Vol.2022-DPS-190 No.35
Vol.2022-CSEC-96 No.35
2022/3/11

トランプカードによる3入力論理関数の秘密計算プロトコル

芳賀 陸雄[1]　林 優一[1,4]　宮原 大輝[2,4]　水木 敬明[3,4]

概要：物理的なカード組を用いて秘密計算を実現するプロトコルをカードベース暗号プロトコルという．カードベース暗号プロトコルでは，赤と黒の 2 種類のカードが使われることが多いが，Niemi と Renvall によって 1999 年に市販のトランプカードを用いたプロトコルが初めて考案されて以来，2 入力の AND や XOR の秘密計算プロトコルがいくつか提案されている．最近では，Koyama らによって 3 入力の AND や多数決関数を秘密計算する効率的なプロトコルが開発されている．本稿では，Koyama らのプロトコルの一般化を目指し，3 入力論理関数を秘密計算する，トランプカードを用いた汎用的なプロトコルを構成する．

1. 序論

物理的なカード組を用いて秘密計算を実現するプロトコルを**カードベース暗号プロトコル** [1,7] という．カードベース暗号の既存研究では多くの場合，表面が $\boxed{\heartsuit}$ と $\boxed{\clubsuit}$ であり，裏面が $\boxed{?}$ で区別がつかないカード組が用いられる．ブール値は以下のように符号化される．

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

2 枚の裏になったカードがビット $x \in \{0,1\}$ を表すとき，この 2 枚のカードを x の**コミットメント**と呼び，以下のように書く．

$$\underbrace{\boxed{?}\boxed{?}}_{x}$$

このコミットメントで入出力を行うプロトコルを**コミット型**プロトコルと呼び，論理関数の値をコミットメントとして出力するプロトコルが数多く提案されている．

1.1 トランプカードを用いたカードベース暗号

$\boxed{\clubsuit}$ や $\boxed{\heartsuit}$ のようなスートからなるカードを用いたカードベース暗号プロトコルは市販のトランプカード 1 セットでは実現することができない．なぜなら，市販のカードにはスートの他に数字が書かれているからである．市販のトランプカードを用いたカードベース暗号は，1999 年に Niemi と Renvall [9] によって初めて提案された．市販のトランプカードはジョーカーを除いて数字とスートの組み合わせが 52 通り存在するので，トランプの 1 セットを 1 から 52 の自然数からなるとみなす．トランプカードを用いたプロトコルでは，ビット $x \in \{0,1\}$ を $1 \le i < j \le 52$ を満たす i と j を用いて以下のように符号化される．

$$\boxed{i}\boxed{j} = 0, \quad \boxed{j}\boxed{i} = 1$$

すなわち，左のカードの数字の方が小さければ 0，左のカードの数字の方が大きければ 1 を表す．ビット x が $\boxed{i}$ と $\boxed{j}$ で表されるコミットメントであるとき，以下のように書く．

$$\underbrace{\boxed{?}\boxed{?}}_{[x]^{\{i,j\}}}$$

ここで，集合 $\{i,j\}$ をコミットメントの**ベース**と呼ぶ．前述の Niemi と Renvall [9] は，具体的にコミット型 2 入力 AND プロトコルを提案し，またコミット型 2 入力 XOR プロトコルとコピープロトコルも提案している．

1.2 既存研究

本稿では，トランプカードを用いた 3 入力論理関数の秘密計算に焦点を絞る．そこで，トランプカードを用いた 3 入力 AND プロトコルに関する既存研究を表 1 に示す．カードベース暗号は実際に手元で実行可能であることに意義があるため，プロトコルは必要なカードの枚数とシャッフルの回数，ステップ数の少なさで評価される．有限のステップで終了するプロトコルを **finite** であるといい，プロトコル内にループが存在し，ステップ数が有限の期待値

[1] 奈良先端科学技術大学院大学, 奈良県生駒市高山町 8916-5
Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara 630-0192, Japan
[2] 電気通信大学, 東京都調布市調布々丘 1-5-1
The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan
[3] 東北大学, 宮城県仙台市青葉区荒巻字青葉 6-3
Tohoku University, Aramaki-Aza-Aoba, Aoba, Sendai 980-8576, Japan
[4] 産業技術総合研究所, 東京都江東区青海 2-3-26
National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan

として表されるプロトコルを **Las Vegas** であるという．表 1 では 4 列目がこれを示す．Koyama ら [4] は Niemi と Revall の 2 入力 AND プロトコルを拡張し，Las Vegas ではあるが，8.5 回（期待値）のシャッフル回数と最小枚数（6 枚）の 3 入力 AND プロトコルを提案している．

表 1 トランプカードを用いた 3 入力 AND プロトコル

	カード枚数	シャッフル回数	有限
Mizuki [5] (2 times)	10	8	✓
Koch et al. [2] (2 times)	6	12 (exp.)	
Koyama et al. [4]	6	8.5 (exp.)	

1.3 本稿の貢献

本稿の貢献は，Niemi と Renvall の 5 枚のカード列を並び替えることで 2 入力 AND プロトコル [9] を構成する方法と，Koyama らのコミットメントによるスワップにより 3 入力 AND プロトコル [4] と 3 入力多数決プロトコル [10] を構成する方法を発展させ，Koyama らと同じ 6 枚のカードと 8.5 回（期待値）のシャッフル回数で汎用的に 3 入力論理関数を秘密計算するプロトコルの構成を提案する．さらに，この手法が適用できない（一部の）3 入力論理関数に対しては，8 枚のカードで構成する方法を提案する．

1.4 本稿の構成

2 節では，カードベース暗号で用いられる操作と，Niemi と Renvall の 2 入力 AND プロトコル，Koyama らの 3 入力論理関数の構成 [10] について説明する．3 節では，Niemi と Renvall，Koyama らのアイディアを一般化することで汎用的な 3 入力論理関数プロトコルを構成する方法を示し，これによって実際に一例として 3 入力 XOR プロトコルを構成する．さらに，この方法では実現できない一部の 3 入力論理関数に対して，2 枚のカードを追加した 8 枚カードの構成法を示して，4 節にてまとめる．

2. 準備

本節では，カードベース暗号の標準的な計算モデル [6] に従い，基本的なカード操作について説明する．次に，カードベース暗号において実用的なシャッフル操作であるランダムカットとランダム 2 等分割カットについて説明する．最後に，Niemi と Renvall の 2 入力 AND プロトコル [9] と Koyama らの 3 入力論理関数の構成方法 [4, 10] を説明する．

2.1 カードベース暗号で使用する操作

ここでは，カードベース暗号で使用される基本的な操作を以下のように定義する．

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 次の対称群を表し，この操作を (perm, π) と書く．

$$\boxed{?}_1\boxed{?}_2\cdots\boxed{?}_n \xrightarrow{(\mathsf{perm},\pi)} \boxed{?}_{\pi^{-1}(1)}\boxed{?}_{\pi^{-1}(2)}\cdots\boxed{?}_{\pi^{-1}(n)}$$

めくる． カード列の左から t 枚目のカードをめくり，カードの色を確認する．この操作を $(\mathsf{turn}, \{t\})$ と書く．

$$\boxed{?}_1\boxed{?}_2\cdots\boxed{?}_t\cdots\boxed{?}_n \xrightarrow{(\mathsf{turn},\{t\})} \boxed{?}_1\boxed{?}_2\cdots\boxed{\clubsuit}_t\cdots\boxed{?}_n$$

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 $\mathcal{F}$ に従って選ばれる置換 $\pi \in \Pi$ を適用する．この操作を $(\mathsf{shuf}, \Pi, \mathcal{F})$ と書く．

$$\boxed{?}_1\boxed{?}_2\cdots\boxed{?}_n \xrightarrow{(\mathsf{shuf},\Pi,\mathcal{F})} \boxed{?}_{\pi^{-1}(1)}\boxed{?}_{\pi^{-1}(2)}\cdots\boxed{?}_{\pi^{-1}(n)}$$

ただし，Π に含まれる置換の内, どの置換が適用されたのかは誰も知り得ないとする．

2.2 ランダムカット

ランダムカットは，カード列を巡回的にシフトさせることでカードをシャッフルする操作である．n 枚のカード列に対してランダムカットを適用すると，カード列の n 通りのうちいずれか 1 つとなり，その生起確率は $1/n$ である．ランダムカットは $< \cdot >$ で表記し，以下のように書く．

$$\left\langle \boxed{?}_1\boxed{?}_2\cdots\boxed{?}_{n-1}\boxed{?}_n \right\rangle \rightarrow \begin{cases} \boxed{?}_1\boxed{?}_2\cdots\boxed{?}_{n-1}\boxed{?}_n \\ \boxed{?}_2\boxed{?}_3\cdots\boxed{?}_n\boxed{?}_1 \\ \vdots \\ \boxed{?}_{n-1}\boxed{?}_n\cdots\boxed{?}_{n-3}\boxed{?}_{n-2} \\ \boxed{?}_n\boxed{?}_1\cdots\boxed{?}_{n-2}\boxed{?}_{n-1} \end{cases}$$

2.3 ランダム 2 等分割カット

ランダム 2 等分割カット (Random Bisection Cut: RBC) は 2009 年に Mizuki と Sone によって提案されたシャッフル操作 [8] である．$2n$ 枚のカードに対して RBC を適用すると，カード列は次のうちのどちらかに遷移し，その生起確率は 1/2 である．RBC は $[\cdot|\cdot]$ で表記し，以下のように書く．

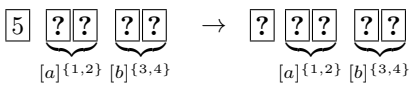
$$\left[\boxed{?}_1\cdots\boxed{?}_n \,\middle|\, \boxed{?}_{n+1}\cdots\boxed{?}_{2n}\right] \rightarrow \begin{cases} \boxed{?}_1\cdots\boxed{?}_n \,|\, \boxed{?}_{n+1}\cdots\boxed{?}_{2n} \\ \boxed{?}_{n+1}\cdots\boxed{?}_{2n} \,|\, \boxed{?}_1\cdots\boxed{?}_n \end{cases}$$

2.4 Niemi と Renvall の 2 入力 AND プロトコル

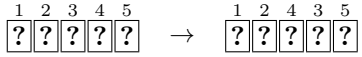
Niemi と Renvall の 2 入力 AND プロトコル [9] は，$a, b \in \{0, 1\}$ のコミットメントとして 4 枚のカードと追加のカード 1 枚，すなわち合計 5 枚を入力として，$a \wedge b$ を

出力する．以下に手順を示す．

(1) 2 つの入力コミットメントと追加のカード $\boxed{5}$ を以下のように置き，裏向きにする．

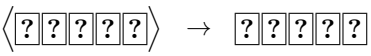


(2) 3 枚目と 4 枚目のカードを並び替える．



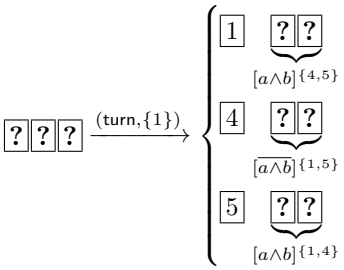
ここで，入力のカード列と並び替え後のカード列は表 2 の 3 列目と 4 列目のようになる．さらに，並び替え後のカード列の $\boxed{5},\boxed{1},\boxed{4}$ の順序に着目すると，$a \wedge b = 1$ のときは $\boxed{5} \to \boxed{1} \to \boxed{4}$ となっており，$a \wedge b = 0$ のときは $\boxed{5} \to \boxed{4} \to \boxed{1}$ となっていることが分かる．そこで，入力の情報を漏らさないように以下の操作を繰り返すことで，$\boxed{2}$ と $\boxed{3}$ のカードを取り除く．

(3) ランダムカットをカード列に適用する．



(4) 1 枚目をめくり，そのカードが $\boxed{2}$ または $\boxed{3}$ であれば取り除く．そうでなければめくったカードを裏返す．カード列から $\boxed{2}$ と $\boxed{3}$ が全て取り除かれるまでステップ (3) と (4) を繰り返す．

(5) $\boxed{2}$ と $\boxed{3}$ のカードが取り除かれると，カード列は表 2 の 5 列目のようになる．ランダムカットをもう 1 度カード列に適用した後に 1 枚目をめくることで以下の出力を得られる（この操作は Koch らによる改良アイディア [2] によるものである）．



1 枚目が $\boxed{4}$ の場合は，$a \wedge b$ の否定が得られる．その場合はコミットメントを構成する 2 枚の順序を入れ替えることで，$a \wedge b$ のコミットメントを得ることができる．

このプロトコルの正当性は表 2 より明らかである．また，ステップ (3) でランダムカットを適用しているため，ステップ (4) でめくるカードはランダムであり，さらに $\boxed{2}$ と $\boxed{3}$ のカードを除くため，ステップ (5) でめくるカードからは入力情報が漏れない．以上より，このプロトコルは正当かつ安全である．

2.5 Koyama らの 3 入力論理関数プロトコル

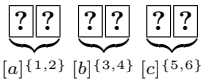
Koyama ら [4, 10] は，3 入力の AND と多数決関数が次のように表現できることに基づき，プロトコルを構築した．

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

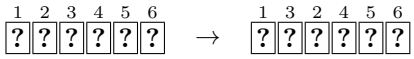
$$\mathsf{maj}(a,b,c) = \begin{cases} \mathrm{OR}(a,b) & \text{if } c = 1 \\ \mathrm{AND}(a,b) & \text{if } c = 0 \end{cases}$$

ここでは，3 入力多数決プロトコルの手順のみ以下に示す．

(1) 3 つの入力コミットメントを置く．

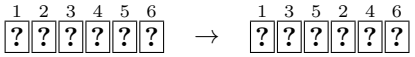


(2) 2 枚目と 3 枚目を入れ替える．

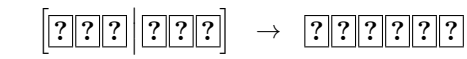


(3) c のコミットメントによるスワップ操作を 1 枚目，2 枚目のカード列と 3 枚目，4 枚目のカード列に次のように適用する．

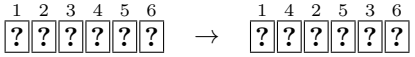
(a) 次のように並び替える．



(b) RBC をカード列に適用する．



(c) 次のように並び替える．



(d) 5 枚目と 6 枚目をめくり，$\boxed{5}\boxed{6}$ ならば何もせず，$\boxed{6}\boxed{5}$ が出たら，1,2 枚目と 3,4 枚目のカードをそれぞれスワップする．

(4) ステップ (3d) でめくった $\boxed{5}$ のカードを追加カードとし，1 枚目から 4 枚目のカードを入力にして Niemi と Renvall のプロトコル [9] のステップ (3), (4), (5) を適用する．

このプロトコルの正当性は表 3 より明らかである．また，ステップ (3b) で RBC を適用しているため，ステップ (3d) でめくる $\boxed{5}\boxed{6}$ の並びの確率は 1/2 であるからステップ (3d) でめくるカードから入力の情報は漏れない．ステップ (4) は Niemi と Renvall のプロトコルと同様である．以上より，このプロトコルは正当かつ安全である．

Vol.2022-DPS-190 No.35
Vol.2022-CSEC-96 No.35
2022/3/11

表 2 入力と並び替え後のカード列と $\boxed{2}$ と $\boxed{3}$ を除いた後

入力 (a,b)	$a \wedge b$	入力カード列	並び替え後	$\boxed{2}$ と $\boxed{3}$ を除いた後
$(0,0)$	0	$\boxed{5}\boxed{1}\boxed{2}\boxed{3}\boxed{4}$	$\boxed{5}\boxed{1}\boxed{3}\boxed{2}\boxed{4}$	$\boxed{1}\boxed{4}\boxed{5}$ or $\boxed{4}\boxed{5}\boxed{1}$ or $\boxed{5}\boxed{1}\boxed{4}$
$(0,1)$	0	$\boxed{5}\boxed{1}\boxed{2}\boxed{4}\boxed{3}$	$\boxed{5}\boxed{1}\boxed{4}\boxed{2}\boxed{3}$	$\boxed{1}\boxed{4}\boxed{5}$ or $\boxed{4}\boxed{5}\boxed{1}$ or $\boxed{5}\boxed{1}\boxed{4}$
$(1,0)$	0	$\boxed{5}\boxed{2}\boxed{1}\boxed{3}\boxed{4}$	$\boxed{5}\boxed{2}\boxed{3}\boxed{1}\boxed{4}$	$\boxed{1}\boxed{4}\boxed{5}$ or $\boxed{4}\boxed{5}\boxed{1}$ or $\boxed{5}\boxed{1}\boxed{4}$
$(1,1)$	1	$\boxed{5}\boxed{2}\boxed{1}\boxed{4}\boxed{3}$	$\boxed{5}\boxed{2}\boxed{4}\boxed{1}\boxed{3}$	$\boxed{1}\boxed{5}\boxed{4}$ or $\boxed{4}\boxed{1}\boxed{5}$ or $\boxed{5}\boxed{4}\boxed{1}$

表 3 3 入力多数決プロトコルの入力と並び替え後のカード列

入力 (a,b,c)	$\mathsf{maj}(a,b,c)$	入力カード列	並び替え後	$\boxed{5}\boxed{1}\boxed{4}$ の並び
$(0,0,0)$	0	$\boxed{1}\boxed{2}\boxed{3}\boxed{4}\boxed{5}\boxed{6}$	$\boxed{5}\boxed{1}\boxed{3}\boxed{2}\boxed{4}$	$\boxed{5}\boxed{1}\boxed{4}$
$(0,0,1)$	0	$\boxed{1}\boxed{2}\boxed{3}\boxed{4}\boxed{6}\boxed{5}$	$\boxed{5}\boxed{3}\boxed{1}\boxed{4}\boxed{2}$	$\boxed{5}\boxed{1}\boxed{4}$
$(0,1,0)$	0	$\boxed{1}\boxed{2}\boxed{4}\boxed{3}\boxed{5}\boxed{6}$	$\boxed{5}\boxed{1}\boxed{4}\boxed{2}\boxed{3}$	$\boxed{5}\boxed{1}\boxed{4}$
$(0,1,1)$	1	$\boxed{1}\boxed{2}\boxed{4}\boxed{3}\boxed{6}\boxed{5}$	$\boxed{5}\boxed{4}\boxed{1}\boxed{3}\boxed{2}$	$\boxed{5}\boxed{4}\boxed{1}$
$(1,0,0)$	0	$\boxed{2}\boxed{1}\boxed{3}\boxed{4}\boxed{5}\boxed{6}$	$\boxed{5}\boxed{2}\boxed{3}\boxed{1}\boxed{4}$	$\boxed{5}\boxed{1}\boxed{4}$
$(1,0,1)$	1	$\boxed{2}\boxed{1}\boxed{3}\boxed{4}\boxed{6}\boxed{5}$	$\boxed{5}\boxed{3}\boxed{2}\boxed{4}\boxed{1}$	$\boxed{5}\boxed{4}\boxed{1}$
$(1,1,0)$	1	$\boxed{2}\boxed{1}\boxed{4}\boxed{3}\boxed{5}\boxed{6}$	$\boxed{5}\boxed{2}\boxed{4}\boxed{1}\boxed{3}$	$\boxed{5}\boxed{4}\boxed{1}$
$(1,1,1)$	1	$\boxed{2}\boxed{1}\boxed{4}\boxed{3}\boxed{6}\boxed{5}$	$\boxed{5}\boxed{4}\boxed{2}\boxed{3}\boxed{1}$	$\boxed{5}\boxed{4}\boxed{1}$

3. 汎用的な 3 入力論理関数の構成方法

本節では，Niemi と Renvall の 5 枚のカード列の並び替えによって 2 入力 AND プロトコル [9] を構成する方法と，Koyama らのコミットメントの値によるスワップ操作によって 3 入力 AND プロトコル [4] と 3 入力多数決プロトコル [10] を構成する方法を一般化し，3 入力論理関数を秘密計算する汎用的なプロトコルの構成方法を提案する．また，一般化のアイディアとその構成のために必要な RBC の操作とその手順を示し，実際に一例として 3 入力 XOR プロトコルを構成する．最後に，前述の手法でプロトコルが構成できない一部の 3 入力論理関数に対して，8 枚のカードでプロトコルを構成する方法を提案する．

3.1 一般化のアイディア

Niemi と Renvall の 5 枚のカード列の並べ替えによって 2 入力 AND プロトコル [9] を構成する方法と，Koyama らのコミットメントの値によるスワップ操作によって 3 入力 AND プロトコル [4] と 3 入力多数決プロトコル [10] を実現する方法を一般化し，c のコミットメントの値を秘匿したまま，その値によって適当な $g(a,b)$ と $h(a,b)$ を選択することで多くの論理関数 $f(a,b,c)$ を構成することが可能である．この一般化のアイディアは以下の式で表現できる．

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

より具体的には，

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

に対してまず，関数 $h(a,b)$ に対応するように並び替えを行い，$c = 1$ の時にはスワップ操作により $g(a,b)$ に対応するように並び替える．

3.1.1 RBC によるスワップ操作

c のコミットメントのブール値を秘匿したままスワップ操作を行うためには，RBC によるスワップ操作が必要である．スワップ操作は $h(a,b)$ から $g(a,b)$ への並べ替え方によって RBC の適用の仕方に対応するカード列が 4 種類存在し，並べ替え方によって適当なものを選ぶ必要がある．

例として $h(a,b)$ に AND，$g(a,b)$ に OR を選択すると，表 5 から分かるように，1 枚目と 2 枚目，3 枚目と 4 枚目のスワップ操作が必要である．ここでは，次の 4 つのスワップ操作 $(A),(B),(C)$ 及び (D) を考える．

(A) 2 枚目と 3 枚目の 2 枚だけをスワップさせる場合，1 枚目と 3 枚目，2 枚目と 4 枚目のような場合にもスワップさせたいカードを 3 枚目と 5 枚目に並べ替えることで，同様の手順でスワップさせることができる．

(a) 次のように並べ替える．

$$\overset{1\ 2\ 3\ 4\ 5\ 6}{\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}} \rightarrow \overset{1\ 4\ 2\ 5\ 3\ 6}{\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}}$$

(b) RBC を適用する．

$$\boxed{?}\boxed{?}\left[\boxed{?}\boxed{?}\middle|\boxed{?}\boxed{?}\right] \rightarrow \begin{cases} \overset{1\ 4\ 2\ 5\ 3\ 6}{\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}} \\ \overset{1\ 4\ 3\ 6\ 2\ 5}{\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}} \end{cases}$$

(c) (a) の逆置換を適用する．

$$\overset{1\ 4\ 2\ 5\ 3\ 6}{\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}} \rightarrow \overset{1\ 2\ 3\ 4\ 5\ 6}{\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}}$$

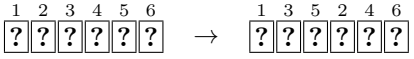
$$\overset{1\ 4\ 3\ 6\ 2\ 5}{\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}} \rightarrow \overset{1\ 3\ 2\ 4\ 5\ 6}{\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}\boxed{?}}$$

(d) 5 枚目と 6 枚目をめくり，$\boxed{5}\boxed{6}$ ならば何もせず，$\boxed{6}\boxed{5}$

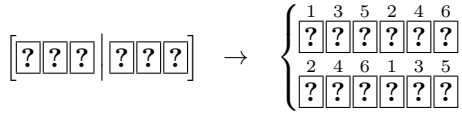
が出たら，2 枚目と 3 枚目のカードをスワップする．

(B) 1 枚目と 2 枚目，3 枚目と 4 枚目をスワップさせる場合

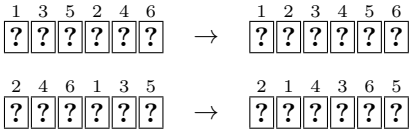
(a) 次のように並べ替える．



(b) RBC を適用する．



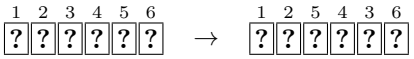
(c) (a) の逆置換を適用する．



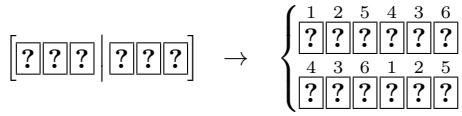
(d) 5 枚目と 6 枚目をめくり，$\boxed{5}\boxed{6}$ ならば何もせず，$\boxed{6}\boxed{5}$ が出たら，1,2 枚目と 3,4 枚目のカードをそれぞれスワップする．

(C) 1 枚目と 4 枚目，2 枚目と 3 枚目をスワップさせる場合

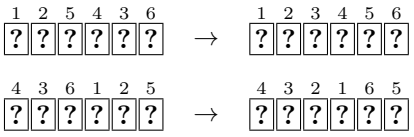
(a) 次のように並べ替える．



(b) RBC を適用する．



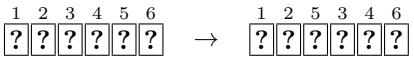
(c) (a) の逆置換を適用する．



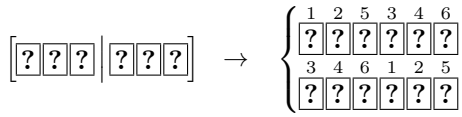
(d) 5 枚目と 6 枚目をめくり，$\boxed{5}\boxed{6}$ ならば何もせず，$\boxed{6}\boxed{5}$ が出たら，1,4 枚目と 2,3 枚目のカードをそれぞれスワップする．

(D) 1 枚目と 3 枚目，2 枚目と 4 枚目をスワップさせる場合

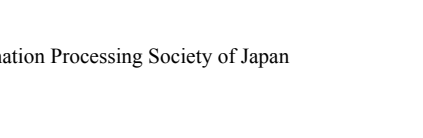
(a) 次のように並べ替える．



(b) RBC を適用する．



(c) (a) の逆置換を適用する．



(d) 5 枚目と 6 枚目をめくり，$\boxed{5}\boxed{6}$ ならば何もせず，$\boxed{6}\boxed{5}$ が出たら，1,3 枚目と 2,4 枚目のカードをそれぞれスワップする．

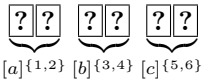
表 4 構成可能なプロトコル

$c=1$ $g(a,b)$	$c=0$ $h(a,b)$	スワップ	3-input Protocols $f(a,b)$
AND	0	(A)	3-AND [4]
1	OR	(A)	3-OR
NAND	OR	(C)	3-XOR
NAND	1	(A)	3-NAND
0	NOR	(A)	3-NOR
AND	NOR	(D)	3-XNOR
OR	AND	(B)	3-major [10]
AND	OR	(B)	3-minor
0	0		0
1	1		1

3.1.2 一般化した手順

以下に一般化した 3 入力論理関数プロトコルの手順を示す．表 4 から適当な $g(a,b)$ と $h(a,b)$ とスワップの種類を選択することで，3 入力論理関数を構成可能である．$h(a,b)$ の並べ替え方は表 5 に示した．

(1) 3 つの入力コミットメントを置く．



(2) $h(a,b)$ の並べ方に置換する．

(3) c のコミットメントによるスワップ操作を行う．

(a) スワップのパターンに対応した置換を行う．
(b) RBC を行う．
(c) (a) の逆置換を行う．
(d) 5 枚目と 6 枚目をめくり，$\boxed{5}\boxed{6}$ ならば何もせず，$\boxed{6}\boxed{5}$ が出たら，RBC のパターンに対応したカードをそれぞれスワップする．

(4) Niemi と Renvall の 2 入力 AND プロトコル [9] のステップ (3), (4), (5) を適用する．

3.2 3 入力 XOR プロトコル

ここでは一例として，3.1 節の手法を用いて 3 入力 XOR プロトコルを構成する．3 入力 XOR プロトコルは以下の式で表現することができる．$g(a,b)$ には NAND(a,b) が，$h(a,b)$ には OR(a,b) が対応する．

表 5 $h(a,b)$ の並べ替え方

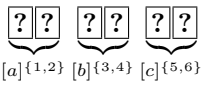
入力 (a,b)	入力カード列	AND(a,b) (perm, (2 3))	OR(a,b) (perm, (1 2 4 3))	NAND(a,b) (perm, (1 4))	NOR(a,b) (perm, (1 3 4 2))
$(0,0)$	1 2 3 4	1 3 2 4	3 1 4 2	4 2 3 1	2 4 1 3
$(0,1)$	1 2 4 3	1 4 2 3	4 1 3 2	3 2 4 1	2 3 1 4
$(1,0)$	2 1 3 4	2 3 1 4	3 2 4 1	4 1 3 2	1 4 2 3
$(1,1)$	2 1 4 3	2 4 1 3	4 2 3 1	3 1 4 2	1 3 2 4

$$\mathrm{XOR}(a,b,c)=\begin{cases}\mathrm{NAND}(a,b) & \text{if } c=1\\ \mathrm{OR}(a,b) & \text{if } c=0\end{cases}$$

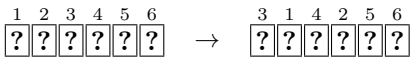
3.2.1 手順

3 入力 XOR プロトコルの計算手順は次のようになる．

(1) 3 つの入力コミットメントを置く．

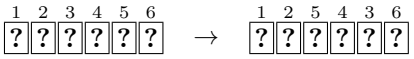


(2) 表 5 の OR の並びに並べ替える．

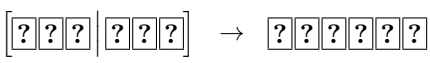


(3) c のコミットメントによるスワップ操作を行う．

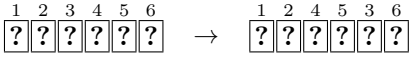
(a) 表 4 より，スワップ (C) の置換を適用する．



(b) RBC を適用する．



(c) (a) の逆置換を適用する．



(d) 5 枚目と 6 枚目をめくり，5 6 ならば何もせず，6 5 が出たら，1,4 枚目と 2,3 枚目のカードをそれぞれスワップする．

(4) (3d) でめくった 5 のカードを追加カードとし，1 枚目から 4 枚目のカードを入力に対して Niemi と Renvall のプロトコル [9] のステップ (3), (4), (5) を適用する．

3.2.2 正当性と安全性

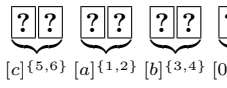
このプロトコルの正当性と安全性を KWH-tree [3] を用いて証明する．KWH-tree はノードにカード列の状態，エッジにカード列に対する操作を記述することでプロトコルを表現する図である．各ノードの確率分布の和が全て入力の確率分布と等しくなる時，そのプロトコルの安全性が証明される．この 3 入力 XOR プロトコルの正当性は表 6 の 5 列目より明らかである．安全性は 3 入力 XOR プロトコルの入力から手順 (2) に至るまでの KWH-tree を図 1 に示したことで証明される．ゆえに，このプロトコルは正当かつ安全である．

3.3 表 4 の関数以外の場合

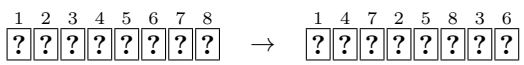
表 4 の $g(a,b)$ や $h(a,b)$ には XOR や XNOR がないが，その理由は表 4 の 2 入力関数と XOR/XNOR を組み合わせて提案手法を適用してもうまくいかないためである．2 入力 XOR プロトコルと 2 入力 XNOR プロトコルは，Niemi と Renvall [9] や Mizuki [5] によって提案されている．表 4 の 2 入力関数と XOR/XNOR では Niemi と Renvall のプロトコルにおけるステップ (4)(5) で取り除くカード，めくるカードが異なるため，表 4 の 2 入力関数と XOR/XNOR を単純に混在させた 3 入力論理関数を構成しようとすると c のコミットメントを秘匿したまま $g(a,b)$, $h(a,b)$ の選択ができないという問題がある．そこで，2 枚のカードを追加し, そのカードに c のコミットメントをコピーすることで，c のコミットメントによるスワップ操作を 2 回適用可能とする方法を提案する．これにより上述の関数が混在する場合の 3 入力論理関数が構成可能であることを示す．以下に式による表現と手順の一例を示す．

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

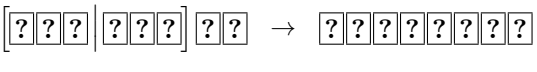
(1) 3 つのコミットメントとコピー用の 2 枚の追加カードを置く．



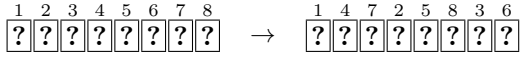
(2) 次のように並べ替える．



(3) RBC をカード列に適用する．



(4) (2) の逆置換を適用する．



(5) 1,2 枚目をめくり，5 6 であれば何もせず，6 5 であれば 4 枚目と 5 枚目，7 枚目と 8 枚目をスワップする．

ここで，7,8 枚目に c のコミットメントがコピーされ，7,8 枚目を使って c のコミットメントによるスワップ

Vol.2022-DPS-190 No.35
Vol.2022-CSEC-96 No.35
2022/3/11

表 6 3 入力 XOR プロトコルの入力と並べ替え後のカード列

入力 (a,b,c)	XOR(a,b,c)	入力カード列	並び替え後	$\boxed{5}\boxed{1}\boxed{4}$ の並び
$(0,0,0)$	0	1 2 3 4 5 6	5 3 1 4 2	5 1 4
$(0,0,1)$	1	1 2 3 4 6 5	5 4 2 3 1	5 4 1
$(0,1,0)$	1	1 2 4 3 5 6	5 4 1 3 2	5 4 1
$(0,1,1)$	1	1 2 4 3 6 5	5 3 2 4 1	5 4 1
$(1,0,0)$	1	2 1 3 4 5 6	5 3 2 4 1	5 4 1
$(1,0,1)$	1	2 1 3 4 6 5	5 4 1 3 2	5 4 1
$(1,1,0)$	1	2 1 4 3 5 6	5 4 2 3 1	5 4 1
$(1,1,1)$	0	2 1 4 3 6 5	5 3 1 4 2	5 1 4

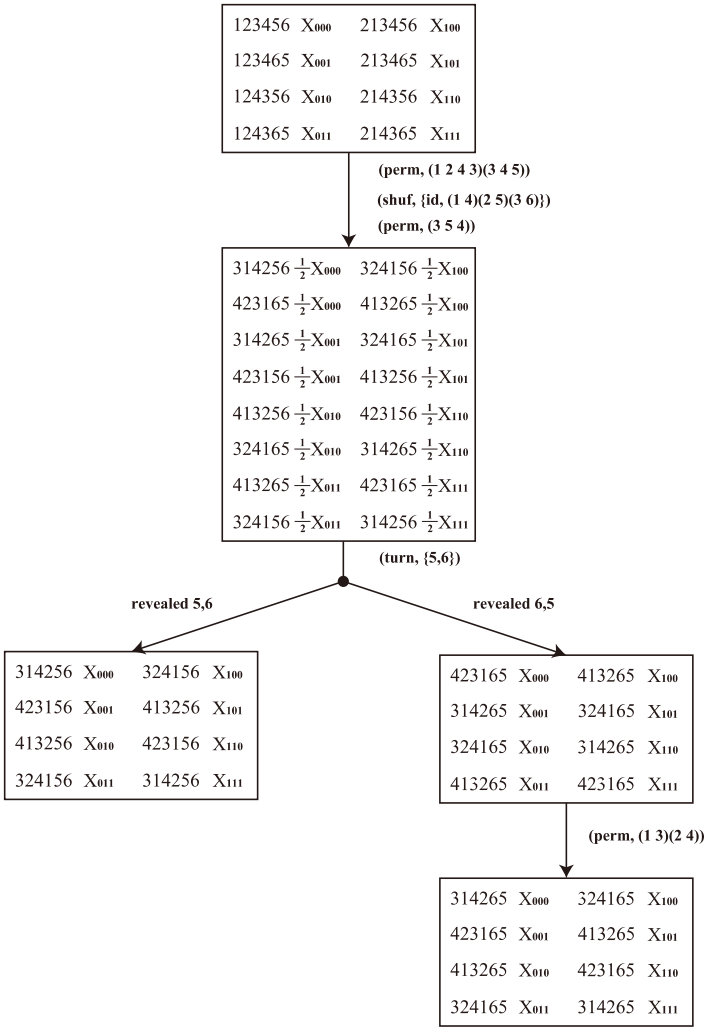


図 1 3 入力 XOR プロトコルの KWH-tree

操作がもう一度可能となる.

(6) 3 枚目から 6 枚目に対してランダムカットを適用する.

$$\overset{1\ 2}{?\,?}\ \Big\langle\ \overset{3\ 4\ 5\ 6}{?\,?\,?\,?}\ \Big\rangle\ \overset{7\ 8}{?\,?}$$

(7) 3 枚目をめくり $\boxed{4}$ が出るまでステップ (6) を繰り返す.

(8) 次のように並べ替える.

$$\overset{1\ 2\ 3\ 4\ 5\ 6\ 7\ 8}{?\,?\,?\,?\,?\,?\,?\,?} \quad \rightarrow \quad \overset{1\ 2\ 3\ 6\ 4\ 7\ 5\ 8}{?\,?\,?\,?\,?\,?\,?\,?}$$

(9) RBC をカード列に適用する.

$$?\,?\,?\,?\,\Big[\,?\,?\,\Big|\,?\,?\,\Big] \quad \rightarrow \quad ?\,?\,?\,?\,?\,?\,?\,?$$

(10) (8) の逆置換を適用する.

$$\overset{1\ 2\ 3\ 4\ 5\ 6\ 7\ 8}{?\,?\,?\,?\,?\,?\,?\,?} \quad \rightarrow \quad \overset{1\ 2\ 3\ 6\ 4\ 7\ 5\ 8}{?\,?\,?\,?\,?\,?\,?\,?}$$

(11) 3 枚目から 6 枚目に対してランダムカットを適用する.

$$\overset{1\ 2}{?\,?}\ \Big\langle\ \overset{3\ 4\ 5\ 6}{?\,?\,?\,?}\ \Big\rangle\ \overset{7\ 8}{?\,?}$$

(12) 3 枚目をめくり，$\boxed{2}$ が出るまで (11) を繰り返す．

(13) 4 枚目をめくり，$\boxed{1}$ なら $\overset{5}{\boxed{?}}\overset{6}{\boxed{?}}$，$\boxed{3}$ なら $\overset{6}{\boxed{?}}\overset{5}{\boxed{?}}$，$\boxed{4}$ なら $\overset{5}{\boxed{?}}\overset{6}{\boxed{?}}$ を出力とする．これは Koch の 2 入力 AND プロトコル [2] と同様のブール値の決め方である．

4. 結論

本稿では，Niemi と Renvall の 5 枚のカード列を並べ替えて 2 入力 AND プロトコル [9] を構成する方法と，Koyama らのコミットメントの値によってスワップ操作を行うことで 3 入力 AND プロトコル [4] と 3 入力多数決プロトコル [10] を構成する方法を一般化し，トランプカードを用いて汎用的な 3 入力論理関数を計算するプロトコルの構成方法を示した．具体的には，表 4 に示した全ての 3 入力論理関数を秘密計算できる．また，表 4 以外の 3 入力論理関数の場合には 2 枚のカードを追加し，c のコミットメントをコピーすることで，c のコミットメントによるスワップ操作を 2 回適用可能とする方法を提案し，これにより表 4 以外の一部の 3 入力論理関数が構成可能であることを示した．

提案したプロトコル構成法により，主要な関数には対応できている[*1]が，一部の特殊な 3 入力論理関数にはまだ対応できていないため，それらも網羅する汎用的な構成法を考案することが今後の課題である．

参考文献

[1] Koch, A.: Cryptographic Protocols from Physical Assumptions, PhD Thesis, Karlsruhe Institute of Technology (2019).
[2] Koch, A., Schrempp, M. and Kirsten, M.: Card-Based Cryptography Meets Formal Verification, *Advances in Cryptology – ASIACRYPT 2019* (Galbraith, S. D. and Moriai, S., eds.), Lecture Notes in Computer Science, Vol. 11921, Cham, Springer International Publishing, pp. 488–517 (2019).
[3] Koch, A., Walzer, S. and Härtel, K.: Card-Based Cryptographic Protocols Using a Minimal Number of Cards, *Advances in Cryptology – ASIACRYPT 2015* (Iwata, T. and Cheon, J. H., eds.), Lecture Notes in Computer Science, Vol. 9452, Berlin, Heidelberg, Springer, pp. 783–807 (2015).
[4] Koyama, H., Miyahara, D., Mizuki, T. and Sone, H.: A Secure Three-Input AND Protocol with a Standard Deck of Minimal Cards, *Computer Science – Theory and Applications* (Santhanam, R. and Musatov, D., eds.), Lecture Notes in Computer Science, Vol. 12730, Cham, Springer, pp. 242–256 (2021).
[5] Mizuki, T.: Efficient and Secure Multiparty Computations Using a Standard Deck of Playing Cards, *Cryptology and Network Security* (Foresti, S. and Persiano, G., eds.), Lecture Notes in Computer Science, Vol. 10052, Cham, Springer, pp. 484–499 (2016).
[6] Mizuki, T. and Shizuya, H.: A formalization of card-based cryptographic protocols via abstract machine, *International Journal of Information Security*, Vol. 13, No. 1, pp. 15–23 (online), DOI: 10.1007/s10207-013-0219-4 (2014).
[7] Mizuki, T. and Shizuya, H.: Computational Model of Card-Based Cryptographic Protocols and Its Applications, *IEICE Trans. Fundamentals*, Vol. E100.A, No. 1, pp. 3–11 (2017).
[8] Mizuki, T. and Sone, H.: Six-Card Secure AND and Four-Card Secure XOR, *Frontiers in Algorithmics* (Deng, X., Hopcroft, J. E. and Xue, J., eds.), Lecture Notes in Computer Science, Vol. 5598, Berlin, Heidelberg, Springer, pp. 358–369 (2009).
[9] Niemi, V. and Renvall, A.: Solitaire Zero-knowledge, *Fundam. Inf.*, Vol. 38, No. 1,2, pp. 181–188 (1999).
[10] 小山寛人, 宮原大輝, 水木敬明, 曽根秀昭：トランプカードを用いる 3 入力 AND と多数決プロトコル, 2022 年暗号と情報セキュリティシンポジウム（SCIS2021），No. 2F2-2, pp. 1–8 (2021).

[*1] 本稿では紙面の都合上示していないが，g や h が XOR や XNOR の場合にも 6 枚でプロトコルが構築できる．