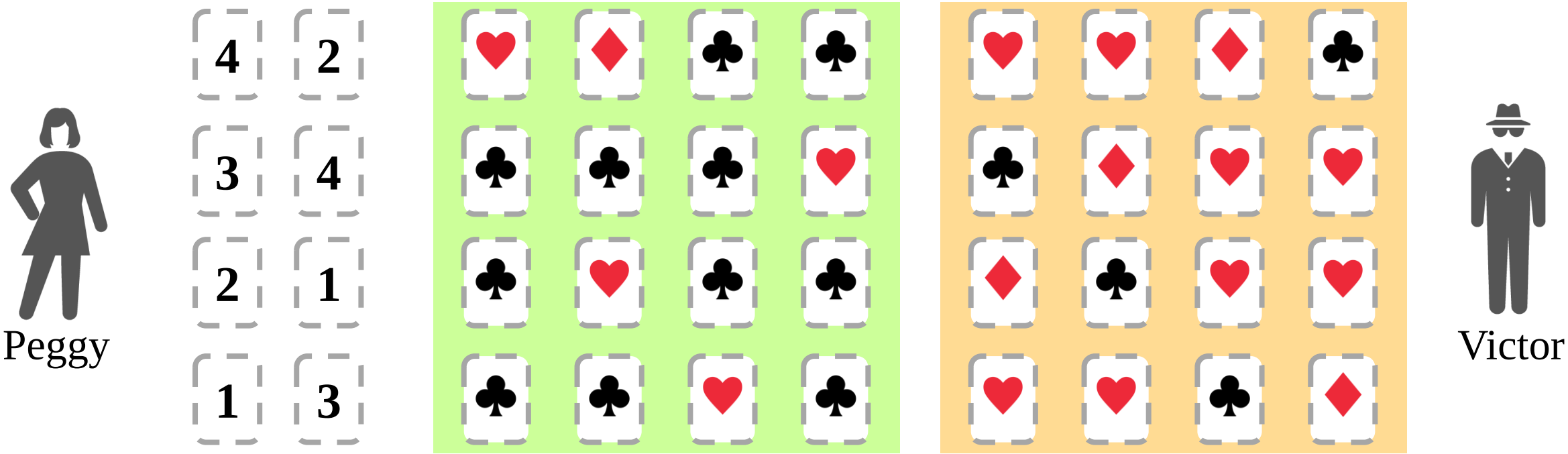
目標



- 緑領域の♦と♥の**右隣り**のカードを交換.　（**空白タイルの移動**）

(2 × 2)パズルに対するカードベースZKP





- 全てのカードを裏にして，行に関するパイルシャッフルを実行.

(2 × 2)パズルに対するカードベースZKP

初期

2	3
1	4

→

2	4
1	3

→

4	2
1	3

→

1	2
4	3

→

1	2
3	4

目標

Peggy

4

3

2

1

Victor

- 全てのカードを裏にして，行に関するパイルシャッフルを実行.

(2 × 2)パズルに対するカードベースZKP

初期

2	3
1	4

→

2	4
1	3

→

4	2
1	3

→

1	2
4	3

→

1	2
3	4

目標

Peggy

1

3

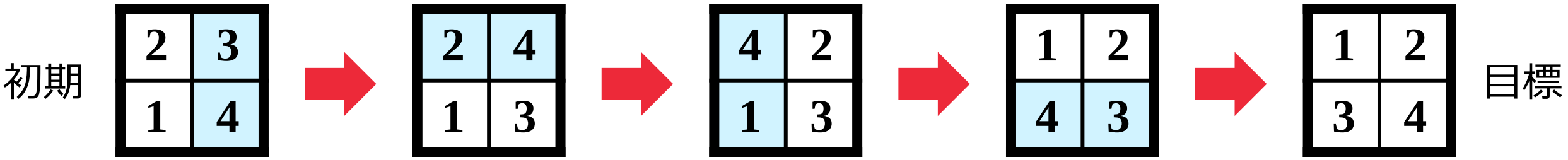
2

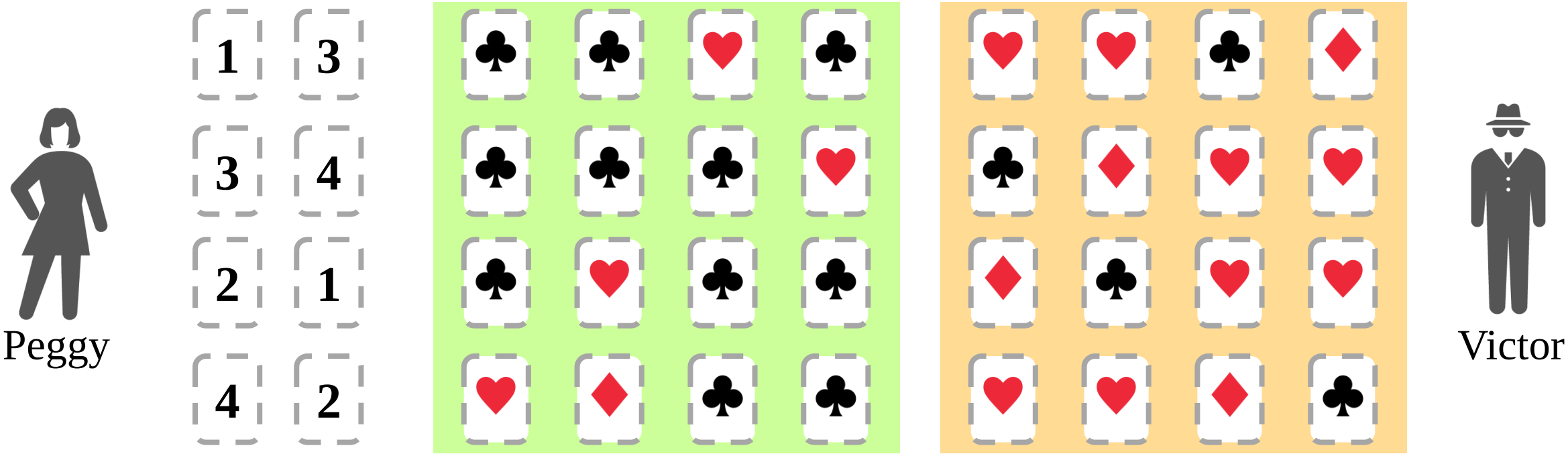
4

Victor

- 全てのカードを裏にして，行に関するパイルシャッフルを実行.

(2 × 2)パズルに対するカードベースZKP

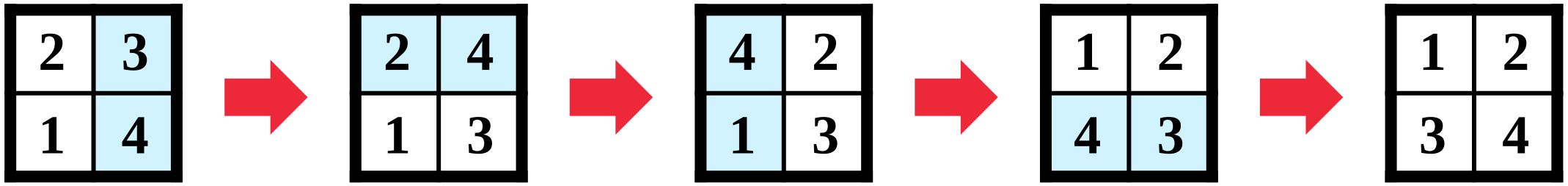




- 全てのカードを裏にして，行に関するパイルシャッフルを実行.

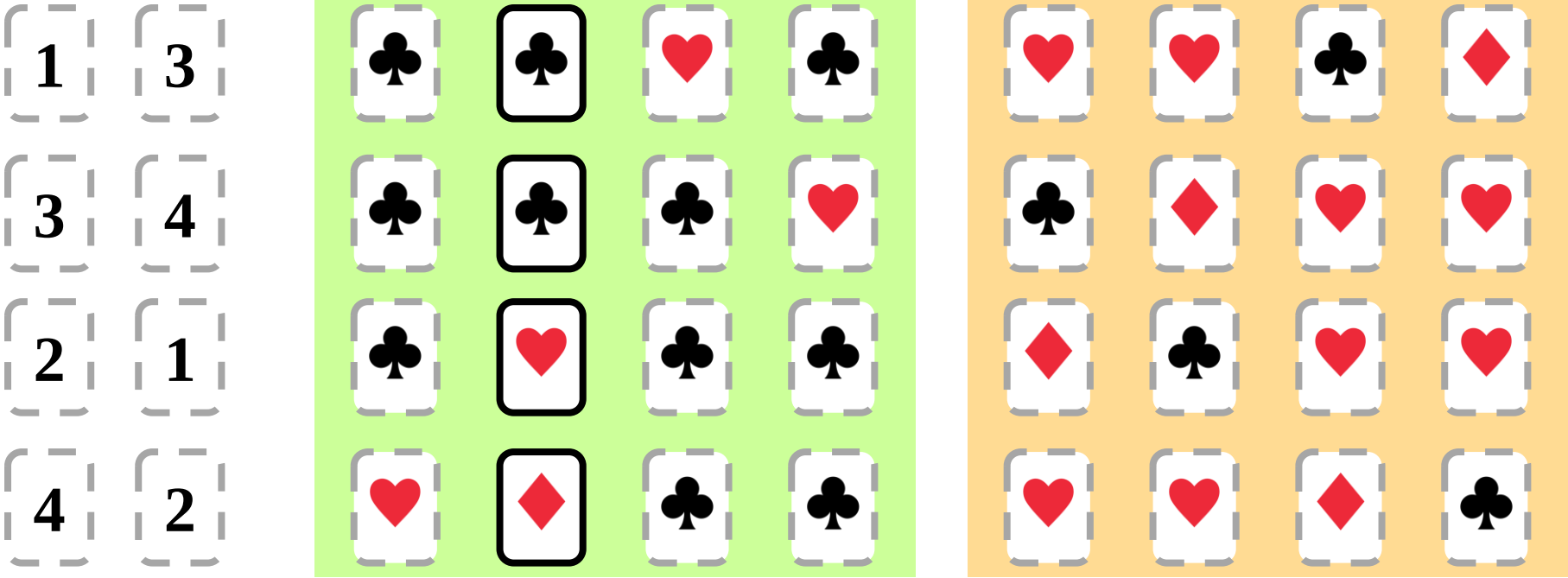
(2 × 2)パズルに対するカードベースZKP

初期



目標

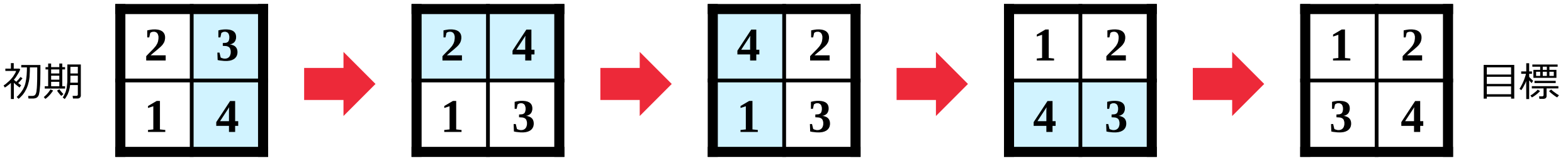
Peggy

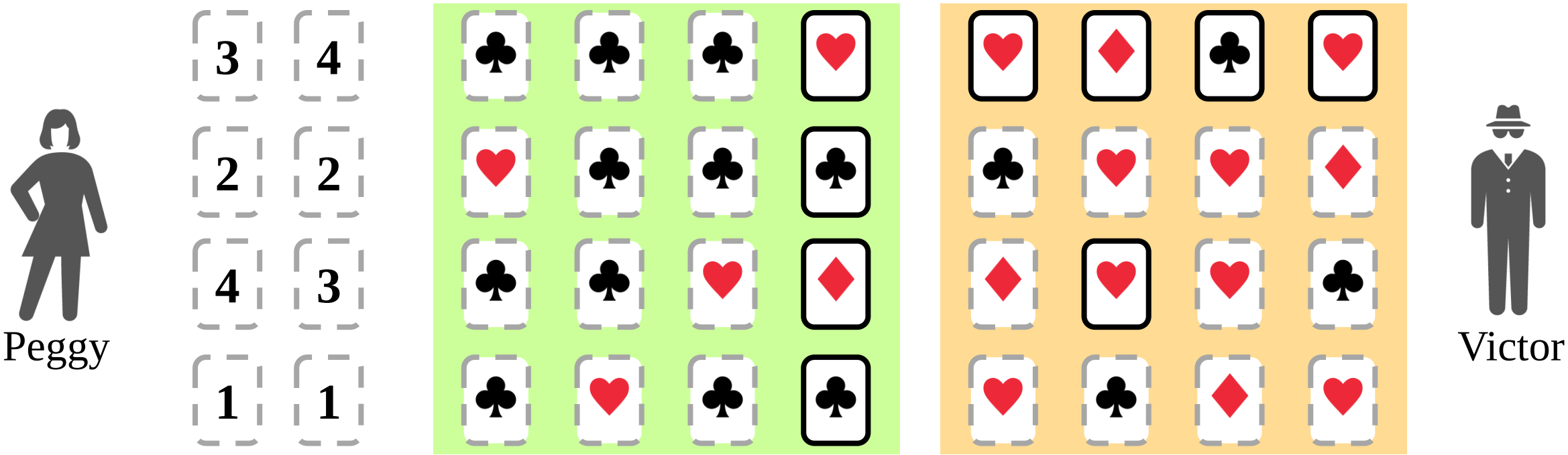


Victor

- ここまで1ステップ．同様の操作を緑領域の2列目に行う．

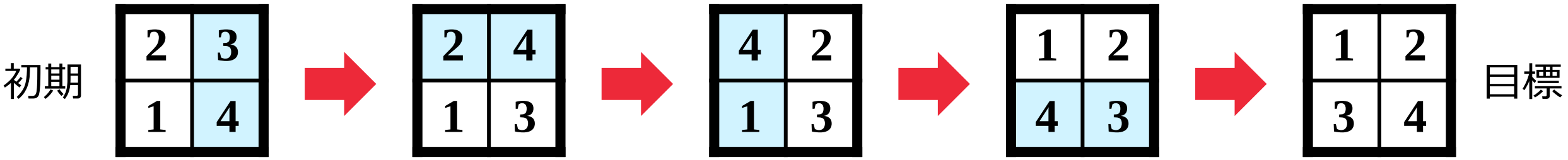
(2 × 2)パズルに対するカードベースZKP

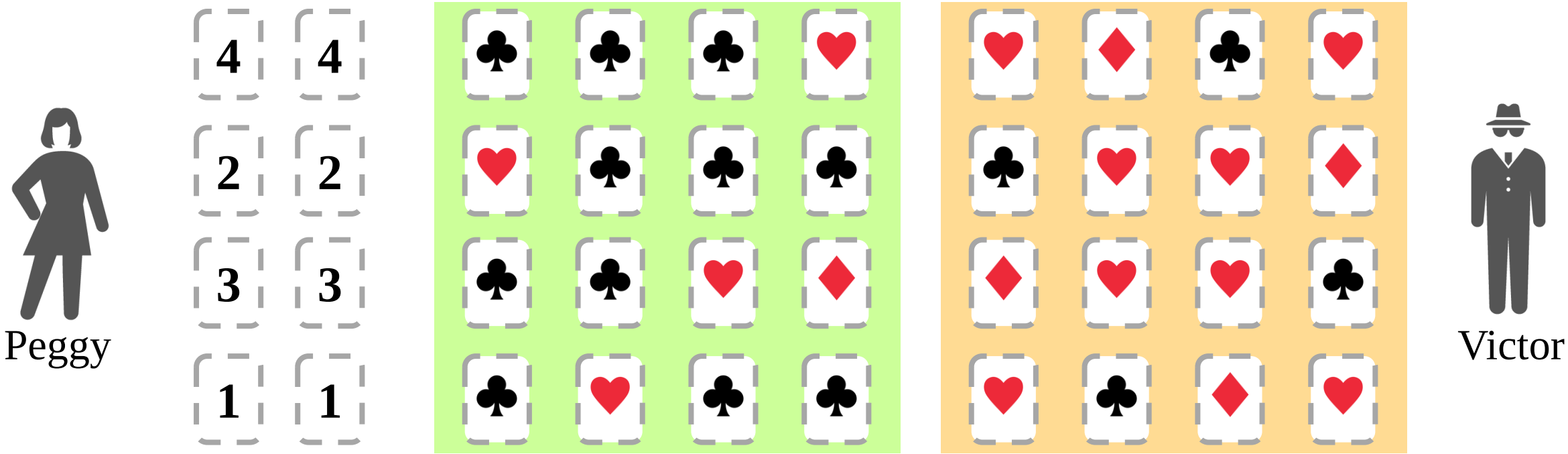




- この操作を緑領域の最後の列まで行う.

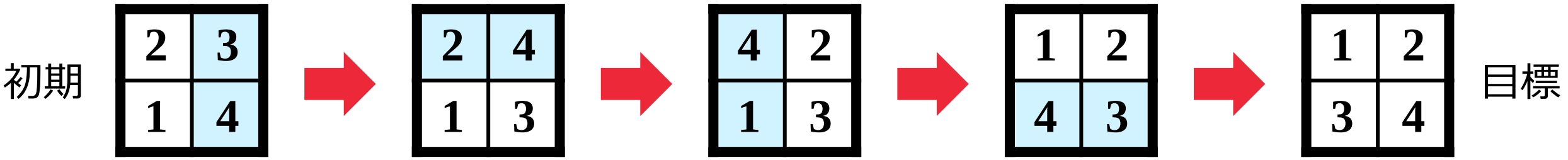
(2 × 2)パズルに対するカードベースZKP

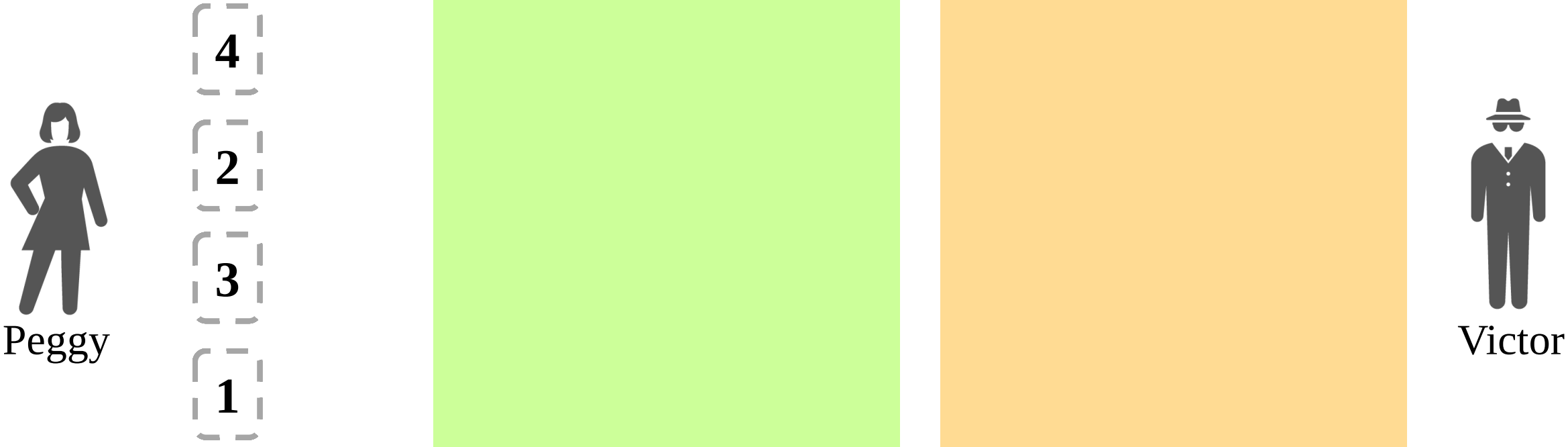




- 最後のスワップ後，カードを裏返して行に関するパイルシャッフルを実行.

(2 × 2)パズルに対するカードベースZKP

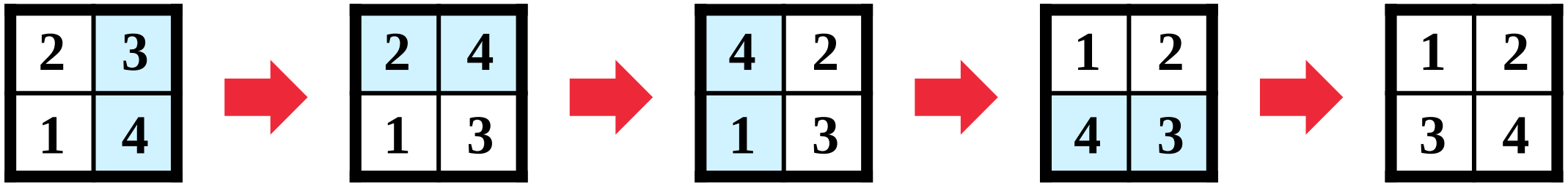




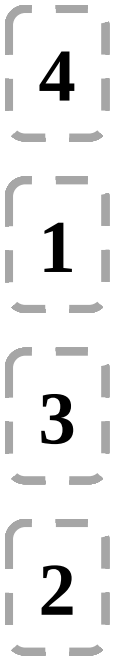
- 最後のスワップ後，カードを裏返して行に関するパイルシャッフルを実行.

(2 × 2)パズルに対するカードベースZKP

初期



目標

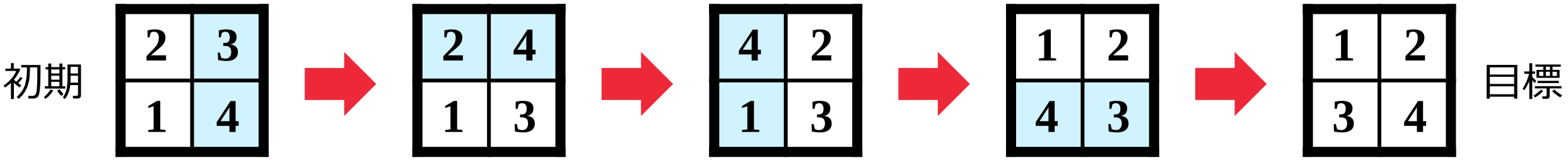


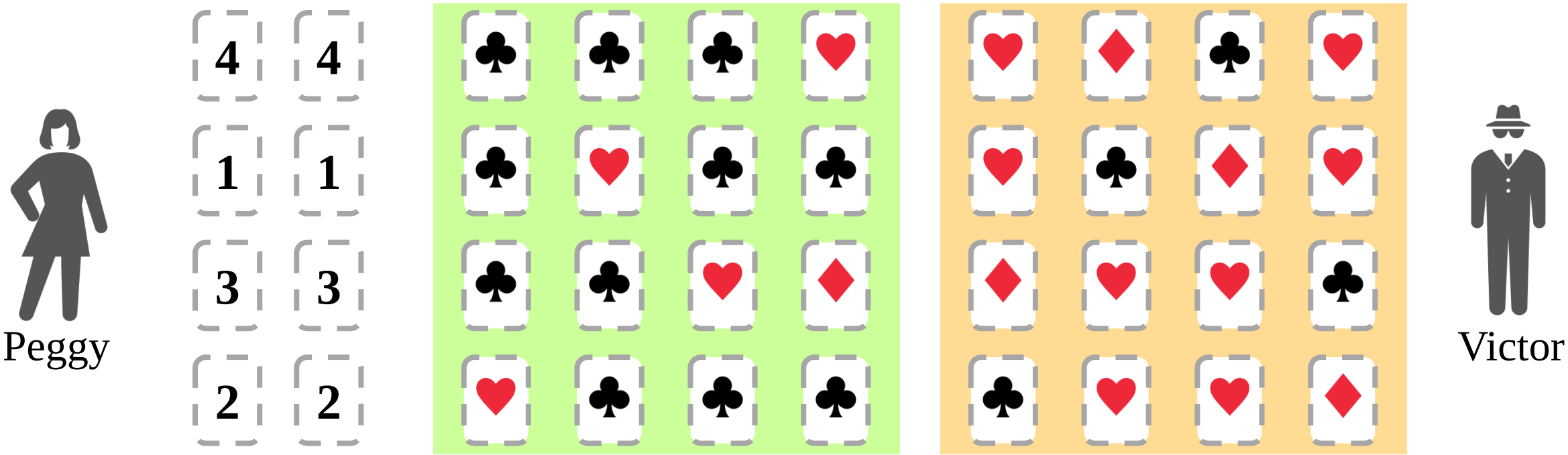
Peggy

Victor

- 最後のスワップ後，カードを裏返して行に関するパイルシャッフルを実行.

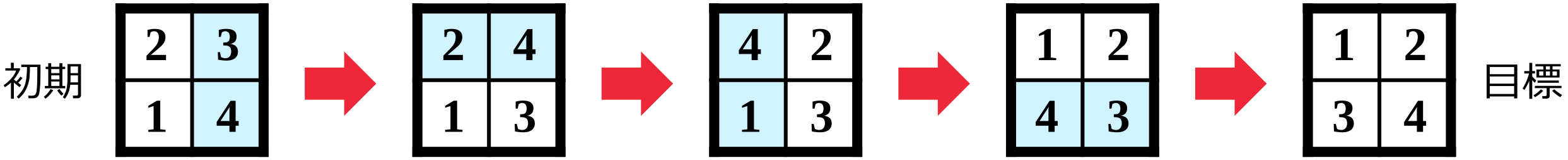
(2 × 2)パズルに対するカードベースZKP

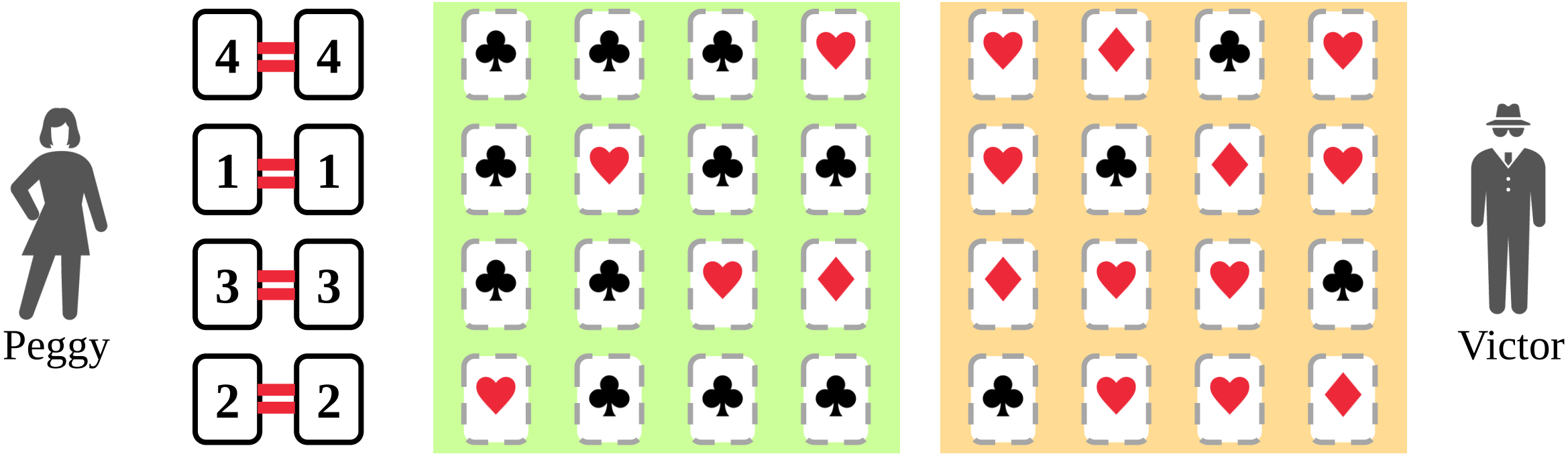




- 最後のスワップ後，カードを裏返して行に関するパイルシャッフルを実行.

(2 × 2)パズルに対するカードベースZKP





- 最後に，**1列目**と**2列目**を表にする．正しい遷移手順なら**数字が一致**している．

$(w \times h)$パズルに対するカードベースZKP

プロトコルの正当性

➢ 完全性

Peggyが遷移手順通りに正しくカードを並べたなら，必ず最後のカードが一致する.

➢ 健全性

Peggyがでたらめにカードを並べたなら，

- 緑領域のカードを開いたとき♦と♥の数が合わない
- オレンジ領域のカードにより，隣接しないタイル移動を行おうとしている
- 最後のカードが一致しない

ことが判明し，Peggyの嘘を**必ず検知できる**.　（**健全性エラー0**）

➢ ゼロ知識性

(2×2)パズルならOK．それ以上の盤面サイズだと**情報漏洩の危険性有**.

ゼロ知識性

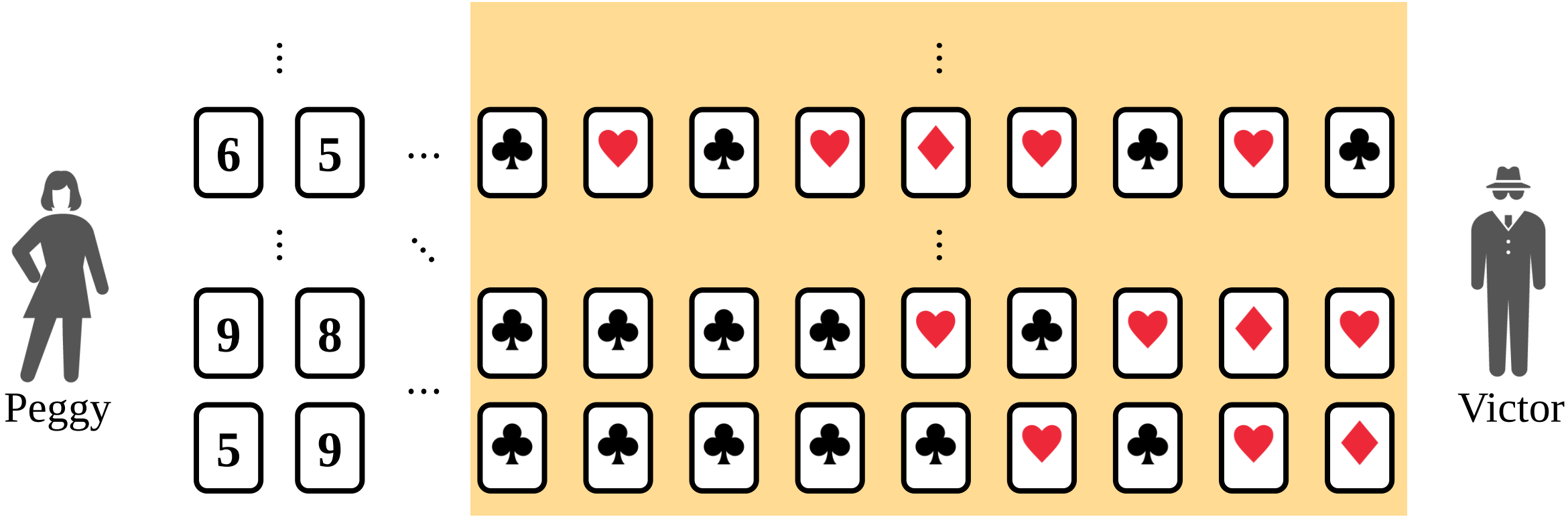
初期

2	8	3
1	6	4
7	9	5

?

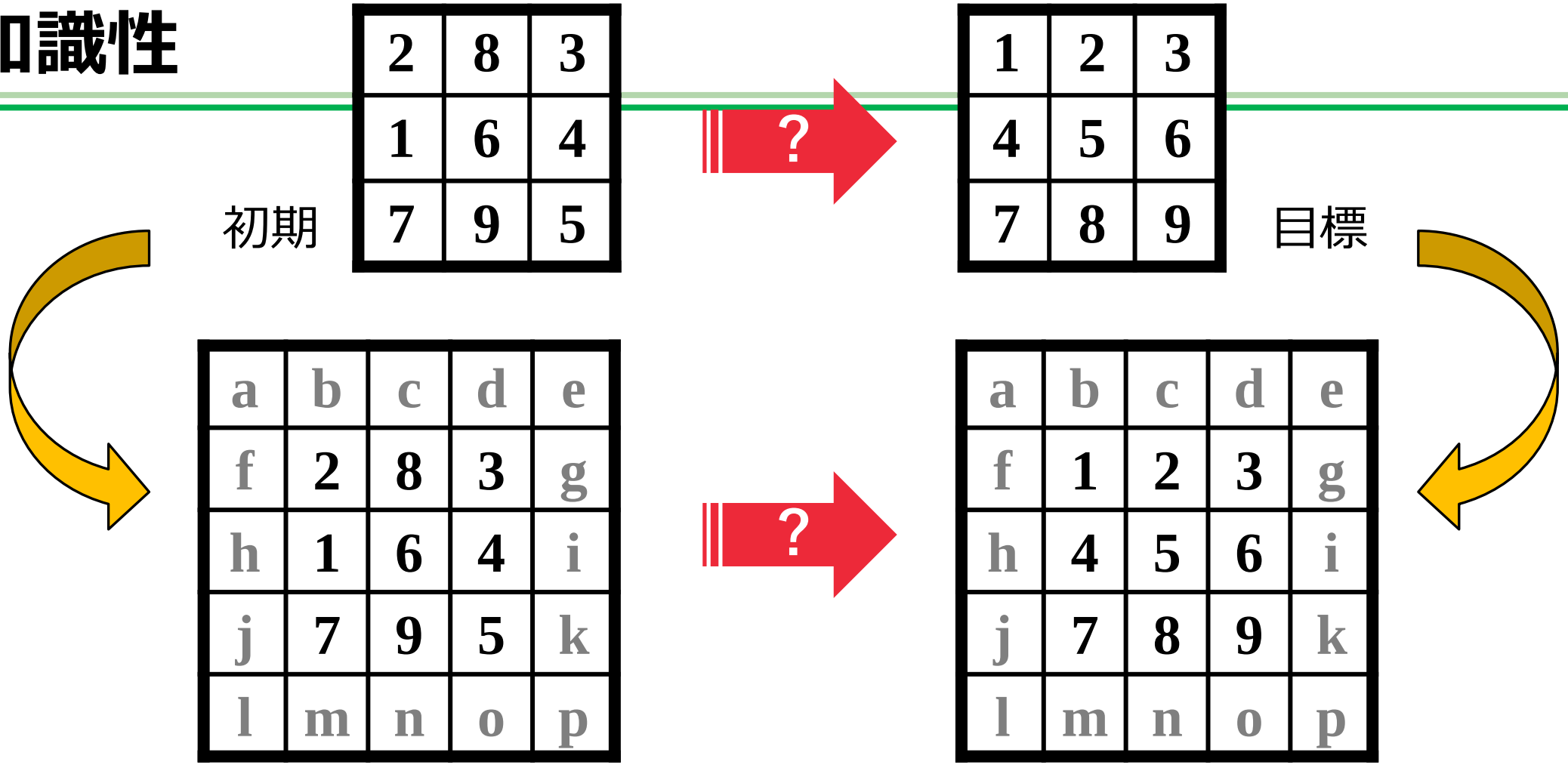
1	2	3
4	5	6
7	8	9

目標



- オレンジ領域の♥の数が**列ごとに異なる → 交換先のタイルが一意に定まる可能性**

ゼロ知識性



- 盤面の周囲にダミーを追加 → 数字タイルは全て4つのタイルと隣接
  （英字が移動するようなカード配置になっていないことは事前にチェック可能）

$(w \times h)$パズルに対するカードベースZKP

プロトコルの正当性

➢ 完全性

Peggyが遷移手順通りに正しくカードを並べたなら，必ず最後のカードが一致する.

➢ 健全性

Peggyがでたらめにカードを並べたなら，

- 緑領域のカードを開いたとき♦と♥の数が合わない
- オレンジ領域のカードにより，隣接しないタイル移動を行おうとしている
- 最後のカードが一致しない

ことが判明し，Peggyの嘘を**必ず検知できる**.　（**健全性エラー0**）

➢ ゼロ知識性

(2×2)パズルならOK．それ以上の盤面サイズだと**情報漏洩の危険性有**.

→**ダミーを置くことでゼロ知識性を保証**.

今後の課題

1) **カード枚数やシャッフル回数の削減**

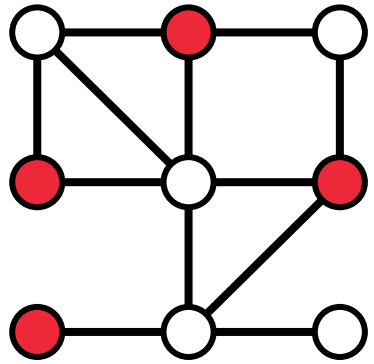
長さℓの遷移手順に対して，今回のZKPは$(wh+\ell+4)(wh+2)+\ell$ 枚のカードを使用し，$3\ell+1$回のパイルシャッフルを行う.
$w=h=\ell=4$の場合であっても，**436枚**のカードと**13回**のシャッフルが必要.
→「簡単に実行できる」とは言い難い.

➡ **より少ないカードとシャッフルで実行できないか？**

2) **他の遷移問題に対するカードベースZKPの構築**

今後の課題

- 情報漏洩無くタイルを交換可能なプロトコルを作成
- 様々な遷移問題は，オブジェクトの交換によって定義可能.



独立集合：グラフ上で互いに隣接しない頂点の集合

トークンと見なす

独立集合遷移問題：与えられた2つの独立集合はルールの下で遷移可能か？

ルール①：**トークン**は隣接した空白頂点に動かせる
ルール②：**トークン**の移動後も**トークン**の集合は独立集合をなす

- 15パズルと同様の手法で，ルール①は情報漏洩無く可能
- ルール②を満たすことをチェックするプロトコルを作成すればOK

➡ 遷移問題に対するカードベースZKPの**フレームワークを作成可能**？

今後の課題

- 良くないお知らせ：多くの遷移問題は**PSPACE完全**
  （独立集合遷移問題・倉庫番・スーパーマリオブラザーズ等）

  ➡ 遷移の最短手順が**入力の指数長**になりうる
  ＝ **指数枚のカードが必要**

- それでも… NP完全（最短手順が多項式長）な遷移問題はいくつかある
  - $(w \times h)$パズル
  - トークンスワッピング・シーケンシャルトークンスワッピング
  - パンケーキソート （カードベースZKPの研究あり [Komano and Mizuki 2022]）
  - 初代ポケモン（敵トレーナーしかいない場合）
  - 独立集合遷移問題の特殊ケース

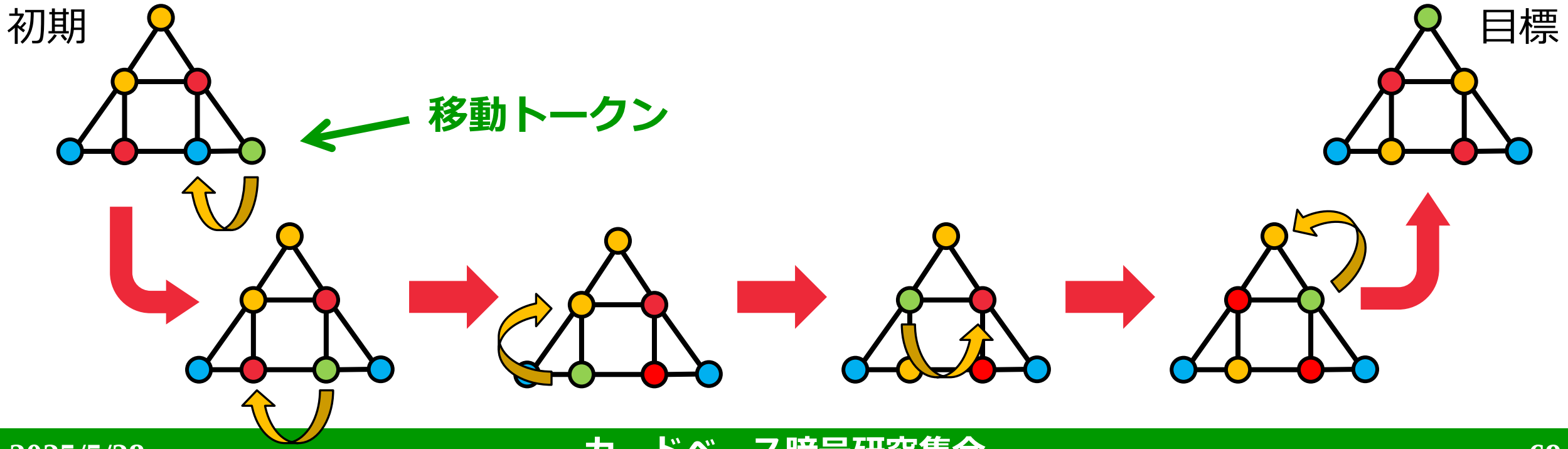
- NP完全な遷移問題を扱っていく？ **皆さんのご意見をお待ちしております**

補足スライド

シーケンシャルトークンスワッピング

- 各頂点に1つずつ色付トークンが載せられたグラフを考える
- 以下のルールの下，トークンの初期配置は目標配置へ遷移可能？

ルール：トークン（**移動トークン**）を1つ選び，それと隣接するトークンを交換する

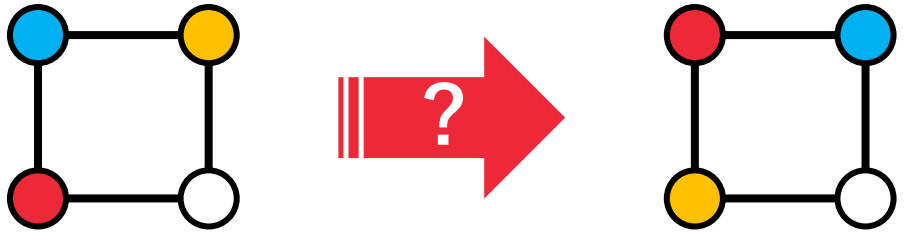


補足スライド

15パズル

ルール：**空白セル**に隣り合った
タイルのみ空白セルに動かせる

シーケンシャルトークンスワッピング



ルール：**移動トークン**を1つ選び，
それと隣接するトークンを交換する

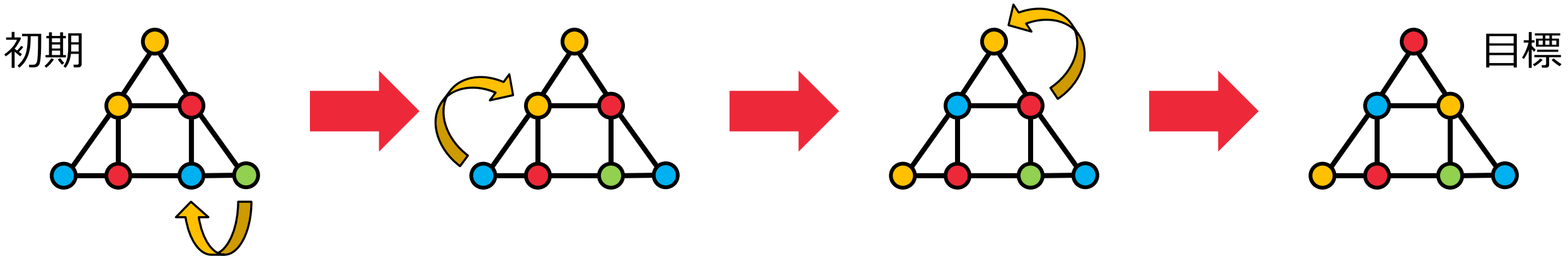
- $(w \times h)$パズルをグリッドグラフ上の色付トークンに変換
- 違いは，変換後において白色のトークンを移動トークンとする必要があるのみ
  （プロトコルで対処可能）

補足スライド

トークンスワッピング

- 各頂点に1つずつ色付トークンが載せられたグラフを考える
- 以下のルールの下，トークンの初期配置は目標配置へ遷移可能？

ルール：隣接するトークンを交換する



- 移動トークンが無い，という点で前述の2つの問題とは異なる
- とはいえ，緑領域の♥の置き方を少し変えるだけでカードベースZKPを構築可