Vol.2021-DPS-186 No.6
Vol.2021-CSEC-92 No.6
2021/3/15

6枚のカードを用いた非コミット型 3入力ANDプロトコル

佐々木 優[1,a)]　宮原 大輝[1,2]　水木 敬明[1,2]　曽根 秀昭[1]

概要：カードベース暗号は，物理的なカード組を用いて秘密計算等を実現する手法であり，通常は黒と赤の2枚のカードを用いてビット値を表現する．これまで，2色カードデッキを用いて2入力のAND関数を秘密計算する手法が数多く提案されてきた．3入力については，2016年にMizukiが提案したプロトコルは6枚のカードを用いて5回のシャッフルを行うことでAND秘密計算を実現している．また，既存の2入力ANDプロトコルを組み合わせて3入力ANDを実現することも可能であるが，一様で閉じているシャッフルを用いるプロトコルの組み合わせでは，7枚のカードを要する．そこで，本稿では6枚のカードを用いた3入力ANDプロトコルを新たに提案する．これは，シャッフル回数が2回（ランダムカットとランダム二等分割カットそれぞれ1回）と少なく，最小のカード枚数で実行できるプロトコルである．

Non-Committed Three-Input AND Protocol Using Six Cards

1. はじめに

物理的なカード組を用いて秘密計算を行う手法を**カードベース暗号**という．カードベース暗号では，裏面が ? のように区別がつかず，表面は ♣ または ♡ のいずれかである2種類のカードを用いる．これらのカードを用いて次のようにブール値を表す．

$$\boxed{\clubsuit}\boxed{\heartsuit} = 0,\ \boxed{\heartsuit}\boxed{\clubsuit} = 1$$

このルールに従って裏返しに置かれたカード2枚を**コミットメント**と呼ぶ．

2入力ANDプロトコルは，2つのコミットメント（$x, y \in \{0,1\}$ のコミットメント）を入力とし，シャッフルやカードをめくる操作などによって $x \wedge y$ の秘密計算を行う．頻繁に使われるシャッフルは**ランダムカット**（RC）（2.2節）と**ランダム二等分割カット**（RBC）（2.3節）である．出力が $x \wedge y$ のコミットメントであるとき，そのようなプロトコルを**コミット型プロトコル**という．

$$\underbrace{\boxed{?}\boxed{?}}_{x}\ \underbrace{\boxed{?}\boxed{?}}_{y} \rightarrow \cdots \rightarrow \underbrace{\boxed{?}\boxed{?}}_{x \wedge y}$$

一方で，出力がコミットメントでないとき，そのようなプロトコルを**非コミット型プロトコル**という．（例えば，ある決められた位置のカードをめくり，その模様によって $x \wedge y = 0$ であるか $x \wedge y = 1$ であるかが分かるようなプロトコルである．）

$$\underbrace{\boxed{?}\boxed{?}}_{x}\ \underbrace{\boxed{?}\boxed{?}}_{y} \rightarrow \cdots \rightarrow x \wedge y$$

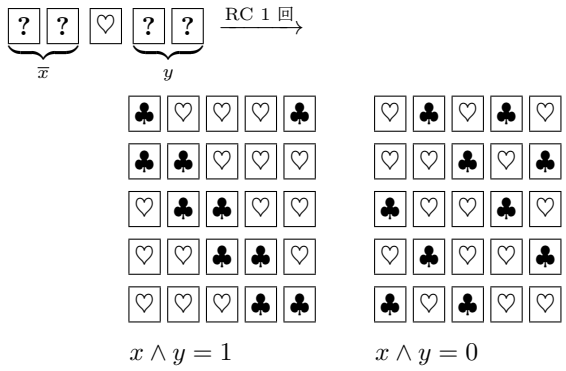
1.1 既存研究：2入力AND

2色カードデッキを用いた2入力ANDプロトコルの既存研究の一部を表1に示す．

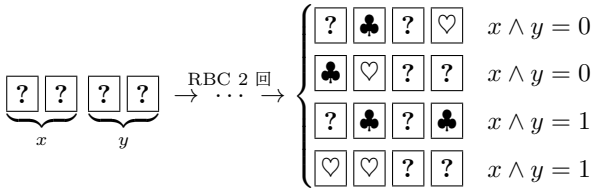
1989年にDen Boerは，Five-card trick [1] と呼ばれる非コミット型の2入力ANDプロトコルを提案した．このプロトコルは，1枚の追加カードを使用し，合計5枚のカードで論理積の秘密計算を行う．もう少し具体的には，x と y のコミットメントと追加カード ♡ を並べ，ランダムカットのシャッフルを適用する．最後にすべてのカードをめくり，3枚の ♡ が連続して並んだとき出力は1を示し，それ以外の並びが得られたとき出力は0を示す．

[1] 東北大学
Tohoku University, Sendai, Miyagi 980–8579, Japan
[2] 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology
[a)] yu.sasaki.r6@dc.tohoku.ac.jp

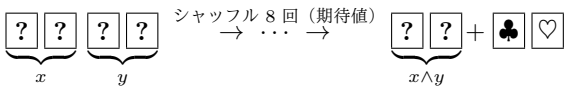
Vol.2021-DPS-186 No.6
Vol.2021-CSEC-92 No.6
2021/3/15



2012 年に Kumamoto ら [2] は Five-card trick からカード 1 枚を減らすことに成功し，4 枚の非コミット型 2 入力 AND プロトコルを考案した．このプロトコルは 2 回のランダム二等分割カットのシャッフルを用いる．

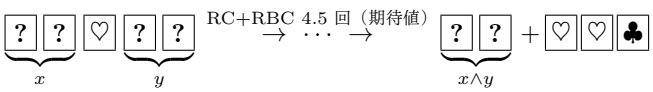


コミット型 AND プロトコルとして，2015 年に Koch ら [3] は 4 枚のカードを使用するプロトコルを提案した．Koch らのプロトコルは，出力のコミットメントを得るときに，コミットメントとは別に模様が分かっているカードが 2 枚得られる．



このプロトコルは一様でないシャッフル（定義は後述）を用いるため，人の手による実装が複雑になる．

Abe ら [4] は，5 枚のカードを用いて，ランダムカットとランダム二等分割カット（これらは一様で閉じたシャッフルである）のみを用いるプロトコルを提案した．さらに，Abe らはこのプロトコルを改良し，シャッフル回数を期待値 7 回から期待値 4.5 回まで削減した [5]．Abe らのプロトコルは，出力のコミットメントを得るときに，コミットメントとは別に模様が分かっているカードが 3 枚得られる．

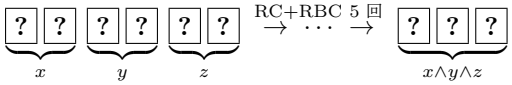


1.2 既存研究：3 入力 AND

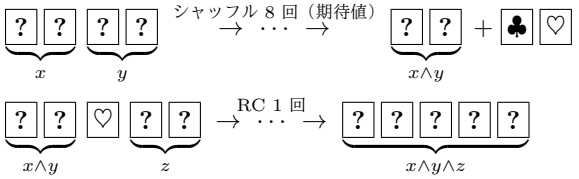
続いて，本稿の主テーマである，3 入力 AND プロトコルを表 2 に示す．

2016 年に Mizuki は 3 入力の論理積を秘密計算することのできるプロトコルを 2 つ提案した [6]．1 つは非コミット型 n 入力 AND プロトコルである．このプロトコルは $2n+1$ 枚のカードを使用し，シャッフルを $n-1$ 回行うことで n 個の論理積を秘密計算する．3 入力の論理積を秘密計算する場合は 7 枚のカード使用し，ランダム二等分割カットを 2 回行う．

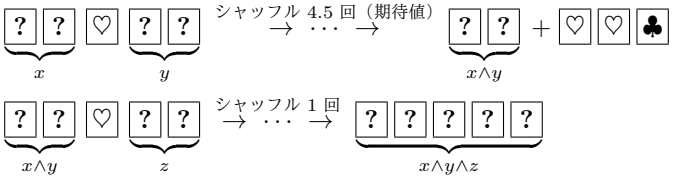
Mizuki が提案したもう 1 つのプロトコルは，6 枚のカードを使用する非コミット型 3 入力 AND プロトコルである．このプロトコルは 5 回のシャッフル（2 回のランダムカットと 3 回のランダム二等分割カット）を行うことで実現される．



また，3 入力の論理積の秘密計算は 2 入力 AND プロトコルの組み合わせによって行うこともできる．表 2 の Koch+Boer は Koch らのプロトコルと Boer のプロトコル（Five-card trick）を 1 回ずつ実行することで，3 入力の論理積を秘密計算する．すなわち，最初に Koch らのプロトコルによって 2 つの論理積を秘密計算し，コミットメントを得た後，そのコミットメントともう 1 つの入力コミットメント及び場所の分かっている $\boxed{\heartsuit}$ を 1 枚を使用して，Five-card trick によって再び論理積を秘密計算する．この方法では，使用するカードが合計 6 枚で済む．しかし，Koch らのプロトコルを使用するので，一様でないシャッフルを用いており操作が複雑になる．



先述の Abe らのプロトコルを用いれば一様で閉じたシャッフルのみで 3 入力 AND を秘密計算することができるが，7 枚のカードを要する．



1.3 本稿の貢献

本稿では，既存のどのプロトコルよりも効率的な 6 枚非コミット型 3 入力 AND プロトコルを新たに提案する．このプロトコルはランダムカットとランダム二等分割カットのみ，すなわち一様で閉じたシャッフルのみを用いる．さらに，Mizuki のプロトコル [6] は必要なシャッフル回数が 5 回であったのに対して，提案プロトコルは 2 回（ランダムカットとランダム二等分割カットをそれぞれ 1 回）と大幅な削減に成功している．表 2 を再度参照されたい．

Vol.2021-DPS-186 No.6
Vol.2021-CSEC-92 No.6
2021/3/15

表 1　既存の 2 入力 AND プロトコル

	形式	カード枚数	シャッフル回数	finite	uniform closed
Den Boer [1]	非コミット型	5	1	✓	✓
Kumamoto ら [2]	非コミット型	4	2	✓	✓
Koch ら [3]	コミット型	4	8		
Abe ら [4]	コミット型	5	7		✓
Abe ら [5]	コミット型	5	4.5		✓

表 2　既存の 3 入力 AND プロトコルと提案プロトコル

	形式	カード枚数	シャッフル回数	finite	uniform closed
Mizuki [6]	非コミット型	7	2	✓	✓
Mizuki [6]	非コミット型	6	5	✓	✓
Koch+Boer	非コミット型	6	9		
Abe+Boer	非コミット型	7	5.5		✓
本稿	**非コミット型**	**6**	**2**	✓	✓

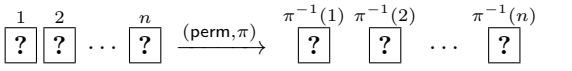
2. 準備

本節ではカードベース暗号で使用される操作について説明する．次に，提案プロトコルで使用する 2 つのシャッフル方法について説明を行う．

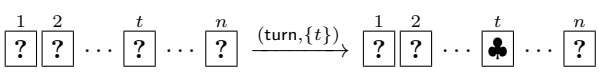
2.1 カードベース暗号で使用される操作

カードベース暗号では，カード列に対して主に次の 3 つの操作が行われる．

並べ替え　カード列に対してある置換 $\pi \in S_n$ を適用する．ここで S_n は n 次の対称群である．この操作を (perm,π) と書く．

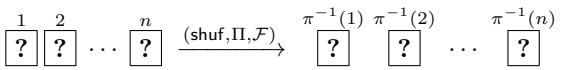


めくる　カード列の左から t 番目をめくってカードの模様を確認する．この操作を $(\mathsf{turn},\{t\})$ と書く．



インデックスの集合 T に対しても同様に (turn,T) を定義する．

シャッフル　カード列に対してある置換集合 $\Pi \subseteq S_n$ から確率分布 $\mathcal{F}$ にしたがって得られる置換 π を適用する．この操作を $(\mathsf{shuf},\Pi,\mathcal{F})$ と書く．

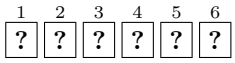


Π に含まれる置換のうち，どれが適用されるかは誰も知り得ないものとする．確率分布 $\mathcal{F}$ が一様（uniform）である場合，一様なシャッフルと呼ぶとともに，表記から $\mathcal{F}$ を省略して (shuf,Π) のように書く．

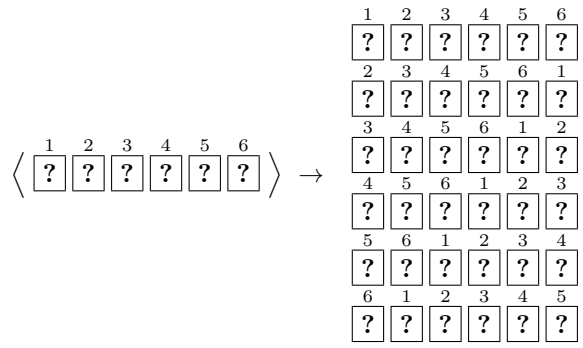
任意の $\tau,\sigma \in \Pi$ に対して $\tau\sigma \in \Pi$ であるとき，そのシャッフルを closed である（閉じている）という．この条件を満たさないとき，non-closed である（閉じていない）という．

2.2 ランダムカット

ランダムカットとは，ランダムな回数分カード列を巡回的にシフトするシャッフル操作のことである．カード列に対してランダムカットを適用するとき，$\langle\cdot\rangle$ と表記する．6 枚のカードにランダムカットを適用する例を次に示す．説明のためにカードの上に番号を振る．



このカード列に対してランダムカットを適用すると，右のカード列のいずれか 1 つが得られる．これらのカード列の生起確率は一様（すなわち 1/6）である．



このランダムカットは巡回置換 $\sigma = (1\,2\,3\,4\,5\,6)$ を用いて

$$(\mathsf{shuf},\{\mathsf{id},\sigma,\sigma^2,\sigma^3,\sigma^4,\sigma^5\})$$

と書くことができる．ここで id は恒等置換を表す．本稿ではこのランダムカットの簡易的な表記として

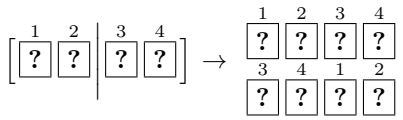
$$(\mathsf{shuf},\mathsf{RC}_{1,2,3,4,5,6})$$

と書くことにする．ランダムカットは人間の手で簡単に実現できることが実験的に確認されている [7]．

2.3 ランダム二等分割カット

ランダム二等分割カット [8] は，2009 年に Mizuki と Sone

によって考案されたシャッフル操作である．これは，カード列を半分に割り，その左右をランダムに入れ替える操作である．カード列にランダム二等分割カットを適用するとき，$[\cdot|\cdot]$ と表記する．例として，4 枚のカード列にランダム二等分割カットを適用すると，右のカード列のいずれかが 1/2 の確率で得られる．



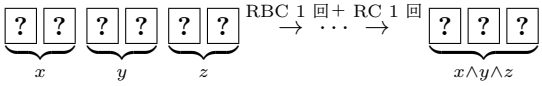
このランダム二等分割カットは，次のように書くことができる．

$$(\mathsf{shuf}, \{\mathsf{id}, (1\,3)(2\,4)\})$$

ランダム二等分割カットは身近な道具を用いて安全に実装できることが知られている [7]．カードの裏面が上下非対称の場合は，ランダムカットを用いて実装することもできる [9]．

3. 提案プロトコル

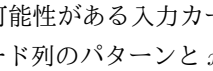
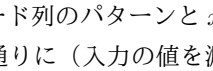
本節では，6 枚のカードを用いて 3 入力 AND を秘密計算する新しいプロトコルを提案する．



このプロトコルはランダム二等分割カット 1 回とランダムカット 1 回を使用する．

まず，3.1 節でプロトコルのアイデアを述べ，3.2 節でプロトコルを記述し，3.3 節でプロトコルの正当性の直感的な理解を与え，3.4 節で正当性と安全性について説明する．

3.1 アイデア

プロトコルの目的は，入力の値 $(x, y, z) \in \{0, 1\}^3$ によって 8 通りの可能性がある入力カード列を，$x \wedge y \wedge z = 1$ に対応するカード列のパターンと $x \wedge y \wedge z = 0$ に対応するパターンの 2 通りに（入力の値を漏らさずに）割り当てることである．したがって，まず始めに $x \wedge y \wedge z = 1$ に対応するカード列と $x \wedge y \wedge z = 0$ に対応するカード列を定める．$(x, y, z) = (1, 1, 1)$ のときの入力カード列は ♣ と ♡ の模様が交互に並ぶカード列であるため，そのようなカード列パターン（  または  ，**相互カード列**と呼ぶ）を $x \wedge y \wedge z = 1$ に対応するものとし，それ以外のカード列パターン（**非相互カード列**）を $x \wedge y \wedge z = 0$ に対応するものと定める．

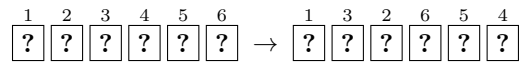
入力とそれに対応するカード列を表 3 の 1 列目と 2 列目に示す．表 3 から，入力のカード列が相互カード列となっているのは $(0, 0, 0)$ と $(1, 1, 1)$ の場合であることが分かる．したがって，$(0, 0, 0)$ の場合の入力カード列を非相互カード列に変更（かつ $(0, 0, 0)$ 以外の対応関係は維持）できれば，3 入力の論理積を得られる．

3.2 記述

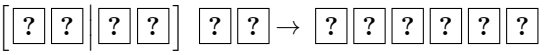
提案プロトコルの具体的な操作を次に示す．

(1) 次の (a)–(c) の操作を行うことで $(\mathsf{shuf}, \{\mathsf{id}, (1\,2)(3\,6)\})$ を適用する．

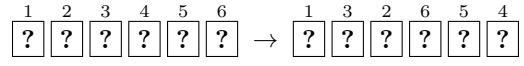
(a) カード列の 2 番目と 3 番目及び 4 番目と 6 番目をそれぞれ入れ替える．



(b) 1 から 4 番目の 4 枚のカードに対してランダム二等分割カットを適用する．

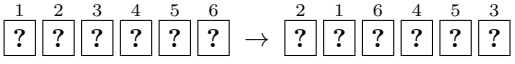


(c) 再び 2 番目と 3 番目及び 4 番目と 6 番目を入れ替える．

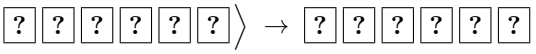


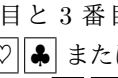
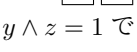
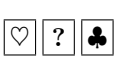
(2) 1 番目のカードをめくり，その模様を確認する．♣ が出た場合，1 番目と 2 番目及び 3 番目と 6 番目をそれぞれ入れ替える．♡ が出た場合は何も行わずに (3) に進む．

♣ が出た場合



(3) 全体にランダムカットを適用する．



(4) 1 番目と 3 番目と 5 番目をめくる．  か  またはその巡回であった場合，$x \wedge y \wedge z = 0$ である．  または  であった場合，$x \wedge y \wedge z = 1$ である．

$x \wedge y \wedge z = 0$

$x \wedge y \wedge z = 1$

3.3 操作の理由

ステップ 1 のシャッフルとステップ 2 の並べ替えに登場する置換 $(1\,2)(3\,6)$ を考え，入力カード列にそれを適用したとすると，表 3 の 3 列目のようになる．このとき，表 3 の 2 列目（入力時のカード列）と 3 列目（$(\mathsf{perm}, (1\,2)(3\,6))$ の適

表 3　入力と対応するカード列

入力	入力時のカード列	(perm,(1 2)(3 6)) の適用
(0,0,0)	♣♡♣♡♣♡	♡♣♡♡♣♣
(0,0,1)	♣♡♣♡♡♣	♡♣♣♡♡♣
(0,1,0)	♣♡♡♣♣♡	♡♣♡♣♣♡
(0,1,1)	♣♡♡♣♡♣	♡♣♣♣♡♡
(1,0,0)	♡♣♣♡♣♡	♣♡♡♡♣♣
(1,0,1)	♡♣♣♡♡♣	♣♡♣♡♡♣
(1,1,0)	♡♣♡♣♣♡	♣♡♡♣♣♡
(1,1,1)	♡♣♡♣♡♣	♣♡♣♣♡♡

表 4　入力とステップ 3 後のカード列

入力	ステップ 3 後のカード列
(0,0,0)	♡♣♡♡♣♣
(0,0,1)	♡♣♣♡♡♣
(0,1,0)	♡♣♡♣♣♡
(0,1,1)	♡♣♣♣♡♡
(1,0,0)	♡♣♣♡♣♡
(1,0,1)	♡♣♣♡♡♣
(1,1,0)	♡♣♡♣♣♡
(1,1,1)	♡♣♡♣♡♣

用後のカード列）は，ステップ 1 で (shuf, {id, (1 2)(3 6)}) を適用した後の遷移可能性をそのまま示していることに注意しよう．表 3 の 3 列目において，$x = 0$ のとき，すなわち上から 4 つのカード列を見ると，入力が $(0,0,1)$ と $(0,1,0)$ と $(0,1,1)$ のときは非相互カード列のまま，$(0,0,0)$ のときは非相互カード列に変換できている．したがって，3 列目（(perm,(1 2)(3 6)) の適用後のカード列）の上から 4 つのカード列と，2 列目（入力時のカード列）の下から 4 つのカード列を組み合わせることができると，表 4 のようなカード列が得られ，$(1,1,1)$ のときのみ相互カード列になる．そのためには，表 3 の各カード列について先頭のカードが ♣ のときに置換 (1 2)(3 6) を適用すればよい．以上が，ステップ 1 でシャッフル (shuf, {id, (1 2)(3 6)}) を適用し，ステップ 2 で先頭をめくり ♣ が出たときに (perm,(1 2)(3 6)) を行う理由である．

ステップ 4 の操作でカード列にランダムカットを適用すると，次のカード列あるいはその巡回が得られる．

(a) ♡♡♣♣♡♣
(b) ♣♣♡♡♣♡
(c) ♡♡♡♣♣♣
(d) ♡♣♡♣♡♣

ここで，入力が $(0,0,0)$ と $(1,0,0)$ の場合は (a) のカード列に，$(0,0,1)$ と $(0,1,0)$ と $(1,0,1)$ と $(1,1,0)$ は (b) に，$(0,1,1)$ は (c) に，$(1,1,1)$ は (d) にそれぞれ対応している．今，すべてのカードをめくってしまうと (a)–(c) が区別でき，情報が漏れる．そこで，カード列の奇数番目に着目する．奇数番目のみをめくると，(a)–(c) は ♡♣♡ か ♣♡♣ またはその巡回が得られる．(d) は ♡♡♡ または ♣♣♣ が得られる．このようにすることで，(a)–(c) が区別できなくなり，安全に $x \wedge y \wedge z$ の値だけを得ることができる．

3.4　正当性と安全性

提案した非コミット型プロトコルの正当性と安全性を KWH-tree [3] を用いて証明する．KWH-tree はカードの状態を表すノードと，それらを結び操作を表すエッジで構成される．各ノードにおいて確率分布の和が入力の確率分布に等しいという条件を満たしつつ KWH-tree を描くことができれば，プロトコルの正当性と安全性は証明される．提案プロトコルの KWH-tree を図 1 に示す．これより提案プロトコルは正当かつ安全である．

出力の安全性について詳しい説明を付け加える．図 1 における最終ノードを，出力の模様によって場合分けを行い，図 2 に示す．出力に関して $x \wedge y \wedge z = 0$ となる如何なる模様が出た場合であっても，入力が $(0,0,0)$，$(0,0,1)$，$(0,1,0)$，$(0,1,1)$，$(1,0,0)$，$(1,0,1)$，$(1,1,0)$ である確率は，$x \wedge y \wedge z = 0$ であるときのそれぞれの条件付き確率と同じである．故に，$x \wedge y \wedge z = 0$ が得られたとしても，出力の模様によって（$x \wedge y \wedge z = 0$ であることを超えた）入力の情報が漏れることはなく，安全である．

4.　おわりに

本稿では，一様で閉じたシャッフルを用いる非コミット型 3 入力 AND プロトコルを提案した．このプロトコルは 6 枚のカードのみを用いており，必要なカード枚数は最小である．6 枚のカードを用いる既存のプロトコル [6] では有限 5 回のシャッフルが必要であったが，提案プロトコルはそれを有限 2 回にまで削減することができた．

参考文献

[1] B. den Boer, “More efficient match-making and satisfiability The five card trick,” Advances in Cryptology–EUROCRYPT ’89, pp. 208–217, 1989.
[2] T. Mizuki, M. Kumamoto, and H. Sone, “The five-card trick can be done with four cards,” ASIACRYPT 2012, pp. 598–606, 2012.
[3] A. Koch, S. Walzer, and K. Härtel, “Card-based cryptographic protocols using a minimal number of cards,” ASIACRYPT 2015, pp. 783–807, 2015.
[4] Y. Abe, Y. Hayashi, T. Mizuki, and H. Sone, “Five-card AND protocol in committed format using only practical shuffles,” APKC 2018, pp. 3–8, 2018.

[5] Y. Abe, Y. Hayashi, T. Mizuki, and H. Sone, “Five-card AND computations in committed format using only uniform cyclic shuffles,” NGC 2021, 2021.
[6] T. Mizuki, “Card-based protocols for securely computing the conjunction of multiple variables,” TCS 2016, pp. 34–44, 2016.
[7] I. Ueda, A. Nishimura, Y. Hayashi, T. Mizuki and H. Sone, “How to implement a random bisection cut,” TPNC 2016, pp.58–69, 2016.
[8] T. Mizuki and H. Sone, “Six-card secure AND and four-card secure XOR,” FAW 2009, pp. 358–369, 2009.
[9] I. Ueda, D. Miyahara, A. Nishimura, Y. Hayashi, T. Mizuki and H. Sone, “Secure implementations of a random bisection cut,” International Journal of Information Security, pp.445–452, 2020.

Vol.2021-DPS-186 No.6
Vol.2021-CSEC-92 No.6
2021/3/15

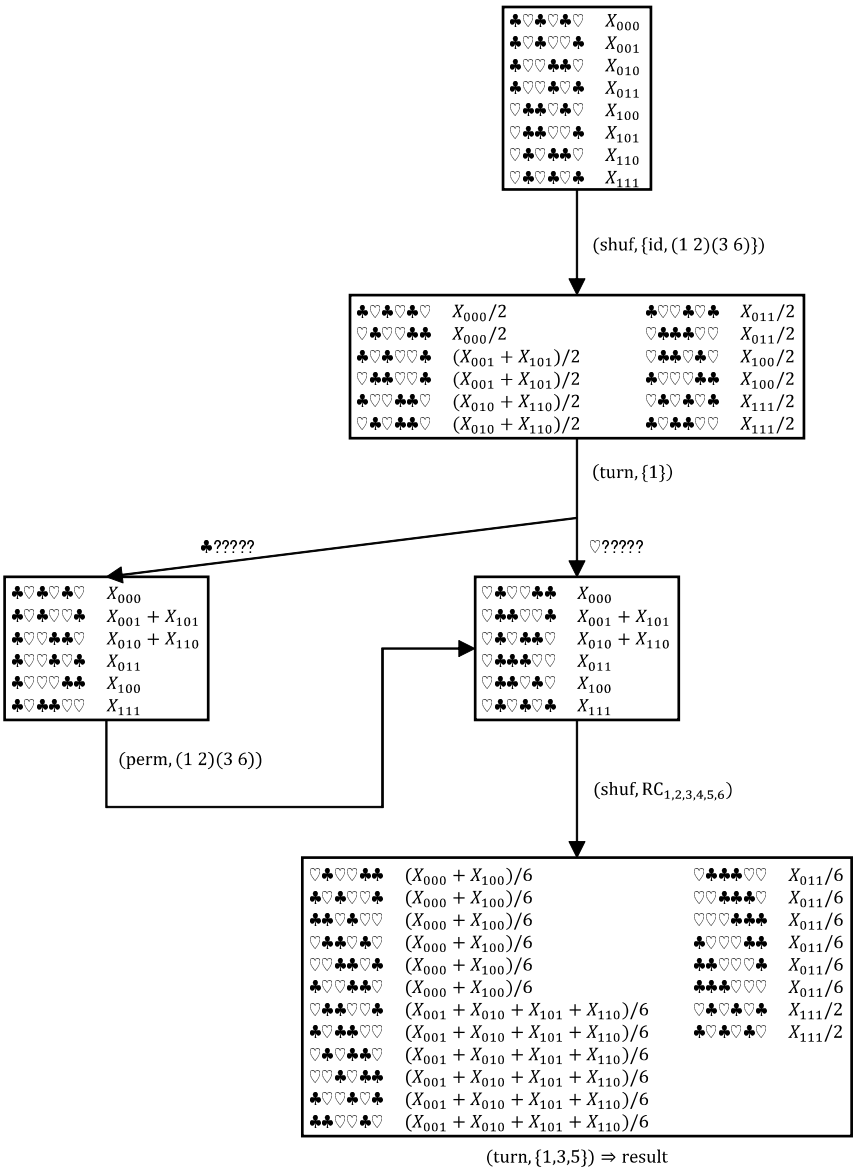


図 1　提案プロトコルの KWH-tree

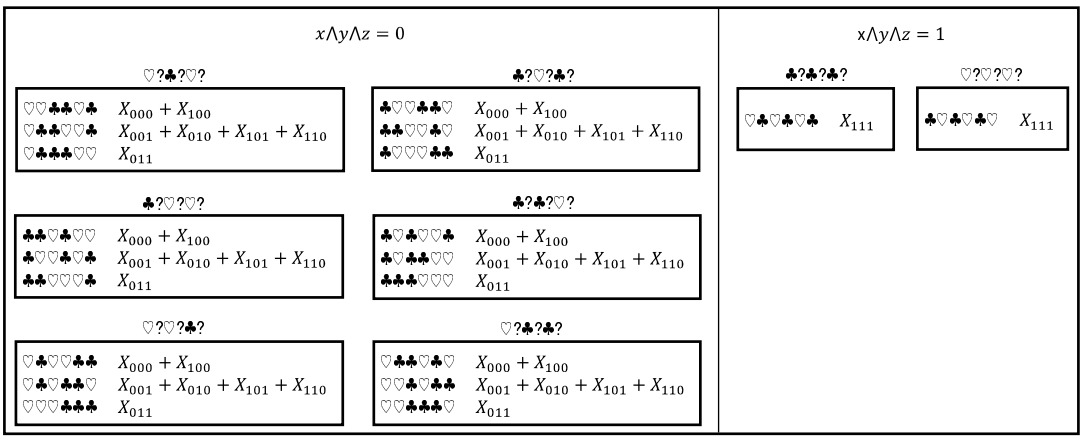


図 2　(turn,{1, 3, 5}) の後の状態