Computer Security Symposium 2019
21 - 24 October 2019

非コミット型カードベースプロトコルと不正開示攻撃の定式化

高島 健[1]　宮原 大輝[1,2]　水木 敬明[3]　曽根 秀昭[3]

概要：物理的なカードを用いて秘密計算を実現するカードベース暗号においては，入力の値を漏らさないことが重要である．ほぼ全てのカードベースプロトコルは，参加するプレイヤーが semi-honest か covert であるという仮定の下，安全である．すなわち，攻撃者は自身の不正が他のプレイヤーに気付かれないと確信できる場合にのみ攻撃を行う．著者らは先行研究として，より能動的な攻撃を仮定し，攻撃者は攻撃が検知されることを厭わず，不正にカードをめくることを想定した．このような攻撃を不正開示攻撃と呼び，これに対し，最大 t 枚のカードをめくられても入力に関する情報が漏れないことを t-安全とする概念を導入した．本稿では，この先行研究を発展させ，t-安全の厳密な定義を与えるため，これまで定式化されていなかった非コミット型プロトコルについてその本質を抽出し，プロトコルの各状態が満たすべき条件を整理する．この定式化の下で，不正開示攻撃を扱うための枠組みを構築し，先行研究で概要を示した AND プロトコルが 1-安全であることを示す．

Formalizing Non-Committed Card-based Protocols and Maliciously Revealing Cards Attack

Ken Takashima[1]　Daiki Miyahara[1,2]　Takaaki Mizuki[3]　Hideaki Sone[3]

Abstract: In card-based protocols, which perform secure multi-party computations using a deck of cards, keeping inputs secret is important. Almost all the existing card-based protocols are secure under the assumption that players are semi-honest or covert; thus, an adversary is willing to attack only if she will not be caught. In our previous work, we considered a more active attack: An adversary will maliciously open the cards even if she will be caught. We called such an attack the maliciously revealing cards attack. Against such an active attack, we introduced the notion of t-secureness, meaning that information about inputs are not leaked even if at most t cards are revealed illegally. In this paper, in order to give a more rigorous definition of t-secureness, by extending our previous work, we investigate the essence of non-committed protocols that has not been formalized until now, and we specify what each status of such a non-committed protocol should satisfy. Based on our new formalization, we construct a framework to handle the maliciously revealing cards attack and we show that the AND protocol proposed in our previous work is surely 1-secure.

1. はじめに

入力を秘密にしたまま計算を行う**秘密計算**が広く知られている．秘密計算における入力は，守るべき情報であるため，物理的なカードを用いて秘密計算を実現する**カードベース暗号**においても，入力の値を漏らさないことは重要である．

例えば，1989 年に den Boer が提案した **five-card trick** [1] は，2 入力の論理積（AND）の秘密計算を物理的なカード組を用いて安全に実現できる．Alice と Bob がそれぞれ，秘密の 1 ビットの入力 $a, b \in \{0, 1\}$ を持っているとし，a, b を 2 色のカードを用いて次のように符号化する．

$$\boxed{\clubsuit}\boxed{\heartsuit} = 0 \text{ and } \boxed{\heartsuit}\boxed{\clubsuit} = 1 \qquad (1)$$

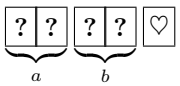
Alice と Bob は自身の入力 a, b を，2 枚のカード (♣, ♡) を使って表し，他のプレイヤーに見えないように裏向きに置く．このようなカード列 $\boxed{?}\boxed{?}$ を**コミットメント**と呼ぶ．

[1] 東北大学大学院情報科学研究科
Graduate School of Information Sciences, Tohoku University
[2] 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology
[3] 東北大学サイバーサイエンスセンター
Cyberscience Center, Tohoku University

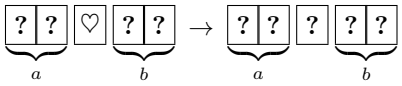
これら 4 枚のカードに 1 枚の ♡ を加え，計 5 枚のカードに対して入れ替え操作やシャッフル操作を行うことで，$a \wedge b$ を得ることができる．詳細な手順を次に示す．

(1) Alice は 2 枚のカードを用いて，自身の入力 a を表すコミットメントを作る．

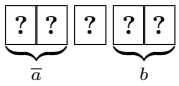
Bob も同様に b のコミットメントを作る．更に，1 枚の ♡ を加え，5 枚のカードの並びは初め，次のようになる．



(2) 右端の ♡ を真ん中に移動し，裏向きにする．

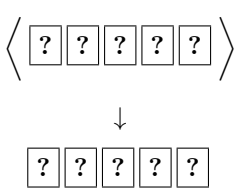


(3) 左から 1 枚目と 2 枚目のカードを入れ替える．(1) のように符号化しているため，この操作は a のコミットメントから a を否定した $\bar{a}$ を作る操作となる．



この時，$a = b = 1$ のときに限り，中央に ♡♡♡ が並ぶことに注意されたい．

(4) カード列に**ランダムカット**を適用する．これはカード列をランダムな量だけ巡回的にシフトするシャッフル操作で，$\langle\cdot\rangle$ で表す．



シフト量は $\{0, 1, 2, 3, 4\}$ 上で均一に分布し，その値を参加者は知らない[*1]．

(5) 5 枚のカードを全て表にする．

- もし（巡回的に）連続する赤のカード ♡♡♡ が現れれば，$a \wedge b = 1$ である．
- もし ♡♡♡ が現れなければ，$a \wedge b = 0$ である．

five-card trick を用いれば，入力 a, b を秘密にしたまま，$a \wedge b$ を得ることができる．このプロトコルを用いると，例えば，カップルは互いの気持ちを秘密に入力して，両者が慕い合っているかどうかを確認することができる．

このように，カード組を用いて，身近な場面で秘密計算を行うことができる．言うまでもなく秘密計算の際には入力の情報が漏れないことが大切である．

[*1] 人間の手で安全にランダムカットが実行できることが知られている [12].

five-card trick では，カードを裏向きにすることで a, b を秘匿し，シャッフル操作によって，($a \wedge b$ の値以外の) 入力の情報を失わせている．言い換えれば, five-card trick では，プレイヤーがプロトコルに従う限り安全であると言える．ほぼ全てのカードベースプロトコル [1], [2], [4], [6], [10], [11] は，参加するプレイヤーが semi-honest か covert であるという仮定の下，安全である．すなわち，攻撃者は自身の不正が他のプレイヤーに気付かれないと確信できる場合にのみ攻撃を行う．多くのカードベースプロトコルでは，他のプレイヤーや聴衆の目の前で全ての操作が行われるため，他者に気付かれずに不正を行うことは難しい．したがって，covert なプレイヤーでさえ，プロトコルに従うこととなる．

それに対し，著者らは先行研究 [13] において，より能動的な攻撃を仮定した．すなわち，攻撃者は攻撃が検知されることを厭わず，不正にカードをめくることを想定した．このような攻撃を**不正開示攻撃**と呼び，これに対し，最大 t 枚のカードをめくられても入力に関する情報が漏れないことを **t-安全**とする概念を導入した．また，秘密分散の考えを応用し，実際に 1-安全な AND プロトコルの概要を示した．

この先行研究 [13] では，紙面のページ数制限の都合もあり，カードベース暗号の計算モデルに則った t-安全の厳密な定義を行うことが出来ておらず，また，1-安全な AND プロトコルの全ての手順と本プロトコルが 1-安全であることを示せていなかった．

本稿では，先行研究を発展させ，t-安全の厳密な定義を与える．それにより，これまで定式化されていなかった非コミット型プロトコル（後述）についてその本質を抽出し，プロトコルの各状態が満たすべき条件を整理することが可能となった．この定式化の下で，不正開示攻撃を扱うための枠組みを構築し，先行研究で概要を示した AND プロトコルが 1-安全であることを示す．

five-card trick は非コミット型プロトコルの代表的なものである．本プロトコルでは，最後に全てのカードを表にして，計算結果を得たことを思い出そう．これに対し，計算結果をコミットメントで得るプロトコル [2], [10], [11] も多く存在し，このようなプロトコルを**コミット型プロトコル**と呼ぶ．本稿では，five-card trick（や [6]）のような複数のカードを表向きにしてそこから計算結果を得るプロトコルを**非コミット型プロトコル**と呼ぶ．

また，Mizuki と Shizuya [8] らは先行研究として，カードの表面に傷が付いている状況に対して，同様に秘密分散法のアイデアを使用していた．この先行研究では，コミット型プロトコルのみを対象としていたため，本稿の内容を応用して，非コミット型プロトコルに対してこのような攻撃に関する考察を行うのも面白いだろう．

本稿の構成は次の通りである．2 節では，カードベース

プロトコルの形式的な記述の仕方を示す．3 節では，非コミット型プロトコルを定式化する．4 節では，不正開示攻撃に対して，t-安全を定義する．5 節では，1-安全 AND プロトコルの手順とその安全性を示す．6 節では，結論を述べる．

2. カードベースプロトコル

本節では，カードベースプロトコルを形式的に記述する．

カードベースプロトコルの計算モデルは抽象機械を介して定式化されている [3], [4], [7], [9]．概して，プロトコルは，カード列とそれに適用する 3 種類の動作 $\mathsf{turn}, \mathsf{perm}, \mathsf{shuf}$ が連続したものから成る．

d 枚のカードから成るカード列を考えよう．カードをめくる動作 (turn, T) はカードの位置を表す集合 $T \subseteq \{1, 2, \ldots, d\}$ に対し，T が示す位置にある全てのカードをめくる操作を表す．次に，(perm, π) は d 次の対称群 S_d における置換 $\pi \in S_d$ に対し，π に従って，カード列の並び順を変える操作を表す．また，(shuf, Π) は置換の集合 $\Pi \subseteq S_d$ に対し，Π から均一な確率で選ばれた置換 π に従って，カード列を並び変える確率的操作を表す．ちょうど d 枚のカードを使用するプロトコルを d **カードプロトコル**と呼ぶ．

1 節で説明した five-card trick [1] を思い出そう．本プロトコルでは，裏面が $\boxed{?}$ である 2 種類のカード（$\boxed{\clubsuit}$ と $\boxed{\heartsuit}$）を使用した．なお，同じ種類のカード同士は区別できない．five-card trick は次のようなカード列から開始した．

$$\underbrace{\boxed{?}\boxed{?}}_{a}\underbrace{\boxed{?}\boxed{?}}_{b}\boxed{\heartsuit} \tag{2}$$

ステップ 2 は形式的には $(\mathsf{perm}, (3\ 4\ 5))$ の後，$(\mathsf{turn}, \{3\})$ を適用したと表される．ステップ 3 は $(\mathsf{perm}, (1\ 2))$ と表される．ステップ 4 ではランダムカットを適用した．これは $\mathsf{RC_5}= \{(1\ 2\ 3\ 4\ 5)^i \mid 1 \le i \le 5\}$ を用いて，$(\mathsf{shuf}, \mathsf{RC_5})$ と表される．ステップ 5 は $(\mathsf{turn}, \{1, 2, 3, 4, 5\})$ と表される．

プロトコルの正当性と安全性を議論するため，プロトコルの**状態**という概念を導入する [4]．例えば，(2) で表された five-card trick の初期状態は，次のように表される．

$$\begin{aligned}
&\clubsuit\heartsuit\clubsuit\heartsuit\heartsuit \ \ (p_{00}, 0, 0, 0)\\
&\clubsuit\heartsuit\heartsuit\clubsuit\heartsuit \ \ (0, p_{01}, 0, 0)\\
&\heartsuit\clubsuit\clubsuit\heartsuit\heartsuit \ \ (0, 0, p_{10}, 0)\\
&\heartsuit\clubsuit\heartsuit\clubsuit\heartsuit \ \ (0, 0, 0, p_{11})
\end{aligned}$$

p_{ij} は，各 $(i, j) \in \{0, 1\}^2$ に対して，入力 (a, b) が (i, j) と等しい確率を表している．言い換えると，$(p_{00}, p_{01}, p_{10}, p_{11})$ は，入力の集合 $\{0, 1\}^2$ 上の確率分布を表している．上に示した状態は，$\clubsuit\heartsuit\clubsuit\heartsuit\heartsuit$ のような**シンボル** $(\clubsuit, \heartsuit)$ の列と $(p_{00}, 0, 0, 0)$ のような**確率トレース** [5] のペアで表される 4 つの**エントリ**で構成されている．例えば，1 つ目のエントリは，$(a, b) = (0, 0)$ かつ $\clubsuit\heartsuit\clubsuit\heartsuit\heartsuit$ が生じる確率は p_{00} であり，$(a, b) \neq (0, 0)$ かつ $\clubsuit\heartsuit\clubsuit\heartsuit\heartsuit$ は生じないことを表している．

five-card trick の状態は図 1 のように，動作によって遷移して行き，最後には，turn によって，10 個の**葉**と呼ばれる状態が作られる．

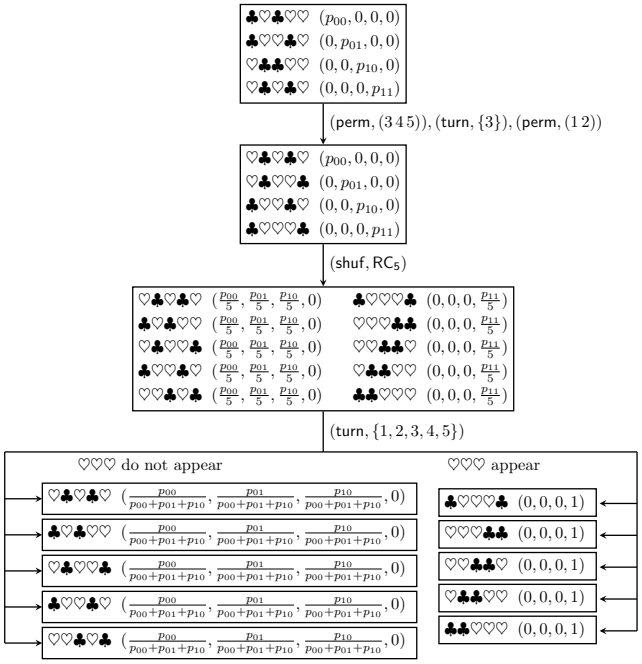


図 1: five-card trick の KWH-tree

図 1 のような木構造を用いたプロトコルの表記方法は，Koch と Walzer [3] によって提案された．本稿では KWH-tree に，Mizuki と Komano [5] によって提案された確率トレースを導入している．

図 1 において，最初 3 つの状態，すなわち葉ではない各状態では，全ての確率トレースの同じ要素の総和が $(p_{00}, p_{01}, p_{10}, p_{11})$ となっている．これにより，入力 a, b に関する情報が漏れていないことが保障される．また，葉の状態では 1 つのエントリのみ存在し，確率トレースの和は $(\frac{p_{00}}{p_{00}+p_{01}+p_{10}}, \frac{p_{01}}{p_{00}+p_{01}+p_{10}}, \frac{p_{10}}{p_{00}+p_{01}+p_{10}}, 0)$ または $(0, 0, 0, 1)$ である．このことは，$a \wedge b$ の値以外の情報が漏れていないことを意味する．

3. 非コミット型プロトコルの定式化

本稿では，これまで形式的に定式化されてこなかった非コミット型プロトコルに着目し，論理関数を計算する非コミット型プロトコルの定式化を行う．

そのためにまず，n 入力プロトコルを定義する．

定義 1　ある整数 $n \ge 2$ に対して $d \ge 2n$ とし，$\mathcal{P}$ を d カードプロトコルとする．もし $\mathcal{P}$ の初期状態が，次に表す 2^n 個のエントリで構成されているならば，$\mathcal{P}$ を n **入力プロトコル**と呼ぶ．

$$\underbrace{\overbrace{\clubsuit\heartsuit\clubsuit\heartsuit\clubsuit\heartsuit\cdots\clubsuit\heartsuit\clubsuit\heartsuit}^{2n\text{ シンボル}}}_{000\ldots00_2}\ \alpha\quad(p_0,0,0,\ldots,0,0)$$

$$\underbrace{\clubsuit\heartsuit\clubsuit\heartsuit\clubsuit\heartsuit\cdots\clubsuit\heartsuit\heartsuit\clubsuit}_{000\ldots01_2}\ \alpha\quad(0,p_1,0,\ldots,0,0)$$

$$\vdots$$

$$\underbrace{\heartsuit\clubsuit\heartsuit\clubsuit\heartsuit\clubsuit\cdots\heartsuit\clubsuit\heartsuit\clubsuit}_{111\ldots11_2}\ \alpha\quad(0,0,0,\ldots,0,p_{2^n-1})$$

α は長さ $d-2n$ の任意のシンボルの列を表す．また，$p_i\,(0\le i\le 2^n-1)$ は n ビットの入力が 2 進数表現の i に等しい時の確率を表す．更に，$(p_0,\ldots,p_{2^n-1})$ を**入力分布**と呼ぶ．

定義 1 で見られるように，我々は暗に $\{0,1\}^n$ と $\{0,1,\ldots,2^n-1\}$ の間の一対一写像を暗に仮定している．したがって本稿では，$b\in\{0,1\}^n$ と $(q_0,\ldots,q_{2^n-1})$ に対して，q_b と書いた場合，下付き文字 b を対応する 10 進数と見なす．

次に，状態に関するいくつかの性質を定義する．

定義 2　入力分布が $(p_0,\ldots,p_{2^n-1})$ である n 入力プロトコルを $\mathcal{P}$ とし，論理関数 $f:\{0,1\}^n\to\{0,1\}$ を考える．

- $\mathcal{P}$ の状態 S の確率トレースの和が $(p_0,\ldots,p_{2^n-1})$ と等しい時，S を **opaque 状態**と呼ぶ．
- 状態 S の確率トレースの和 $(q_0,\ldots,q_{2^n-1})$ が各 $b\in\{0,1\}^n$ に対して次を満たす時，S を **0 出力状態**と呼ぶ．

$$\begin{cases} q_b=\frac{p_b}{\sum_{i\in f^{-1}(0)}p_i} & \text{if } f(b)=0\\ q_b=0 & \text{if } f(b)=1\end{cases}$$

$f^{-1}(0)$ は f による 0 の原像である．

- 状態 S の確率トレースの和 $(q_0,\ldots,q_{2^n-1})$ が各 $b\in\{0,1\}^n$ に対して次を満たす時，S を **1 出力状態**と呼ぶ．

$$\begin{cases} q_b=0 & \text{if } f(b)=0\\ q_b=\frac{p_b}{\sum_{i\in f^{-1}(1)}p_i} & \text{if } f(b)=1\end{cases}$$

次に，非コミット型プロトコルを定義する．

定義 3　n 入力プロトコル $\mathcal{P}$ と論理関数 $f:\{0,1\}^n\to\{0,1\}$ を考える．次を満たす時，$\mathcal{P}$ を **f を計算する非コミット型プロトコル**と呼ぶ．

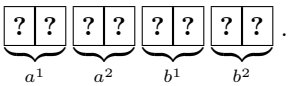
- 全ての葉の状態が 0 出力状態か 1 出力状態であり，それ以外の状態が全て opaque 状態である．
- KWH-tree の高さの期待値が有限の値である．

five-card trick が定義 3 を満たすことは容易に検証できる．

4. 不正開示攻撃に関する定式化

先に述べたように，本稿では，攻撃者がプロトコルを無視して不正にカードをめくるような能動攻撃を考える．

任意の（定義 1 で定義される）n 入力プロトコルはこのような不正開示攻撃に対して安全ではないことに注意しよう．なぜなら，攻撃者はプロトコル開始直後，入力の値を表すコミットメントを表にすることで，入力の情報を不正に得ることができる．本稿では，入力のコミットメントの値を隠すために，秘密分散法の考え方を利用する．例えば，Alice は秘密の入力 a のコミットメントをそのまま置くのではなく，$a=a^1\oplus a^2$ を満たす a^1 と a^2 のコミットメントを置く．ここで，Alice の秘密の入力 a は a^1 と a^2 にランダムに分散されている．Bob も同様に b^1 と b^2 のコミットメントを置く．



このような入力のカード列に対して，高々 1 枚のカードを不正に表にされたとしても，a と b に関する情報は一切漏れない．このアイデアを拡張し，次の定義 4 に示される，$(n,t+1)$ 入力プロトコルを導入する．なお，これ以降，$b\in\{0,1\}^n$ に対して，$b[i]$ は n ビット列 b の i 番目のビットを表す．

定義 4　整数 $n\ge 2,t\ge 1$ に対し，$d\ge 2n(t+1)$ とし，$\mathcal{P}$ を d カードプロトコルとする．$\mathcal{P}$ の初期状態が次に含まれる全てのエントリで構成されているならば，$\mathcal{P}$ を $(n,t+1)$ **入力プロトコル**と呼ぶ．

$$\bigcup_{b\in\{0,1\}^n}\Big\{(x_1^1\ldots x_1^{t+1}x_2^1\ldots x_2^{t+1}\ldots x_n^1\ldots x_n^{t+1}\alpha,\ (0,\ldots,0,\frac{p_b}{n2^t},0,\ldots,0))\ \Big|\ \bigoplus_{j=1}^{t+1}x_i^j=b[i],1\le i\le n\Big\}$$

ここで，$x_i^j\in\{0,1\}$ は $0=\clubsuit\heartsuit,1=\heartsuit\clubsuit$ という符号化に基づいたシンボルの列と見なし，α は長さ $d-2n(t+1)$ の任意のシンボルの列である．

次に，定義 5 に則って，定義 6 では t-安全を定義する．

定義 5　$\mathcal{P}$ を $(n,t+1)$ 入力プロトコル，$f:\{0,1\}^n\to\{0,1\}$ を論理関数とする．定義 2 と同様に opaque 状態，0 出力状態，1 出力状態を定義し，定義 3 と同様に，f を計算する $(n,t+1)$ 入力プロトコルを定義する．

定義 6　$\mathcal{P}$ を f を計算する $(n,t+1)$ 入力非コミット型プロトコルとする．$\mathcal{P}$ の各状態に任意の $(\mathsf{turn},T),|T|\le t$ を適用した時に生成される状態が opaque 状態，0 出力状態，1 出力状態のいずれかであるならば，$\mathcal{P}$ は t-安全であるという．

5. 1-安全 AND プロトコル

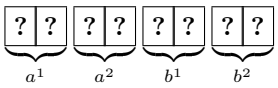
本節では，先行研究 [13] で概要を示した 1-安全 AND プロトコルの完全な手順を示す．5.1 節では改めて，本プロトコルの概要を示す．5.2 節，5.3 節，5.4 節では，フェーズ 1，フェーズ 2，フェーズ 3 の詳細をそれぞれ示す．

5.1 1-安全 AND プロトコルの概要

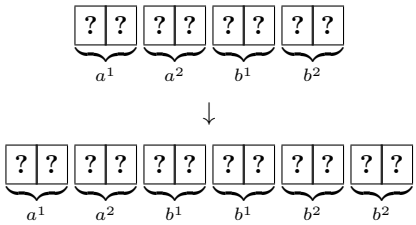
2 入力の 1-安全 AND プロトコルを構成するために，$(2,2)$-入力プロトコルを使用する．

したがって，Alice と Bob は $a = a^1 \oplus a^2, b = b^1 \oplus b^2$ を満たす $a^1, a^2, b^1, b^2 \in \{0,1\}$ を入力する．この時，$a \wedge b = (a^1 \wedge b^1) \oplus (a^1 \wedge b^2) \oplus (a^2 \wedge b^1) \oplus (a^2 \wedge b^2)$ となる．これに基づき，以下のような手順で計算を行う．

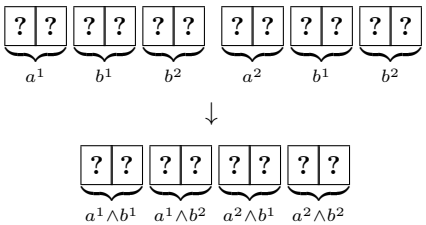
セットアップ 定義 4 を満たすように，Alice は a^1, a^2 を表す 2 つのコミットメントを，Bob は b^1, b^2 を表すコミットメントを作る．



フェーズ 1 既存の COPY プロトコル [10] を用いて，b^1 と b^2 のコミットメントをそれぞれ 2 つに複製する [10].



フェーズ 2 既存の AND プロトコル [8] を用いて，a^1, b^1, b^2 を表すコミットメントから $a^1 \wedge b^1, a^1 \wedge b^2$ を表すコミットメントを作る．同様に，a^2, b^1, b^2 のコミットメントから $a^2 \wedge b^1, a^2 \wedge b^2$ のコミットメントを作る．

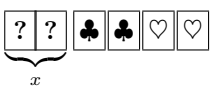


フェーズ 3 $(a^1 \wedge b^1) \oplus (a^1 \wedge b^2) \oplus (a^2 \wedge b^1) \oplus (a^2 \wedge b^2)$ を計算する．

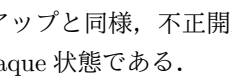
初めに，セットアップの安全性を確認しよう．入力 $(a^1, a^2, b^1, b^2) \in \{0,1\}^4$ には 16 通りの可能性があり，初期状態は表 1 の最初の 4 列と最後の 1 列に表されている．ここで，もし $(\mathsf{turn}, \{i\})$ が適用されても，4 ビットのうち，最大 1 ビットしか明らかにならないことに注意されたい．初期状態に $(\mathsf{turn}, \{i\})$ を適用した後にできる状態が必ず opaque 状態になることは容易に検証できる．

5.2 フェーズ 1

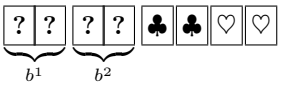
フェーズ 1 では，既存の COPY プロトコル [10] を用いて，b^1 と b^2 のコミットメントを複製する．



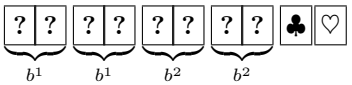
↓



COPY プロトコルを 2 回行うことで，以下のカード列を得る．



↓



既存 COPY プロトコル [10] の KWH-tree を図 2 に示した．これにより，プロトコルの手順と，その正当性と安全性を確認できる．

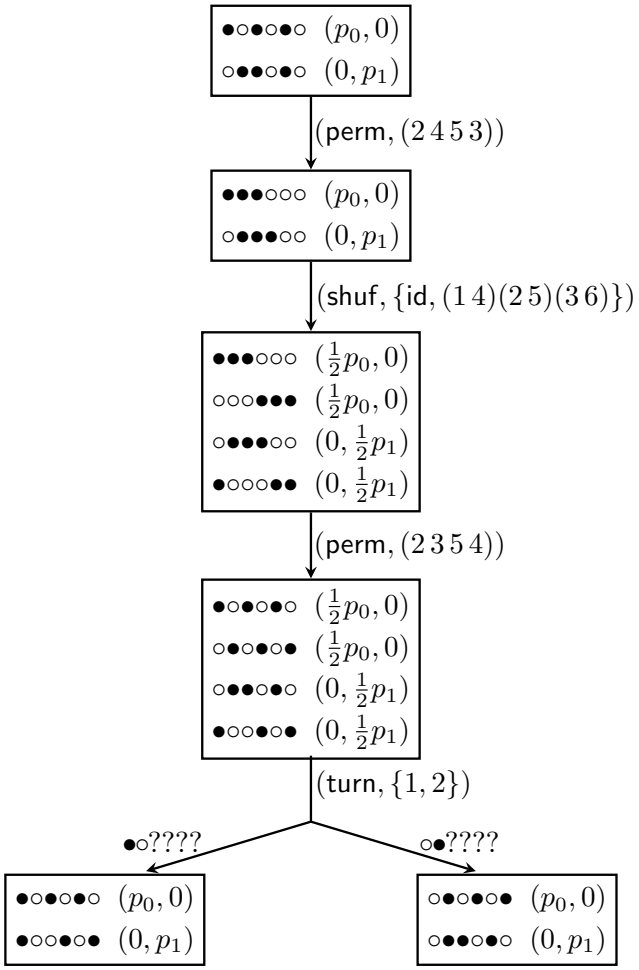


図 2: 既存の COPY プロトコル [10] の KWH-tree（♣ = ●, ♡ = ○ と表記）

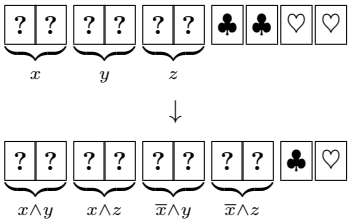
フェーズ 1 の実行中，$(\mathsf{turn}, \{i\})$ によってカードが表にされても，b^1 と b^2 のうち，最大 1 ビットの情報しか漏れないため，セットアップと同様，不正開示攻撃によって生成される状態は opaque 状態である．

表 1: 1-安全 AND プロトコルの状態を得るための真理値表

a^1	a^2	b^1	b^2	$a^1 \wedge b^1$	$a^1 \wedge b^2$	$a^2 \wedge b^1$	$a^2 \wedge b^2$	$\overline{a^1} \wedge b^1$	$\overline{a^1} \wedge b^2$	$\overline{a^2} \wedge b^1$	$\overline{a^2} \wedge b^2$	Prob. trace
0	0	0	0	0	0	0	0	0	0	0	0	$(p_{00}/4, 0, 0, 0)$
0	0	1	1	0	0	0	0	1	1	1	1	$(p_{00}/4, 0, 0, 0)$
1	1	0	0	0	0	0	0	0	0	0	0	$(p_{00}/4, 0, 0, 0)$
1	1	1	1	1	1	1	1	0	0	0	0	$(p_{00}/4, 0, 0, 0)$
0	0	0	1	0	0	0	0	0	1	0	1	$(0, p_{01}/4, 0, 0)$
0	0	1	0	0	0	0	0	1	0	1	0	$(0, p_{01}/4, 0, 0)$
1	1	0	1	0	1	0	1	0	0	0	0	$(0, p_{01}/4, 0, 0)$
1	1	1	0	1	0	1	0	0	0	0	0	$(0, p_{01}/4, 0, 0)$
0	1	0	0	0	0	0	0	0	0	0	0	$(0, 0, p_{10}/4, 0)$
0	1	1	1	0	0	1	1	1	1	0	0	$(0, 0, p_{10}/4, 0)$
1	0	0	0	0	0	0	0	0	0	0	0	$(0, 0, p_{10}/4, 0)$
1	0	1	1	1	1	0	0	0	0	1	1	$(0, 0, p_{10}/4, 0)$
0	1	0	1	0	0	0	1	0	1	0	0	$(0, 0, 0, p_{11}/4)$
0	1	1	0	0	0	1	0	1	0	0	0	$(0, 0, 0, p_{11}/4)$
1	0	0	1	0	1	0	0	0	0	0	1	$(0, 0, 0, p_{11}/4)$
1	0	1	0	1	0	0	0	0	0	1	0	$(0, 0, 0, p_{11}/4)$

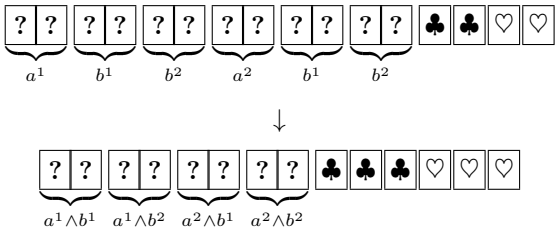
5.3 フェーズ 2

フェーズ 2 では既存の AND プロトコルを用いる [8].



$\overline{x} \wedge y$ と $\overline{x} \wedge z$ のコミットメントには，シャッフルを適用した後で表にする．

2 回 AND プロトコル [8] を適用することで，次のカード列を得る．



この既存 AND プロトコル [8] の KWH-tree を図 3 に示した．ただし，入力分布は本稿の状況に合わせている．これにより，プロトコルの手順と，その正当性と安全性を確認できる．

フェーズ 2 の実行中，$(\mathsf{turn}, \{i\})$ によって明らかになるのは，a^1, a^2, b^1, b^2, $a^1 \wedge b^1, a^1 \wedge b^2, a^2 \wedge b^1, a^2 \wedge b^2$, $\overline{a^1} \wedge b^1, \overline{a^1} \wedge b^2, \overline{a^2} \wedge b^1, \overline{a^2} \wedge b^2$ の中の最大 1 ビットのみである．表 1 を用いて，不正開示攻撃によって生じる状態が全て opaque 状態であることを確認できる．

5.4 フェーズ 3

フェーズ 3 では，$a^1 \wedge b^1, a^1 \wedge b^2, a^2 \wedge b^1, a^2 \wedge b^2$ の 4 つのコミットメントから，$(a^1 \wedge b^1) \oplus (a^1 \wedge b^2) \oplus (a^2 \wedge b^1) \oplus (a^2 \wedge b^2)$ を計算する．この計算を行う “4 ビット XOR サブプロトコル” は以下の手順で行われる．

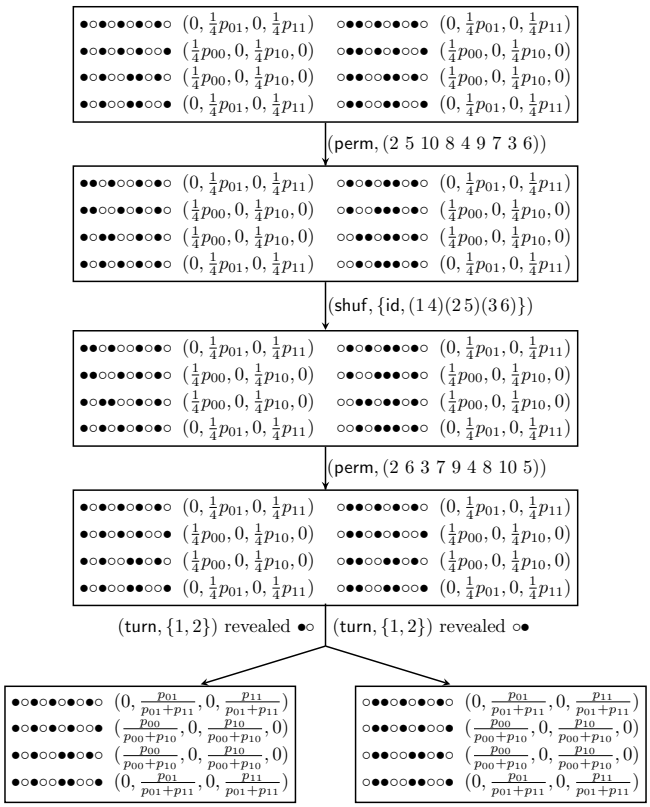
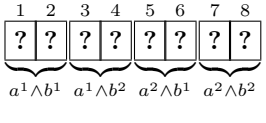
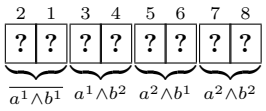


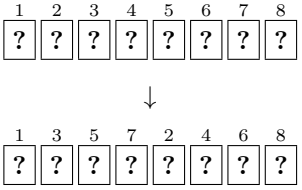
図 3: 既存の AND プロトコル [8] の KWH-tree

(1) $(\mathsf{perm}, (1\ 2))$ によって，$a^1 \wedge b^1$ のコミットメントに NOT を適用する．

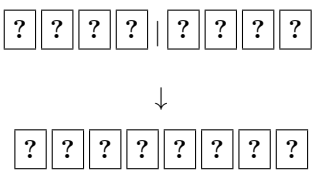




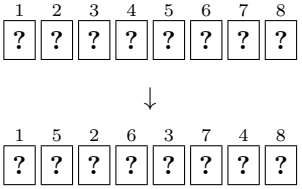
(2) $(\mathsf{perm}, (2\ 5\ 3)(4\ 6\ 7))$ を適用し，カード列を並び替える．



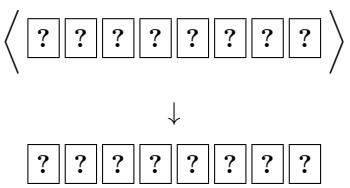
(3) $[\cdot|\cdot]$ と表されるランダム二等分割カット [10] を適用する．これにより，$(\mathsf{shuf}, \{\mathsf{id}, (1\,5)(2\,6)(3\,7)(4\,8)\})$ が適用される．



(4) $(\mathsf{perm}, (2\,3\,5)(4\,7\,6))$ により，ステップ 2 の逆置換を適用する．



(5) $(\mathsf{shuf}, \mathsf{RC}_8)$ により，ランダムカットを適用する．RC_5 と同様に RC_8 を定義する．



(6) $(\mathsf{turn}, \{1,2,3,4,5,6,7,8\})$ により，全てのカードを表にする．連続する同じ色のカードを区切って，コミットメントを識別する．例えば，♣♣♡♣♡♣♡♡ というカード列は，♣♣ と ♡♡ の間に区切りを入れることで，♣|♣♡♣♡♣♡|♡ のように表せ，巡回的に並び替えて，♣♡|♣♡|♣♡|♡♣ のようにコミットメントを識別できる．

- 奇数個の 1 のコミットメントが含まれているならば，$a \wedge b = 0$ である．
- 偶数個の 1 のコミットメントが含まれているならば，$a \wedge b = 1$ である．

図 4 に 4 ビット XOR サブプロトコルの KWH-tree を示した．five-card trick と同様に，本プロトコルの正当性と安全性を検証できる．

フェーズ 3 の実行中，$(\mathsf{turn}, \{i\})$ によって明らかになるのは，$a^1 \wedge b^1, a^1 \wedge b^2, a^2 \wedge b^1, a^2 \wedge b^2$ の中の最大 1 ビットのみである．表 1 を用いて，葉以外の状態に不正開示攻撃が行われた時に生じる状態が全て opaque 状態であることを確認できる．また，葉の状態に不正開示攻撃が行われた時に生じる状態が 0 出力状態または 1 出力状態であることは明らかである．

したがって，セットアップからフェーズ 3 までの間，常に定義 6 を満たしていたため，本プロトコルは 1-安全であることが示された．

6. おわりに

本稿では，先行研究を発展させ，非コミット型プロトコルの定式化を行った．更に，不正開示攻撃に対して，t-安全を定義し，1-安全プロトコルの手順を示した．また，本プロトコルが 1-安全であることを示した．

参考文献

[1] den Boer, B.: More efficient match-making and satisfiability the five card trick. In: Quisquater, J.J., Vandewalle, J. (eds.) Advances in Cryptology — EUROCRYPT '89. Lecture Notes in Computer Science, vol. 434, pp. 208–217. Springer, Berlin, Heidelberg (1990)

[2] Ishikawa, R., Chida, E., Mizuki, T.: Efficient card-based protocols for generating a hidden random permutation without fixed points. In: Calude, C.S., Dinneen, M.J. (eds.) Unconventional Computation and Natural Computation. Lecture Notes in Computer Science, vol. 9252, pp. 215–226. Springer, Cham (2015)

[3] Koch, A., Walzer, S.: Foundations for actively secure card-based cryptography. Cryptology ePrint Archive, Report 2017/423 (2017), https://eprint.iacr.org/2017/423

[4] Koch, A., Walzer, S., Härtel, K.: Card-based cryptographic protocols using a minimal number of cards. In: Iwata, T., Cheon, J.H. (eds.) Advances in Cryptology – ASIACRYPT 2015. Lecture Notes in Computer Science, vol. 9452, pp. 783–807. Springer, Berlin, Heidelberg (2015)

[5] Mizuki, T., Komano, Y.: Analysis of information leakage due to operative errors in card-based protocols. In: Iliopoulos, C., Sung, W., Leong, H.W. (eds.) Combinatorial Algorithms. Lecture Notes in Computer Science, vol. 10979, pp. 250–262. Springer, Cham (2018)

[6] Mizuki, T., Kumamoto, M., Sone, H.: The five-card trick can be done with four cards. In: Wang, X., Sako, K. (eds.) Advances in Cryptology – ASIACRYPT 2012. Lecture Notes in Computer Science, vol. 7658, pp. 598–606. Springer, Berlin, Heidelberg (2012)

[7] Mizuki, T., Shizuya, H.: A formalization of card-based cryptographic protocols via abstract machine. International Journal of Information Security 13(1), 15–23 (2014)

[8] Mizuki, T., Shizuya, H.: Practical card-based cryptography. In: Ferro, A., Luccio, F., Widmayer, P. (eds.) Fun with Algorithms. Lecture Notes in Computer Science, vol. 8496, pp. 313–324. Springer, Cham (2014)

[9] Mizuki, T., Shizuya, H.: Computational model of card-based cryptographic protocols and its applications. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E100.A(1), 3–11

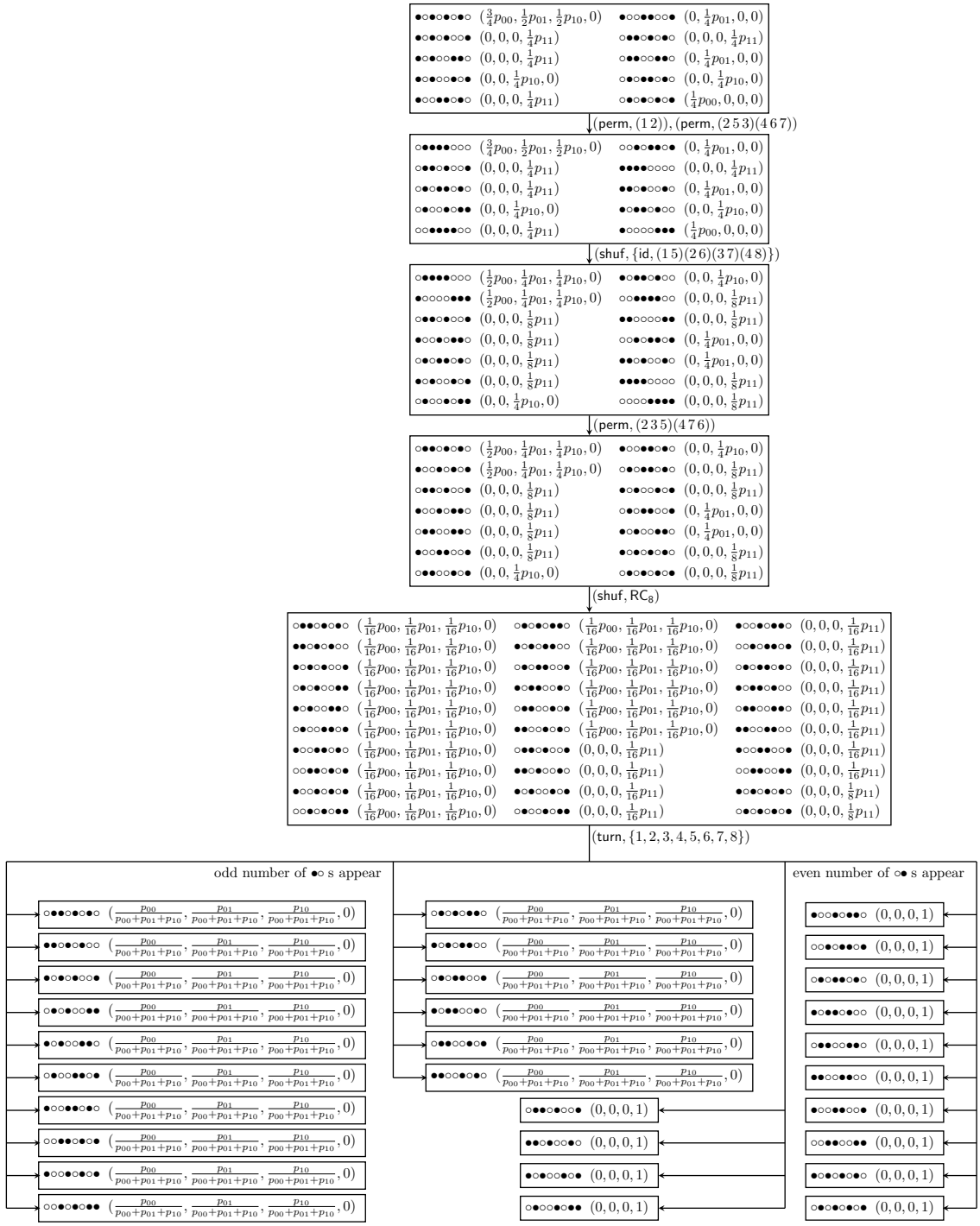


図 4: 4 ビット XOR サブプロトコルの KWH-tree

(2017)

[10] Mizuki, T., Sone, H.: Six-card secure AND and four-card secure XOR. In: Deng, X., Hopcroft, J.E., Xue, J. (eds.) Frontiers in Algorithmics. Lecture Notes in Computer Science, vol. 5598, pp. 358–369. Springer, Berlin, Heidelberg (2009)

[11] Niemi, V., Renvall, A.: Secure multiparty computations without computers. Theoretical Computer Science 191(1–2), 173–183 (1998)

[12] Ueda, I., Nishimura, A., Hayashi, Y., Mizuki, T., Sone, H.: How to implement a random bisection cut. In: Martín-Vide, C., Mizuki, T., Vega-Rodríguez, M.A. (eds.) Theory and Practice of Natural Computing. Lecture Notes in Computer Science, vol. 10071, pp. 58–69. Springer, Cham (2016)

[13] 高島健, 宮原大輝, 水木敬明, 曽根秀昭: 不正開示攻撃を考慮したカードベース AND プロトコル. 2019 年電子情報通信学会ソサイエティ大会 (2019)