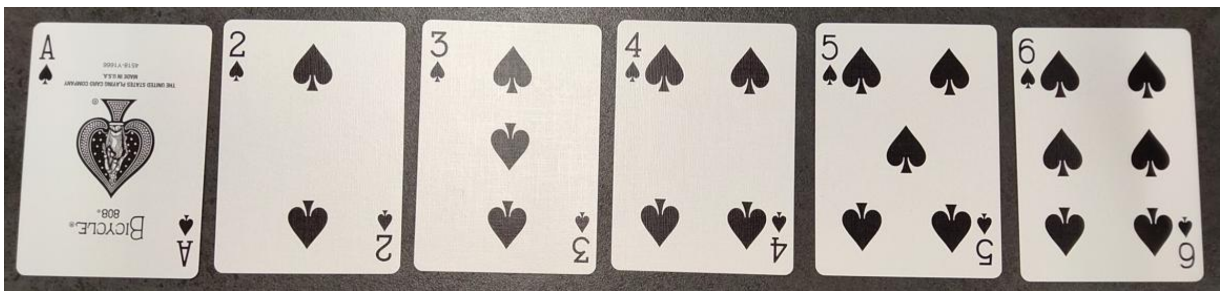
Six Standard Playing Cards Are Sufficient for All Three-Input Boolean Functions

Masanori Kabutomori and Takaaki Mizuki

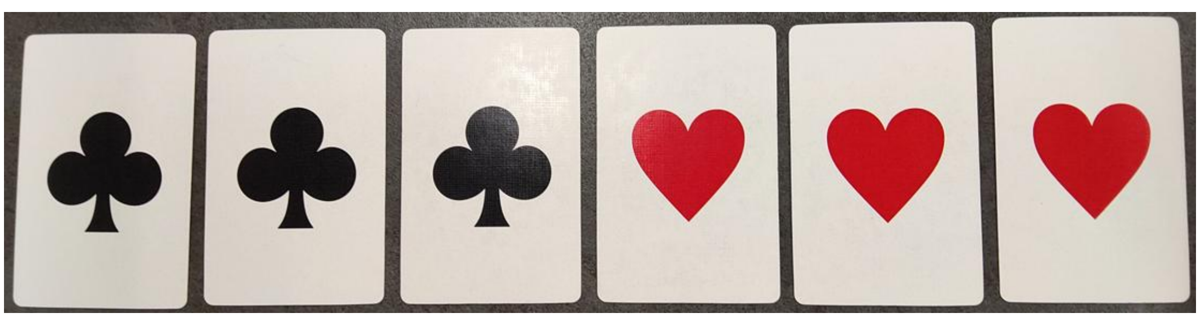
Tohoku University

Six *Standard Playing Cards* Are Sufficient for All Three-Input Boolean Functions

Standard playing cards

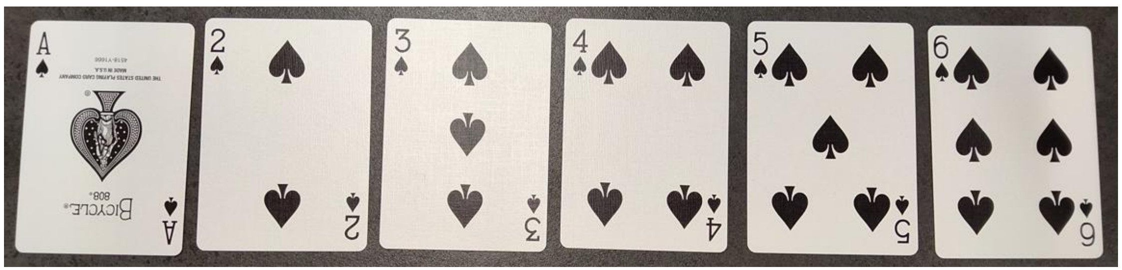


Two-color deck of cards

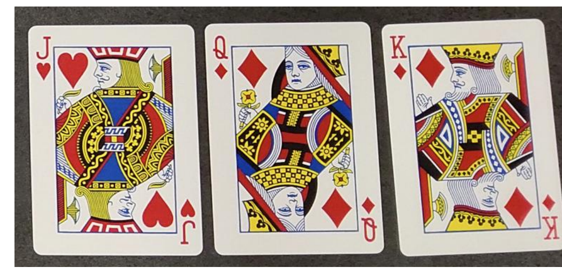


Six *Standard Playing Cards* Are Sufficient for All Three-Input Boolean Functions

Standard playing cards



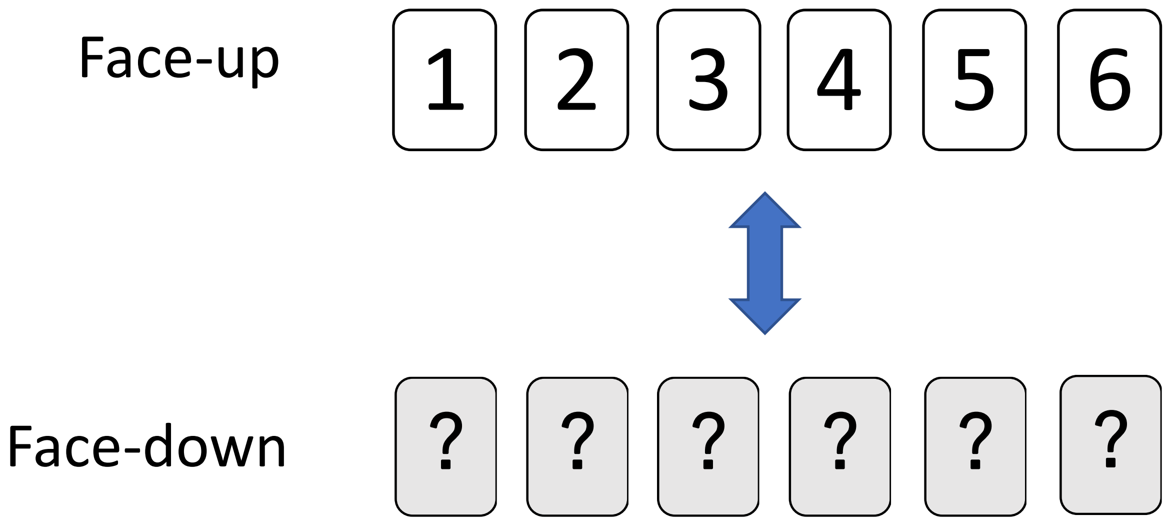
…



1 2 3 4 5 6 … 50 51 52

Numbered cards

Six Standard Playing *Cards* Are Sufficient for All Three-Input Boolean Functions



Six Standard Playing Cards Are Sufficient for All *Three-Input Boolean Functions*

$a \in \{0,1\}$

Alice

$b \in \{0,1\}$

Bob

$c \in \{0,1\}$

Carol

Secure computation

$f(a, b, c)$

Six Standard Playing Cards Are Sufficient for All Three-Input Boolean Functions

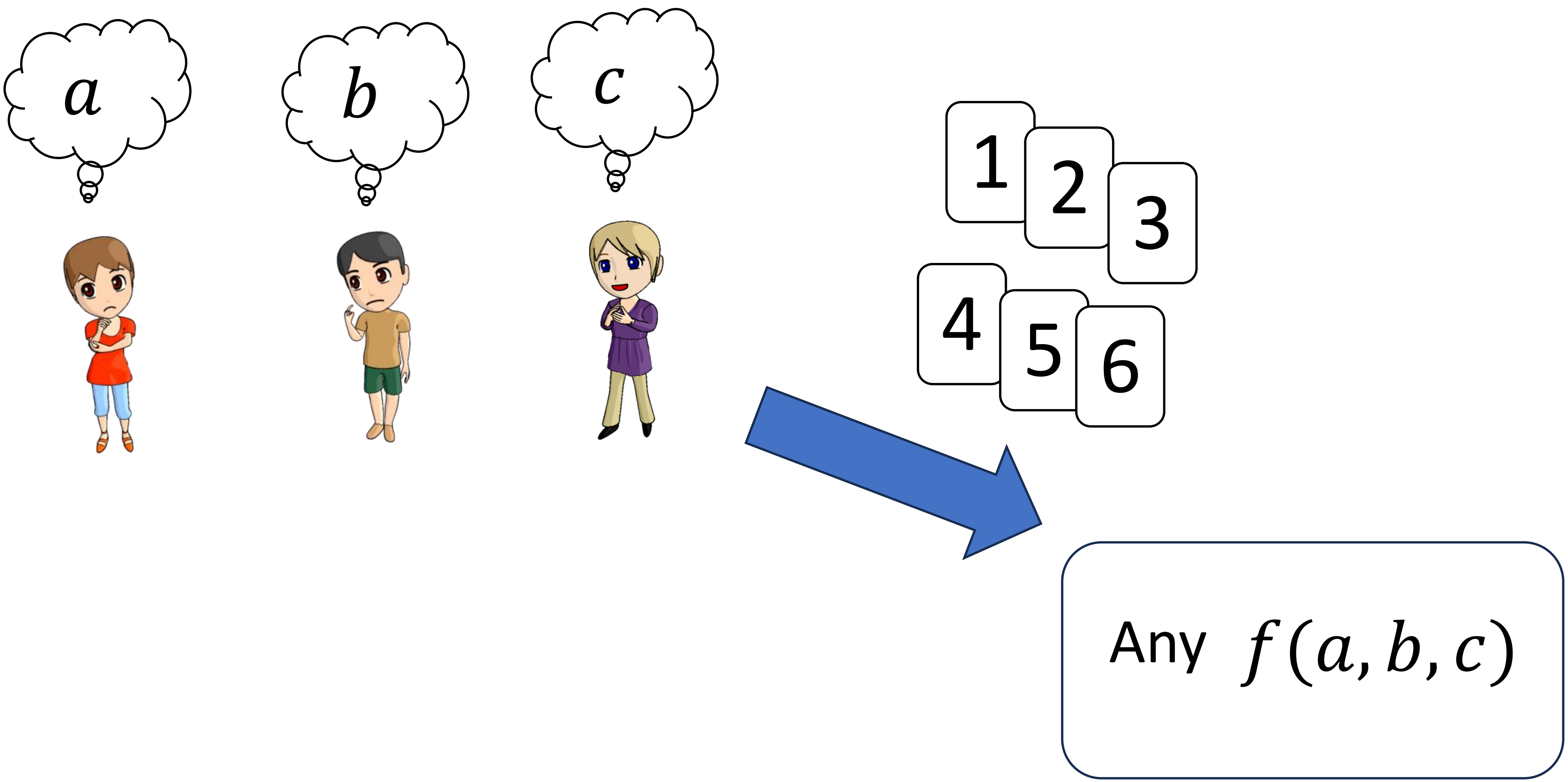


Table of Contents

Table of Contents

1. Introduction
2. Haga et al's Pro
3. Analysis by NPN Equivalence classes
4. Proposed Protocols
5. Conclusion

Encoding and Commitment
Know Results
Our Contribution

Encoding

For $i < j$,

$$\boxed{i}\,\boxed{j} = 0$$

$$\boxed{j}\,\boxed{i} = 1$$

Example:

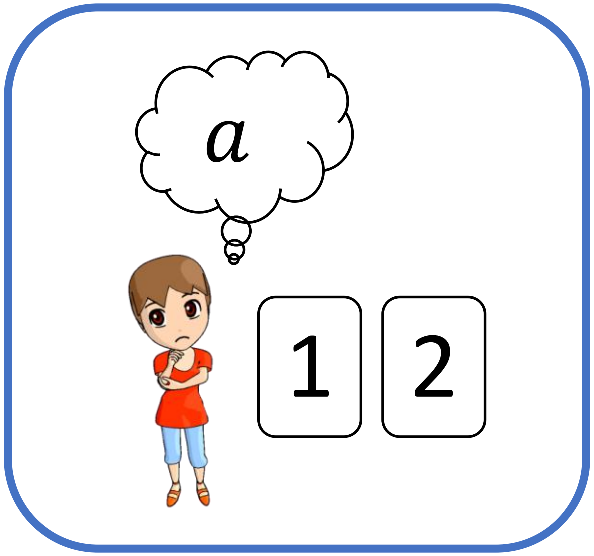
$$\boxed{1}\,\boxed{2} = 0$$

$$\boxed{2}\,\boxed{1} = 1$$

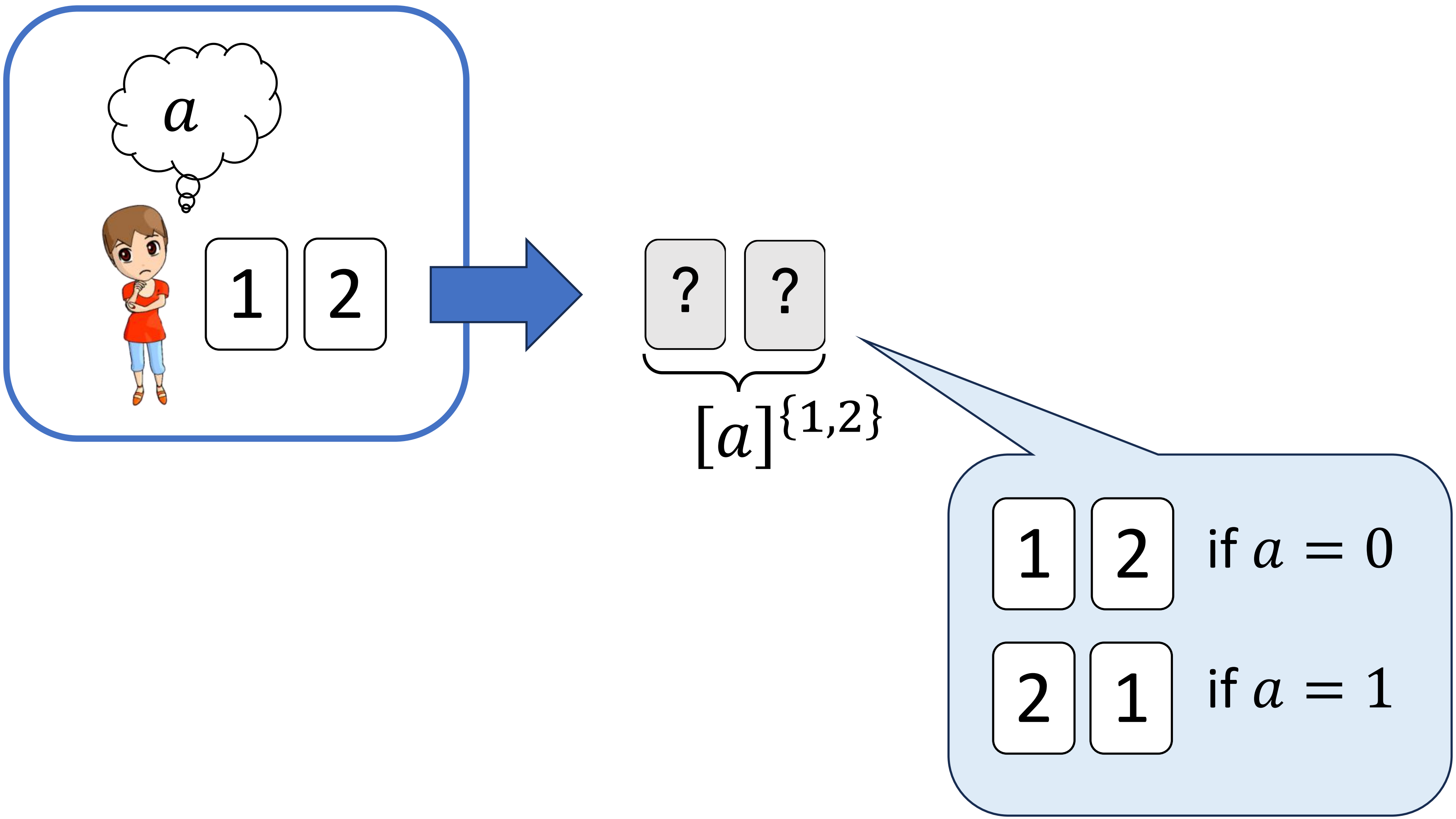
$$\boxed{1}\,\boxed{4} = 0$$

$$\boxed{4}\,\boxed{1} = 1$$

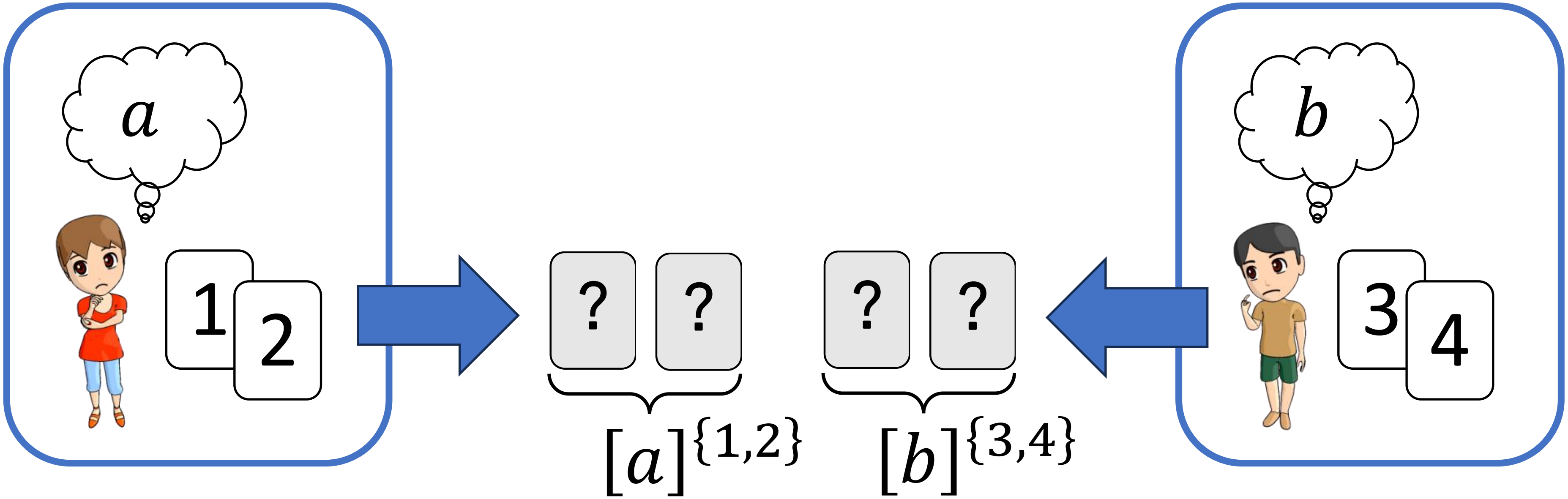
Commitment



Commitment

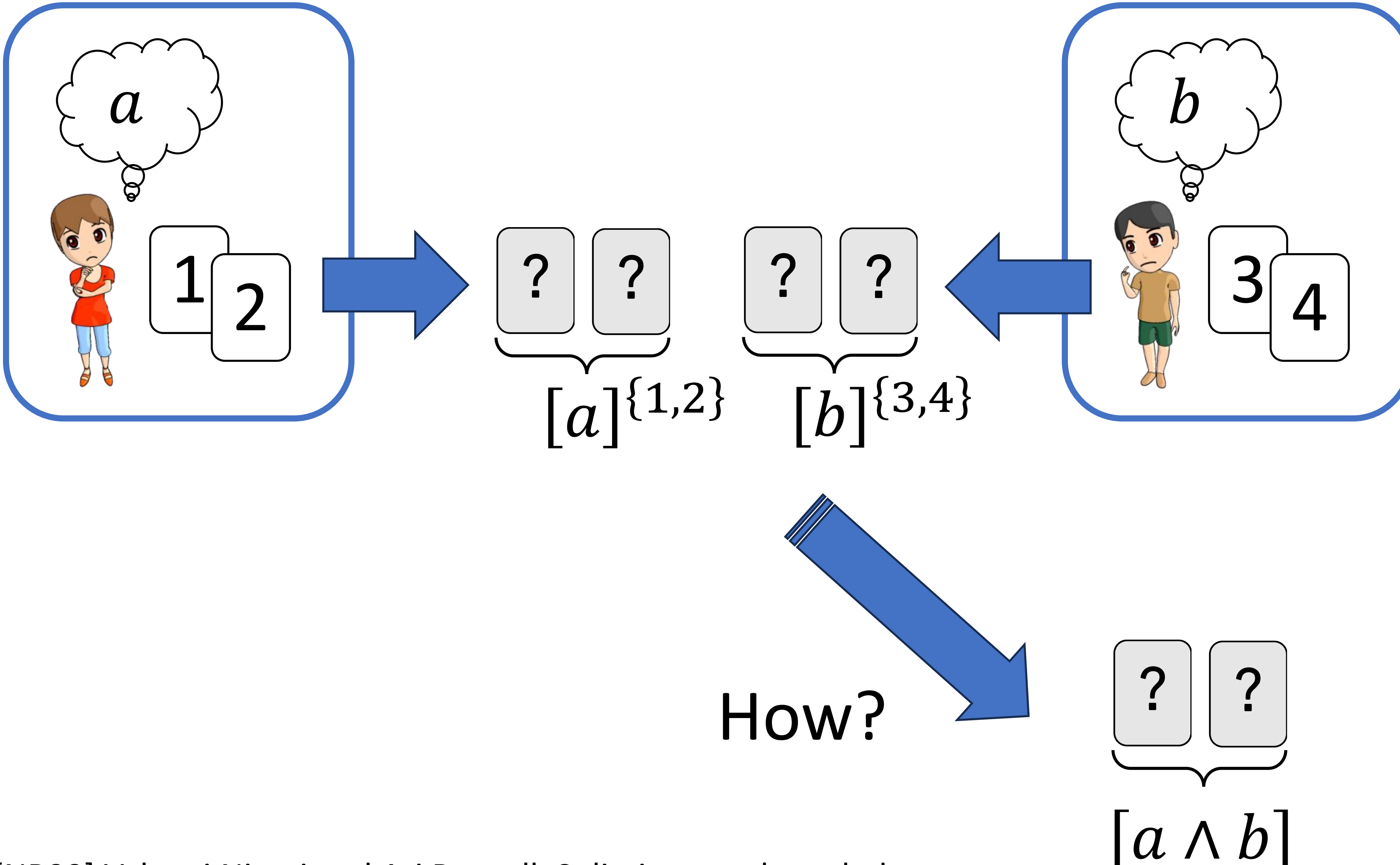


AND protocol [NR99]



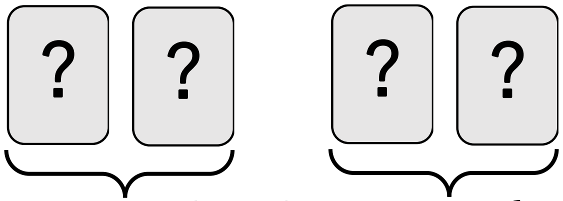
[NR99] Valtteri Niemi and Ari Renvall. Solitaire zero-knowledge. Fundam.Inf., 38(1,2):181–188, 1999

AND protocol [NR99]



[NR99] Valtteri Niemi and Ari Renvall. Solitaire zero-knowledge. Fundam.Inf., 38(1,2):181–188, 1999

Idea



$[a]^{\{1,2\}}$ $[b]^{\{3,4\}}$

$(\boldsymbol{a}, \boldsymbol{b})$				
(0,0)	1	2	3	4
(0,1)	1	2	4	3
(1,0)	2	1	3	4
(1,1)	2	1	4	3

$\underbrace{\boxed{?}\ \boxed{?}}_{[a]^{\{1,2\}}}$ $\underbrace{\boxed{?}\ \boxed{?}}_{[b]^{\{3,4\}}}$

(a,b)				
(0,0)	1	2	3	4
(0,1)	1	2	4	3
(1,0)	2	1	3	4
(1,1)	2	1	4	3

If we ignore $\boxed{2}\ \boxed{3}$

(a,b)				
(0,0)	1	2	3	4
(0,1)	1	2	4	3
(1,0)	2	1	3	4
(1,1)	2	1	4	3

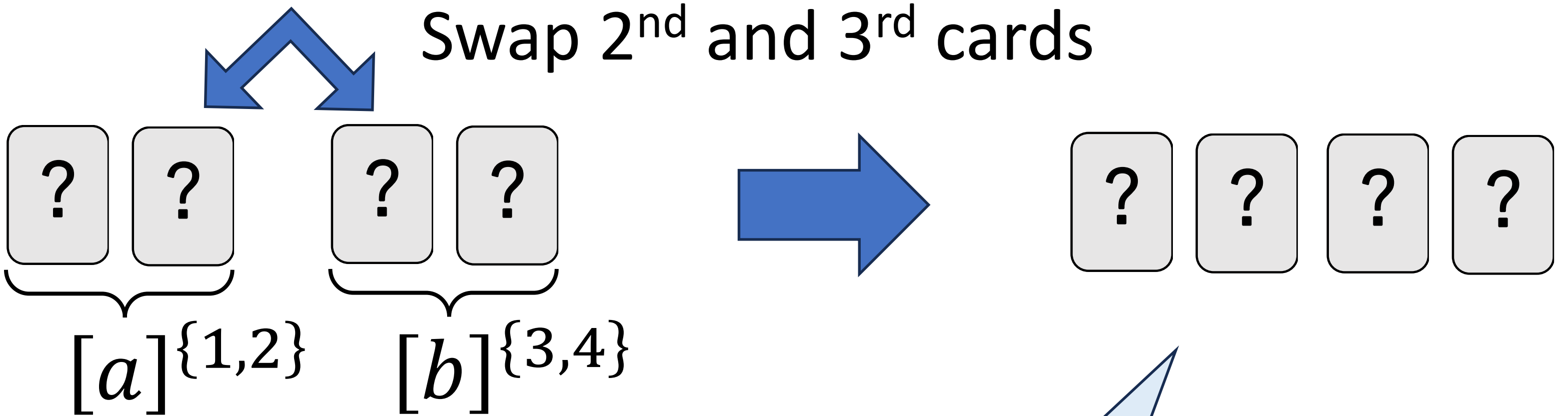
$\boxed{1}\ \boxed{4} = 0$

$\boxed{1}\ \boxed{4} = 0$

$\boxed{1}\ \boxed{4} = 0$

$\boxed{1}\ \boxed{4} = 0$

Swap 2nd and 3rd cards



(a,b)				
(0,0)	1	2	3	4
(0,1)	1	2	4	3
(1,0)	2	1	3	4
(1,1)	2	1	4	3

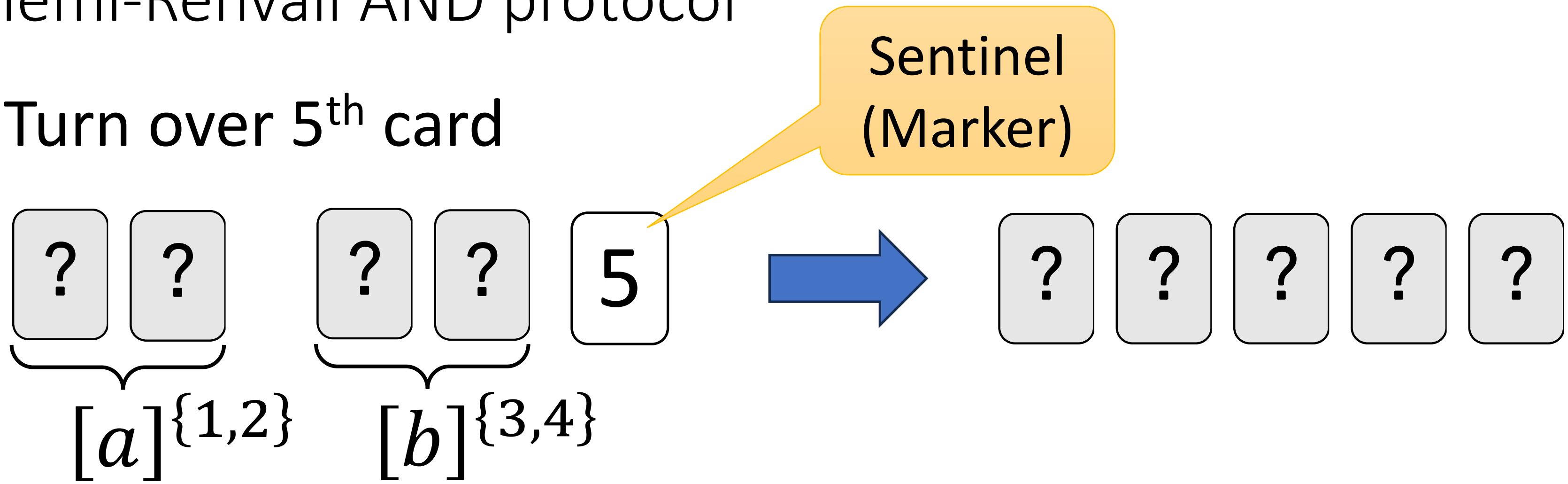
(a,b)				
(0,0)	1	3	2	4
(0,1)	1	4	2	3
(1,0)	2	3	1	4
(1,1)	2	4	1	3

$$\left.\begin{array}{l} \boxed{1}\,\boxed{4} = 0 \\ \boxed{1}\,\boxed{4} = 0 \\ \boxed{1}\,\boxed{4} = 0 \\ \boxed{4}\,\boxed{1} = 1 \end{array}\right\} a \wedge b$$

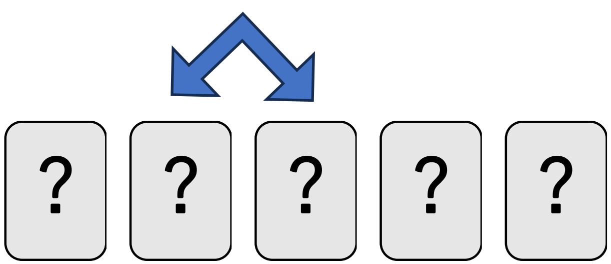
We want to remove $\boxed{2}\,\boxed{3}$

Niemi-Renvall AND protocol

1. Turn over 5^{th} card

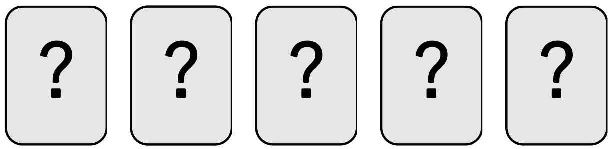


2. Swap 2^{nd} and 3^{rd} cards

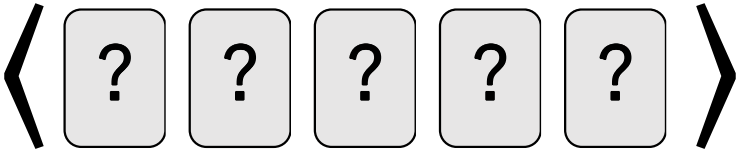


3. Remove $\boxed{2}\boxed{3}$ by random cuts

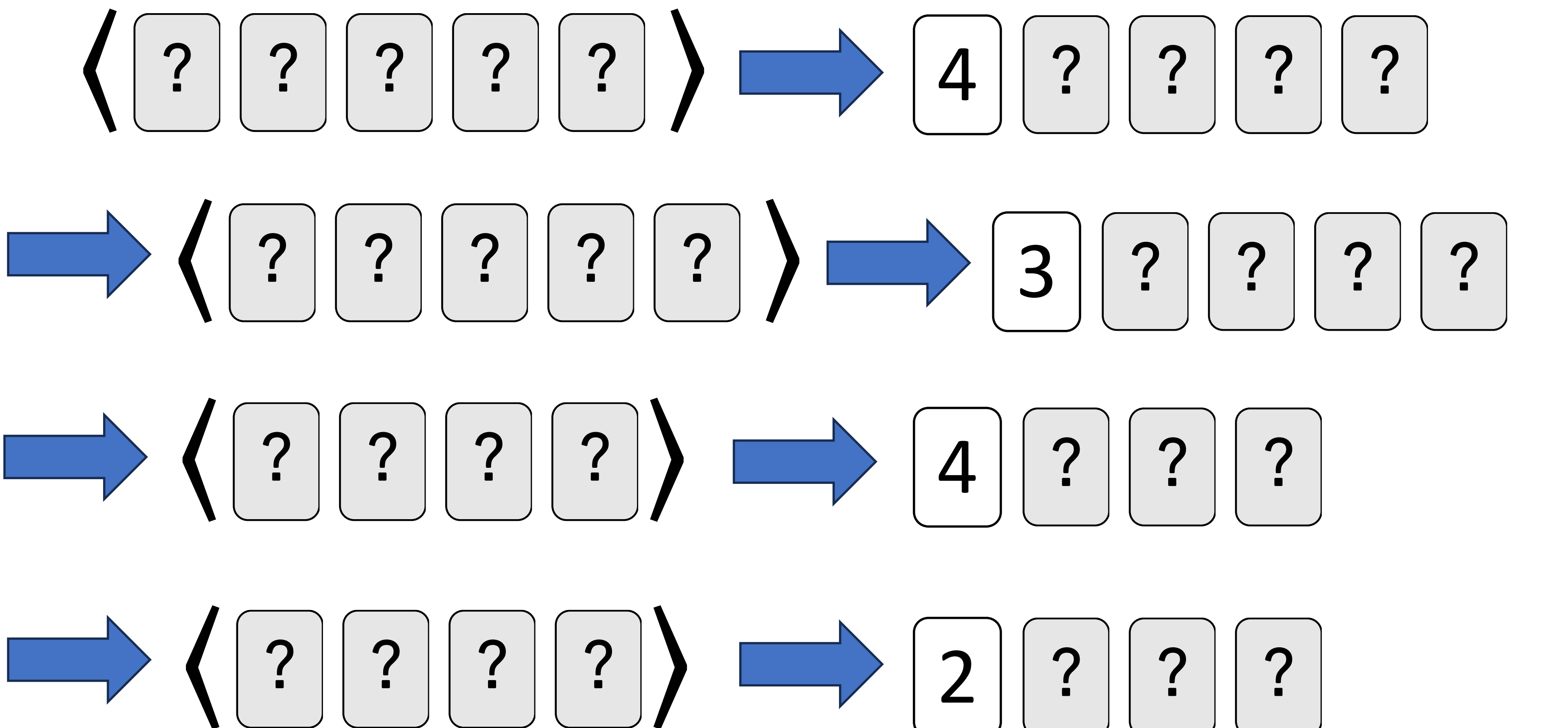
A ***random cut*** is a shuffle that cyclically shifts the sequence by a uniformly random offset.



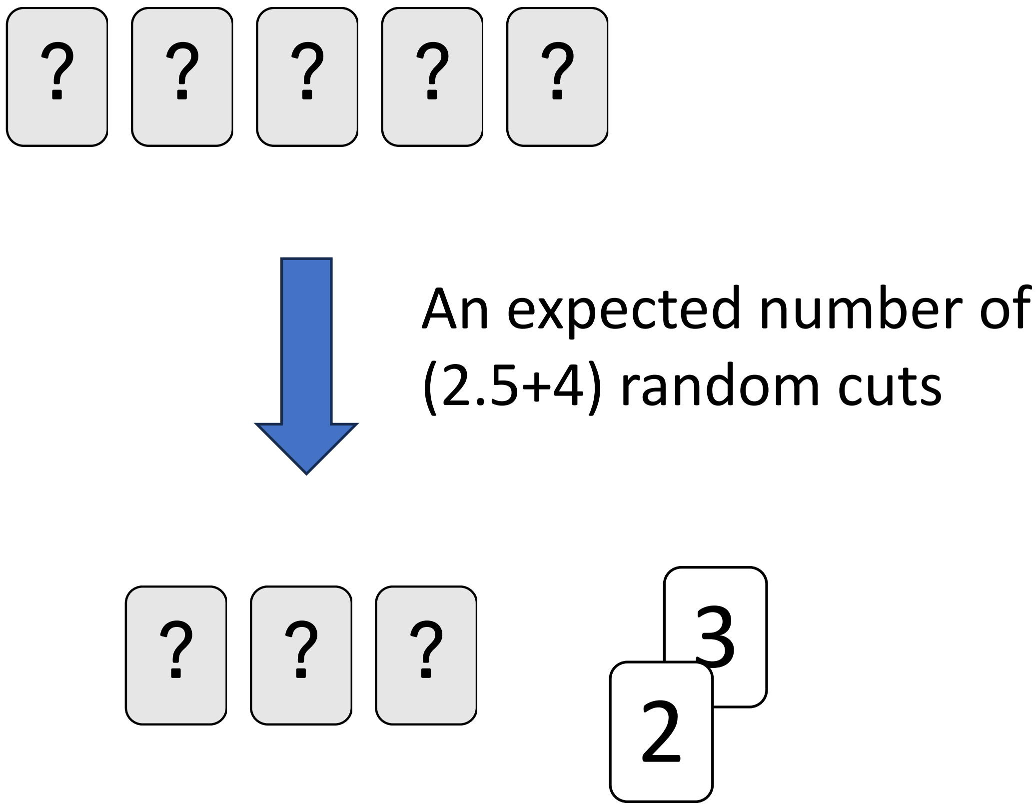
We use the following expression:



Reveal 1st card after applying a random cut until $\boxed{2}\,\boxed{3}$ are found



3. Remove 2 3 by random cuts



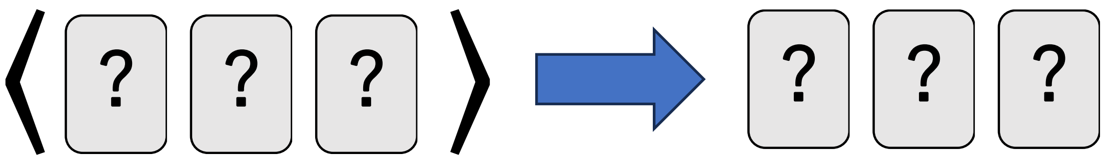
We can remove them without leaking any information.

1. Turn over 5^{th} card

2. Swap 2^{nd} and 3^{rd} cards

3. Remove $\boxed{2}\boxed{3}$ by random cuts

4. Apply a random cut



(a,b)	Sequence										
$(0,0)$	1	4	5	or	5	1	4	or	4	5	1
$(0,1)$	1	4	5	or	5	1	4	or	4	5	1 .
$(1,0)$	1	4	5	or	5	1	4	or	4	5	1
$(1,1)$	1	5	4	or	5	4	1	or	4	1	5

1. Turn over 5th card

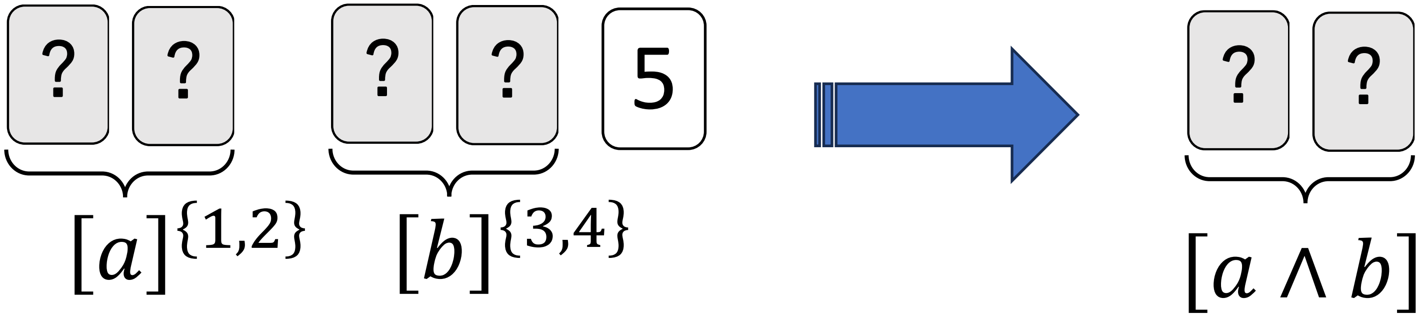
2. Swap 2nd and 3rd cards

3. Remove $\boxed{2}\boxed{3}$ by random cuts

4. Apply a random cut

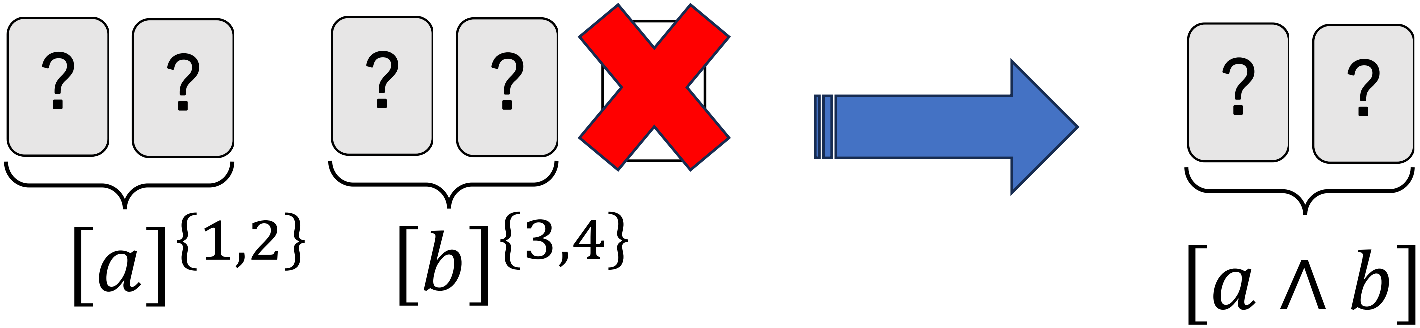
5. Reveal 1st card

$$\boxed{1}\ \underbrace{\boxed{?}\boxed{?}}_{[a\wedge b]^{\{4,5\}}} \quad \text{or} \quad \boxed{5}\ \underbrace{\boxed{?}\boxed{?}}_{[a\wedge b]^{\{1,4\}}} \quad \text{or} \quad \boxed{4}\ \underbrace{\boxed{?}\boxed{?}}_{[\overline{a\wedge b}]^{\{1,5\}}}$$



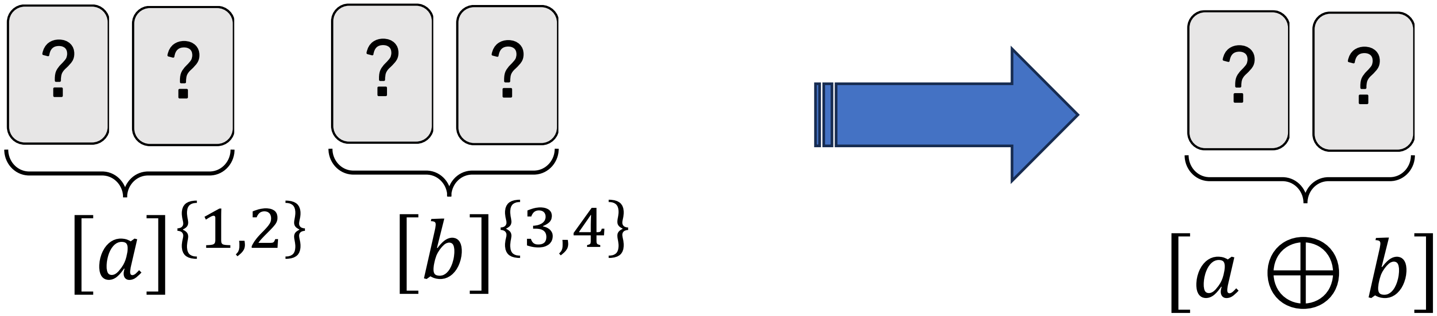
Function	# of cards	# of shuffles	Authors
AND	5	7.5 (exp)	Niemi-Renvall [NR99]

[NR99] Valtteri Niemi and Ari Renvall. Solitaire zero-knowledge. Fundam.Inf., 38(1,2):181–188, 1999



Function	# of cards	# of shuffles	Authors
AND	5	7.5 (exp)	Niemi-Renvall [NR99]
	4	6 (exp)	Koch et al. [KSK21]

[KSK21] Alexander Koch, Michael Schrempp, and Michael Kirsten. Card-based cryptography meets formal verification. New Gener. Com-put., 39(1):115–158, 2021.

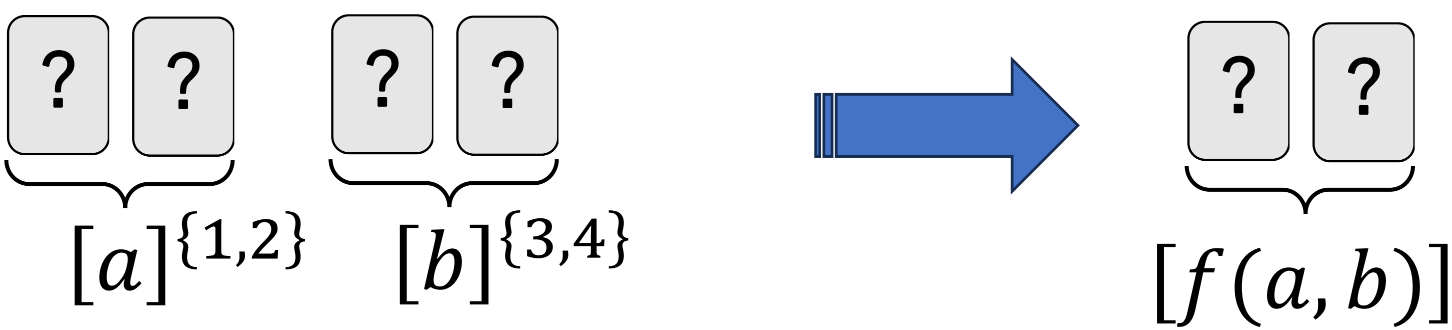


Function	# of cards	# of shuffles	Authors
AND	5	7.5 (exp)	Niemi-Renvall [NR99]
	4	6 (exp)	Koch et al. [KSK21]
XOR	4	7 (exp)	Niemi-Renvall [NR99]
	4	1	Mizuki [Miz16]

[Miz16] Takaaki Mizuki. Efficient and secure multiparty computations using a standard deck of playing cards. CANS 2026, vol.10052, LNCS, pp.484–499, 2016. Springer.

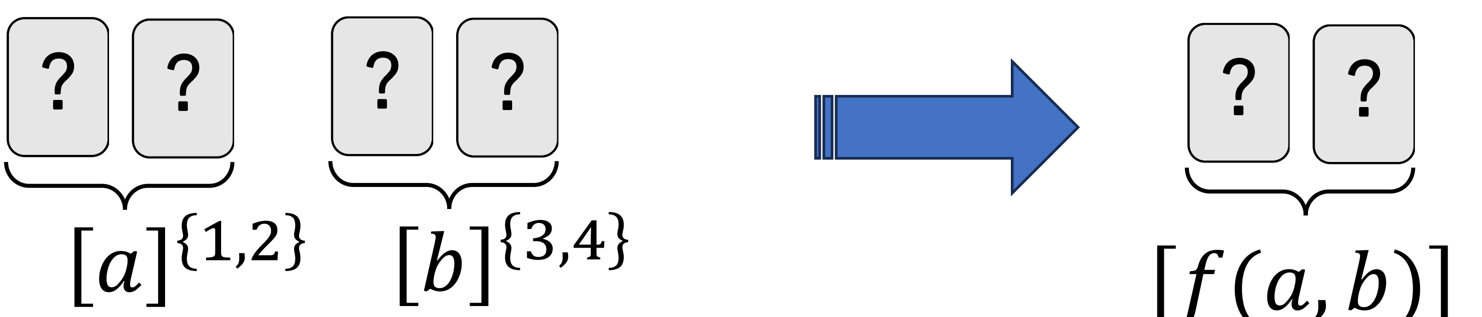
Function	# of cards	# of shuffles	Authors
AND	5	7.5 (exp)	Niemi-Renvall [NR99]
	4	6 (exp)	Koch et al. [KSK21]
XOR	4	7 (exp)	Niemi-Renvall [NR99]
	4	1	Mizuki [Miz16]

Card-minimal protocols for 2 inputs



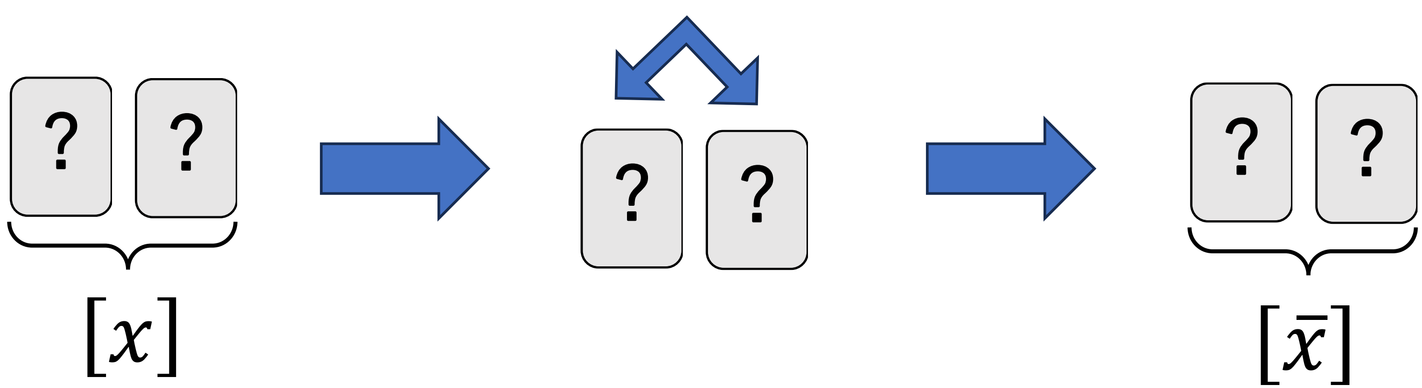
Function	# of cards	# of shuffles	Authors
AND	5	7.5 (exp)	Niemi-Renvall [NR99]
	4	6 (exp)	Koch et al. [KSK21]
XOR	4	7 (exp)	Niemi-Renvall [NR99]
	4	1	Mizuki [Miz16]

A 4-card protocol for any 2-input function $f: \{0,1\}^2 \to \{0,1\}$ can be constructed by AND, XOR, and NOT.



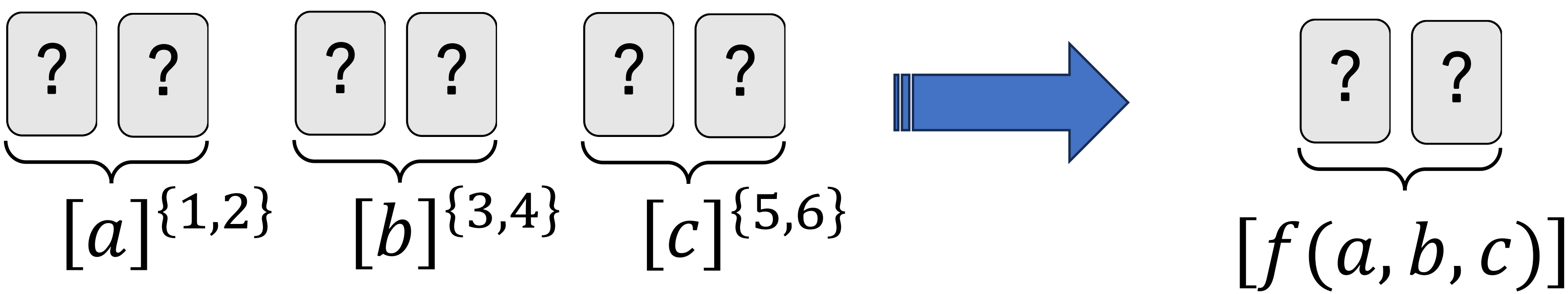
Function	# of cards	# of shuffles	Authors
AND	5	7.5 (exp)	Niemi-Renvall [NR99]
	4	6 (exp)	Koch et al. [KSK21]
XOR	4	7 (exp)	Niemi-Renvall [NR99]
	4	1	Mizuki [Miz16]

A 4-card protocol for any 2-input function $f: \{0,1\}^2 \rightarrow \{0,1\}$ can be constructed by AND, XOR, and **NOT**.

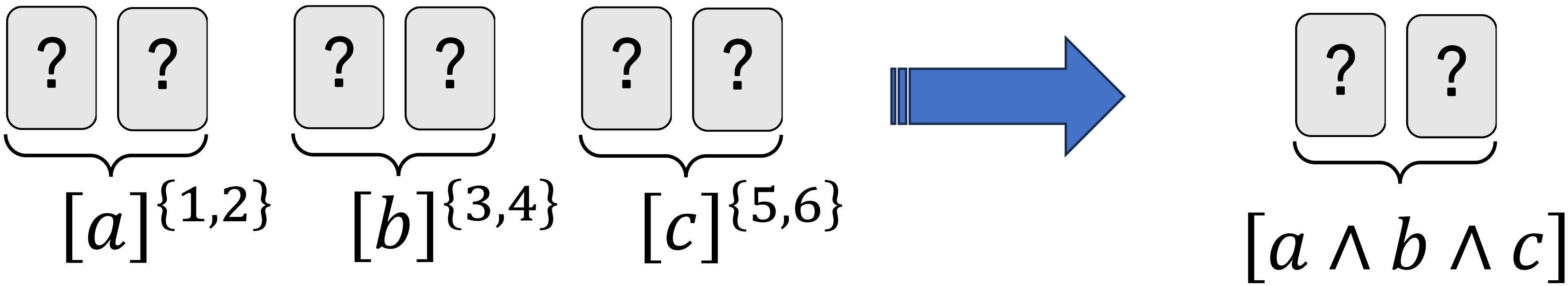


The problem of finding card-minimal protocols for **all 2-input** functions was already resolved.

How about **3-input** functions $f\colon \{0,1\}^3 \to \{0,1\}$?

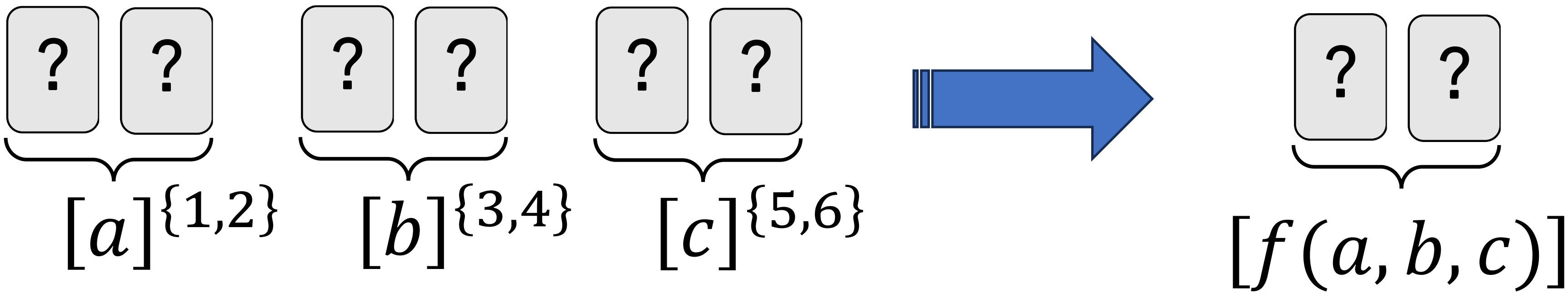


We want to construct a 6-card protocol (without any additional card except for 3 input commitments).



Function	# of cards	# of shuffles	Authors
3-AND	6	8.5 (exp)	Koyama et al. [KMMS21]

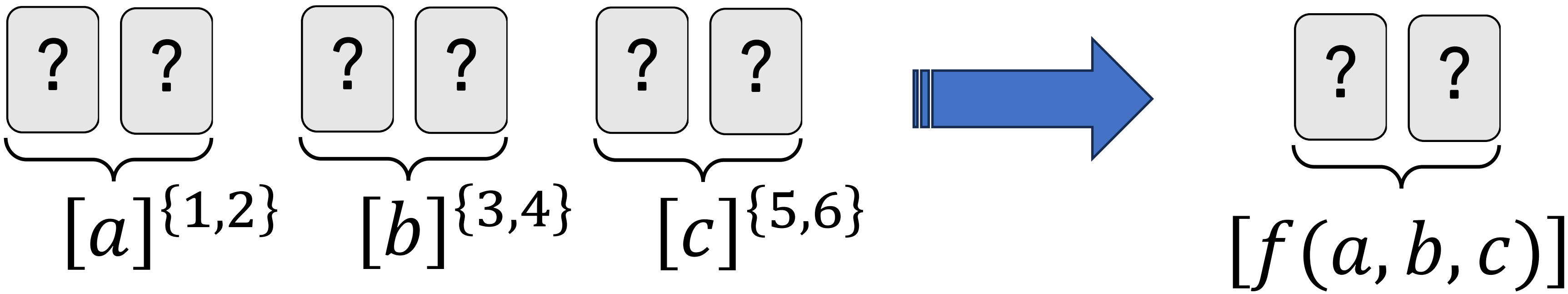
[KMMS21] Hiroto Koyama, Daiki Miyahara, Takaaki Mizuki, and Hideaki Sone. A secure three-input AND protocol with a standard deck of minimal cards. CSR 2021, vol.12730, LNCS, pp.242–256, 2021. Springer.



Function	# of cards	# of shuffles	Authors
3-AND	6	8.5 (exp)	Koyama et al. [KMMS21]
140 functions	6	8.5 (exp)	Haga et al. [HHMM22]

out of 256

[HHMM22] Rikuo Haga, Yuichi Hayashi, Daiki Miyahara, and Takaaki Mizuki. Card-minimal protocols for three-input functions with standard playing cards. AFRICACRYPT 2022, vol.13503, LNCS, pp.448–468, 2022. Springer



Function	# of cards	# of shuffles	Authors
3-AND	**6**	**8.5 (exp)**	**Koyama et al. [KMMS21]**
140 functions	**6**	**8.5 (exp)**	**Haga et al. [HHMM22]**

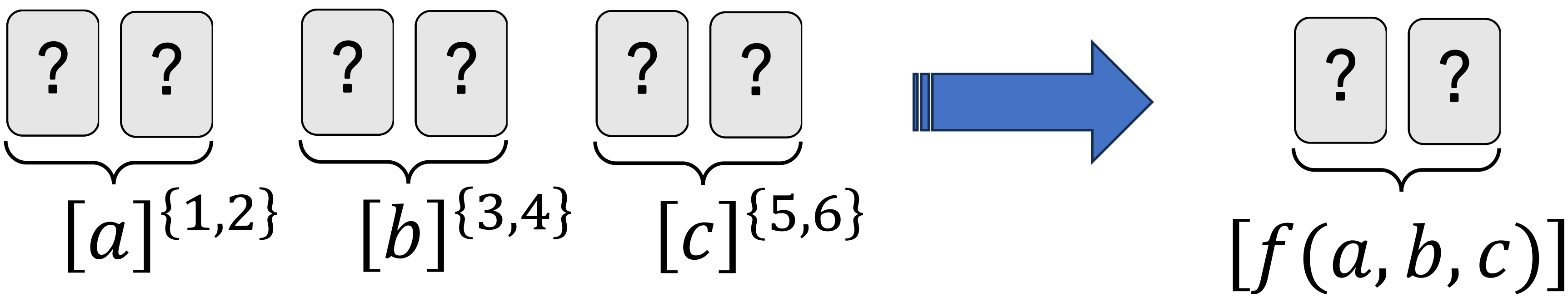
Open Problem:

Can 6-card protocols be constructed for the remaining **116** 3-input functions?

(Note that there are 256 3-input functions)

Our contribution

We show that, for the remaining **116** 3-input functions, 6-card protocols can be constructed.



Any function
$f:\{0,1\}^3 \rightarrow \{0,1\}$

Table of Contents

Idea

$\underbrace{\boxed{?}\ \boxed{?}}_{[a]^{\{1,2\}}}\ \underbrace{\boxed{?}\ \boxed{?}}_{[b]^{\{3,4\}}}$ $\xrightarrow{\text{Swap 2}^{\text{nd}}\text{ and 3}^{\text{rd}}\text{ cards}}$ $\boxed{?}\ \boxed{?}\ \boxed{?}\ \boxed{?}$

(a,b)				
(0,0)	1	2	3	4
(0,1)	1	2	4	3
(1,0)	2	1	3	4
(1,1)	2	1	4	3

$\boxed{1}\ \boxed{4} = 0$

$\boxed{1}\ \boxed{4} = 0$

$\boxed{1}\ \boxed{4} = 0$

$\boxed{1}\ \boxed{4} = 0$

Computes 0 (constant)

(a,b)				
(0,0)	1	3	2	4
(0,1)	1	4	2	3
(1,0)	2	3	1	4
(1,1)	2	4	1	3

$\boxed{1}\ \boxed{4} = 0$

$\boxed{1}\ \boxed{4} = 0$

$\boxed{1}\ \boxed{4} = 0$

$\boxed{4}\ \boxed{1} = 1$

Computes $a \wedge b$

Idea

$\underbrace{?\ ?}_{[a]^{\{1,2\}}}\ \underbrace{?\ ?}_{[b]^{\{3,4\}}}$

if $c = 1$

Swap 2nd and 3rd cards

Computes $a \wedge b$

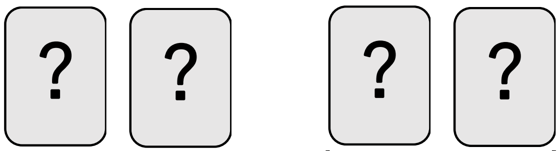
$\underbrace{?\ ?}_{[c]^{\{5,6\}}}$

if $c = 0$

Do nothing

Computes 0 (constant)

Idea

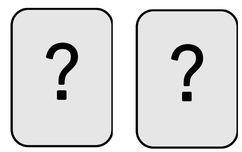


$[a]^{\{1,2\}}$ $[b]^{\{3,4\}}$

if $c = 1$

Swap 2nd and 3rd cards

Computes $a \wedge b$



$[c]^{\{5,6\}}$

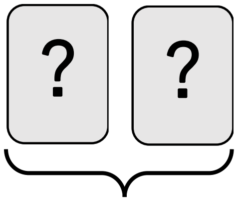
if $c = 0$

Do nothing

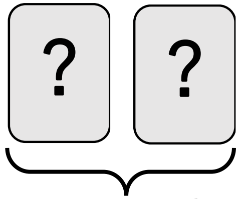
Computes 0 (constant)

$$\begin{cases} 0 & \text{if } c = 0 \\ a \wedge b & \text{if } c = 1 \end{cases}$$

Idea



$[a]^{\{1,2\}}$

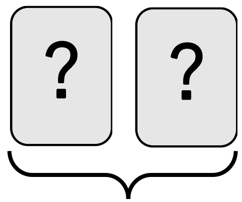


$[b]^{\{3,4\}}$

if $c = 1$

Swap 2nd and 3rd cards

Computes $a \wedge b$



$[c]^{\{5,6\}}$

if $c = 0$

Do nothing

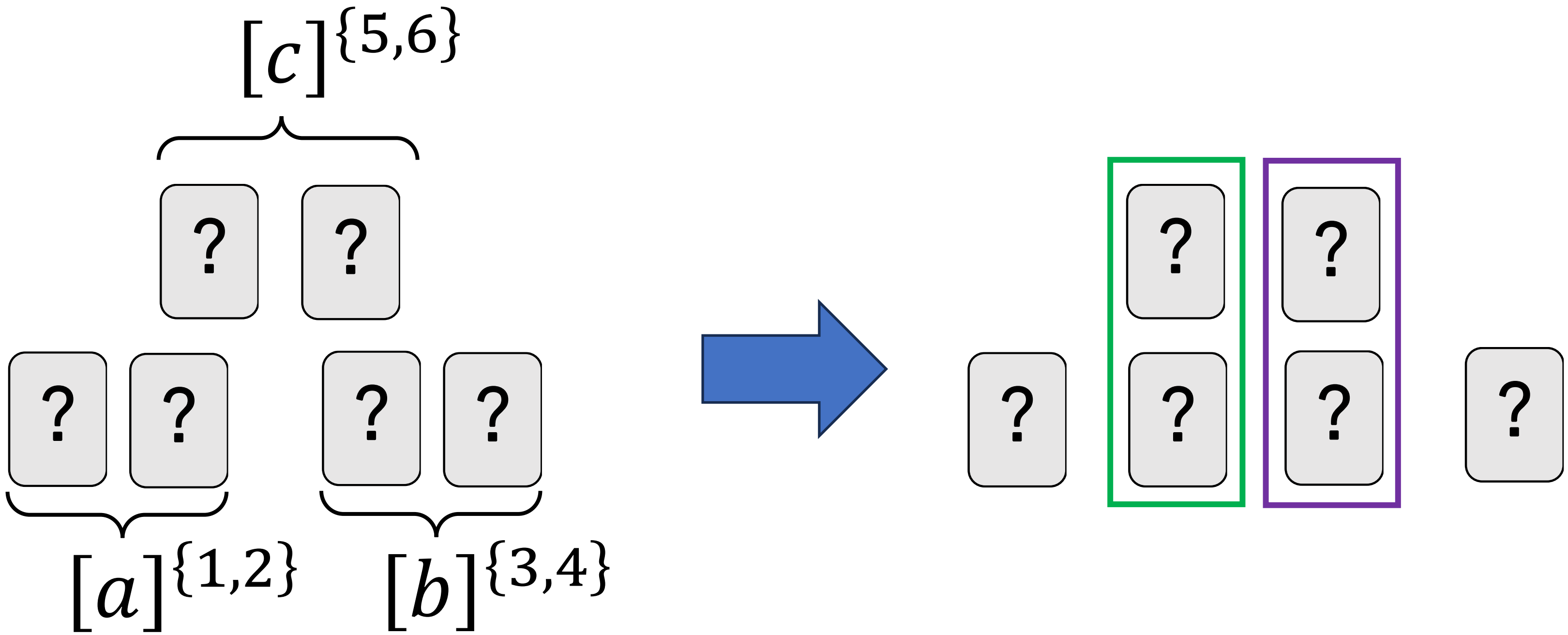
Computes 0 (constant)

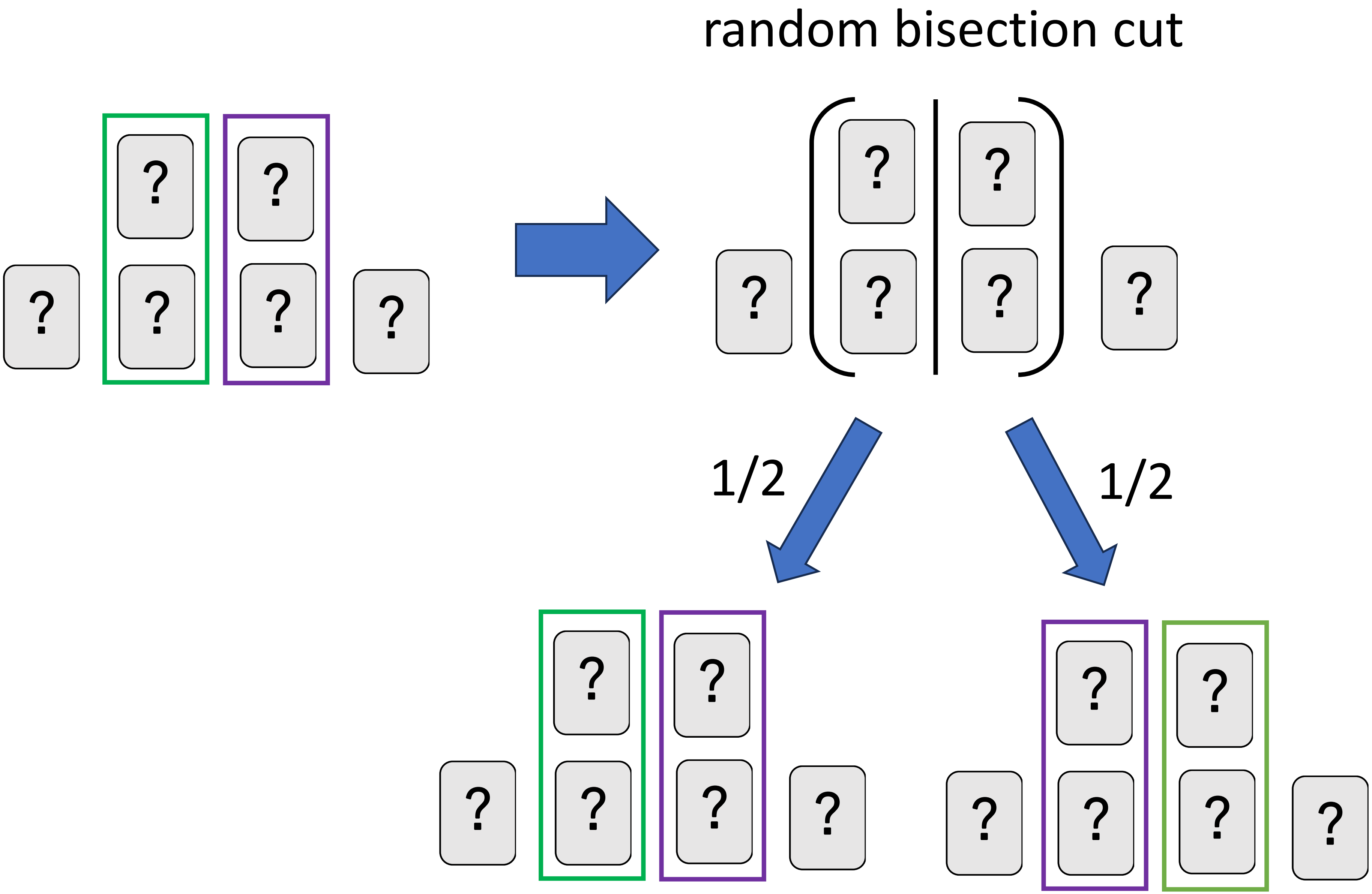
$$\begin{cases} 0 & \text{if } c = 0 \\ a \wedge b & \text{if } c = 1 \end{cases}$$

$$= \boldsymbol{a \wedge b \wedge c}$$

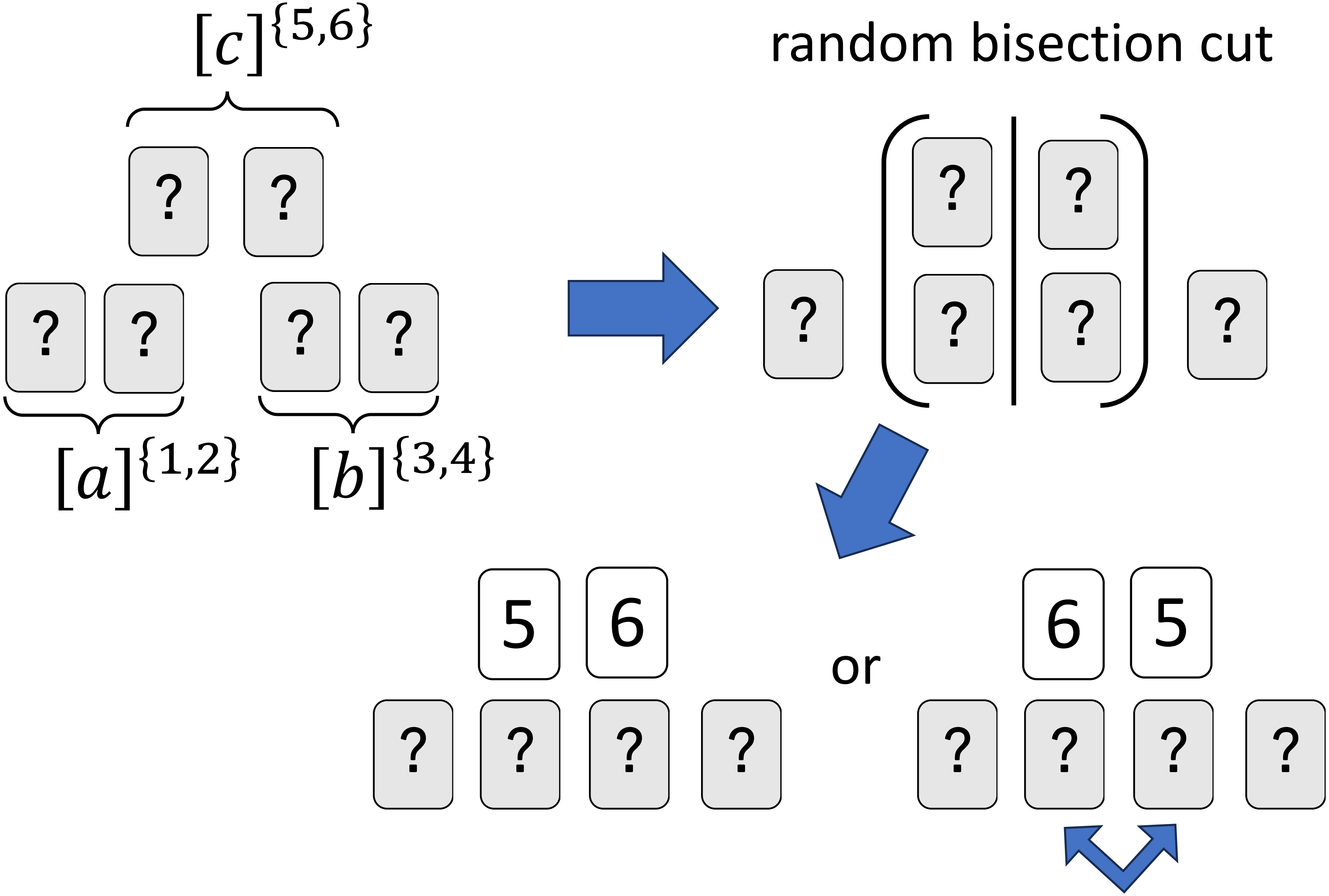
How to swap 2nd and 3rd only if $c = 1$

(without leaking any information)




random bisection cut
?
1/2
1/2

How to swap 2nd and 3rd only if $c = 1$



Outline of Haga et al.'s protocol

$$\underbrace{\boxed{?}\,\boxed{?}}_{[a]^{\{1,2\}}} \quad \underbrace{\boxed{?}\,\boxed{?}}_{[b]^{\{3,4\}}} \quad \underbrace{\boxed{?}\,\boxed{?}}_{[c]^{\{5,6\}}}$$

1. Swap some cards depending on c

$$\underbrace{\boxed{?}\,\boxed{?}}_{[c]}$$

$$\underline{\boxed{?}\,\boxed{?}\,\boxed{?}\,\boxed{?}}$$

2. Remove $\boxed{2}\,\boxed{3}$ by random cuts $\left\langle \boxed{?}\,\boxed{?}\,\boxed{?}\,\boxed{?}\,\boxed{5} \right\rangle$

3. Apply a random cut

4. Reveal 1st card

$$\underbrace{\boxed{?}\,\boxed{?}}_{[f(a,b,c)]}$$

Haga et al.'s method computes those in yellow.

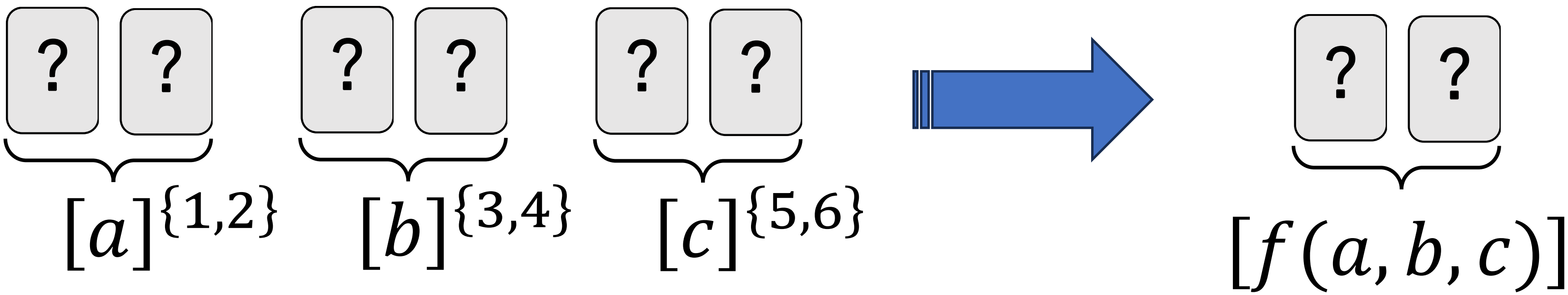
$$f(a,b,c) = \begin{cases} g(a,b) & \text{if } c = 0 \\ h(a,b) & \text{if } c = 1 \end{cases}$$

140 out of 256 functions

$h \backslash g$	0	$a \wedge b$	$a \wedge \overline{b}$	a	$\overline{a} \wedge b$	b	$a \oplus b$	$a \vee b$	$\overline{a \vee b}$	$\overline{a \oplus b}$	$\overline{b}$	$a \vee \overline{b}$	$\overline{a}$	$\overline{a} \vee b$	$\overline{a \wedge b}$	1
0	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓	✓	✓	✓	✓
$a \wedge b$	✓		✓	✓	✓	✓		✓	✓						✓	✓
$a \wedge \overline{b}$	✓	✓		✓	✓				✓		✓	✓		✓		✓
a	✓	✓	✓	✓		✓		✓			✓	✓	✓			✓
$\overline{a} \wedge b$	✓	✓	✓			✓			✓			✓	✓	✓		✓
b	✓	✓		✓	✓	✓		✓			✓		✓	✓		✓
$a \oplus b$																
$a \vee b$	✓	✓		✓		✓			✓			✓		✓	✓	✓
$\overline{a \vee b}$	✓	✓	✓		✓			✓			✓		✓		✓	✓
$\overline{a \oplus b}$																
$\overline{b}$	✓		✓	✓		✓			✓		✓	✓	✓		✓	✓
$a \vee \overline{b}$	✓		✓	✓	✓			✓			✓			✓	✓	✓
$\overline{a}$	✓			✓	✓	✓			✓		✓		✓	✓	✓	✓
$\overline{a} \vee b$	✓		✓		✓	✓		✓				✓	✓		✓	✓
$\overline{a \wedge b}$	✓	✓						✓	✓		✓	✓	✓	✓		✓
1	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓	✓	✓	✓	✓

Table of Contents

Assume that



Negation of inputs: $f(\overline{a},b,c), f(\overline{a},b,\overline{c})$, ...

Permutation of input order: $f(b,c,a), f(c,b,a)$, ...

Negation of output: $\overline{f}(a,b,c)$

All these functions can also be securely computed (because NOT and permutation of inputs are trivial).

Negation of inputs: $f(\overline{a}, b, c), f(\overline{a}, b, \overline{c})$, ...
Permutation of input order: $f(b, c, a), f(c, b, a)$, ...
Negation of output: $\overline{f}(a, b, c)$

We consider ***NPN equivalence classes***.

It suffices to deal with a representative function in each class.

It is well known that there are 14 NPN equivalence classes for 3-input functions.

It is well known that there are 14 NPN equivalence classes for 3-input functions.

Class	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14
Rep.	0	And3	$a \wedge b$	XorAnd	OrAnd	a	Onehot	Majority	Equality	Dot	Mux	AndXor	$a \oplus b$	Xor3

$$\begin{aligned}
\mathsf{And3}(a,b,c) &= a \wedge b \wedge c \\
\mathsf{XorAnd}(a,b,c) &= a \wedge (b \oplus c) \\
\mathsf{OrAnd}(a,b,c) &= a \wedge (b \vee c) \\
\mathsf{Onehot}(a,b,c) &= (a \wedge \overline{b} \wedge \overline{c}) \oplus (\overline{a} \wedge b \wedge \overline{c}) \oplus (\overline{a} \wedge \overline{b} \wedge c) \\
\mathsf{Majority}(a,b,c) &= (a \wedge b) \vee (b \wedge c) \vee (a \wedge c) \\
\mathsf{Equality}(a,b,c) &= (a \wedge b \wedge c) \oplus (\overline{a} \wedge \overline{b} \wedge \overline{c}) \\
\mathsf{Dot}(a,b,c) &= a \oplus (c \vee (a \wedge b)) \\
\mathsf{Mux}(a,b,c) &= (a \wedge b) \vee (\overline{a} \wedge c) \\
\mathsf{AndXor}(a,b,c) &= a \oplus (b \wedge c) \\
\mathsf{Xor3}(a,b,c) &= a \oplus b \oplus c\,.
\end{aligned}$$

$h \backslash g$	0	$a \wedge b$	$a \wedge \overline{b}$	a	$\overline{a} \wedge b$	b	$a \oplus b$	$a \vee b$	$\overline{a \vee b}$	$\overline{a \oplus b}$	$\overline{b}$	$a \vee \overline{b}$	$\overline{a}$	$\overline{a} \vee b$	$\overline{a \wedge b}$	1
0	#1	#2	#2	#3	#2	#3	#4	#5	#2	#4	#3	#5	#3	#5	#5	#6
$a \wedge b$	#2	#3	#4	#5	#4	#5	#10	#8	#9	#7	#10	#11	#10	#11	#12	#5
$a \wedge \overline{b}$	#2	#4	#3	#5	#9	#10	#7	#11	#4	#10	#5	#8	#10	#12	#11	#5
a	#3	#5	#5	#6	#10	#11	#12	#5	#10	#12	#11	#5	#13	#10	#10	#3
$\overline{a} \wedge b$	#2	#4	#9	#10	#3	#5	#7	#11	#4	#10	#10	#12	#5	#8	#11	#5
b	#3	#5	#10	#11	#5	#6	#12	#5	#10	#12	#13	#10	#11	#5	#10	#3
$a \oplus b$	#4	#10	#7	#12	#7	#12	#13	#7	#10	#14	#12	#10	#12	#10	#7	#4
$a \vee b$	#5	#8	#11	#5	#11	#5	#7	#3	#12	#10	#10	#4	#10	#4	#9	#2
$\overline{a \vee b}$	#2	#9	#4	#10	#4	#10	#10	#12	#3	#7	#5	#11	#5	#11	#8	#5
$\overline{a \oplus b}$	#4	#7	#10	#12	#10	#12	#14	#10	#7	#13	#12	#7	#12	#7	#10	#4
$\overline{b}$	#3	#10	#5	#11	#10	#13	#12	#10	#5	#12	#6	#5	#11	#10	#5	#3
$a \vee \overline{b}$	#5	#11	#8	#5	#12	#10	#10	#4	#11	#7	#5	#3	#10	#9	#4	#2
$\overline{a}$	#3	#10	#10	#13	#5	#11	#12	#10	#5	#12	#11	#10	#6	#5	#5	#3
$\overline{a} \vee b$	#5	#11	#12	#10	#8	#5	#10	#4	#11	#7	#10	#9	#5	#3	#4	#2
$\overline{a \wedge b}$	#5	#12	#11	#10	#11	#10	#7	#9	#8	#10	#5	#4	#5	#4	#3	#2
1	#6	#5	#5	#3	#5	#3	#4	#2	#5	#4	#3	#2	#3	#2	#2	#1

#1～#6, #8, #9, #11～#13 are covered.

#7, #10, and #14 are uncovered.

#7, #10, and #14 are uncovered.

#7:

Onehot $= (\boldsymbol{a} \wedge \overline{\boldsymbol{b}} \wedge \overline{\boldsymbol{c}}) \oplus (\overline{\boldsymbol{a}} \wedge \boldsymbol{b} \wedge \overline{\boldsymbol{c}}) \oplus (\overline{\boldsymbol{a}} \wedge \overline{\boldsymbol{b}} \wedge \boldsymbol{c})$

#10:

Dot $= \boldsymbol{a} \oplus (\boldsymbol{c} \vee (\boldsymbol{a} \wedge \boldsymbol{b}))$

#14:

Xor3 $= \boldsymbol{a} \oplus \boldsymbol{b} \oplus \boldsymbol{c}$

2-input XOR twice

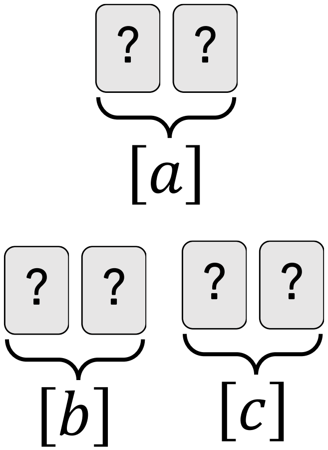
The truly unresolved classes are #7 (Onehot) and #10 (Dot).
We will construct 6-card protocols for these two functions.

Table of Contents

$$\mathrm{Onehot}(a,b,c) = \begin{cases} b \oplus c & \text{if } a = 0 \\ \overline{b \vee c} & \text{if } a = 1 \end{cases}$$

$$\underbrace{\boxed{?}\ \boxed{?}}_{[a]^{\{1,2\}}} \quad \underbrace{\boxed{?}\ \boxed{?}}_{[b]^{\{3,4\}}} \quad \underbrace{\boxed{?}\ \boxed{?}}_{[c]^{\{5,6\}}}$$

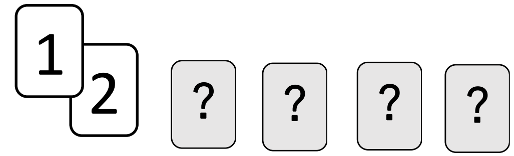
(b,c)				
(0,0)	3	4	5	6
(0,1)	3	4	6	5
(1,0)	4	3	5	6
(1,1)	4	3	6	5



Swap 1st and 4th cards only if $a = 0$:

(b,c)				
(0,0)	3	4	5	6
(0,1)	3	4	6	5
(1,0)	4	3	5	6
(1,1)	4	3	6	5

$a = 0$



(b,c)				
(0,0)	6	4	5	3
(0,1)	5	4	6	3
(1,0)	6	3	5	4
(1,1)	5	3	6	4

$a = 1$

(b,c)				
(0,0)	3	4	5	6
(0,1)	3	4	6	5
(1,0)	4	3	5	6
(1,1)	4	3	6	5

$a = 0$

$(\boldsymbol{b}, \boldsymbol{c})$				
(0,0)	6	4	5	3
(0,1)	5	4	6	3
(1,0)	6	3	5	4
(1,1)	5	3	6	4

$a = 1$

$(\boldsymbol{b}, \boldsymbol{c})$				
(0,0)	3	4	5	6
(0,1)	3	4	6	5
(1,0)	4	3	5	6
(1,1)	4	3	6	5

Find $\boxed{6}$ (or $\boxed{3}$):

$\left\langle \boxed{?}\ \boxed{?}\ \boxed{?}\ \boxed{?} \right\rangle$

$\boxed{1}\ \boxed{2}$

$\boxed{6}\ \boxed{?}\ \boxed{?}\ \boxed{?}$

$(\boldsymbol{b}, \boldsymbol{c})$				
(0,0)	6	4	5	3
(0,1)	6	3	5	4
(1,0)	6	3	5	4
(1,1)	6	4	5	3

$(\boldsymbol{b}, \boldsymbol{c})$				
(0,0)	6	3	4	5
(0,1)	6	5	3	4
(1,0)	6	4	3	5
(1,1)	6	5	4	3

1
2

6 ? ? ?

$a = 0$

(b, c)				
(0,0)	6	4	5	3
(0,1)	6	3	5	4
(1,0)	6	3	5	4
(1,1)	6	4	5	3

$a = 1$

(b, c)				
(0,0)	6	3	4	5
(0,1)	6	5	3	4
(1,0)	6	4	3	5
(1,1)	6	5	4	3

Move $\boxed{1}$:

2

6 ? 1 ? ?

(b, c)					
(0,0)	6	4	1	5	3
(0,1)	6	3	1	5	4
(1,0)	6	3	1	5	4
(1,1)	6	4	1	5	3

(b, c)					
(0,0)	6	3	1	4	5
(0,1)	6	5	1	3	4
(1,0)	6	4	1	3	5
(1,1)	6	5	1	4	3

Sentinel

$\boxed{2}$

$\boxed{6}$ $\boxed{?}$ $\boxed{1}$ $\boxed{?}$ $\boxed{?}$

$a = 0$

(b, c)					
(0,0)	6	4	1	5	3
(0,1)	6	3	1	5	4
(1,0)	6	3	1	5	4
(1,1)	6	4	1	5	3

$\boxed{1}\boxed{3} = 0$

$\boxed{3}\boxed{1} = 1$

$\boxed{3}\boxed{1} = 1$

$\boxed{1}\boxed{3} = 0$

$b \oplus c$

$a = 1$

(b, c)					
(0,0)	6	3	1	4	5
(0,1)	6	5	1	3	4
(1,0)	6	4	1	3	5
(1,1)	6	5	1	4	3

$\boxed{3}\boxed{1} = 1$

$\boxed{1}\boxed{3} = 0$

$\boxed{1}\boxed{3} = 0$

$\boxed{1}\boxed{3} = 0$

$\overline{b \vee c}$

Removing $\boxed{4}$ $\boxed{5}$ results in $\mathrm{Onehot}(a, b, c) = \begin{cases} b \oplus c & \text{if } a = 0 \\ \overline{b \vee c} & \text{if } a = 1 \end{cases}$

Our protocol for Onehot

$\underbrace{\boxed{?}\boxed{?}}_{[a]}$

$\underbrace{\boxed{?}\boxed{?}}_{[b]}$ $\underbrace{\boxed{?}\boxed{?}}_{[c]}$

1. Swap 1st and 4th cards if $a = 0$
2. Find $\boxed{6}$ (or $\boxed{3}$) $\boxed{6}\ \boxed{?}\ \boxed{?}\ \boxed{?}$
3. Insert $\boxed{1}$ $\boxed{6}\ \boxed{?}\ \boxed{1}\ \boxed{?}\ \boxed{?}$
4. Remove $\boxed{4}\boxed{5}$ $\boxed{?}\ \boxed{?}\ \boxed{?}$
5. Apply a random cut
6. Reveal 1st card

$\underbrace{\boxed{?}\boxed{?}}_{[\mathrm{Onehot}(a,b,c)]}$

Similarly, we can construct a protocol for **Dot** $= \boldsymbol{a} \oplus \left(\boldsymbol{c} \vee (\boldsymbol{a} \wedge \boldsymbol{b})\right)$

Table of Contents

We proposed 6-card protocols for Onehot and Dot, resolving the card-minimality problem for 3-input functions.

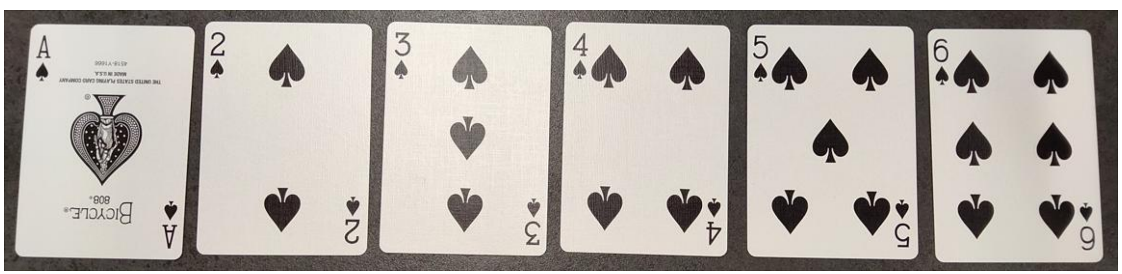
class	#1	#2	#3	#4	#5	#6	#7
Rep.	0	And3	$a \wedge b$	XorAnd	OrAnd	a	Onehot
Haga et al.	8.5	8.5	8.5	8.5	8.5	8.5	-
Trivial/ Known	0	-	6	7	-	0	-
This paper	-	-	-	-	-	-	10.5

class	#8	#9	#10	#11	#12	#13	#14
Rep.	Majority	Equality	Dot	Mux	AndXor	$a \oplus b$	Xor3
Haga et al.	8.5	8.5	-	8.5	8.5	8.5	-
Trivial/ Known	-	-	-	-	7	1	2
This paper	-	-	10.5	-	-	-	-

Future work:

- Reduce the number of shuffles
- Card-minimal problem for more than 3 inputs
  - As an upper bound, $2n + 8$ cards have been shown to be sufficient for any n-input function [SM19]

Enjoy secure computations with six cards!



[SM19] Kazumasa Shinagawa and Takaaki Mizuki. Secure computation ofany Boolean function based on any deck of cards. FAW 2019, vol.11458, LNCS, pp.63–75, 2019. Springer.