多色カードを用いた効率的な対称関数プロトコル

2024/5/22

産学連携と数理・暗号分野連携によるカードベース暗号の深化と新境地

高橋由紘 (茨城大)

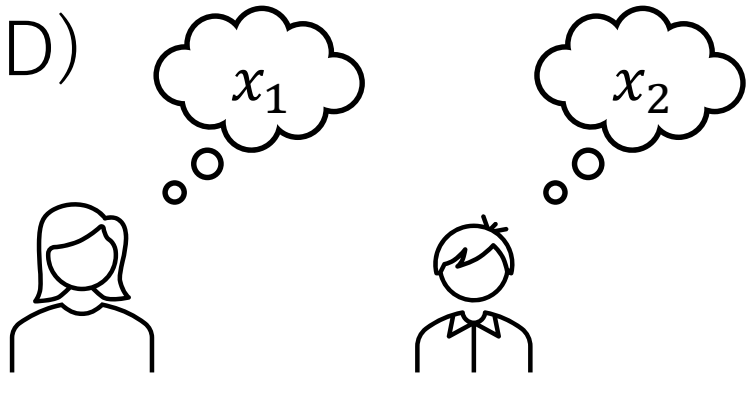
共同研究者：品川和雅（茨城大）　四方隼人（東北大）　水木敬明（東北大）

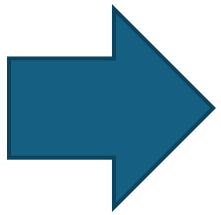
国際会議APKC2024にて発表予定

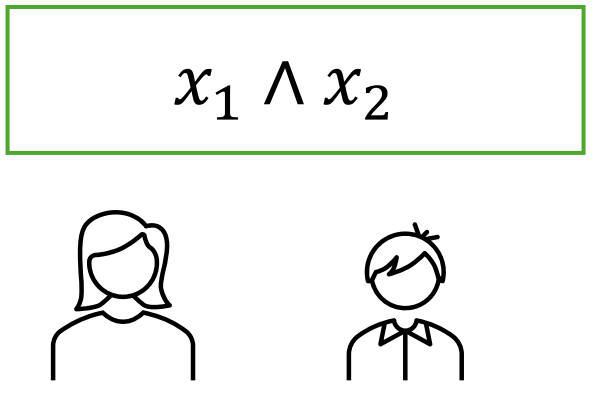
秘密計算

入力を秘密にしたまま計算する暗号技術

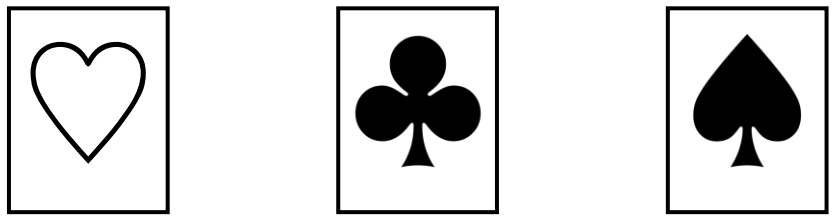
(ex. AND)





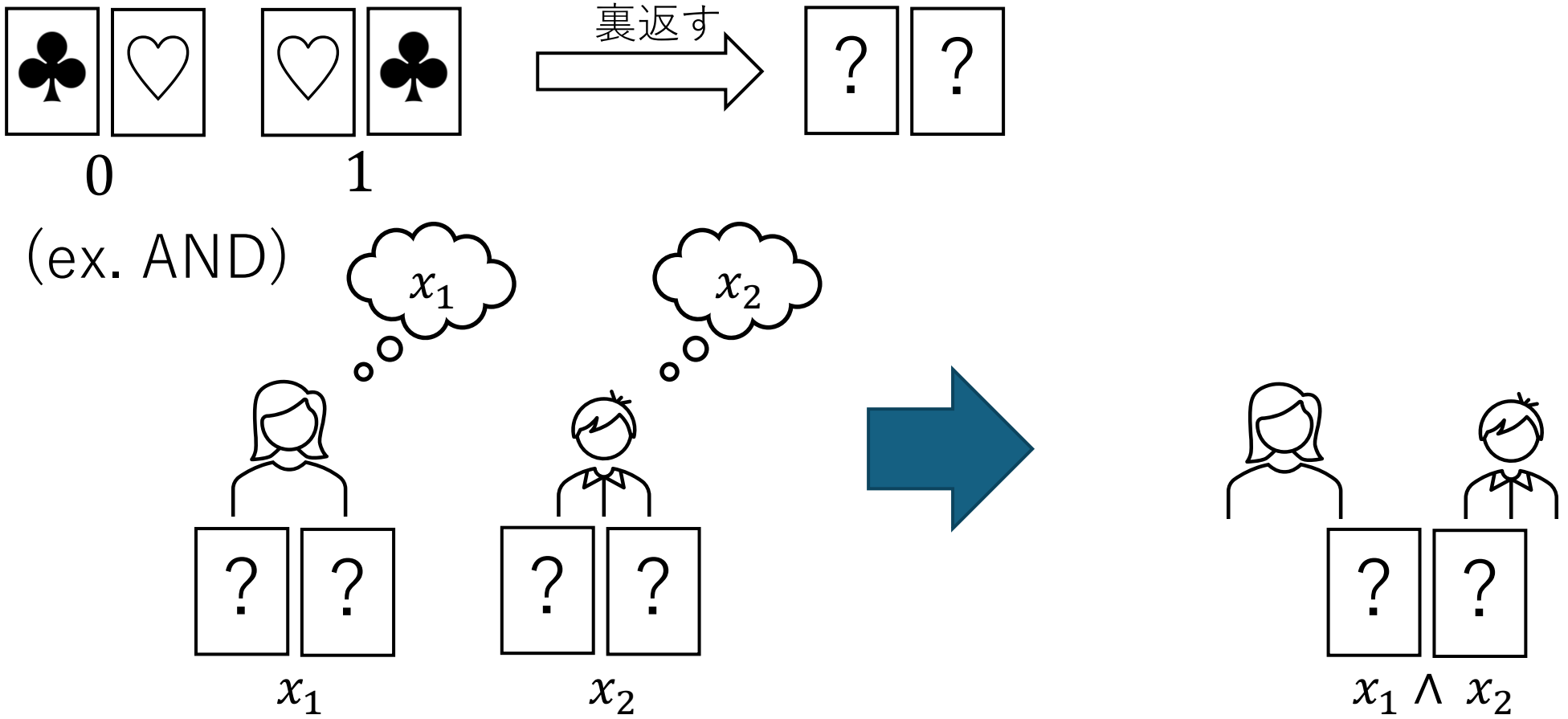


物理的なカードを用いて秘密計算を行うことを**カードベース暗号**と呼ぶ



カードベース暗号

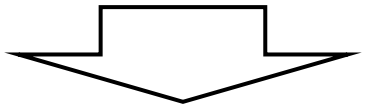
物理的なカードを用いて秘密計算を実行する



対称関数プロトコル

対称関数とは・・・

n変数関数$f:\{0,1\}^n \to R$が対称関数であるとは、入力の順序を任意に入れ替えても、出力値が変わらない関数



出力は、**ビットの和**に依存するので、ある関数$g:\{0,1,\dots n\} \to R$を用いて

$$f(x_1, x_2, \dots, x_n) = g(\sum x_i)$$

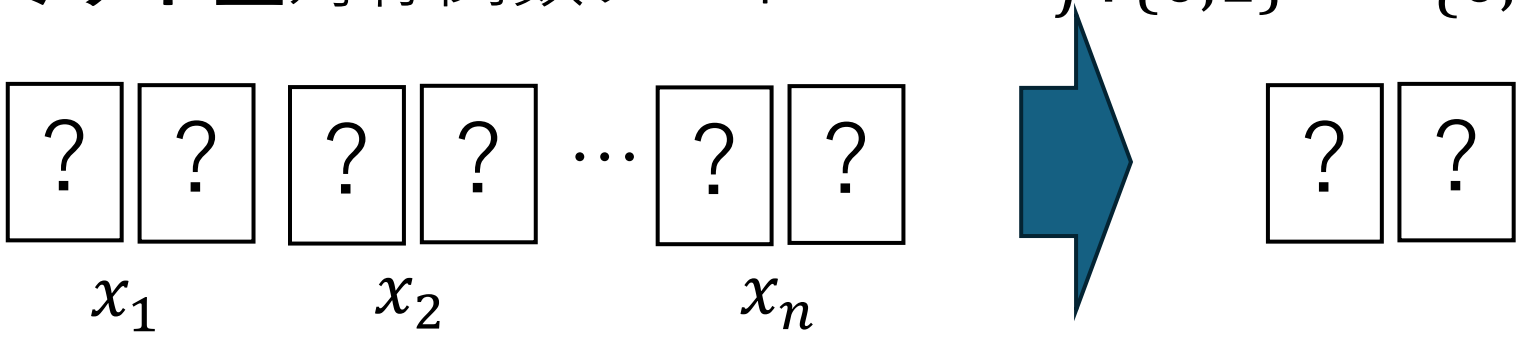
と表せる

(例)

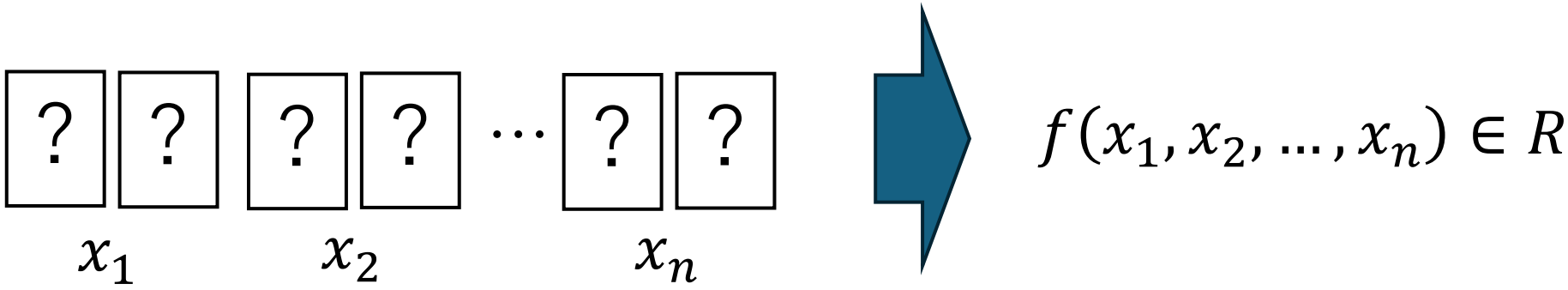
$$f(0,1,1) = f(1,0,1) = f(1,1,0) = g(2)$$

対称関数プロトコル

コミット型対称関数プロトコル… $f:\{0,1\}^n \to \{0,1\}$



非コミット型対称関数プロトコル…$f:\{0,1\}^n \to R$ (Rは有限集合)



新たな**非コミット型プロトコル**を提案する

貢献

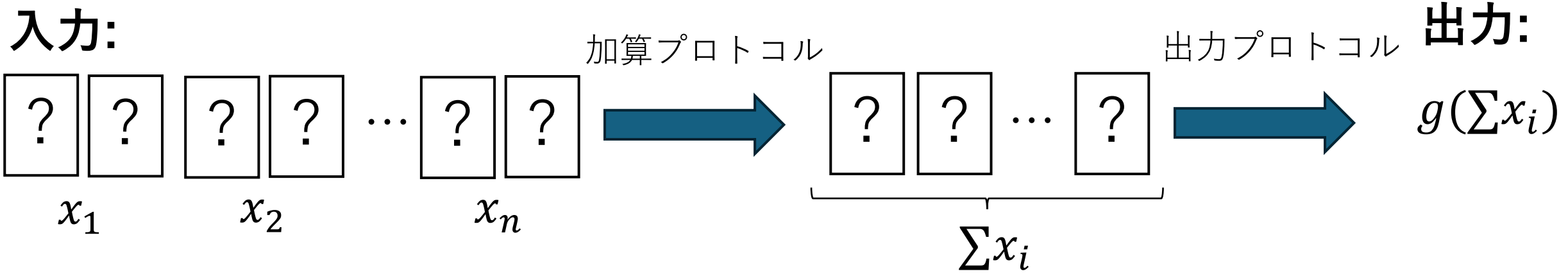
対称関数$f:\{0,1\}^n \to R$を計算する非コミット型プロトコル

考案者	色数	カード枚数	シャッフル回数	有限時間	シャッフル
Ruangwises-Itoh(TAMC2020)	2	$2n+2$	$n+\|R\|-2$	○	PShift
提案方式①	**4**	$\boldsymbol{2n}$	$\boldsymbol{2n+\|R\|-1+(n-2)\sum_{i}^{n-1}\frac{1}{i}}$	×	**RC**
提案方式②	**3**	$\boldsymbol{2n+1}$	$\boldsymbol{n+\|R\|}$	○	**RC and RBC**
提案方式③	**4**	$\boldsymbol{2n}$	$\boldsymbol{n+\|R\|+8}$	×	**RC and RBC**

- 提案方式①：$2n$枚のLas Vegasプロトコル(RCのみ)
- 提案方式②：$2n+1$枚の有限時間プロトコル
- 提案方式③：シャッフル回数を削減した$2n$枚のLas Vegasプロトコル

対称関数プロトコル

対称関数$f: \{0,1\}^n \to R$に対する**非コミット型**プロトコルの流れ



本論文では、3つの**加算プロトコル**を提案し、手順を簡略化する

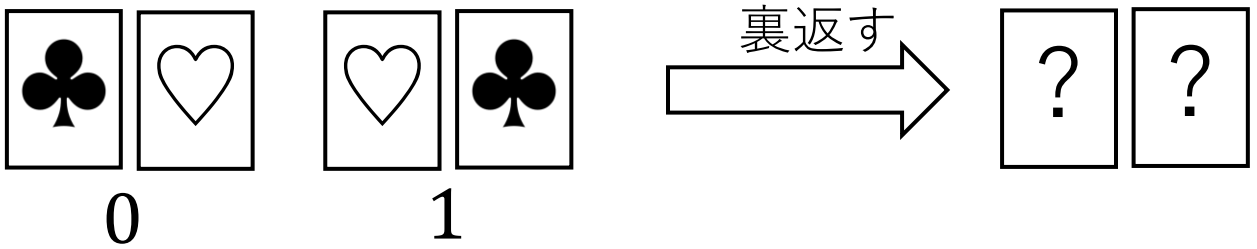
準備

- **1ビットの符号化**
- **整数の符号化**
- **シャッフル**

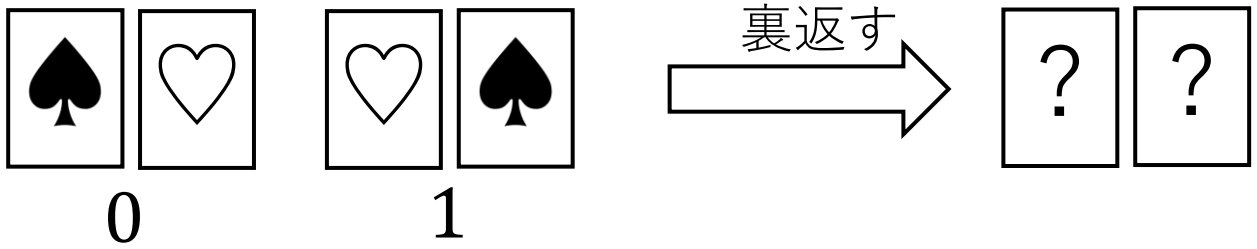
1ビットの符号化

3つのビット符号化方式を用いる

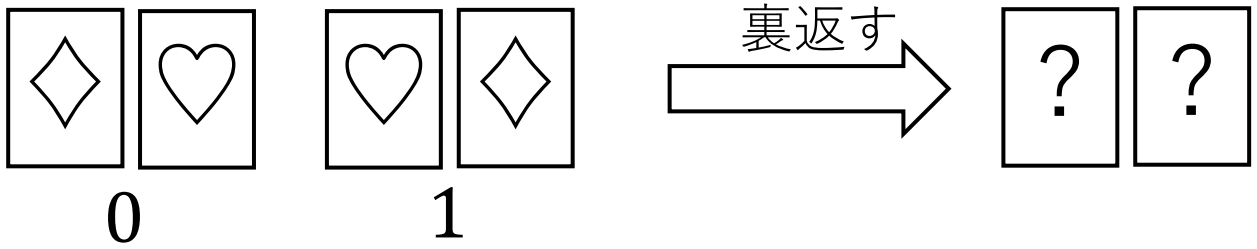
①♣と♡の2枚のカードを用いてビットを符号化



②♠と♡の2枚のカードを用いてビットを符号化



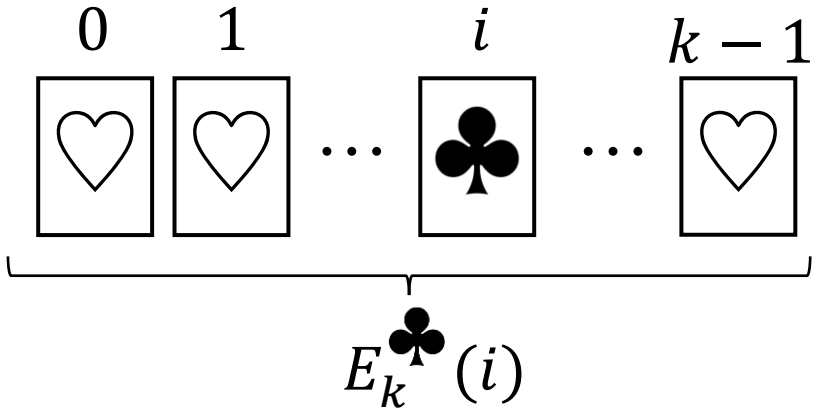
③♢と♡の2枚のカードを用いてビットを符号化



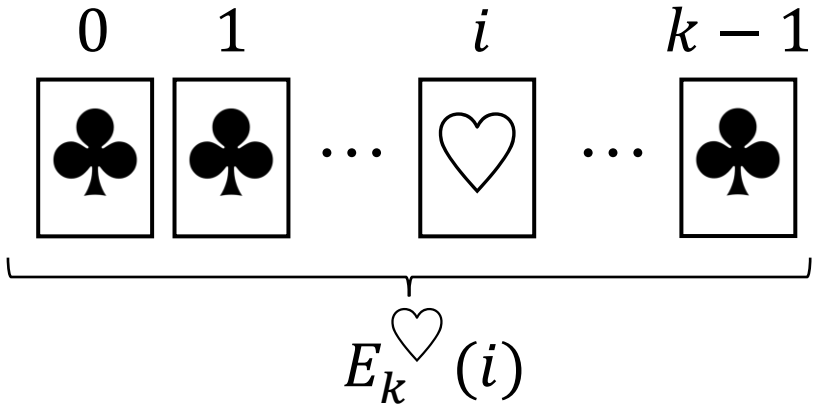
整数の符号化

1枚の♣と$k-1$枚の♡を用いて整数$i \bmod k$を符号化

i番目に♣を置くことで整数を符号化
これを$E_k^{\clubsuit}(i)$と表す

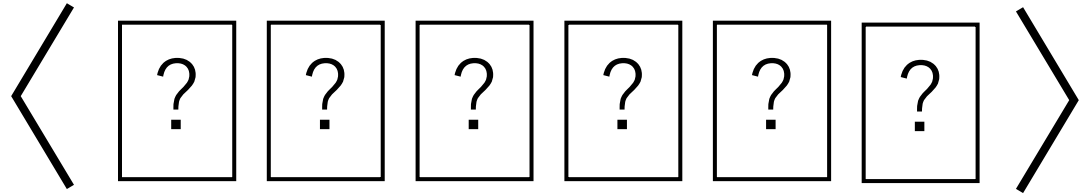


♣と♡が逆のとき、 $E_k^{\heartsuit}(i)$と表す



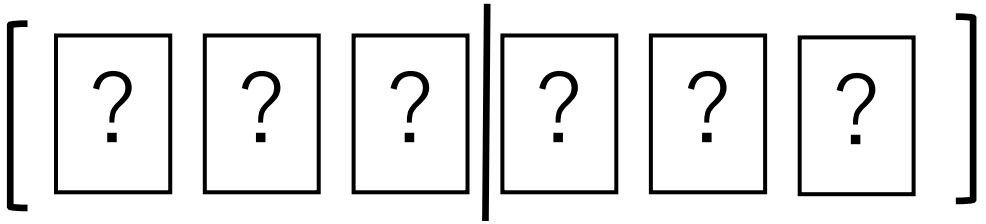
シャッフル

- **ランダムカット**

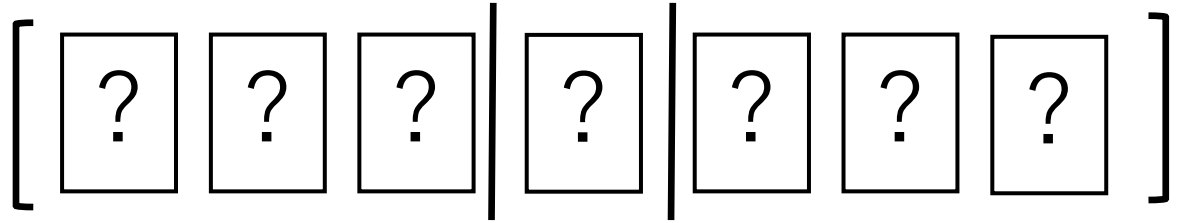


- **ランダム二等分割カット**

偶数枚のとき…

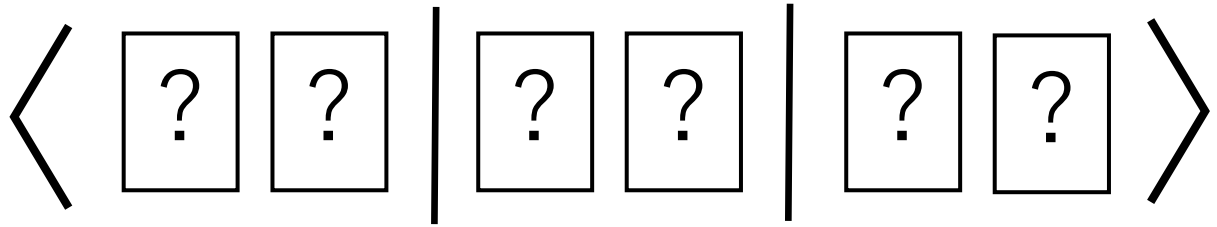


奇数枚のとき…

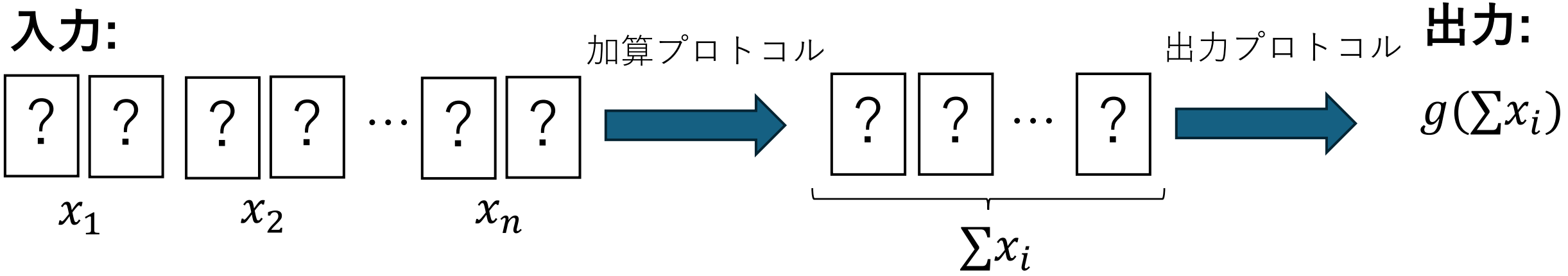


シャッフル

- **パイルシフティングシャッフル**



対称関数プロトコル

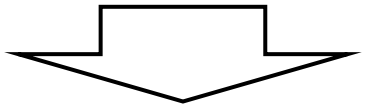


全てのプロトコルに共通する**出力プロトコル**について説明する

出力プロトコル

対称関数$f\{0,1\}^n \to R$を計算する一般的な手法

加算プロトコルを用いて全ビットを加算



整数符号化された加算結果$\sum_{i=1}^{n} x_i$から対称関数の結果を出力

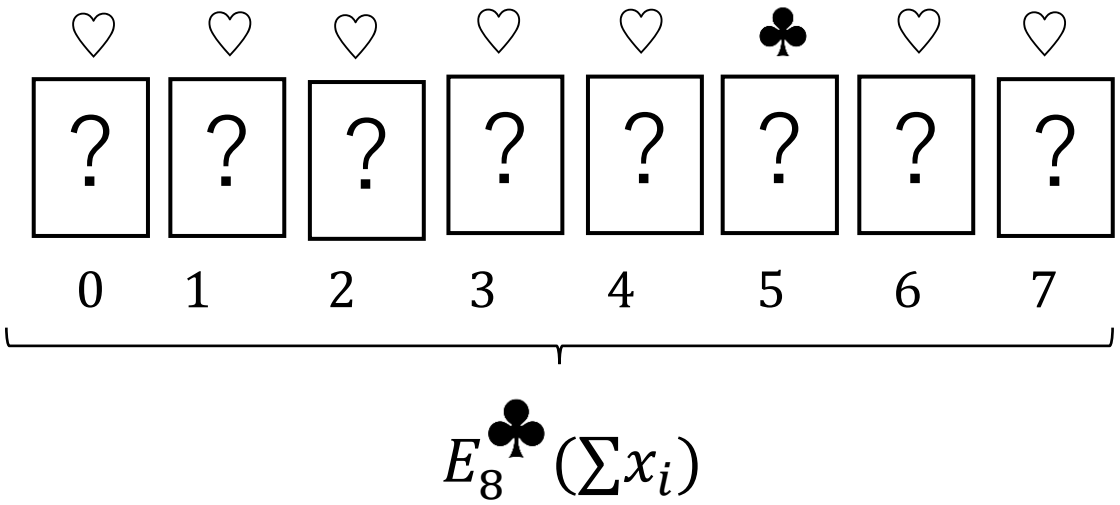
Ruangwises-Itohによって考案されたプロトコルを用いる

Ruangwises and T. Itoh. Securely computing the nvariable equality function with 2n cards. In TAMC 2020

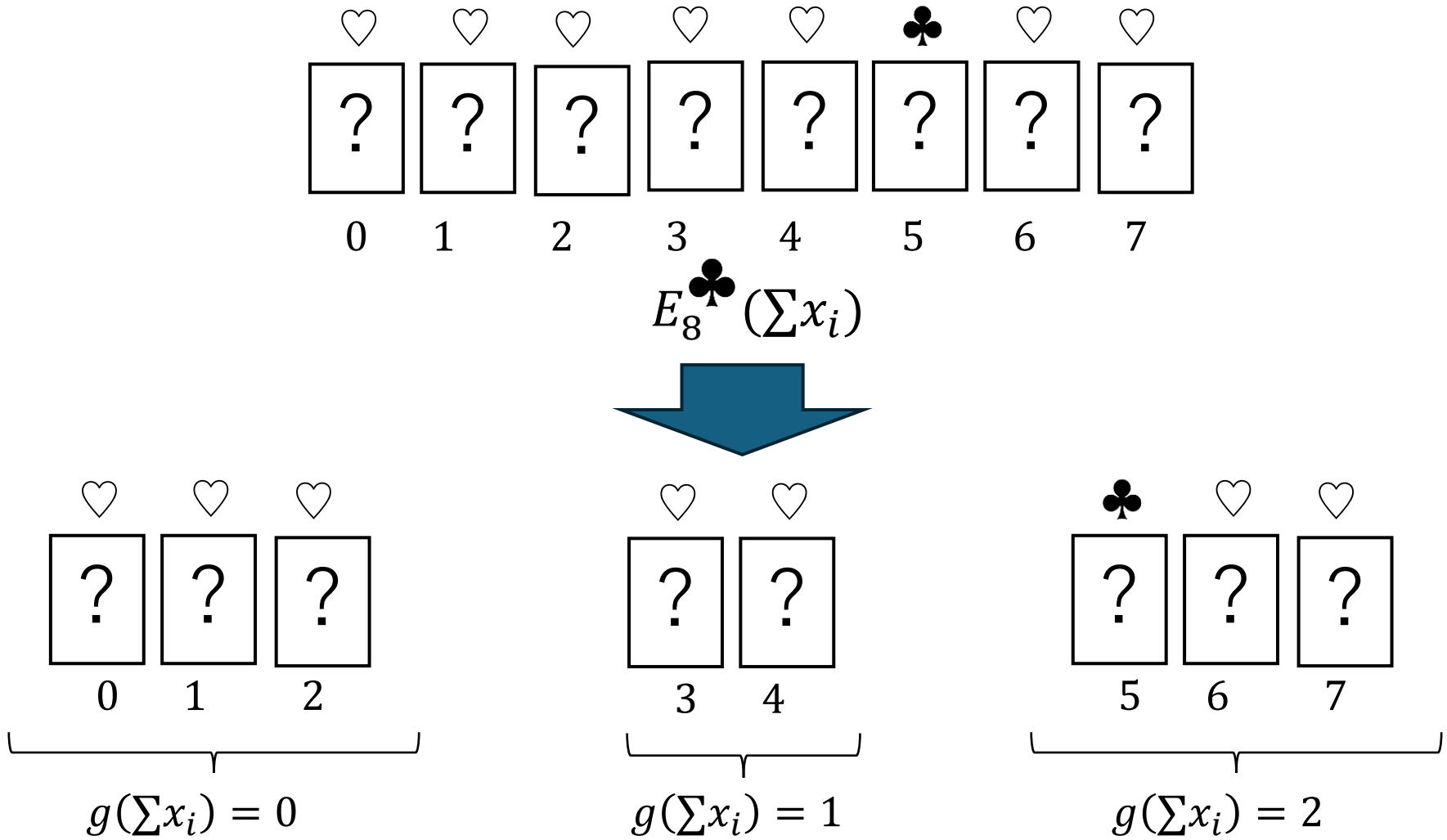
出力プロトコル

例)

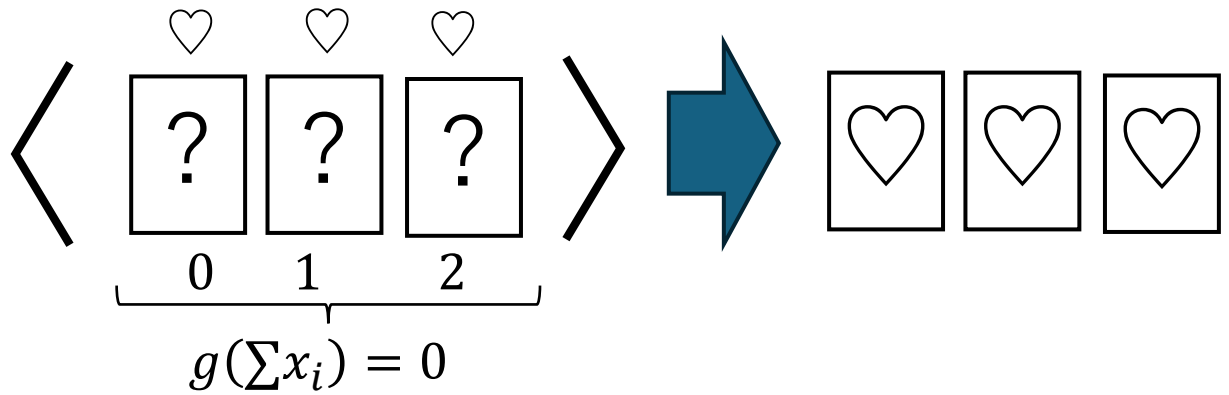
$$f(x_1, \ldots, x_7) = g(\sum x_i) = \begin{cases} 0 \text{ if } \sum x_i = 0,1,2 \\ 1 \text{ if } \sum x_i = 3,4 \\ 2 \text{ if } \sum x_i = 5,6,7 \end{cases}$$



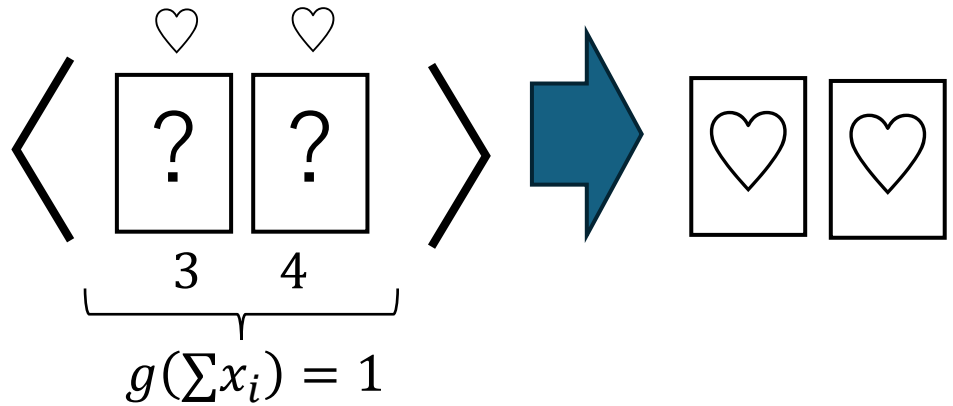
1. カード列をそれぞれの出力ごとに分ける



2. $g(\sum x_i) = 0$の束にランダムカットを施し、もし♣があれば、0を出力する



3. $g(\sum x_i) = 1$の束にランダムカットを施し、もし♣があれば、1を出力する



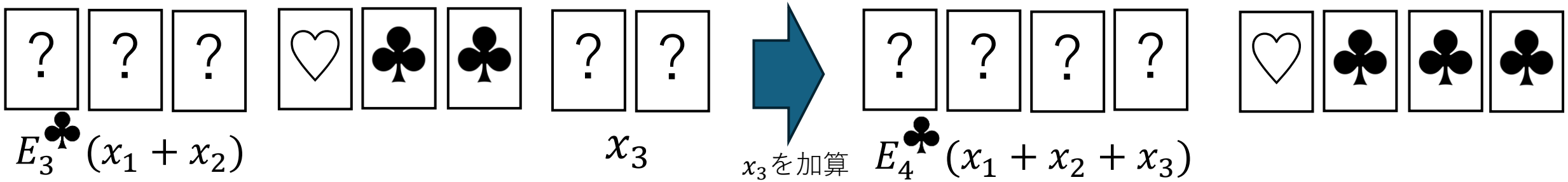
4. この時点で何も出力していない場合、2を出力する

既存プロトコル

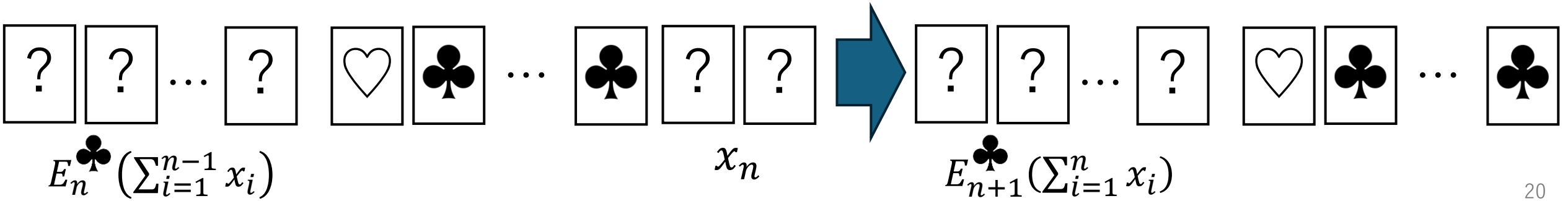
既存加算プロトコル

- **Ruangwises-Itohの加算プロトコル**

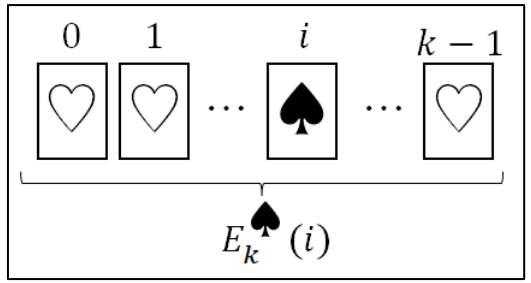
$2n+2$枚の加算プロトコル[Ruangwises-Itoh TAMC20]

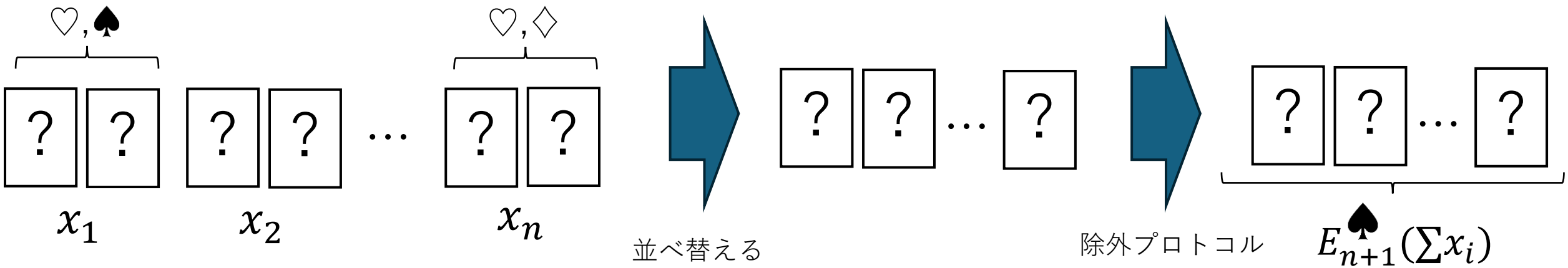


…



提案加算プロトコル①の流れ





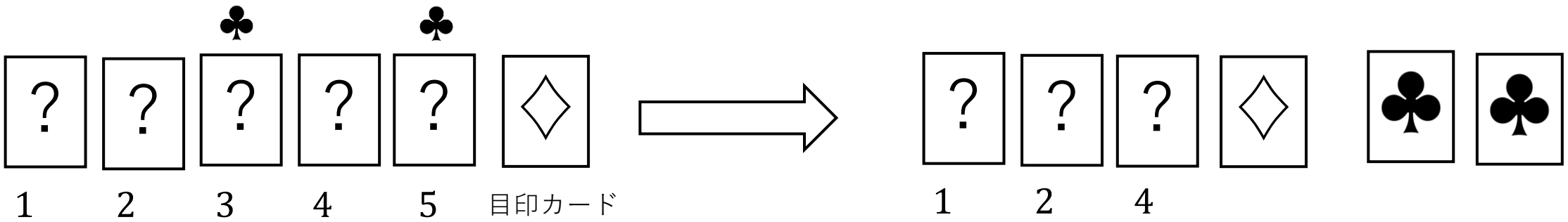
カード枚数：$2n$枚(4色)

シャッフル回数：$2n + |R| - 1 + (n-2)\sum_i^{n-1}\frac{1}{i}$回(Las Vegas)

考案者	色数	カード枚数	シャッフル回数	有限時間	シャッフル
Ruangwises-Itoh(TAMC2020)	2	$2n+2$	$n+\|R\|-2$	○	PShift
提案方式①	**4**	$\boldsymbol{2n}$	$\boldsymbol{2n+\|R\|-1+(n-2)\sum_{i}^{n-1}\frac{1}{i}}$	×	**RC**
提案方式②	3	$2n+1$	$n+\|R\|$	○	RC and RBC
提案方式③	4	$2n$	$n+\|R\|+8$	×	RC and RBC

除外プロトコル

[Takashima et al. 20] によって提案されたプロトコル

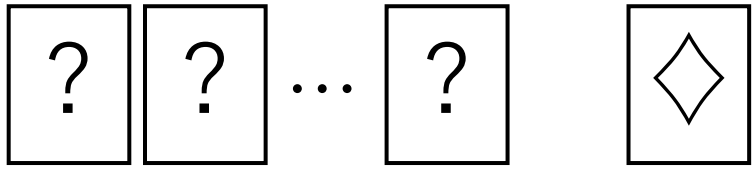


順番を保持したまま、特定のカードを除去する

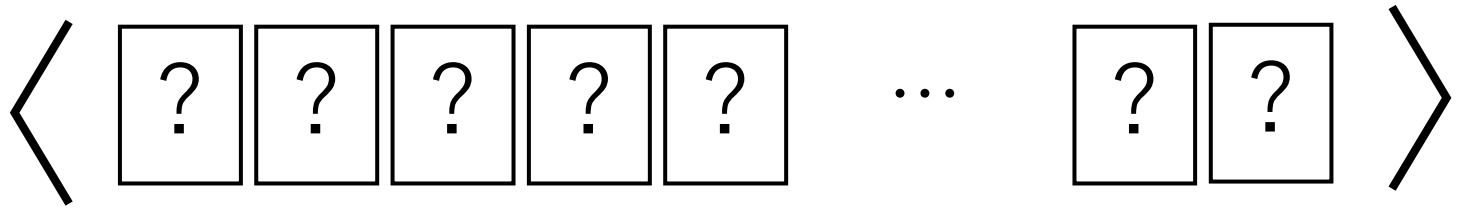
KenTakashima, Yuta Abe, Tatsuya Sasaki, Daiki Miyahara, Kazumasa Shinagawa, Takaaki Mizuki, and Hideaki Sone. 2020. Card-based protocols for secure ranking computations. Theor. Comput. Sci. 845 (2020),

除外プロトコル

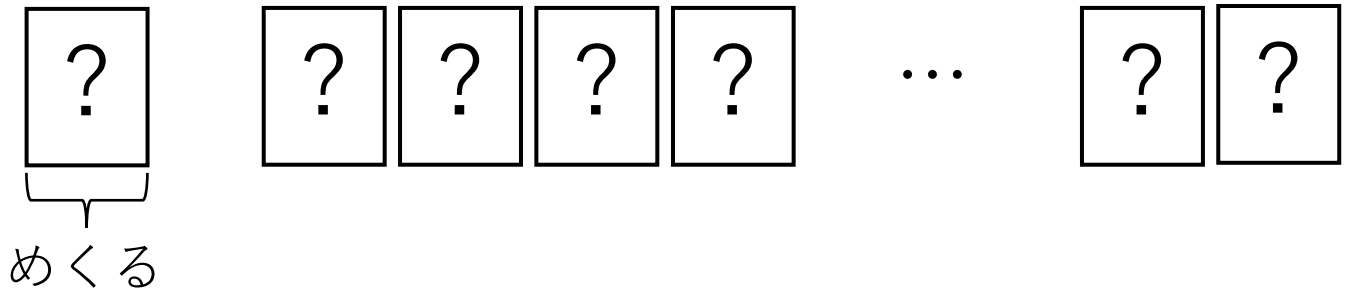
1. 以下のようにカードを並べる



2. ♢カードを裏返し、ランダムカットを適用する



3. ランダムカットを適用したカード列の左端のカードを開く

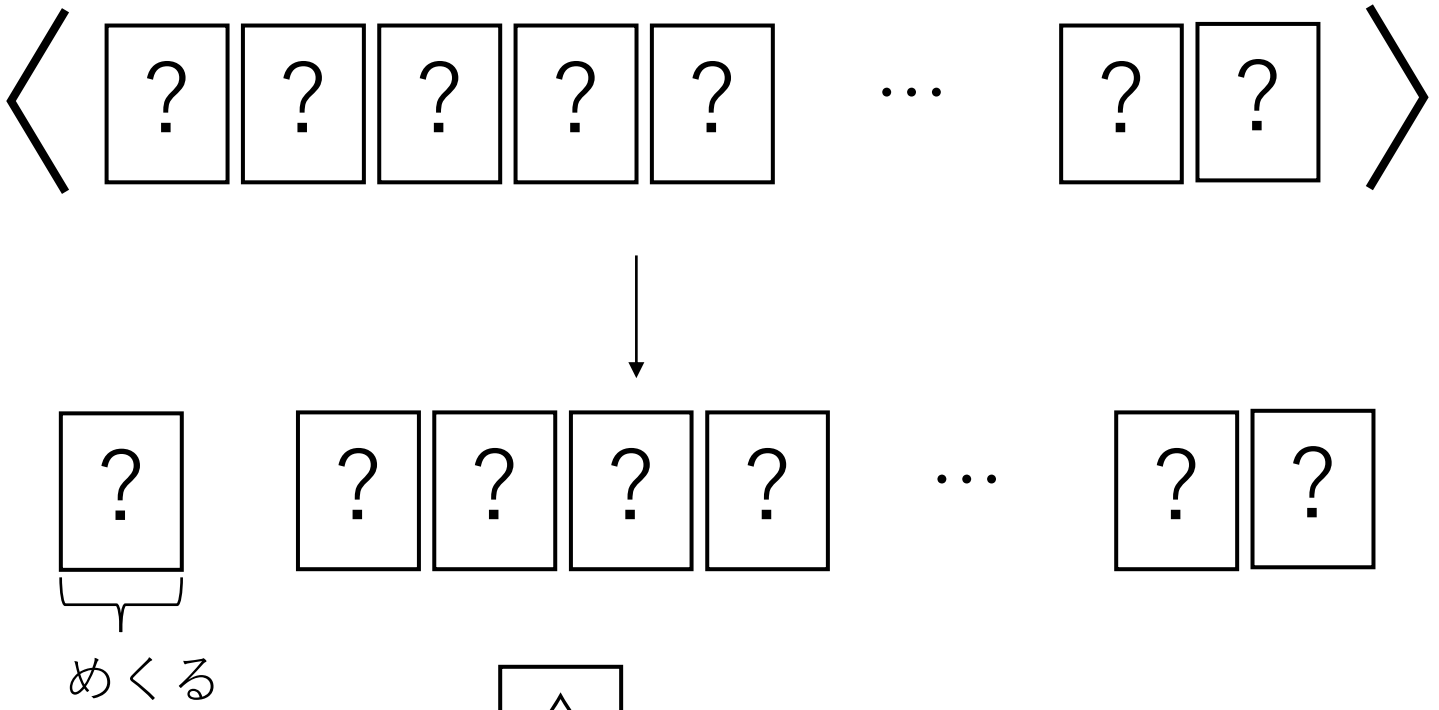


(a) めくったカードが  の場合、カードを取り除き、手順1に戻る

(b) めくったカードが  でない場合、裏返して元に戻し、手順1に戻る

この手順を、  がk枚なくなるま繰り返す

4.ランダムカットを適用して左端のカードを開く

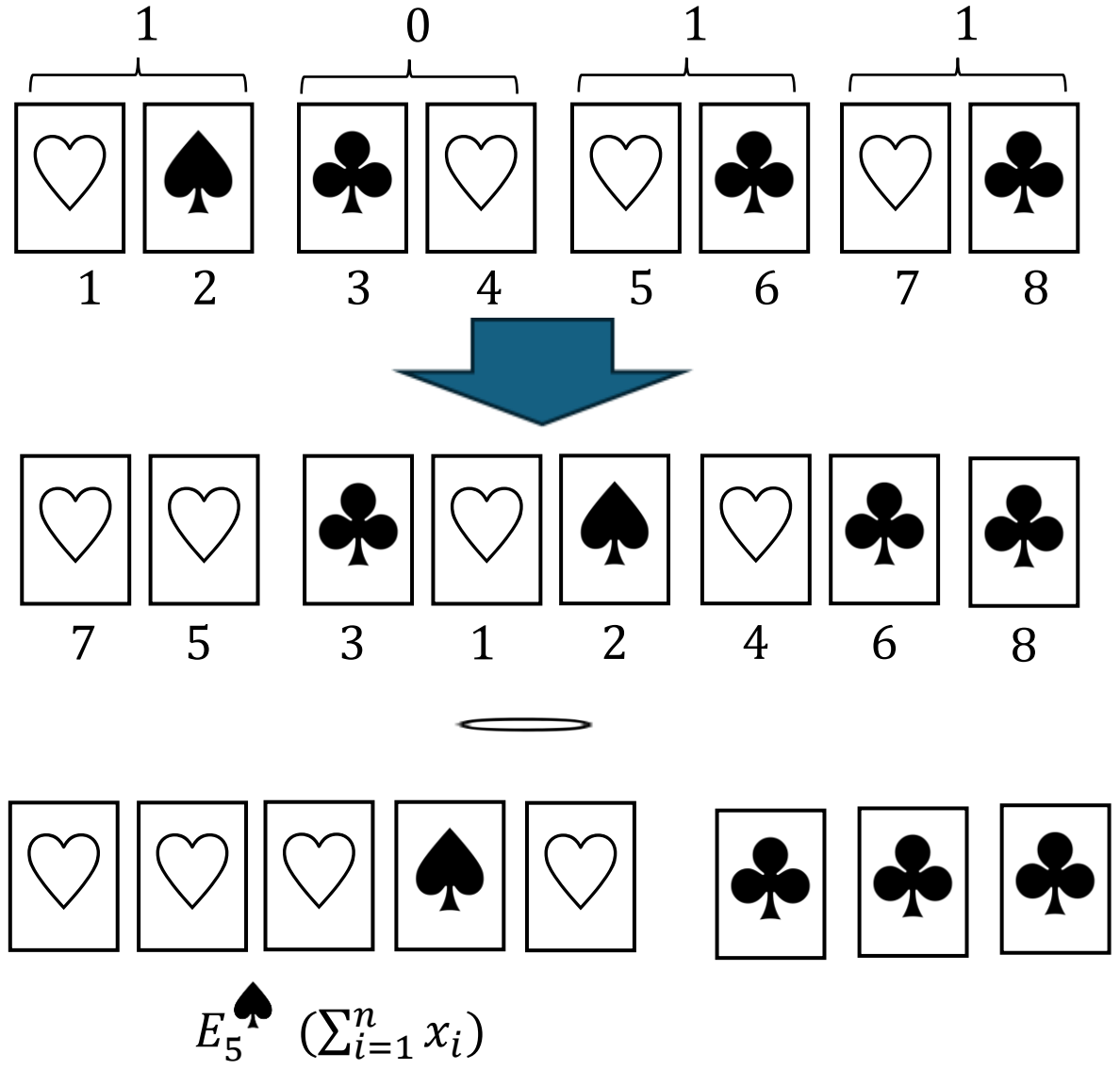


(a) めくったカードが ♢ の場合、カードを取り除き手順を終了

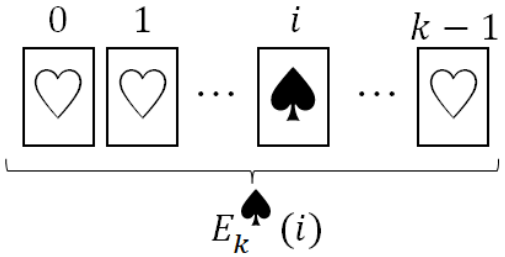
(b) めくったカードが ♢ ではない場合、裏返して元に戻し、
手順4を繰り返す

以上より、♣をk枚取り除き、元のカード列に戻せた

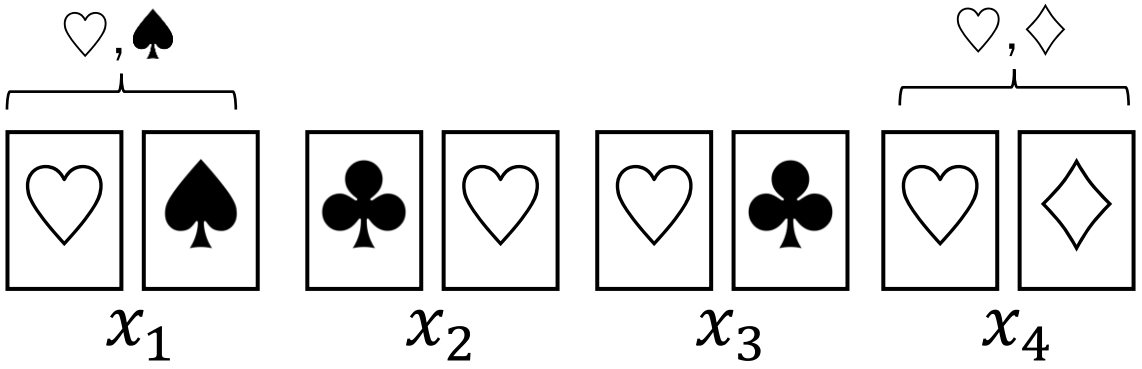
提案加算プロトコル①のアイデア



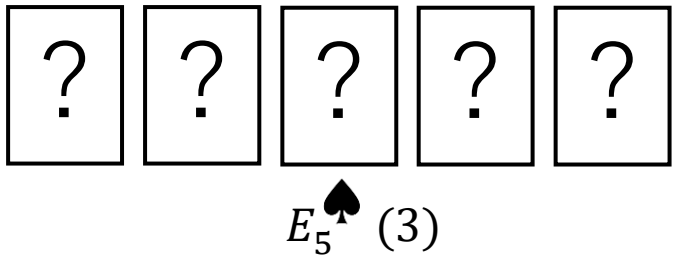
提案加算プロトコル①

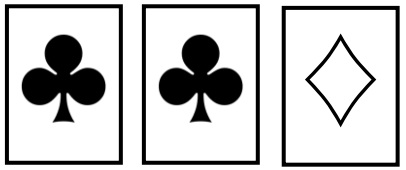


ex)$(x_1, x_2, x_3, x_4) = (1,0,1,1)$



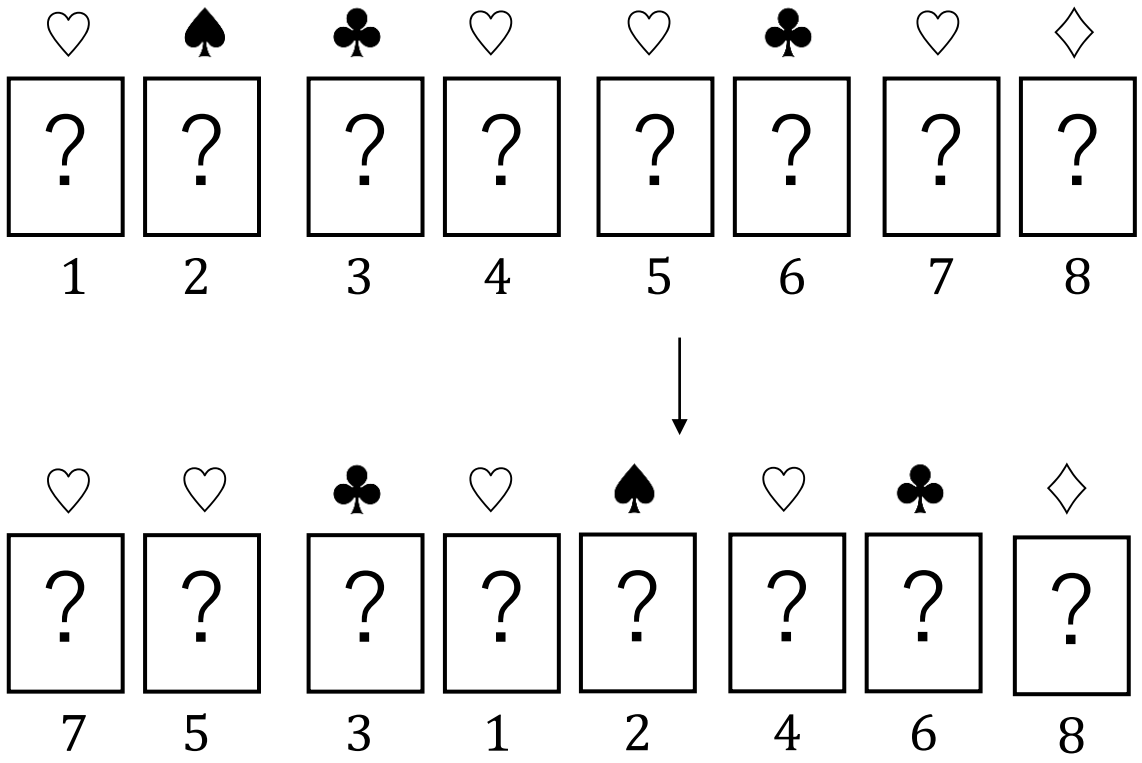
除外プロトコル



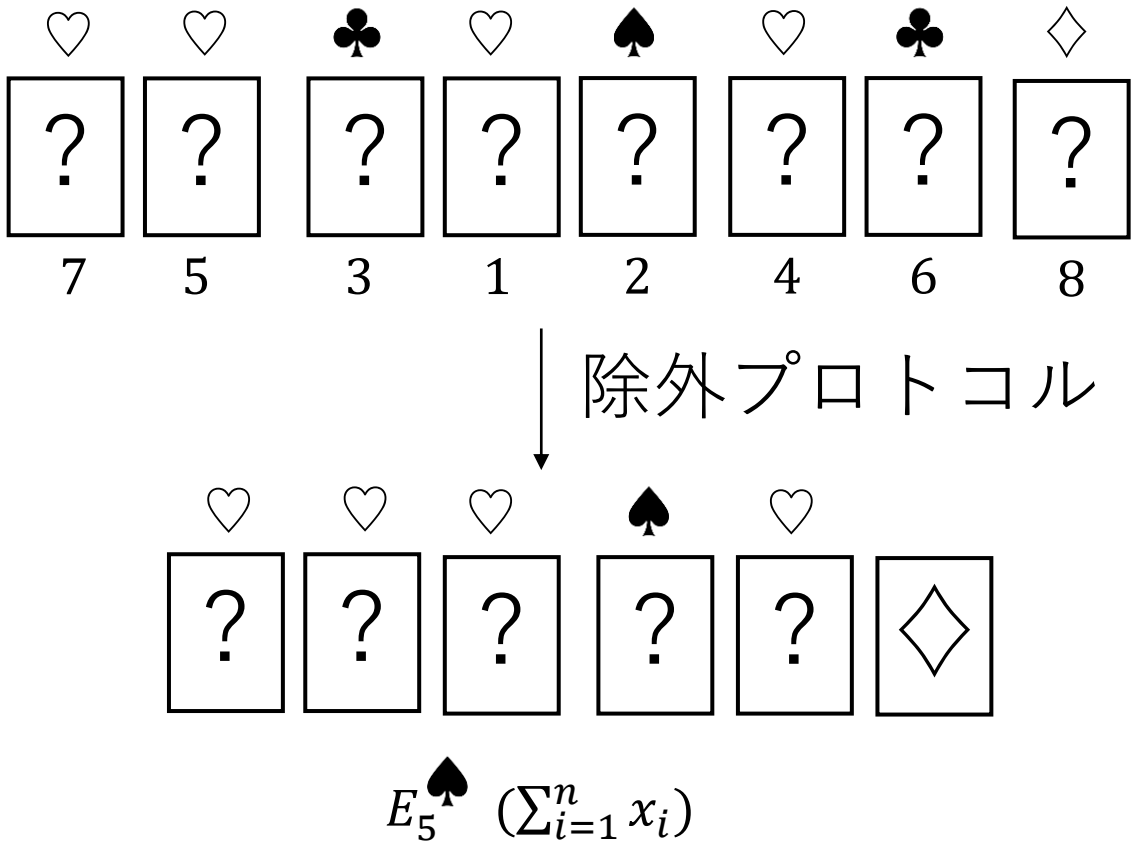


提案加算プロトコル①

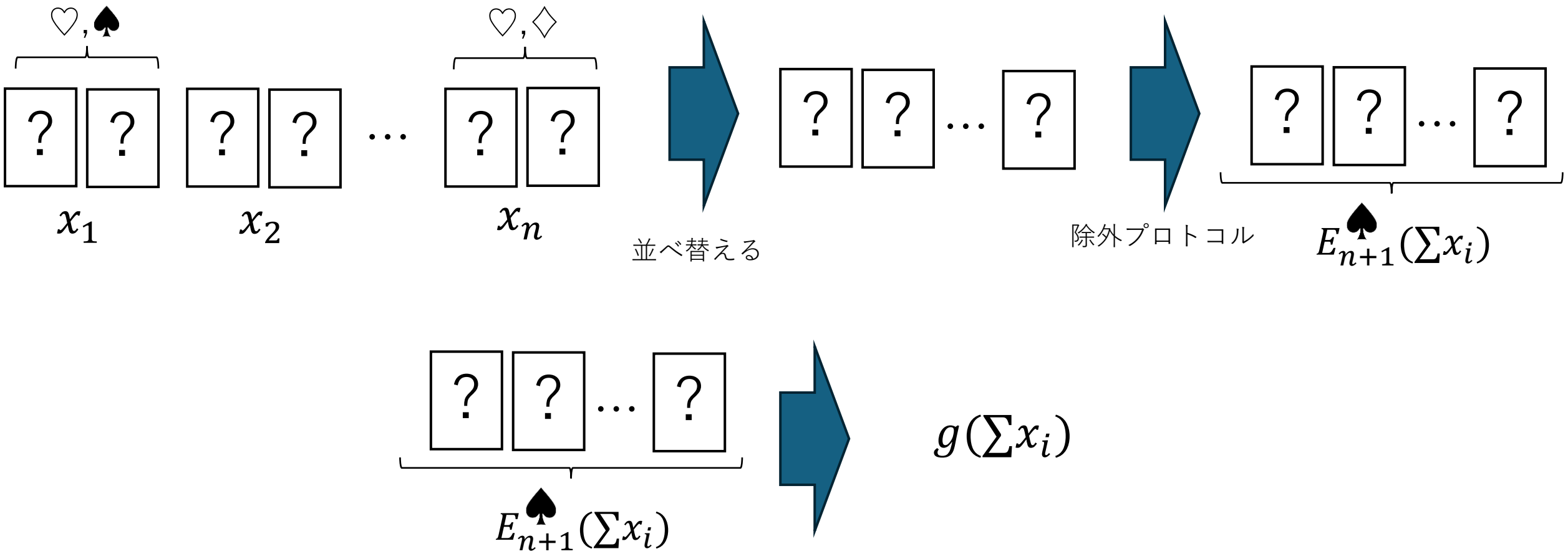
1. x_1を中央に置き、以下のように入力されたカードを両脇に置く



2. このカード列から除外プロトコルを用いて♣カードを取り除き、♢が先頭に来るように並べ替える

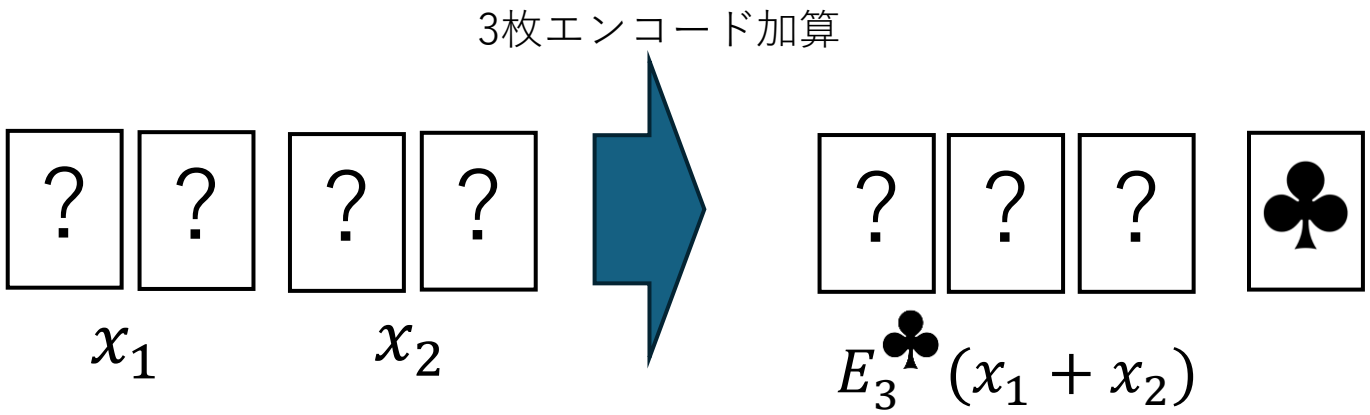


提案プロトコル①

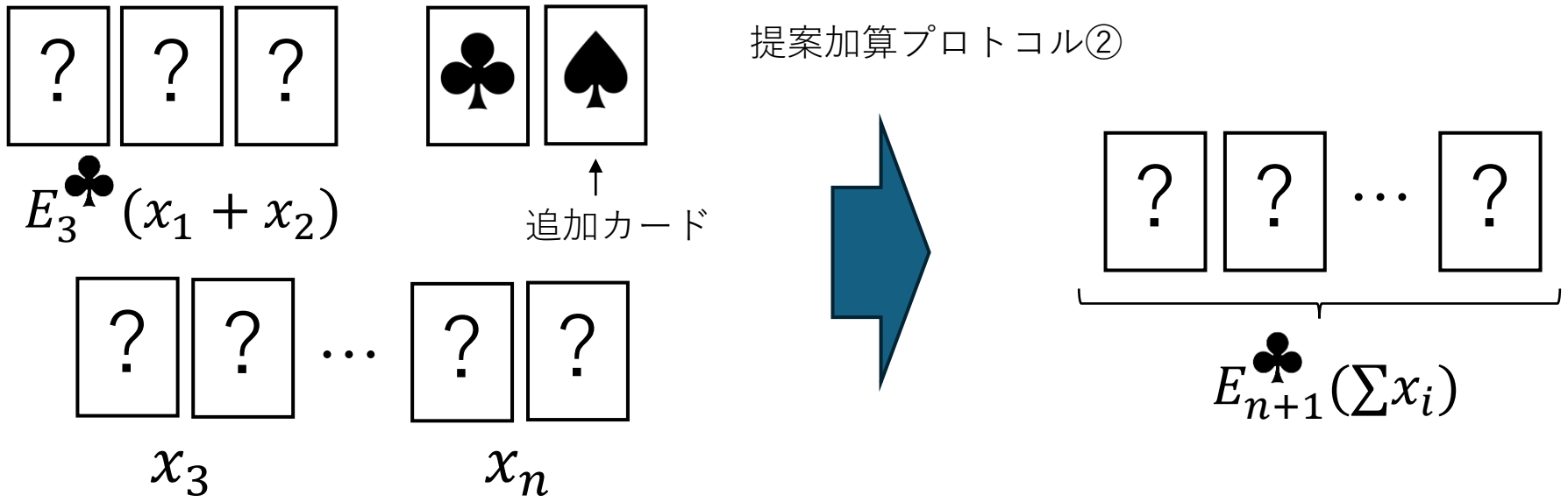

♡,♠
♡,♢
? ? ? ? … ? ?
x_1
x_2
x_n
並べ替える
? ? … ?
除外プロトコル
? ? … ?
$E_{n+1}^{\spadesuit}(\sum x_i)$
? ? … ?
$E_{n+1}^{\spadesuit}(\sum x_i)$
出力プロトコル
$g(\sum x_i)$

考案者	色数	カード枚数	シャッフル回数	有限時間	シャッフル
Ruangwises-Itoh(TAMC2020)	2	$2n+2$	$n+\|R\|-2$	○	PShift
提案方式①	4	$2n$	$2n+\|R\|-1+(n-2)\sum_{i}^{n-1}\frac{1}{i}$	×	RC
提案方式②	**3**	$\boldsymbol{2n+1}$	$\boldsymbol{n+\|R\|}$	○	**RC and RBC**
提案方式③	4	$2n$	$n+\|R\|+8$	×	RC and RBC

提案プロトコル②の流れ

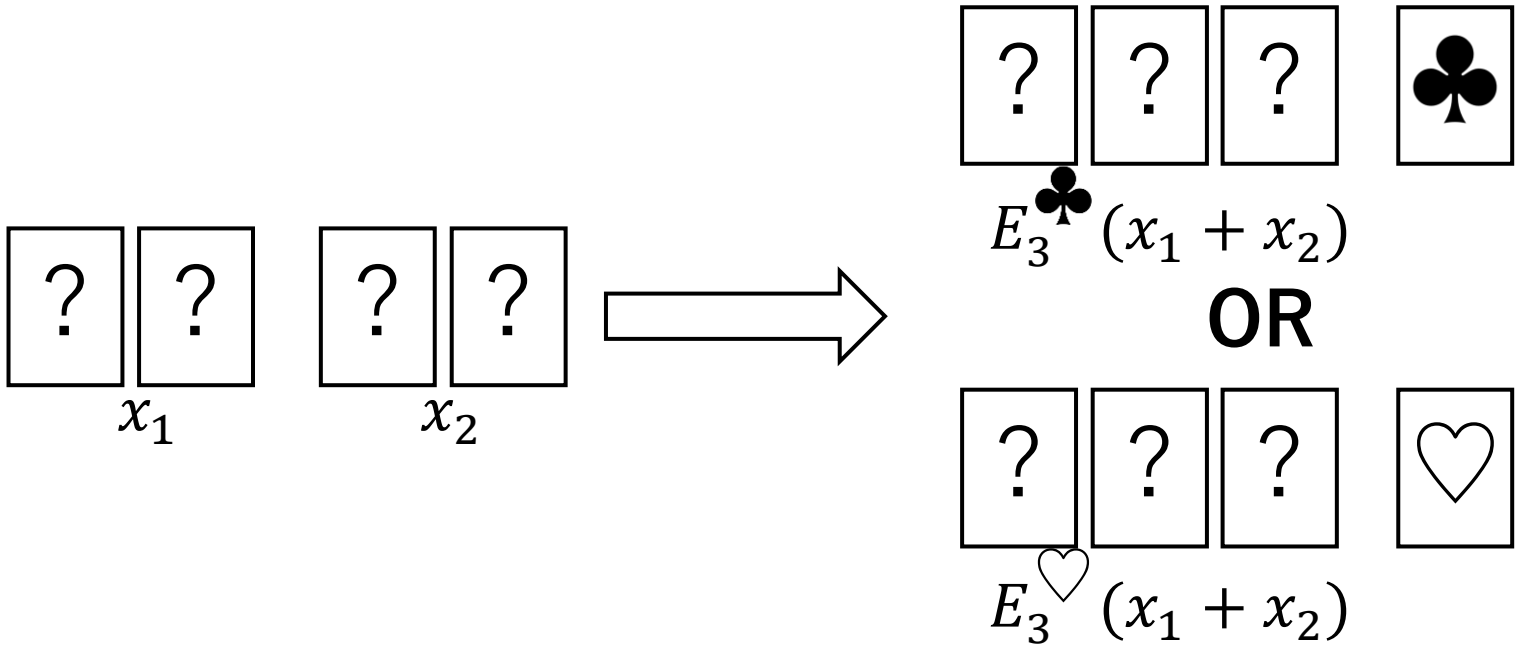


カード枚数：$2n + 1$枚(3色)
シャッフル回数：$n + |R|$



3枚エンコード加算

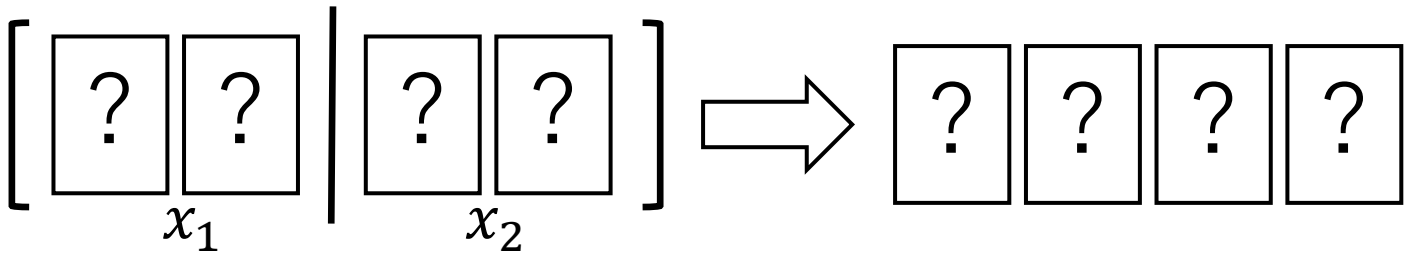
[Shikata et al. 22]によって提案されたプロトコル



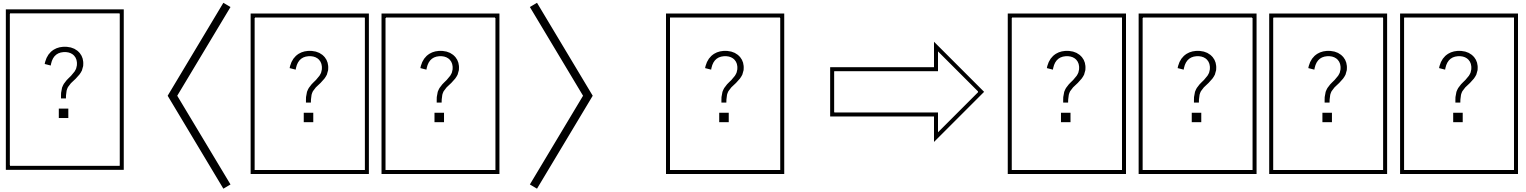
H. Shikata, K. Toyoda, D. Miyahara, and T. Mizuki. Card-minimal protocols for symmetric boolean functions of more than seven inputs. ICTAC 2022,

3枚エンコード加算

1. ランダム二等分割カットを適用

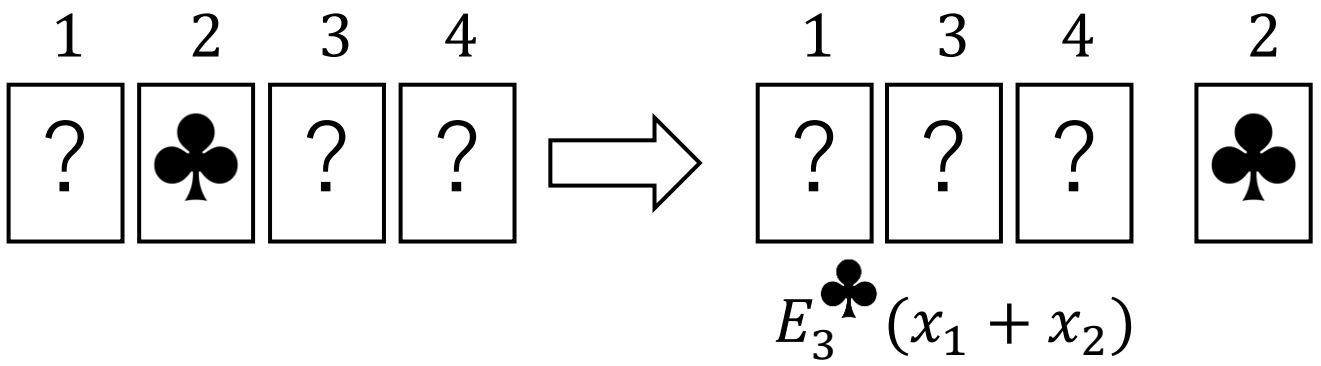


2. 中央の2枚に対してランダムカットを適用

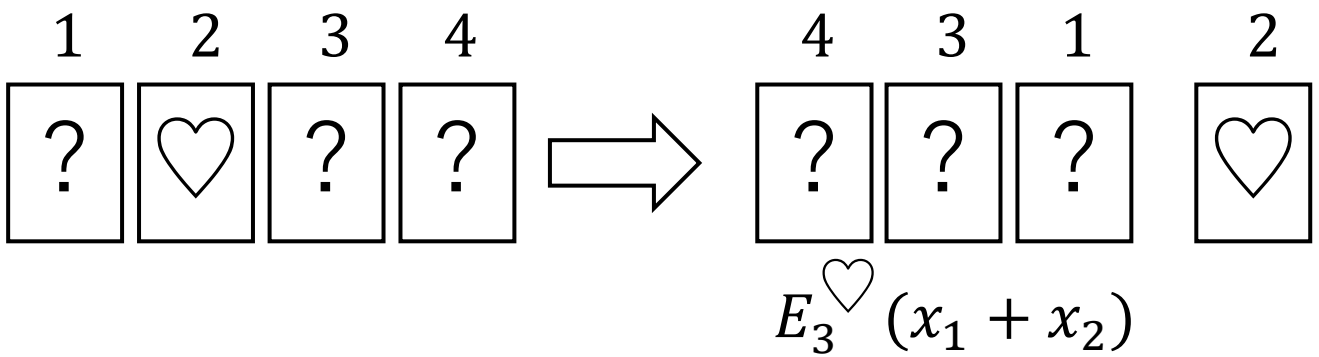


3. 左から2枚目のカードを開く

・開いたカードが ♣ のとき



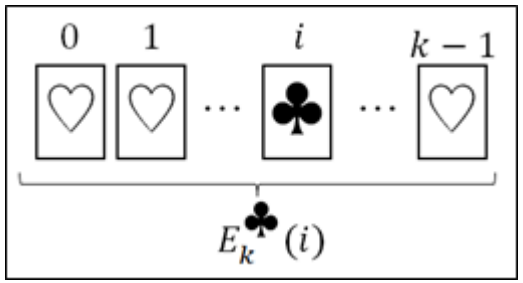
・開いたカードが ♡ のとき

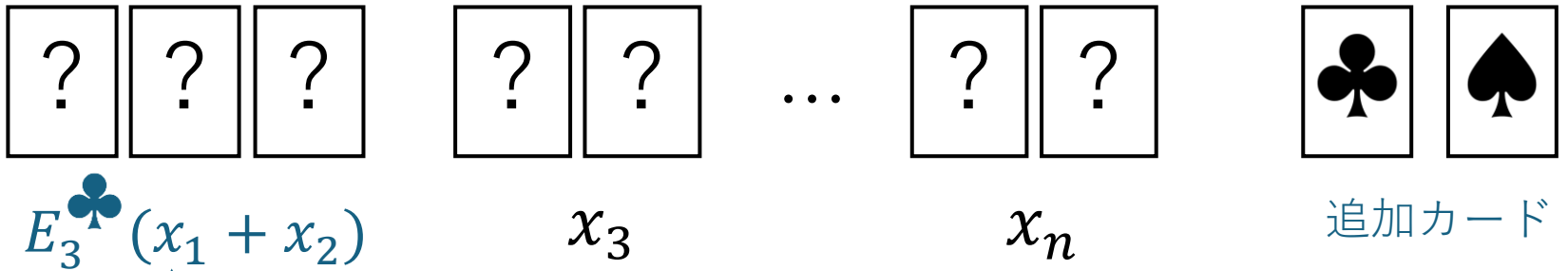


ランダムに出力

追加カードなしで
$x_1 + x_2$を加算

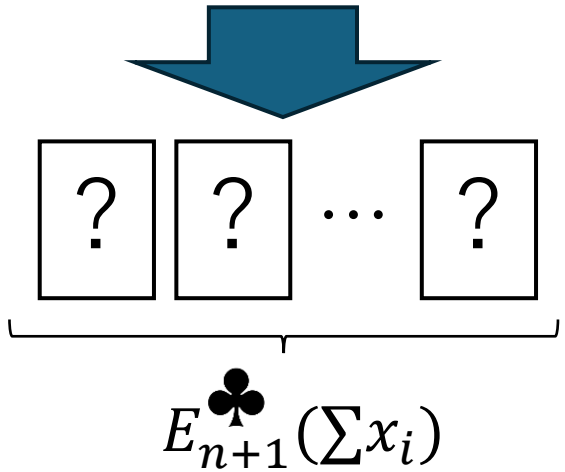
提案加算プロトコル②



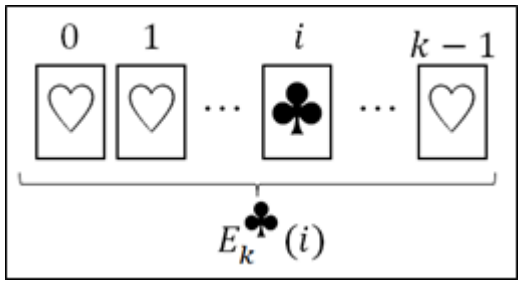


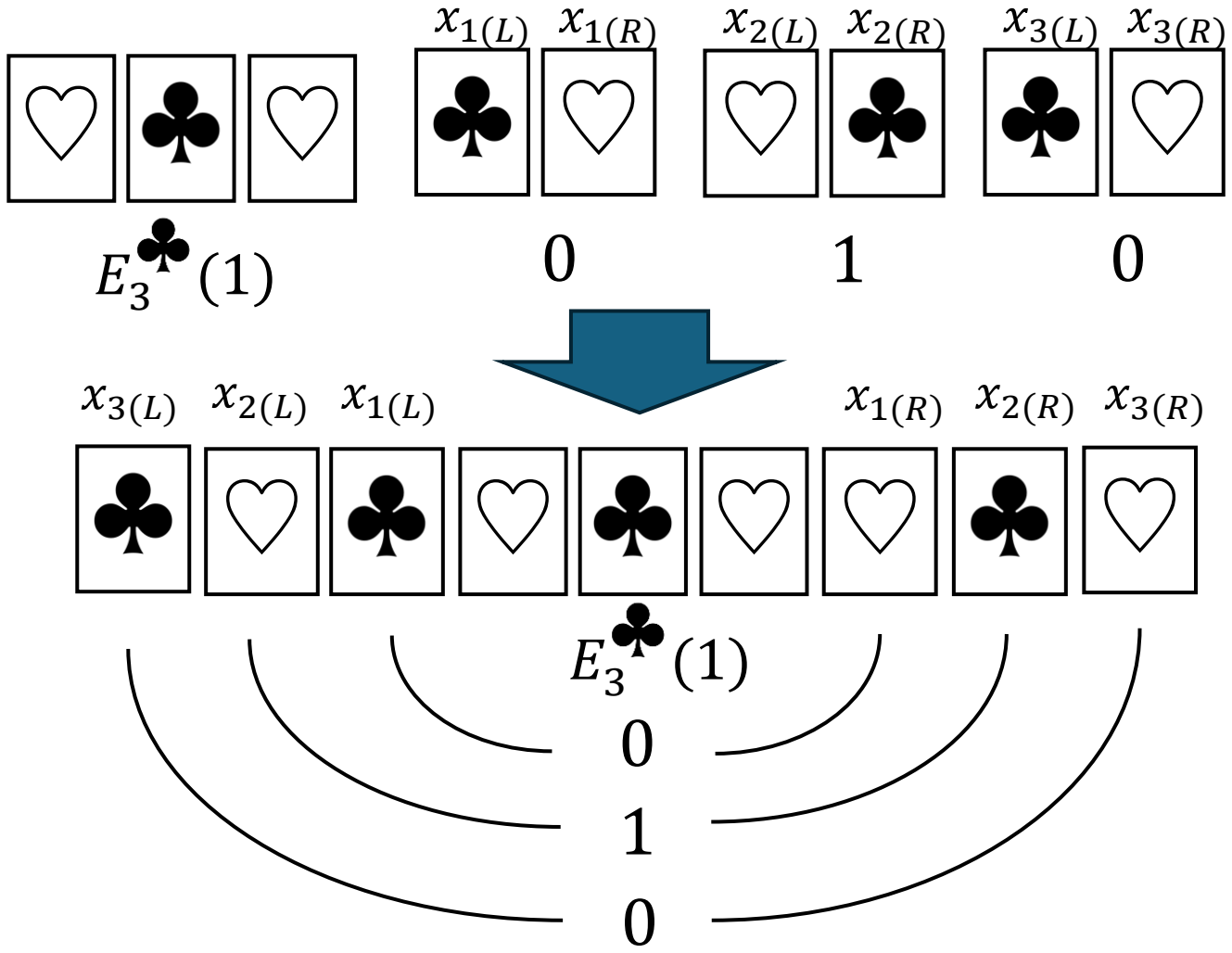
※ 整数符号化であればよい($E_2^{\heartsuit}(x_1)$、$E_k^{\clubsuit}(\sum y_i)$など)

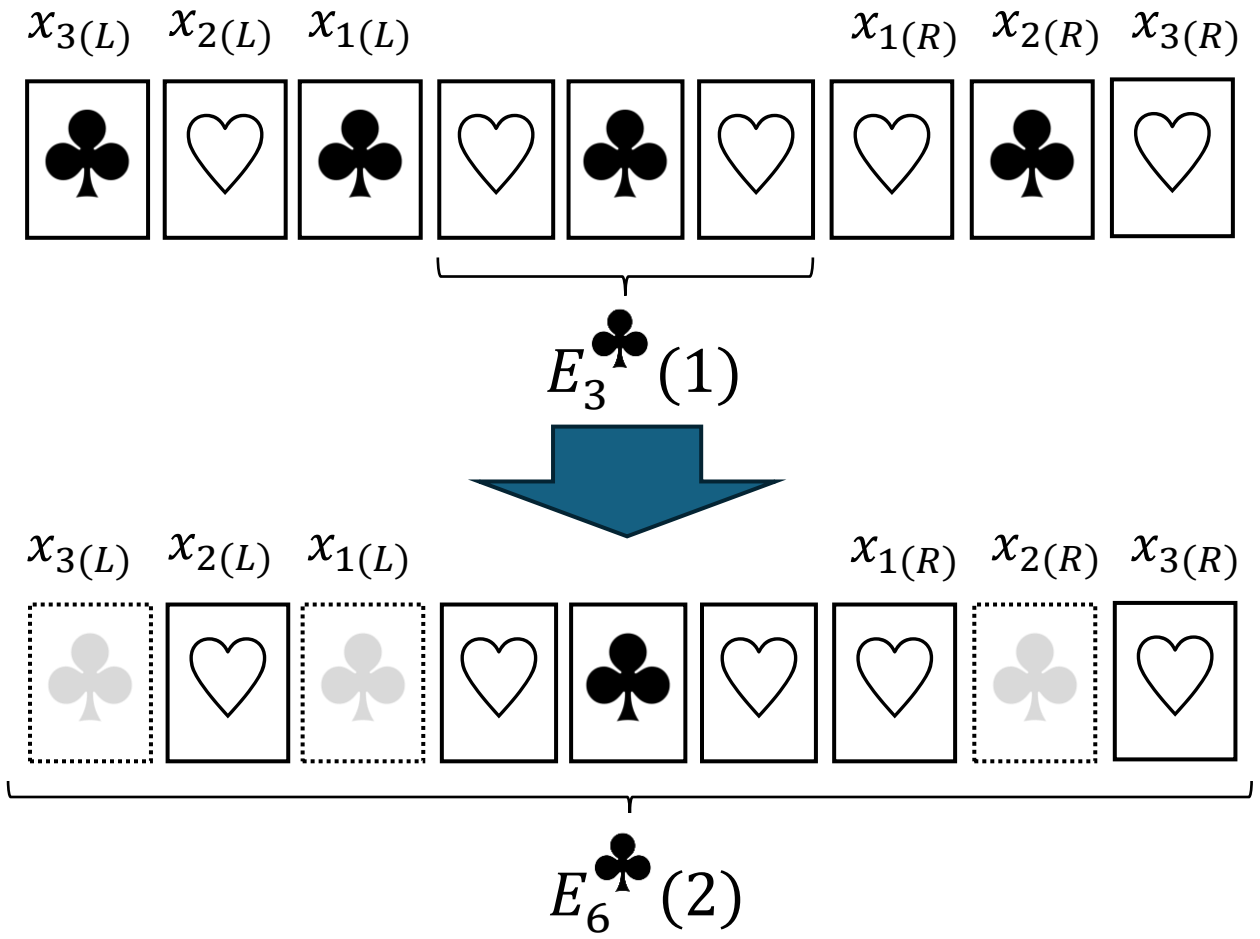
※ 異なる種類であればよい(♡♣、♢♠など)



提案加算プロトコル②のアイデア



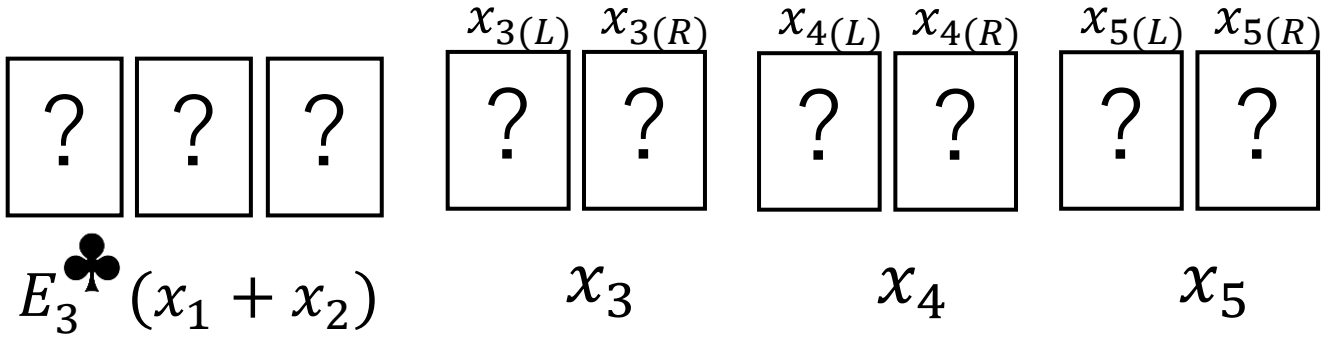




・x_1, x_2, x_3をそのままめくると入力が漏れる
・ランダム二等分割カットでランダマイズしてからオープンする

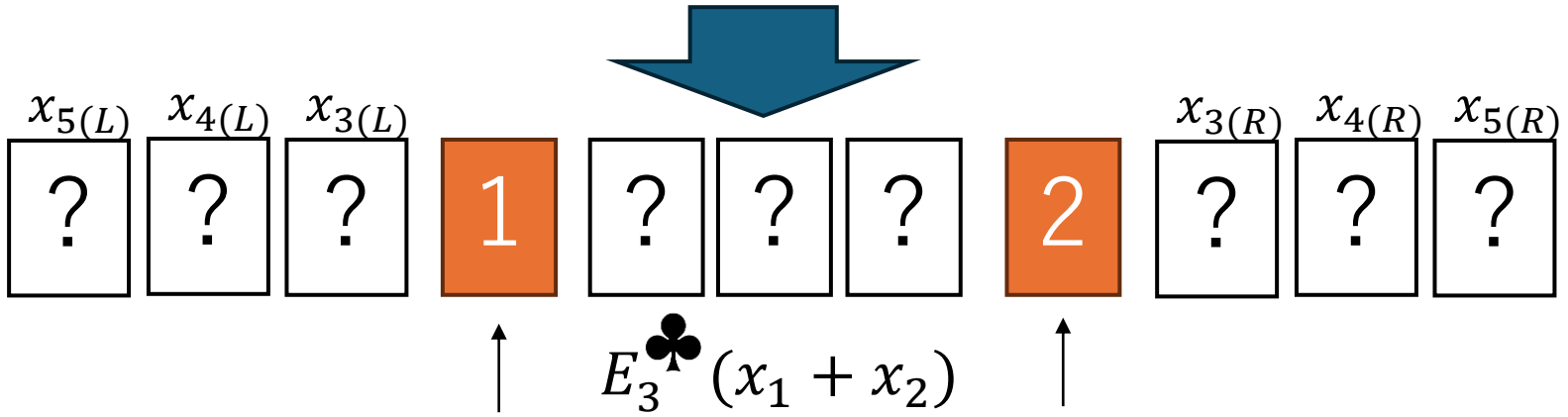
提案加算プロトコル②

1. カードを並べ替える

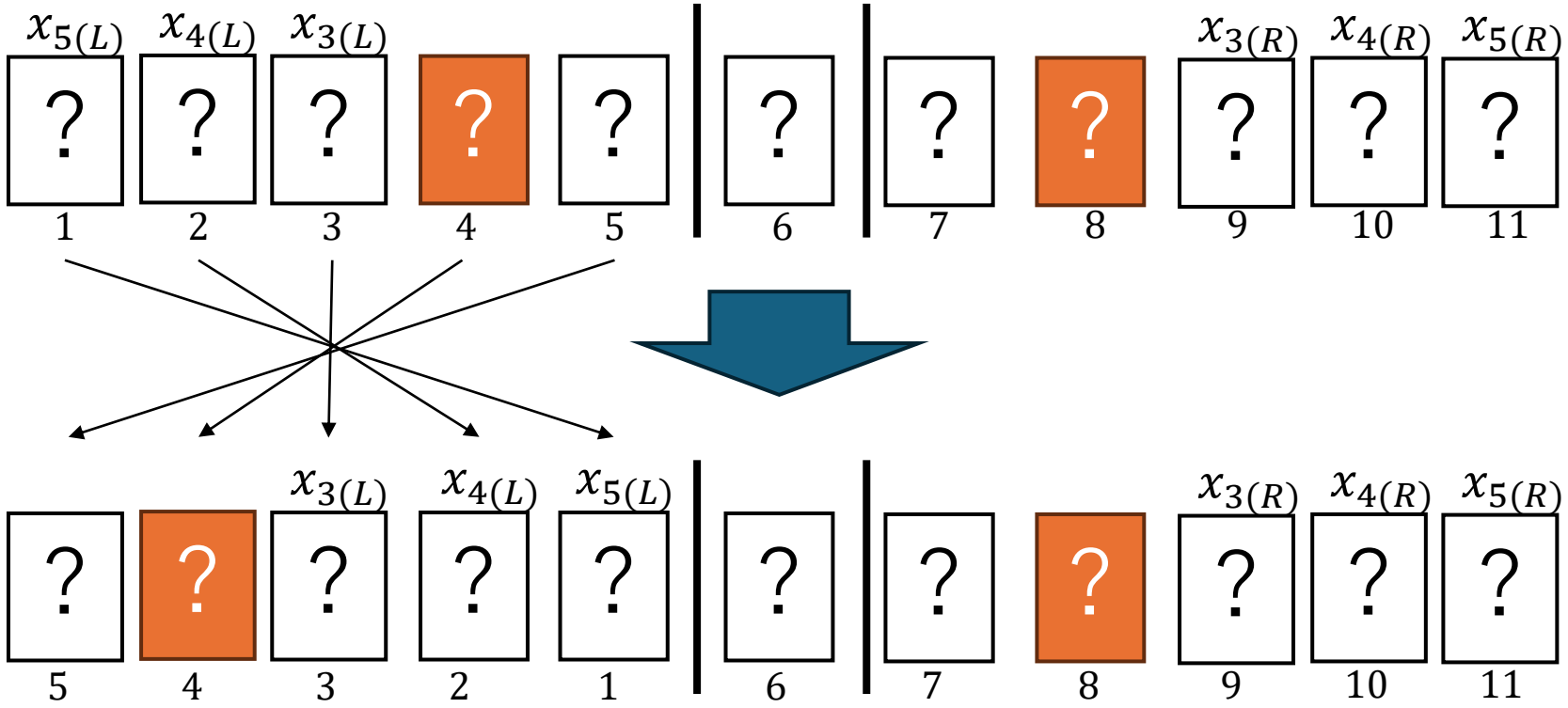


ナンバーカード

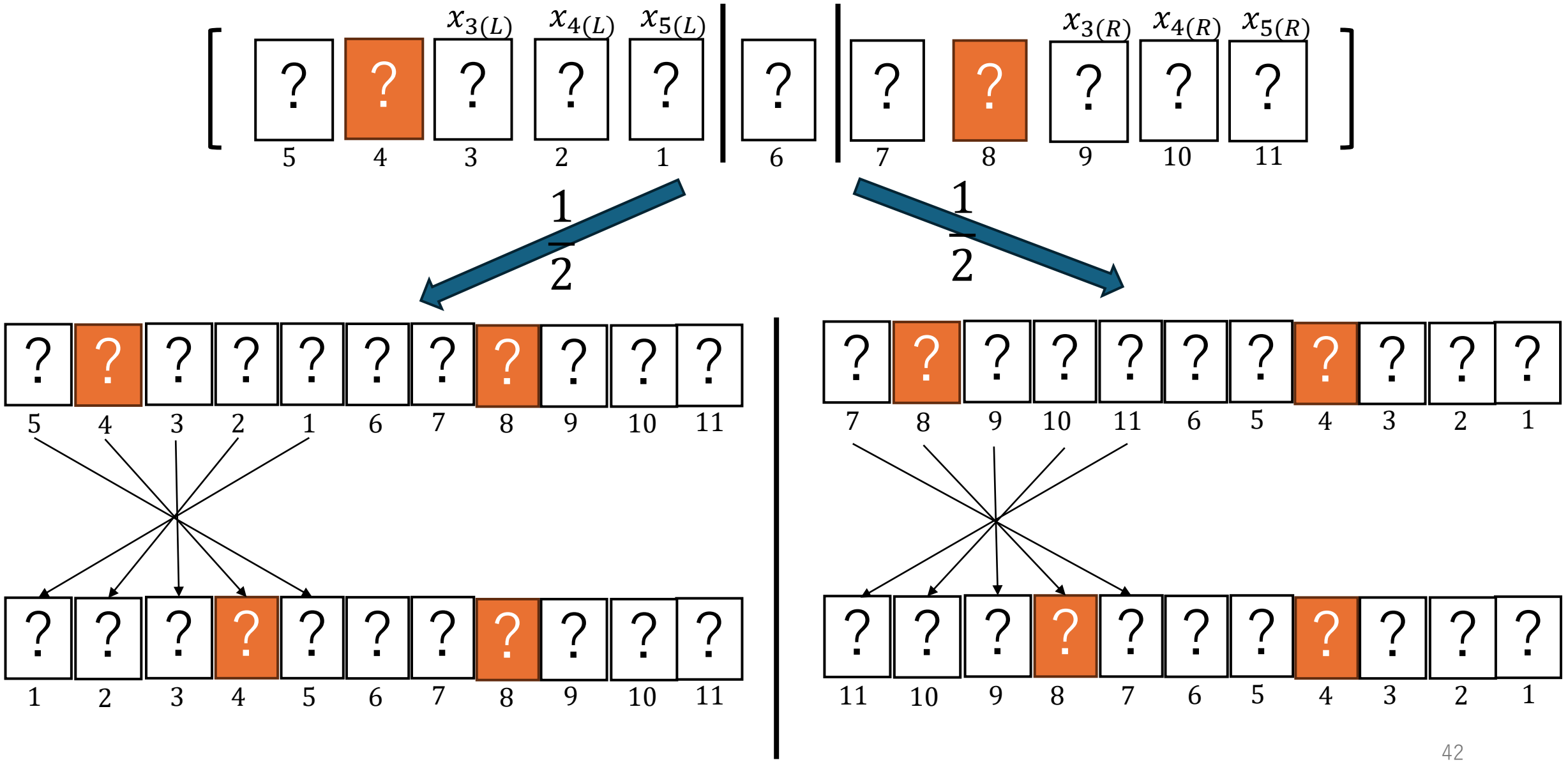
（実際は ♣ ♠ ）



2. カードを以下のように分割し、左側のカード列を並べ替える

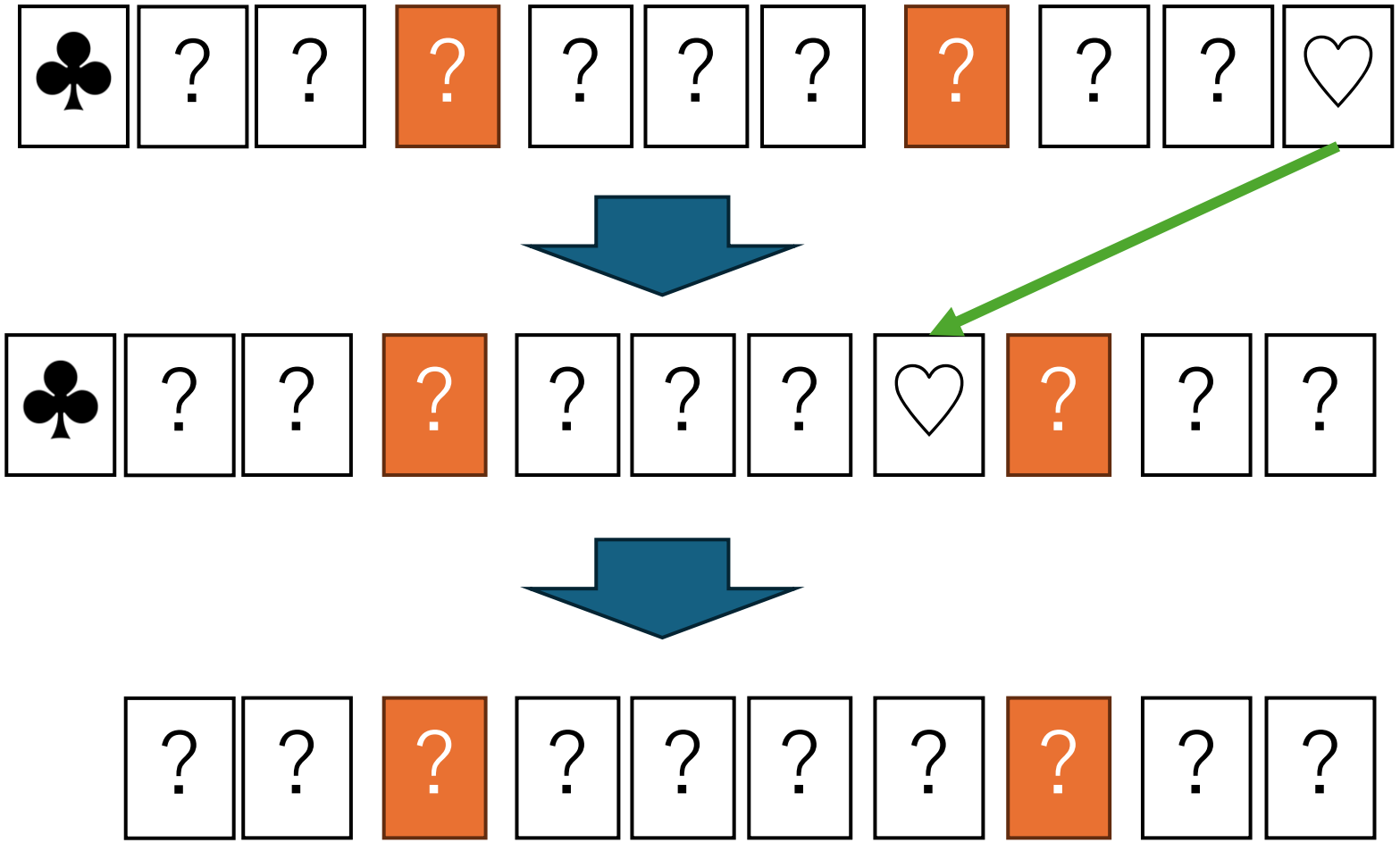


3. ランダム二等分割カットを実行し、並べ替える

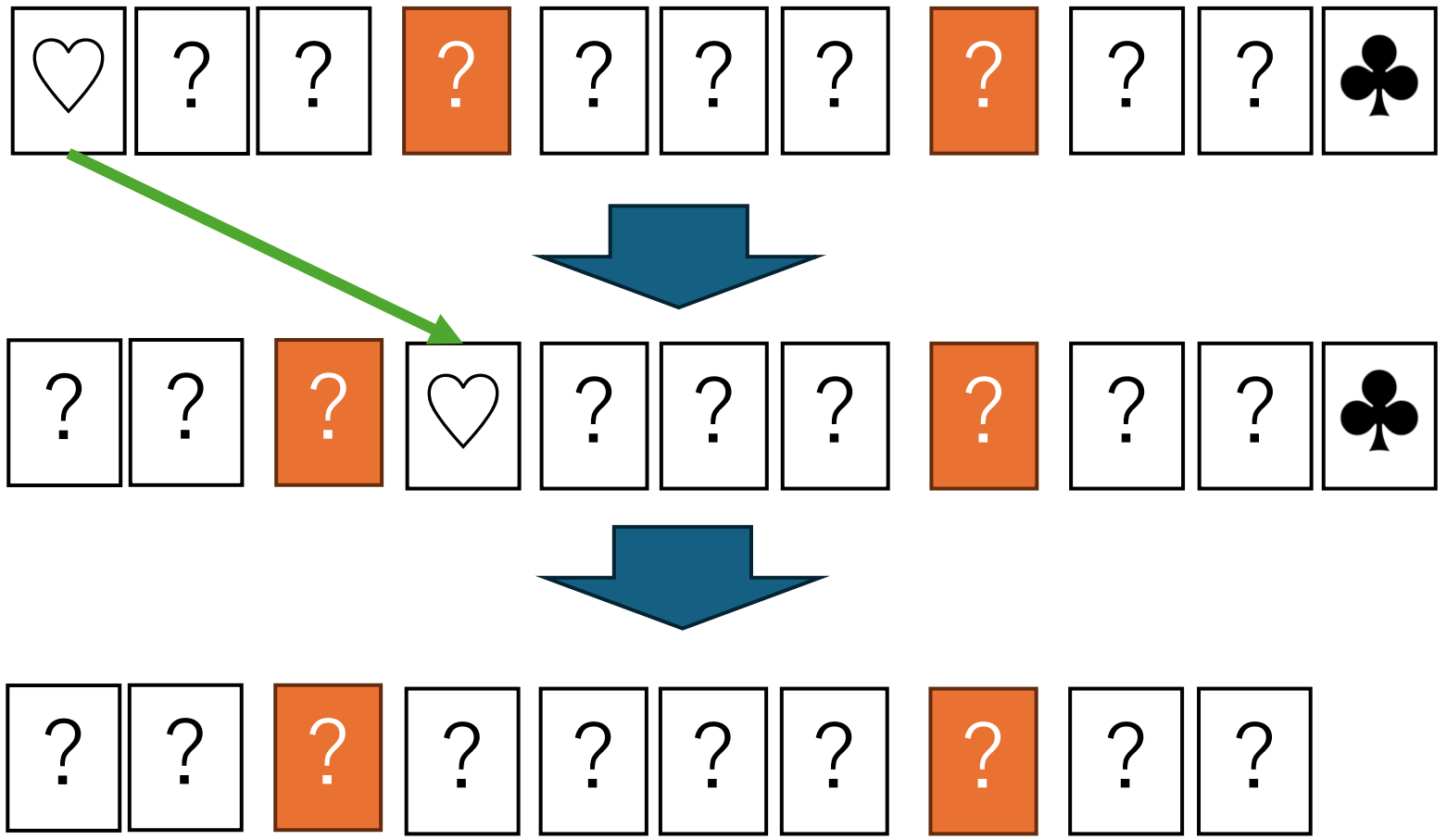


4. 両端のカードを開く

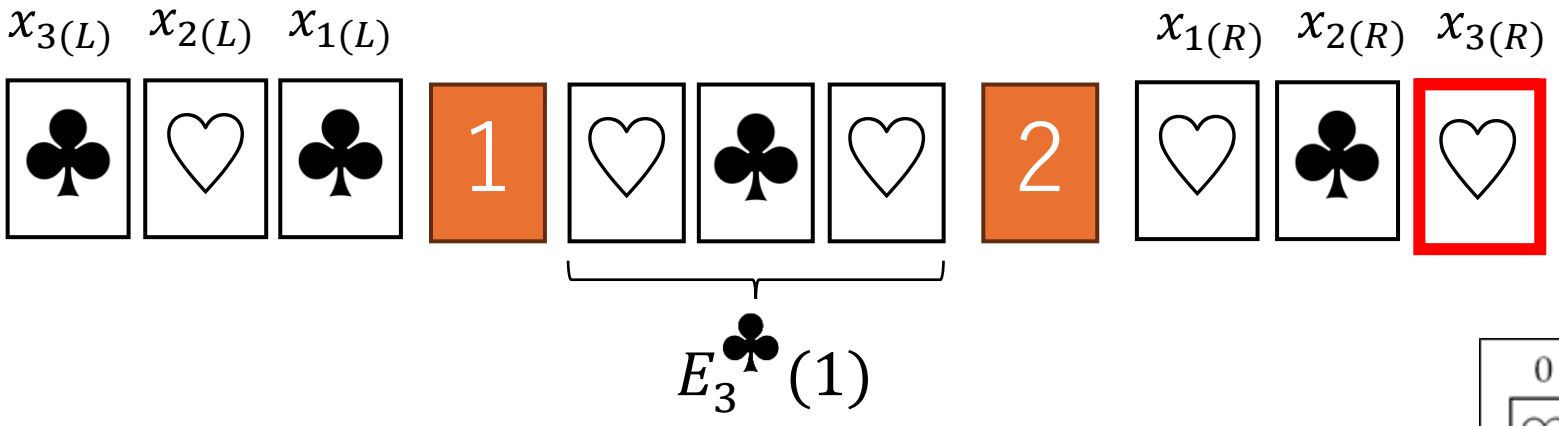
開いたカードが以下のとき

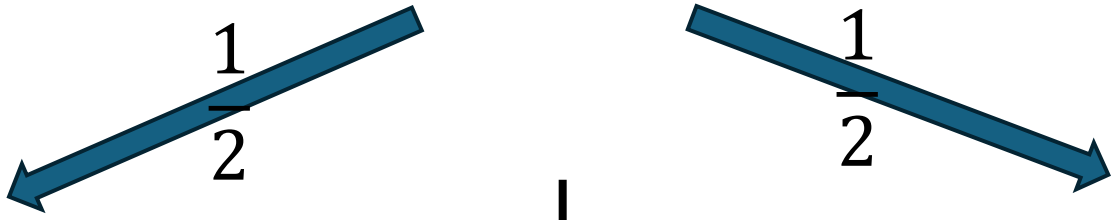


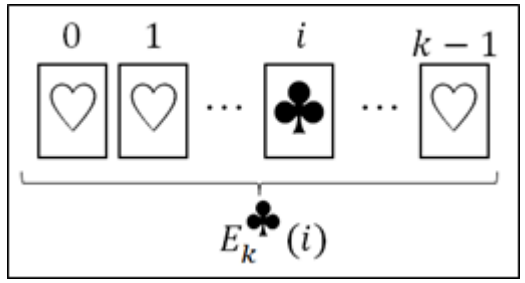
開いたカードが以下のとき

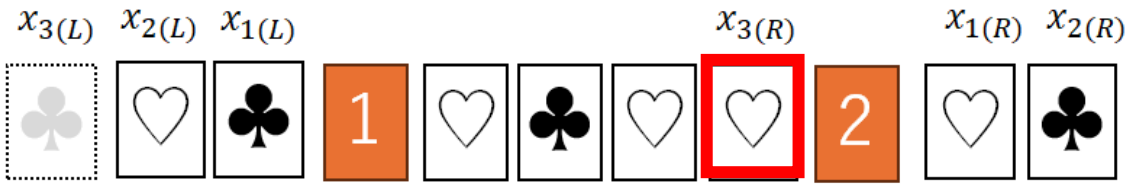


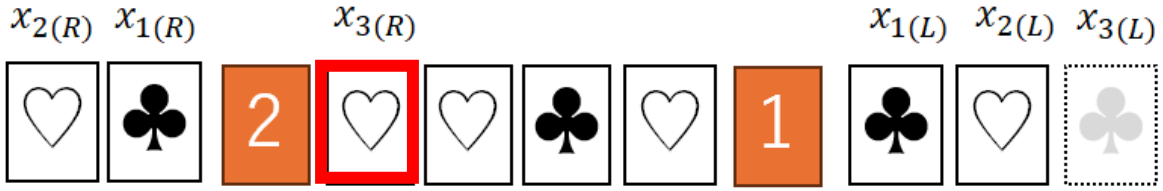
この手順で、x_5のコミットメントが加算された


$x_{3(L)}$ $x_{2(L)}$ $x_{1(L)}$
1
2
$x_{1(R)}$ $x_{2(R)}$ $x_{3(R)}$
$E_3^{\clubsuit}(1)$

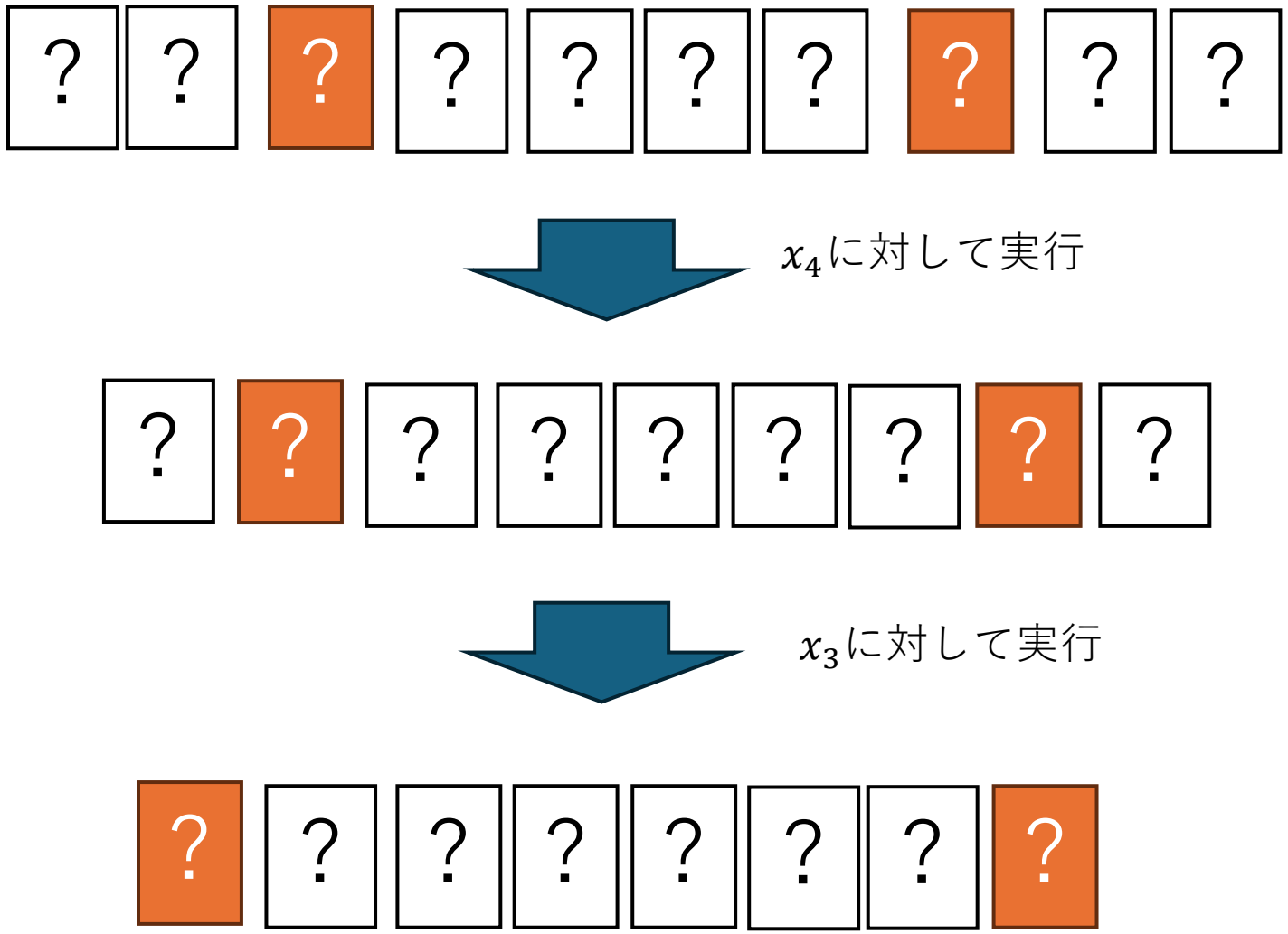

$\frac{1}{2}$
$\frac{1}{2}$


0
1
i
$k-1$
$E_k^{\clubsuit}(i)$

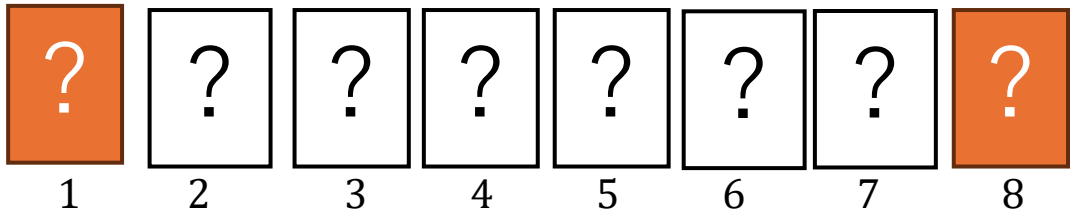

$x_{3(L)}$ $x_{2(L)}$ $x_{1(L)}$
1
$x_{3(R)}$
2
$x_{1(R)}$ $x_{2(R)}$

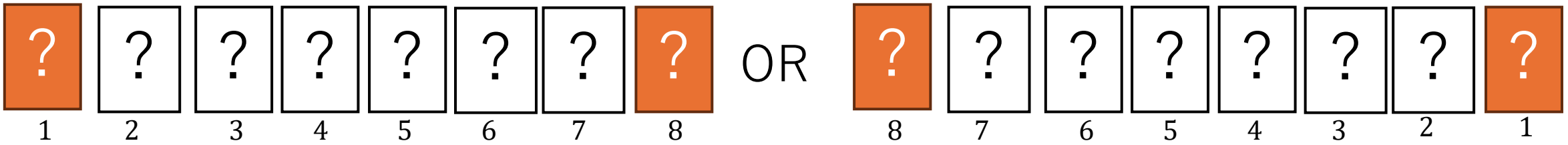

$x_{2(R)}$ $x_{1(R)}$
2
$x_{3(R)}$
1
$x_{1(L)}$ $x_{2(L)}$ $x_{3(L)}$

5. 同様の手順をx_3, x_4に対しても実行する



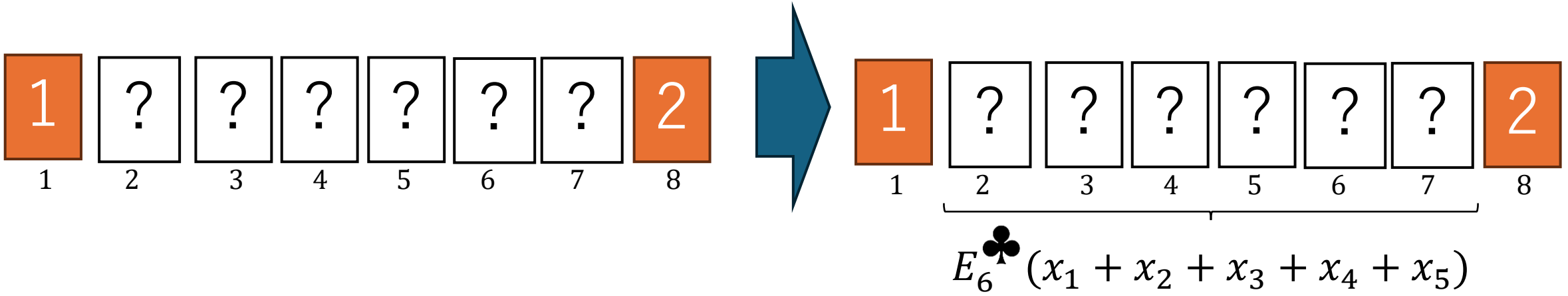
6. 手順2,3を実行する



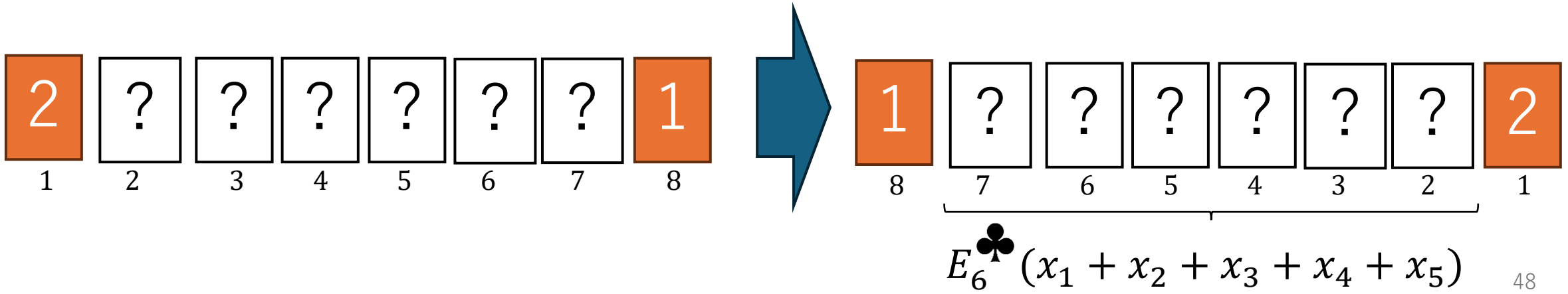


7. ナンバーカードを開く

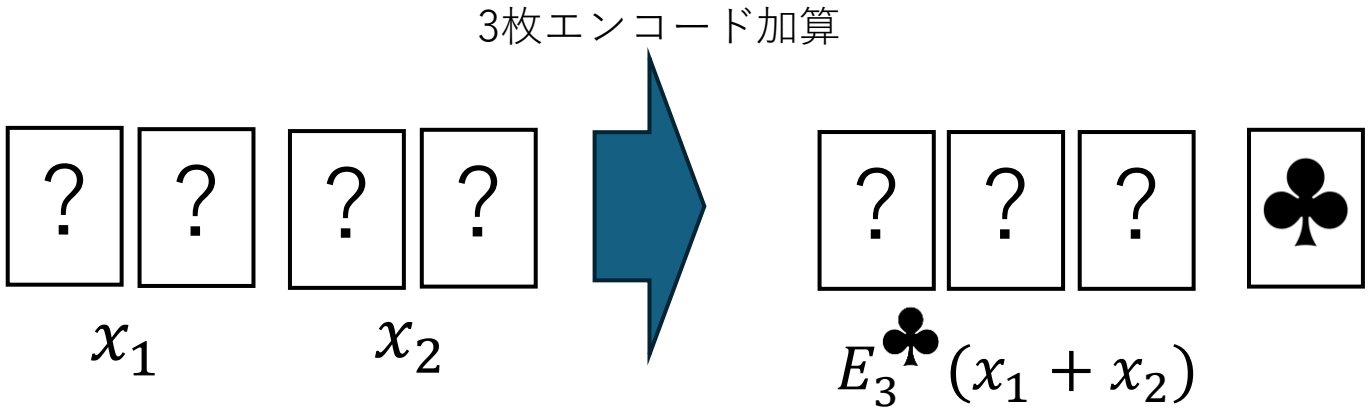
(a)以下の状態の時

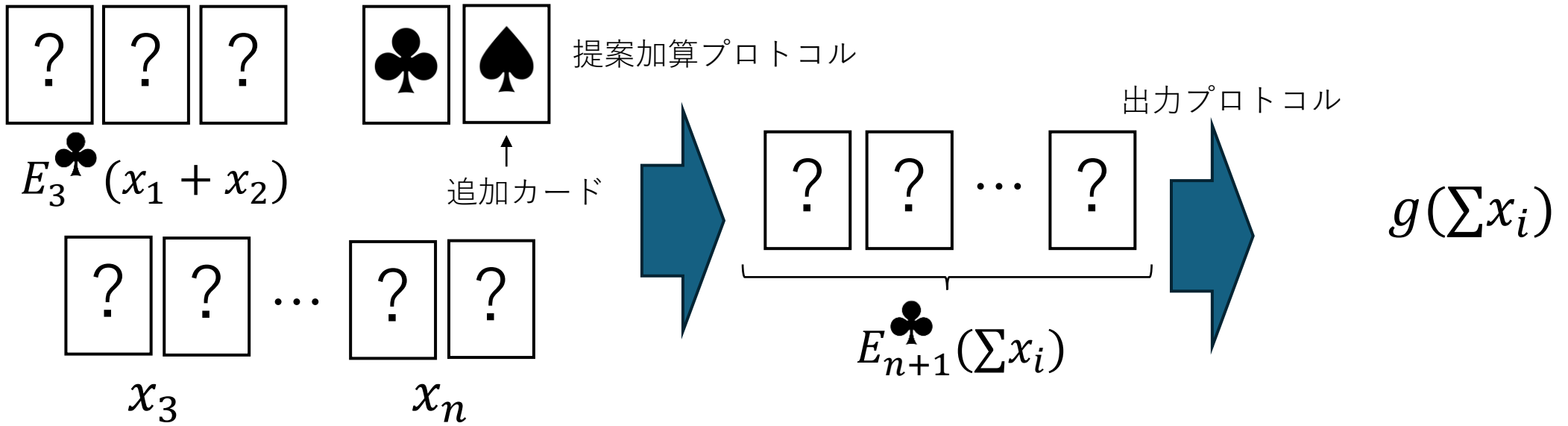


(b)以下の状態の時



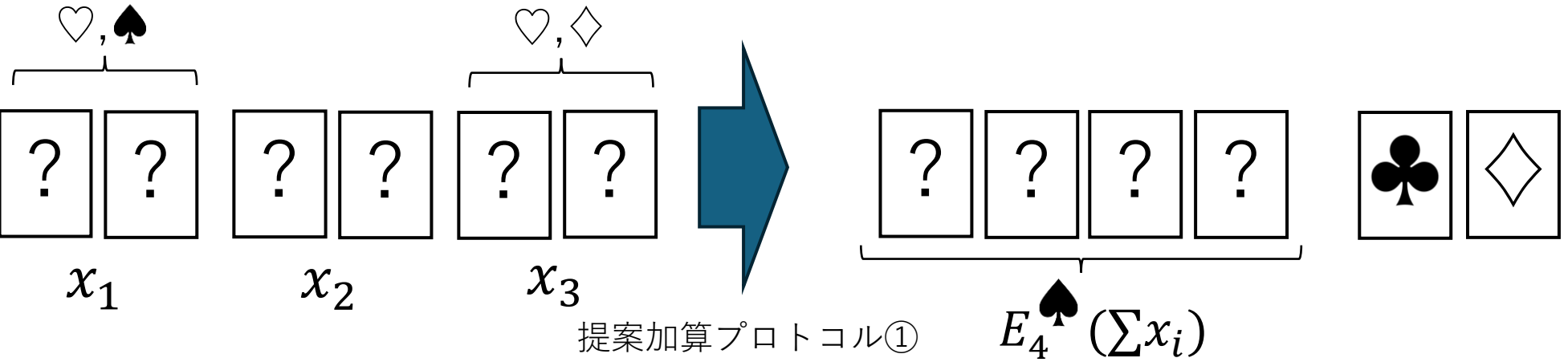
提案プロトコル②の流れ



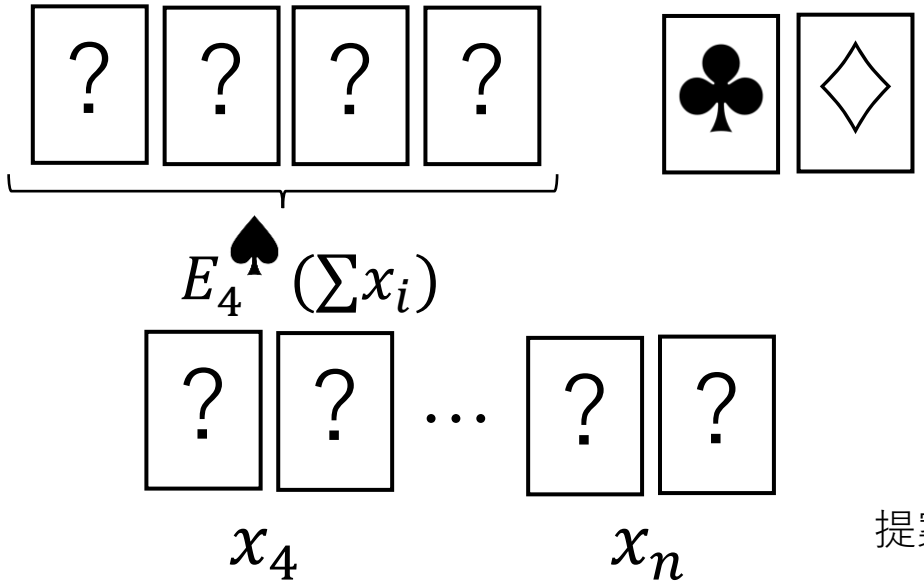


考案者	色数	カード枚数	シャッフル回数	有限時間	シャッフル
Ruangwises-Itoh(TAMC2020)	2	$2n+2$	$n+\|R\|-2$	○	PShift
提案方式①	4	$2n$	$2n+\|R\|-1+(n-2)\sum_{i}^{n-1}\frac{1}{i}$	×	RC
提案方式②	3	$2n+1$	$n+\|R\|$	○	RC and RBC
提案方式③	**4**	$\boldsymbol{2n}$	$\boldsymbol{n+\|R\|+8}$	×	**RC and RBC**

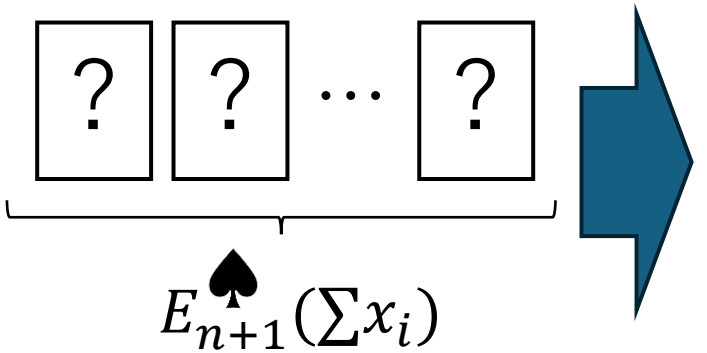
提案プロトコル③の流れ



提案加算プロトコル①



提案加算プロトコル②



出力プロトコル

$g(\sum x_i)$

まとめ

成果

- 4色$2n$枚のLas Vegasプロトコル（RCのみ）
- 3色$2n+1$枚の有限時間プロトコル
- シャッフル回数を削減した4色$2n$枚のLas Vegasプロトコル

現在の進捗

- 2色$2n+1$枚のLas Vegasプロトコル

未解決問題

- 2色$2n+1$枚の有限時間プロトコルは構成可能か？
- 2色または3色$2n$枚のLas Vegasプロトコルは構成可能か？