ボールと袋を用いた秘密計算

宮原 大輝[1] 駒野 雄一[2] 水木 敬明[3] 曽根 秀昭[3]

概要：2 人の暗号学者が，彼らの秘密の入力から論理積（AND）の秘密計算を行いたいという状況を考えよう．彼らはキッチンにいるとし，そこにはカレーが入った 3 つの鍋といくつかの具材が存在している．興味深いことに，彼らは入力にしたがって具材を鍋に秘密に投入することで，カレーの味から論理積の値だけを得ることができる．秘密計算のために料理を行うのは手間がかかるため，本稿では具材と（カレーが入った）鍋をそれぞれ，ボールと袋に置き換える．これらは扱いやすい上に，高等学校教育における確率の分野にも登場するほど身近な道具である．したがって，本稿において取り組む問題は次のように一般化される：ボールと袋を用いることで秘密計算を達成できるのであろうか．この問題はコンピュータサイエンス（CS）アンプラグド— コンピュータを用いることなく情報科学分野を教えるための学習法— の分野であると捉えることもできる．既存の CS アンプラグドな秘密計算法として，カードベースな方式やコインベースな方式が与えられているが，本稿ではボールと袋の物理的な性質に基づく方式を与える．すなわち，ボールが袋に入った途端に順番が（袋の中で）ランダムになる性質を秘密計算に応用する．本稿は，秘密計算の文脈においてボールと袋を用いる枠組みを与える初めてのものであり，容易に実行可能な論理積の秘密計算法を提案する．生徒・学生にとって，我々のボールベースな秘密計算方式が身近なものであり，暗号や情報セキュリティ分野を勉強するきっかけとなることを期待している．

Secure Multiparty Computations Using Balls and Bags

DAIKI MIYAHARA[1] YUICHI KOMANO[2] TAKAAKI MIZUKI[3] HIDEAKI SONE[3]

1. 初めに

3 人の暗号学者が今まさにカレーを料理しようとしている．彼らはそれぞれ，カレーに合う具材として人参と玉ねぎを持参してきた．彼らはその具材を自費で準備してきたのかもしれない．もしくは，彼らの中には，アメリカ国家安全保障局（NSA）から援助を受けたものがいるのかもしれない．3 人の暗号学者は，NSA と関りを持つことは個人の自由だと考えている．しかし，口に入れるものが NSA から援助を受けたものなのかどうかについてははっきりさせたい．キッチンには，具材（すなわち人参と玉ねぎ）とカレーの入った鍋がある．そこで彼らは，彼ら全員が自費で準備したのかどうかを明らかにするために，具材と鍋を用いた AND の秘密計算を行うことにした．本稿では，これを**料理する暗号学者の問題**（Cooking Cryptographers Problem）と呼ぶことにする．これは，**食事する暗号学者の問題**（Dining Cryptographers Problem）[1] から取って名付けた．

1.1 料理する暗号学者

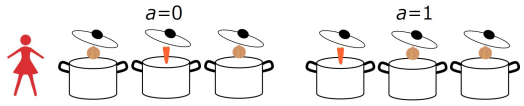
簡単のため，2 人の暗号学者の場合を考えよう．アリスとボブがそれぞれ，秘密の入力 $a, b \in \{0, 1\}$ を持っていると仮定する（NSA の援助を受けている場合は 1 で，そうでない場合は 0）．目的は，2 入力の論理積関数 $f(a, b) = a \wedge b$ を，出力以外の情報を一切漏らさずに出力を得ることである．ここはキッチンであることを思い出そう．2 つの人参と 4 つの玉ねぎ，そして 3 つの鍋を用いて，安全な論理積計算を次のように行うことができる．

(1) カレーの入った 3 つの鍋がキッチンのカウンターにある．アリスとボブはそれぞれ，1 つの人参と 2 つの玉ねぎを手に持っている．

(2) $a = 0$ の場合，アリスは人参を 2 番目の鍋に，玉ねぎを残りの鍋に秘密に（どの鍋に人参を入れたのかをボ

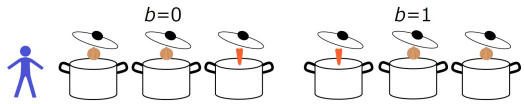
[1] 東北大学大学院情報科学研究科
[2] 株式会社東芝
[3] 東北大学サイバーサイエンスセンター

ブに見せないように）入れる．$a = 1$ の場合，人参を 1 番目の鍋に入れ，玉ねぎを残りの鍋に入れる．

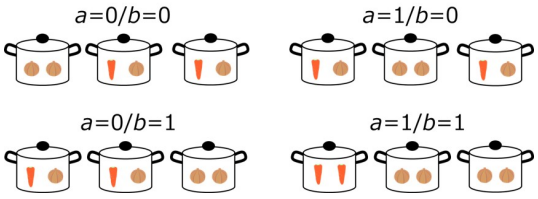


具材は鍋の底に沈むため，誰も鍋に入った具材を見ることはできない．

(3) $b = 0$ の場合，ボブは人参を 3 番目の鍋に，玉ねぎを残りの鍋に秘密に入れる．$b = 1$ の場合，ボブは人参を 1 番目の鍋に入れ，玉ねぎを残りの鍋に入れる．



同じ鍋に 2 つの人参が入るのは $a = b = 1$ の場合だけであることに注意しよう．



(4) アリスとボブは 3 つの鍋をシャッフルし，その 3 つの鍋の順番が誰にも分からないようにする．

(5) 3 つの鍋に入ったカレーを煮た後，彼らはカレーを食べる．もし人参だけが入ったカレーがあった場合，$f(a,b) = a \wedge b = 1$ である．そうでなければ，$a \wedge b = 0$ である．

1.2 貢献

本稿では，上述した 2 者間のプロトコルを，一般的な仮定の下で定式化する．秘密計算を行う度にカレーを料理することは現実的でないため，具材と（カレーの入った）鍋をそれぞれ，色付きのボール（赤と白）と袋に置き換える．ボールと袋は準備しやすいだけではなく，高等教育において確率を学ぶ際に用いられるほど身近な道具である．したがって，ボールと袋を用いる方式はより人に優しいと言える．

袋に入ったボールの色は外から見えないものとする．本稿で我々が取り組む問題は次のように一般化される．**ボールと袋を用いることで秘密計算を達成できるのであろうか．** 1.4 節で見るように，身近な道具を用いた秘密計算はこれまでにいくつか考案されている．例えば，物理的なカード組を用いた方式 [1–3] が挙げられる．本稿では，ボールと袋の物理的な性質に基づく秘密計算方式を初めて与える．すなわち，ボールが袋に入った途端に順番が（袋の中で）ランダムになる性質を秘密計算に応用する．用いる操作は単純なもの（すなわち，ボールを袋に入れる，袋の順番をシャッフルする，袋からボールを取り出す）であるため，容易に実行可能である．

具体的には，2 入力以上の論理積の秘密計算を行うシンプルなプロトコルを構成する．さらに，ボールと袋を用いた秘密計算プロトコルとその安全性の形式的な定義を与える．プロトコルの完全性と安全性を容易に確かめる方法として，本稿ではプロトコルの確率トレースを示したダイアグラム [4] を導入する．

1.3 構成

本稿の構成は次の通りである．2 節では，ボールと袋を用いたプロトコルの形式的な定義を与える．3 節では，プロトコルのステータス遷移を示すダイアグラムを構成する．4 節では，1.1 節で示した 2 入力の AND プロトコルの形式的な記述を与え，そのプロトコルのダイアグラムを示す．5 節では，上述の AND プロトコルを拡張し，多入力の AND プロトコルを構成する．6 節では，本稿で提案したプロトコルの効率性や open problem を議論する．7 節で本稿をまとめる．

1.4 関連研究

物理的な道具を用いることで，秘密計算 [5]，ゼロ知識証明 [6]，否認可能な投票 [7]，及び視覚復号型秘密分散 [8] のような暗号プロトコルを構成できる．これらは**コンピュータを用いない暗号**と呼ばれ，代数学のような数学知識を用いずに原理を理解できることが魅力的である．したがって，教材としてそれらを用いることができる．

上で見たように，本稿では秘密計算を扱う．Yao [9] が秘密計算の概念を 1982 年に提案して以来，秘密分散や準同型暗号を用いた（コンピュータベースの）秘密計算プロトコルが提案されてきた．2016 年には Araki ら [10] によって効率的な手法が提案されている．一方，日常的な道具だけを用いた**コンピュータを用いない秘密計算**も同様に提案されてきている．例えば，カード組 [1–3]，PEZ ディスペンサ [11]，開封防止シール付きの封筒 [12]，視覚復号型秘密分散シート [13]，及びコイン [4] を用いた方式が提案されている．

本稿が提案する方式は，*CS アンプラグド*[*1] の分野であると言える．CS アンプラグドとは，ニュージーランド発祥のものであり，コンピュータを用いることなく情報科学分野を教えるための学習法を集めたものである．そのような学習法の 1 つとして知られているペルシャ人のコインフリップ [14] と同様に，ボールと袋を用いた秘密計算が生徒・学生にとって暗号を学ぶきっかけとなることを期待している．

ボールと袋を用いた秘密計算方式は，袋を壺と捉えることで，**壺問題**の変種であると言うこともできる．壺問題とは，確率と統計における思考実験の 1 つであり，色付きの

[*1] https://csunplugged.org/

いくつかのボールが入った壺から 1 つまたは複数のボールを取り出す．そこから，ある色または別の色を取り出す確率，もしくは他の特性を決定することが壺問題の目的である．壺問題の例として二項分布等が知られているが，本稿によって，秘密計算も壺問題の例として加わることになる．

2. ボールと袋を用いる方式の定式化

本節では，ボールと袋を用いた秘密計算の定式化を行う．

2.1 表記

1.1 節で提案した AND プロトコルを思い出そう．以降では，具材と鍋をそれぞれ，ボールと袋に置き換える．すなわち，図 1 に示す通り，2 つの赤のボール，4 つの白のボール，及び 3 つの袋を用いる．

● と ○ をそれぞれ，赤のボールと白のボールを表すものとする．図 1 を見ると，2 種類の（ボールの）多重集合，すなわち袋とトレイを定義する必要がある．$\{\!\{\cdot\}\!\}$ を**不可視多重集合**とする．これは，ボールを入れる袋を表す．また，$[\cdot]$ を**可視多重集合**とする．これは，いくつかのボールを保管する領域（トレイ）を表し，プロトコルの実行において，初期状態のボールの集合を表したり，対応する袋から取り出したボールを保存するためのものである．

これらの記号を用いると，1.1 節のプロトコルの最初では，3 つの空の袋 $\{\!\{\}\!\}, \{\!\{\}\!\}, \{\!\{\}\!\}$ と，1 つのトレイ $[\bullet, \bullet, \circ, \circ, \circ, \circ]$ がテーブル上にあると表現できる．これを次のように組として表すことにしよう．

$$([\bullet, \bullet, \circ, \circ, \circ, \circ]; \{\!\{\}\!\}, \{\!\{\}\!\}, \{\!\{\}\!\}; [], [], []). \tag{1}$$

このような組を**コンフィグレーション**と呼ぶことにする．コンフィグレーションの最初の成分（最初のセミコロンの前）は，プロトコル中で利用可能なボールを表す．2 番目の成分（最初のセミコロンと次のセミコロンの間）は，袋の列を表す．最後の成分は，袋から取り出されたボールを保存するトレイの列を表す．

これらの記号を用いて 2 者間プロトコルを再度見ていこう．ステップ 2 では，アリスは赤 1 つと白 2 つのボールを持ち，すなわち $[\bullet, \circ, \circ]$ のボールを持ち，a の値に応じて 3 つの袋に入れる．この時，どの色のボールがどの袋に入るかは外から見えないように，手のひらを閉じながら袋に入れる．その結果，上のコンフィグレーションは次のように変化する．

$$([\bullet, \bullet, \circ, \circ, \circ, \circ]; \{\!\{\}\!\}, \{\!\{\}\!\}, \{\!\{\}\!\}; [], [], []) \rightarrow \begin{cases} ([\bullet, \circ, \circ]; \{\!\{\circ\}\!\}, \{\!\{\bullet\}\!\}, \{\!\{\circ\}\!\}; [], [], []), & \text{if } a = 0, \\ ([\bullet, \circ, \circ]; \{\!\{\bullet\}\!\}, \{\!\{\circ\}\!\}, \{\!\{\circ\}\!\}; [], [], []), & \text{if } a = 1. \end{cases} \tag{2}$$

ステップ 3 でボブは，同様に b の値に応じてボールを入れる．このとき，4 つの可能性がある．

$$\begin{aligned} &([]; \{\!\{\circ, \circ\}\!\}, \{\!\{\bullet, \circ\}\!\}, \{\!\{\bullet, \circ\}\!\}; [], [], []), && \text{if } (a, b) = (0, 0), \\ &([]; \{\!\{\bullet, \circ\}\!\}, \{\!\{\bullet, \circ\}\!\}, \{\!\{\circ, \circ\}\!\}; [], [], []), && \text{if } (a, b) = (0, 1), \\ &([]; \{\!\{\bullet, \circ\}\!\}, \{\!\{\circ, \circ\}\!\}, \{\!\{\bullet, \circ\}\!\}; [], [], []), && \text{if } (a, b) = (1, 0), \\ &([]; \{\!\{\bullet, \bullet\}\!\}, \{\!\{\circ, \circ\}\!\}, \{\!\{\circ, \circ\}\!\}; [], [], []), && \text{if } (a, b) = (1, 1), \end{aligned}$$

本稿ではこの状態を次のように表す．

$$\begin{aligned} &([]; \{\!\{\circ, \circ\}\!\}, \{\!\{\bullet, \circ\}\!\}, \{\!\{\bullet, \circ\}\!\}; [], [], []) && (p_{00}, 0, 0, 0), \\ &([]; \{\!\{\bullet, \circ\}\!\}, \{\!\{\bullet, \circ\}\!\}, \{\!\{\circ, \circ\}\!\}; [], [], []) && (0, p_{01}, 0, 0), \\ &([]; \{\!\{\bullet, \circ\}\!\}, \{\!\{\circ, \circ\}\!\}, \{\!\{\bullet, \circ\}\!\}; [], [], []) && (0, 0, p_{10}, 0), \\ &([]; \{\!\{\bullet, \bullet\}\!\}, \{\!\{\circ, \circ\}\!\}, \{\!\{\circ, \circ\}\!\}; [], [], []) && (0, 0, 0, p_{11}), \end{aligned} \tag{3}$$

ここで p_{ij} は，入力 (a, b) が (i, j) である確率を表し，右側にある 4 項組 $(q_{00}, q_{01}, q_{10}, q_{11})$ は，現在のコンフィグレーションが左側のものでかつ対応する入力となる条件付き確率を意味する．

ステップ 4 では，3 つの袋をシャッフルする．その結果，コンフィグレーションは次のようになる．

$$\begin{aligned} &([]; \{\!\{\circ, \circ\}\!\}, \{\!\{\bullet, \circ\}\!\}, \{\!\{\bullet, \circ\}\!\}; [], [], []) && (\tfrac{p_{00}}{3}, \tfrac{p_{01}}{3}, \tfrac{p_{10}}{3}, 0), \\ &([]; \{\!\{\bullet, \circ\}\!\}, \{\!\{\bullet, \circ\}\!\}, \{\!\{\circ, \circ\}\!\}; [], [], []) && (\tfrac{p_{00}}{3}, \tfrac{p_{01}}{3}, \tfrac{p_{10}}{3}, 0), \\ &([]; \{\!\{\bullet, \circ\}\!\}, \{\!\{\circ, \circ\}\!\}, \{\!\{\bullet, \circ\}\!\}; [], [], []) && (\tfrac{p_{00}}{3}, \tfrac{p_{01}}{3}, \tfrac{p_{10}}{3}, 0), \\ &([]; \{\!\{\bullet, \bullet\}\!\}, \{\!\{\circ, \circ\}\!\}, \{\!\{\circ, \circ\}\!\}; [], [], []) && (0, 0, 0, \tfrac{p_{11}}{3}), \\ &([]; \{\!\{\circ, \circ\}\!\}, \{\!\{\bullet, \bullet\}\!\}, \{\!\{\circ, \circ\}\!\}; [], [], []) && (0, 0, 0, \tfrac{p_{11}}{3}), \\ &([]; \{\!\{\circ, \circ\}\!\}, \{\!\{\circ, \circ\}\!\}, \{\!\{\bullet, \bullet\}\!\}; [], [], []) && (0, 0, 0, \tfrac{p_{11}}{3}). \end{aligned} \tag{4}$$

最後に，3 つの袋から全てのボールを取り出す．この取り出されたボールは，コンフィグレーションの 3 つ目の成分（可視多重集合の列）で表される．

$$\begin{aligned} &([], \{\!\{\}\!\}, \{\!\{\}\!\}, \{\!\{\}\!\}; [\circ, \circ], [\bullet, \circ], [\bullet, \circ]) && \left(\tfrac{p_{00}}{p_0}, \tfrac{p_{01}}{p_0}, \tfrac{p_{10}}{p_0}, 0\right), \\ &([], \{\!\{\}\!\}, \{\!\{\}\!\}, \{\!\{\}\!\}; [\bullet, \circ], [\bullet, \circ], [\circ, \circ]) && \left(\tfrac{p_{00}}{p_0}, \tfrac{p_{01}}{p_0}, \tfrac{p_{10}}{p_0}, 0\right), \\ &([], \{\!\{\}\!\}, \{\!\{\}\!\}, \{\!\{\}\!\}; [\bullet, \circ], [\circ, \circ], [\bullet, \circ]) && \left(\tfrac{p_{00}}{p_0}, \tfrac{p_{01}}{p_0}, \tfrac{p_{10}}{p_0}, 0\right), \\ &([], \{\!\{\}\!\}, \{\!\{\}\!\}, \{\!\{\}\!\}; [\bullet, \bullet], [\circ, \circ], [\circ, \circ]) && (0, 0, 0, 1), \\ &([], \{\!\{\}\!\}, \{\!\{\}\!\}, \{\!\{\}\!\}; [\circ, \circ], [\bullet, \bullet], [\circ, \circ]) && (0, 0, 0, 1), \\ &([], \{\!\{\}\!\}, \{\!\{\}\!\}, \{\!\{\}\!\}; [\circ, \circ], [\circ, \circ], [\bullet, \bullet]) && (0, 0, 0, 1), \end{aligned} \tag{5}$$

ここで $p_0 = p_{00} + p_{01} + p_{10}$ とする．3 つ目の成分，すなわちそれぞれの袋から取り出されたボールを見ると，明らかに，同じ袋から 2 つの赤のボールが取り出されたかどうかで，出力以外の情報を一切漏らさずに $a \wedge b$ の値を得ることができる．

2.2 定義

ボールと袋の形式的な定義を導入する．既に述べたように，● と ○ はそれぞれ，赤のボールと白のボールを表す．全てのボールの大きさは等しいものとする．

2.1 節でも述べたように，本稿では “[]” を可視多重集

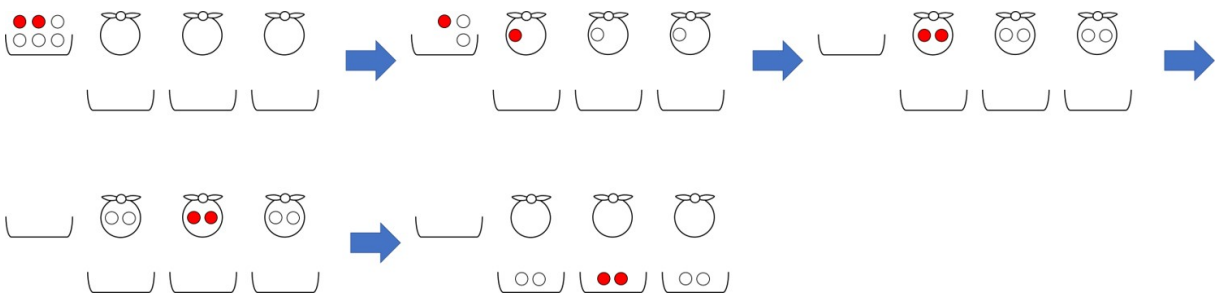


図 1: 1.1 節で提案した 2 入力 AND プロトコルの流れ（$a = b = 1$ の場合）.

合とし，“{{}}” を不可視多重集合とする．ℓ を自然数とし，$\mathsf{b}_i \in \{\bullet, \circ\}, 1 \le i \le \ell$，からなるボールの可視多重集合 $[\mathsf{b}_1, \ldots, \mathsf{b}_\ell]$ を**トレイ**と呼び，トレイの上にボールが置かれるものとする．また同様に，ボールの不可視多重集合 $\{\!\{\mathsf{b}_1, \ldots, \mathsf{b}_\ell\}\!\}$ を**袋**と呼び，袋の中にボールが入るものとする．

集合に対する操作を，集合の組に対して拡張したものを導入する．$\mathbf{X} = (X_1, \ldots, X_k)$ と $\mathbf{Y} = (Y_1, \ldots, Y_k)$ を，同じ長さ k の多重集合の組としたとき，次の操作を定義する．

$$\begin{aligned} \mathbf{X} \cup \mathbf{Y} \quad &:= (X_1 \cup Y_1, X_2 \cup Y_2, \ldots, X_k \cup Y_k) \\ \mathsf{union}(\mathbf{X}) \quad &:= X_1 \cup X_2 \cup \cdots \cup X_k \end{aligned}$$

次に，（2.1 節で記述した）**コンフィグレーション**を定義する．

定義 1 (Configuration) D をボールの可視多重集合とし，k を（袋の個数を表す）自然数とする．次を満たす 3 項組 $(T_0\,;\mathbf{B}\,;\mathbf{T}) = (T_0; B_1, \ldots, B_k; T_1, \ldots, T_k)$ をコンフィグレーションと呼ぶ．

- T_0 はトレイを表す．プロトコルの開始前には，全てのボールがここに置かれる．
- $\mathbf{B} = (B_1, \ldots, B_k)$ は k 個の袋の列である．
- $\mathbf{T} = (T_1, \ldots, T_k)$ は k 個のトレイの列であり，T_i には B_i から取り出されたボールが置かれる．
- $T_0 \cup (B_1 \cup \cdots \cup B_k) \cup (T_1 \cup \cdots \cup T_k) = D$ である．

また，このように D と k を固定して得られる全てのコンフィグレーションからなる集合を $C^{(D,k)}$ と書く．

コンフィグレーション $C = (T_0; B_1, \ldots, B_k; T_1, \ldots, T_k)$ に対し，T_0 や $T_1, \ldots, T_k$ のトレイのボールは目に見えるが，$B_1, \ldots, B_k$ の袋に入っているボールは外からその色が分からない（ただし個数は分かるとする）．これを踏まえて，**可視列** $\mathsf{vis}(C)$ を次のように定義する．

$$\mathsf{vis}(C) := (T_0; |B_1|, \ldots, |B_k|; T_1, \ldots, T_k)$$

また，全ての可視列から成る集合を次のように表す．

$$\mathsf{Vis}^{(D,k)} = \left\{\mathsf{vis}(C) \mid C \in C^{(D,k)}\right\}$$

以上の用語を用いて，色付きのボールと袋を用いた秘密計算**プロトコル**を定義する．

定義 2 (Protocol) プロトコルは 6 項組 $\mathcal{P} = (D, k, n, U, Q, A)$ として表される．ここで，

- D はプロトコルにおいて使えるボールの集合を表し，k は袋の個数を表す．したがって，初期コンフィグレーションは $\mathsf{union}(\mathbf{B}^0 \cup \mathbf{T}^0) = \phi$ かつ $|\mathbf{B}^0| = |\mathbf{T}^0| = k$ なる $\mathbf{B}^0$ と $\mathbf{T}^0$ を用いて $C^0 = (D, \mathbf{B}^0, \mathbf{T}^0)$ となる．
- n は 2 以上の自然数であり，プロトコルに参加するプレイヤーの人数を表す．
- U はプレイヤーが持つ入力の集合である．例えば，各プレイヤーの入力が 1 ビットのときは $U = \{0,1\}^n$ であり，以下では簡単のため，そのように仮定する．
- Q は状態集合を表し，初期状態 q_0 と最終状態 q_f を含む．
- $A : \left(Q \setminus \{q_f\}\right) \times \mathsf{Vis}^{(D,k)} \to Q \times \mathsf{Action}$ は動作関数であり，次の状態と現在のコンフィグレーションに対する操作を決定する．Action は次の操作から成る．以下では，コンフィグレーション $C = (T_0; \mathbf{B}; \mathbf{T}) = (T_0; B_1, \ldots, B_k; T_1, \ldots, T_k)$ に対する操作として記述する．
  - $(\mathsf{PublicPut}, \mathsf{b}, p)$．ここで，$\mathsf{b} \in T_0$ かつ $p \in \{1, 2, \ldots, k\}$ である．これは，トレイ T_0 の上にあるボール b を，p 番目の袋 B_p に（どのボールがどの袋に入るか分かる状態で）入れる操作である．この操作によって，C は次のコンフィグレーション C' に変化する．

    $$C' = (T_0 \setminus [\mathbf{b}]; B_1, \ldots, B_{p-1}, B_p \cup \{\!\{\mathbf{b}\}\!\}, B_{p+1}, \ldots, B_k; \mathbf{T})$$

    この操作を実行するプレイヤーは，その袋にボール b を入れる前に，b を他のプレイヤーに見せなければならない．
  - $(\mathsf{PrivatePut}, i, \mathbf{I}_0, \mathbf{I}_1)$．ここで，$1 \le i \le n$ であり，i 番目のプレイヤーが自分の入力 $x_i \in \{0,1\}$ に基づき操作を行う．また，$\mathsf{union}(\mathbf{I}_0) = \mathsf{union}(\mathbf{I}_1) \subseteq T_0$ であり，$\mathbf{I}_0 = (I_0^1, \ldots, I_0^k), \mathbf{I}_1 = (I_1^1, \ldots, I_1^k)$ とするとき，全ての i で $|I_0^i| = |I_1^i|$ である．これは，式 (2) にもある通り，入力 x_i を持つ i 番目のプレイヤーが，トレイ T_0 からボールを取り出し，それらを $\mathbf{I}_{x_i}$ が指定するように $\mathbf{B}$ に秘密に入れる操作である．この操作によってコンフィグレーションは次のように変化する．

    $$C' = (T_0 \setminus \mathsf{union}(\mathbf{I}_x); \mathbf{B} \cup \mathbf{I}_x; \mathbf{T})$$

    この操作において，$\mathsf{union}(\mathbf{I}_0) = \mathsf{union}(\mathbf{I}_1)$ なので，$\mathbf{I}_0$

の ● の数と $\mathbf{I}_1$ の ● の数は等しく，○ の数も同様であり，$x_i = 0$ であろうと $x_i = 1$ であろうと，同じボール集合を手に持つことに注意しよう．

– (Shuf, R)．ここで，$R \subseteq \{1, 2, \ldots, k\}$ である．これは，式 (4) にもある通り，R によって示された位置の袋をシャッフルする操作であり，シャッフル後の袋の順番は誰にも分からなくなる．この操作によってコンフィグレーションは次のように変化する．

$$C' = \left(T_0; B_{\pi^{-1}(1)}, B_{\pi^{-1}(2)}, \ldots, B_{\pi^{-1}(k)}, ; \mathbf{T}\right)$$

ここで π は，R 以外の位置は全て不動点であるような置換の中から一様ランダムに選ばれた置換である．

– (Take, p)．ここで，$p \in \{1, 2, \ldots, k\}$ である．これは，式 (5) にもある通り，p 番目の袋 B_p からボールを 1 つ取り出す操作であり，取り出されたボールはトレイ T_p の上に置かれる．この操作によってコンフィグレーションは次のように変化する．

$$\begin{aligned} &C' = (T_0; \mathbf{B}'; \mathbf{T}'), \text{ where} \\ &\mathbf{B}' = (B_1, \ldots, B_{p-1}, B_p \setminus \{\!\{\mathsf{b}\}\!\}, T_{p+1}, \ldots, T_k), \text{ and} \\ &\mathbf{T}' = (T_1, \ldots, T_{p-1}, T_p \cup [\mathsf{b}], T_{p+1}, \ldots, T_k) \end{aligned}$$

ここで，$\mathsf{b} \in B_p$ は取り出されたボールを表す．B_p からどのボールが取り出されるかはランダムであることに注意しよう．この操作によって，B_p の中にある他のボールに関する情報が漏れることはないものとする．プロトコル中で全ての袋から全てのボールを取り出す場合，単に (TakeAll) と書くことにする．

– (Move, b, p)．ここで，$\mathsf{b} \in T_p$ かつ $p \in \{1, 2, \ldots, k\}$ である．これは，p 番目にあるトレイの上にあるボール b を，トレイ T_0 に移す操作である．この操作によってコンフィグレーションは次のように変化する．

$$C' = (T_0 \cup [\mathsf{b}]; \mathbf{B}; T_1, \ldots, T_{p-1}, T_p \cup \{\!\{\mathbf{b}\}\!\}, T_{p+1}, \ldots, T_k)$$

– (Result, e)．ここで，e は何らかの式である．この操作は，プロトコルの状態が q_f に遷移したときに発生し，e を出力して終了することを表す．

プロトコル $\mathcal{P} = (D, k, n, U, Q, A)$ は入力 $x = (x_1, \ldots, x_n) \in U$ が与えられると，次のように動作する．まず D と k に対応するボールや袋がテーブルに置いてあり，すなわち初期コンフィグレーションは $\mathsf{union}(\mathbf{B}^0 \cup \mathbf{T}^0) = \phi$ かつ $|\mathbf{B}^0| = |\mathbf{T}^0| = k$ なる $\mathbf{B}^0$ と $\mathbf{T}^0$ に対し $C^0 = (D, \mathbf{B}^0, \mathbf{T}^0)$ であり，各プレイヤーは入力 x_i を秘密に持ち，プロトコルの状態は q_0 である．次に動作関数 $A(C^0, q_0)$ の出力に従い，コンフィグレーションと状態が遷移する．以降もコンフィグレーションと状態を変えながら，状態が q_f に移行するまで A が適用され，(Result, e) の操作とともに終了する．

入力 $(x_1, \ldots, x_n) \in U$ に対しプロトコル $\mathcal{P}$ を実行したとき，状態 q_f で Result により出力される e が，常にある関数 $f(x_1, \ldots, x_n)$ の値と等価であるとき，$\mathcal{P}$ は f に対して**正当性を満たす**と言う．以下では簡単のため，計算したい関数 f は，ブール関数 $f : \{0,1\}^n \to \{0,1\}$ であるとする．

次に，プロトコル $\mathcal{P}$ の**安全性**を定義する．ここでいくつかの用語を与える．プロトコル $\mathcal{P}$ の（有限の）実行を考えてみよう．初期から最終に至る全ての可視列を列挙したもの $\left(\mathsf{vis}\left(C^0\right), \mathsf{vis}\left(C^1\right), \ldots, \mathsf{vis}\left(C^f\right)\right)$ を**可視列トレース**と呼ぶことにする．ここで，$\mathsf{vis}(C^{i-1})$ はある操作によって $\mathsf{vis}(C^i)$ に遷移されるものとする．

定義 3 (Security) $\mathcal{P} = (D, k, n, U, Q, A)$ を f に対して正当なプロトコルとする．V を $\mathcal{P}$ の可視列トレースの確率変数，M を $\mathcal{P}$ の入力の分布，F を f の出力を表す確率変数とする．全ての $x \in U$ と可視列トレース v に対して

$$\Pr[M = x \mid F = 0] = \Pr[M = x \mid V = v, F = 0]$$

かつ

$$\Pr[M = x \mid F = 1] = \Pr[M = x \mid V = v, F = 1]$$

であるとき，$\mathcal{P}$ は f の下で**安全である**と言う．

$\mathcal{P}$ の安全性は，袋から取り出されたボールによって出力以外の情報が漏れないことを直感的には意味する．

プロトコルの正当性と安全性を確かめるために，本稿ではプロトコルの状態遷移を示したダイアグラムを 3 節で構成する．このダイアグラムはカードベースプロトコルに対して考案されたもの [15] であり，次に示す**確率トレース**を用いた拡張版 [16] が提案されている．

定義 4 (Probability Trace) $\mathcal{P} = (D, k, n, U, Q, A)$ をプロトコルとする．このとき，入力 $x \in U$ を何らかの順序で $1 \le x \le |U|$ の自然数とみなし，

$$q_{x,v} = \Pr[X = x, G_v = C \mid V = v]$$

を満たす $|U|$ 項組 $(q_{1,v}, q_{2,v}, \ldots, q_{|U|,v})$ を，コンフィグレーション C と可視列トレース v に対する確率トレースと呼ぶ．ここで，X，G_v，V はそれぞれ，入力を表す確率変数，可視列トレース v を見たときのコンフィグレーションを表す確率変数，可視列トレースの確率変数を表す．

3. ダイアグラム

Koch，Walzer 及び Härtel [15] はカードベース暗号の状態遷移を表すダイアグラムを構成した．その後，確率トレースを利用した拡張ダイアグラム [16] が提案されている．本節では，ボールと袋を用いたプロトコルの状態遷移を表す拡張ダイアグラムを構成する．

そのようなダイアグラムの例を図 2 に示す．その図において，コンフィグレーションと確率トレースの組を含む“箱”を**ステータス**と呼ぶことにする．各ステータスは可

視列トレースの接頭辞に関連している．2.1 節で述べた通り，コンフィグレーションの横にある確率トレースは，可視列トレースの接頭辞が与えられた時に，現在のコンフィグレーションがそのコンフィグレーションである条件付き確率を表す．

図 2 におけるステータスは，次のように遷移している．

- 1 番上にあるステータスは，C^0 と $(p_{00}, p_{01}, p_{10}, p_{11})$ の組から成る．
- 最初（とその次の）操作は PrivatePut である．説明のため，$x_1 = 0$ とする．この時，$\mathbf{I}_0^1 = ([\circ], [\bullet], [\circ])$ が袋に秘密に入れられる．$(x_1, x_2) = (0, 0)$ あるいは $(0, 1)$ なので，確率トレース $(p_{00}, p_{01}, p_{10}, p_{11})$ は $(p_{00}, p_{01}, 0, 0)$ となる．
- 3 番目の操作は Shuf である．操作の結果，3 つのコンフィグレーションが可能性として考えられ，それぞれ等確率（1/3）で生起する．このことは，3 つのコンフィグレーションに対する確率トレース中の係数から見てとれる．
- 一般に，Take は 2 つの可視列を生み出す．すなわち，$\bullet$ もしくは $\circ$ が取り出された場合の 2 つである．しかしながら，図 2 における 4 つ目の動作は TakeAll であり，6 つの可視列が生み出される．例えば，TakeAll によって $([\circ, \circ], [\bullet, \circ], [\bullet, \circ])$ が取り出されたとしよう．それらが取り出される確率は $\frac{p_{00}+p_{01}+p_{10}}{3}$ であるため，入力が (1,1) ではないことが分かる．これは出力が “$x_1 \wedge x_2 = 0$” となることを表す．

  このとき，入力が (0,0)，(0,1)，(1,0) であるそれぞれの条件付確率の分布は，プロトコル開始前に $x_1 \wedge x_2 = 0$ であることを知ったときのその分布と同じであるので，出力以外の情報は全く漏れない．

4. 2 入力 AND プロトコル

本節では，2.1 節で紹介した 2 つの入力 $(x_1, x_2) \in \{0,1\}^2$ の論理積を秘密計算するプロトコルの形式的な記述を与える．まず初めに，その AND プロトコルの原理を説明し，その後にプロトコルの記述を与える．AND プロトコルのダイアグラムを図 2 に示す．正当性や安全性はその図から確かめることができる．

4.1 原理

2.1 節で紹介した AND プロトコルの原理を説明する．このプロトコルにおいて，Alice と Bob は入力が 1 の場合に（最初の）袋 B_1 に $\bullet$ を入れ，入力が 0 の場合は異なる袋 B_2 と B_3 に $\bullet$ を入れることになる．もし袋が 2 つしかない場合，Alice と Bob はお互いの入力が 0 の場合も同じ袋 B_2 に $\bullet$ を入れることになる．したがって，入力が (1,1) の場合も (0,0) の場合も $\{\!\{\bullet, \bullet\}\!\}$ が存在してしまう．このことから，（このアイデアを基にする限り）袋は 3 つ必要なことが分かる．

形式的には，x_1 を持つ Alice は次の動作を行う．

$$\big(\text{PrivatePut}, 1, ([\circ], [\bullet], [\circ]), ([\bullet], [\circ], [\circ])\big)$$

その結果，コンフィグレーションは式 (2) の通りとなる．それから，x_2 を持つ Bob は次の動作を行う．

$$\big(\text{PrivatePut}, 2, ([\circ], [\circ], [\bullet]), ([\bullet], [\circ], [\circ])\big)$$

その結果，コンフィグレーションは式 (3) の通りとなる．したがって，3 つの袋をシャッフルした後に全てのボールを取り出すことで論理積の値だけを得ることができる．すなわち，もし $\{\!\{\bullet, \bullet\}\!\}$ があれば 1 であり，そうでなければ 0 である．出力以外の情報は漏れない．

4.2 記述

プロトコルの形式的な記述を次に示す．

---

2 入力 AND プロトコル $\big([\bullet, \bullet, \circ, \circ, \circ, \circ], 3, 2, \{0,1\}^2, Q, A\big)$

Steps:

(1) (PrivatePut, 1, ([∘], [•], [∘]), ([•], [∘], [∘]))
(2) (PrivatePut, 2, ([∘], [∘], [•]), ([•], [∘], [∘]))
(3) (Shuf, {1, 2, 3})
(4) (TakeAll)
(5) **if** visible seq. includes [•, •] **then**
(6) (Result, “$x_1 \wedge x_2 = 1$”)
(7) **else then**
(8) (Result, “$x_1 \wedge x_2 = 0$”)

---

5. 多入力 AND プロトコル

本節では，多入力の AND プロトコル，すなわち $f(x_1, \ldots, x_n) = x_1 \wedge \cdots \wedge x_n$ を秘密計算するプロトコルの形式的な記述を与える．ここで，$x_i \in \{0,1\}$ である．

5.1 アイデア

4.1 節でも述べた通り，2 入力 AND プロトコルでは，2 人のプレイヤーの入力が 1 の場合においてのみ，同じ袋に $\bullet$ を秘密に入れていた．この原理を単に拡張することで，多入力 AND プロトコルを構成できるのかどうかを考えてみよう．例として，3 人のプレイヤー P_1, P_2, P_3 を仮定し，P_i は入力 $x_i \in \{0,1\}$ を持つものとする．4 つの（空の）袋 B_1, B_2, B_3, B_4 を用意し，プレイヤー P_i の持つ入力 x_i が 1 の場合，（最初の）袋 B_1 に $\bullet$ を入れ（0 の場合は $i+1$ 番目の袋 B_{i+1}），$\circ$ を残りの袋に入れさせる．そうすると，コンフィグレーションは次のようになる．（スペースの都合上，空のトレイは省略する．）

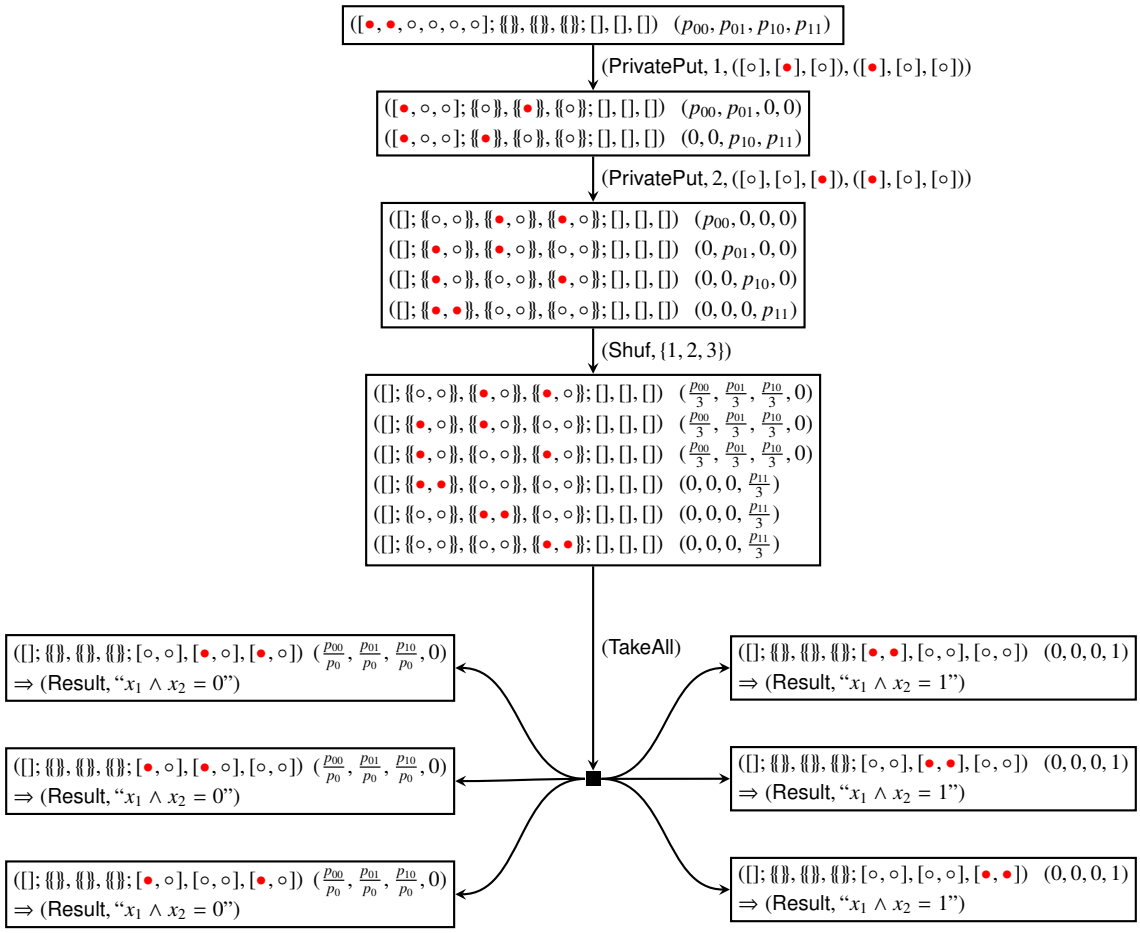


図 2: 2.1 節で紹介した $x_1 \wedge x_2$ を計算する AND プロトコルのダイアグラム．ここで，$p_0 = p_{00} + p_{01} + p_{10}$ である．

$$
\begin{aligned}
&(;\{\!\{\circ,\circ,\circ\}\!\},\{\!\{\bullet,\circ,\circ\}\!\},\{\!\{\bullet,\circ,\circ\}\!\},\{\!\{\bullet,\circ,\circ\}\!\};) && (p_{000},0,0,0,0,0,0,0)\\
&(;\{\!\{\bullet,\circ,\circ\}\!\},\{\!\{\bullet,\circ,\circ\}\!\},\{\!\{\bullet,\circ,\circ\}\!\},\{\!\{\circ,\circ,\circ\}\!\};) && (0,p_{001},0,0,0,0,0,0)\\
&(;\{\!\{\bullet,\circ,\circ\}\!\},\{\!\{\bullet,\circ,\circ\}\!\},\{\!\{\circ,\circ,\circ\}\!\},\{\!\{\bullet,\circ,\circ\}\!\};) && (0,0,p_{010},0,0,0,0,0)\\
&(;\{\!\{\bullet,\circ,\circ\}\!\},\{\!\{\circ,\circ,\circ\}\!\},\{\!\{\bullet,\circ,\circ\}\!\},\{\!\{\bullet,\circ,\circ\}\!\};) && (0,0,0,p_{100},0,0,0,0)\\
&(;\{\!\{\bullet,\bullet,\circ\}\!\},\{\!\{\bullet,\circ,\circ\}\!\},\{\!\{\circ,\circ,\circ\}\!\},\{\!\{\circ,\circ,\circ\}\!\};) && (0,0,0,0,p_{011},0,0,0)\\
&(;\{\!\{\bullet,\bullet,\circ\}\!\},\{\!\{\circ,\circ,\circ\}\!\},\{\!\{\bullet,\circ,\circ\}\!\},\{\!\{\circ,\circ,\circ\}\!\};) && (0,0,0,0,0,p_{101},0,0)\\
&(;\{\!\{\bullet,\bullet,\circ\}\!\},\{\!\{\circ,\circ,\circ\}\!\},\{\!\{\circ,\circ,\circ\}\!\},\{\!\{\bullet,\circ,\circ\}\!\};) && (0,0,0,0,0,0,p_{110},0)\\
&(;\{\!\{\bullet,\bullet,\bullet\}\!\},\{\!\{\circ,\circ,\circ\}\!\},\{\!\{\circ,\circ,\circ\}\!\},\{\!\{\circ,\circ,\circ\}\!\};) && (0,0,0,0,0,0,0,p_{111})
\end{aligned}
\tag{6}
$$

したがって，4 つの袋をシャッフルした後に全てのボールを取り出すと，$\{\!\{\bullet,\bullet,\bullet\}\!\}$ の袋があれば論理積の値は 1 であり，そうでなければ 0 であることが分かる．しかしながら，式 (6) から分かる通り，袋に入っている $\bullet$ の数から入力に関する情報が漏れてしまう．例えば，$\{\!\{\bullet,\bullet,\circ\}\!\}$ の袋がある場合，これは 3 つの入力の内の 2 つが 1 であることを意味する．すなわち，これは論理積の安全な計算ではない．

4 節で紹介した AND プロトコルのステップ 4 に戻ってみよう．そこでの操作 (TakeAll) を (Take, 1) に置き換える．すなわち，式 (4) の（最初の）袋 B_1 からボールを 1 つ取り出す．このとき，$\bullet$ が 1/3 の確率で取り出され，$\circ$ が 2/3 の確率で取り出される．もし $\bullet$ が取り出された場合，式 (4) のコンフィグレーションは次のように変化する．[*2]

$$
\begin{aligned}
&([];\{\!\{\circ\}\!\},\{\!\{\bullet,\circ\}\!\},\{\!\{\circ,\circ\}\!\};[\bullet],[],[]) && (\tfrac{p_{00}}{2},\tfrac{p_{01}}{2},\tfrac{p_{10}}{2},0),\\
&([];\{\!\{\circ\}\!\},\{\!\{\circ,\circ\}\!\},\{\!\{\bullet,\circ\}\!\};[\bullet],[],[]) && (\tfrac{p_{00}}{2},\tfrac{p_{01}}{2},\tfrac{p_{10}}{2},0),\\
&([];\{\!\{\bullet\}\!\},\{\!\{\circ,\circ\}\!\},\{\!\{\circ,\circ\}\!\};[\bullet],[],[]) && (0,0,0,p_{11}).
\end{aligned}
\tag{7}
$$

(Take, 1) は入力に関する情報を漏らさないことに注意しよう．その後，4 番目の袋を追加し，3 人目のプレイヤー P_3 は次の動作を行う．

$$\big(\text{PrivatePut}, 3, ([\circ],[],[],[\bullet,\circ]),([\bullet],[],[],[\circ,\circ])\big)$$

すなわち，$x_3 = 0$ の場合，式 (7) のコンフィグレーションは次のように変化する．

$$
\begin{aligned}
&(;\{\!\{\circ,\circ\}\!\},\{\!\{\bullet,\circ\}\!\},\{\!\{\circ,\circ\}\!\},\{\!\{\bullet,\circ\}\!\};) && (\tfrac{p_{000}}{2},0,\tfrac{p_{010}}{2},\tfrac{p_{100}}{2},0,0,0,0)\\
&(;\{\!\{\circ,\circ\}\!\},\{\!\{\circ,\circ\}\!\},\{\!\{\bullet,\circ\}\!\},\{\!\{\bullet,\circ\}\!\};) && (\tfrac{p_{000}}{2},0,\tfrac{p_{010}}{2},\tfrac{p_{100}}{2},0,0,0,0)\\
&(;\{\!\{\bullet,\circ\}\!\},\{\!\{\circ,\circ\}\!\},\{\!\{\circ,\circ\}\!\},\{\!\{\bullet,\circ\}\!\};) && (0,0,0,0,0,0,p_{110},0)
\end{aligned}
$$

$x_3 = 1$ の場合，式 (7) のコンフィグレーションは次のように変化する．

$$
\begin{aligned}
&(;\{\!\{\bullet,\circ\}\!\},\{\!\{\bullet,\circ\}\!\},\{\!\{\circ,\circ\}\!\},\{\!\{\circ,\circ\}\!\};) && (0,\tfrac{p_{001}}{2},0,0,\tfrac{p_{011}}{2},\tfrac{p_{101}}{2},0,0)\\
&(;\{\!\{\bullet,\circ\}\!\},\{\!\{\circ,\circ\}\!\},\{\!\{\bullet,\circ\}\!\},\{\!\{\circ,\circ\}\!\};) && (0,\tfrac{p_{001}}{2},0,0,\tfrac{p_{011}}{2},\tfrac{p_{101}}{2},0,0)\\
&(;\{\!\{\bullet,\bullet\}\!\},\{\!\{\circ,\circ\}\!\},\{\!\{\circ,\circ\}\!\},\{\!\{\circ,\circ\}\!\};) && (0,0,0,0,0,0,0,p_{111})
\end{aligned}
$$

[*2] 取り出されたボールが $\circ$ の場合，そのボールを B_1 に戻し $\bullet$ が取り出されるまでシャッフルとボールを取り出すことを繰り返す．

上から分かる通り，論理積の値が 1 のときだけ $\{\!\{\bullet,\bullet\}\!\}$ が存在する．0 の場合，4 つの袋は $\{\!\{\bullet,\circ\}\!\},\{\!\{\bullet,\circ\}\!\},\{\!\{\circ,\circ\}\!\},\{\!\{\circ,\circ\}\!\}$ を並び替えたものの 1 つとなっている．したがって，$(\mathsf{Shuf},\{1,2,3,4\})$ と $(\mathsf{TakeAll})$ を行うことで，出力以外の情報を漏らすことなく 3 つの入力 $x_1 \wedge x_2 \wedge x_3$ の論理積の値だけを得ることができる．

次に，$n=4$ の場合を考えよう．$n=3$ の場合と同様に，上の操作 $(\mathsf{TakeAll})$ を $(\mathsf{Take},1)$ に置き換え，その結果 $\bullet$ が取り出されたとする．その結果，コンフィグレーションは次のように変化する．

$$\begin{aligned}
&(;\{\!\{\circ\}\!\},\{\!\{\bullet,\circ\}\!\},\{\!\{\circ,\circ\}\!\},\{\!\{\circ,\circ\}\!\};) \quad (\tfrac{p_{000}}{3},\tfrac{p_{001}}{3},\tfrac{p_{010}}{3},\tfrac{p_{100}}{3},\tfrac{p_{011}}{3},\tfrac{p_{101}}{3},\tfrac{p_{110}}{3},0),\\
&(;\{\!\{\circ\}\!\},\{\!\{\circ,\circ\}\!\},\{\!\{\bullet,\circ\}\!\},\{\!\{\circ,\circ\}\!\};) \quad (\tfrac{p_{000}}{3},\tfrac{p_{001}}{3},\tfrac{p_{010}}{3},\tfrac{p_{100}}{3},\tfrac{p_{011}}{3},\tfrac{p_{101}}{3},\tfrac{p_{110}}{3},0),\\
&(;\{\!\{\circ\}\!\},\{\!\{\circ,\circ\}\!\},\{\!\{\circ,\circ\}\!\},\{\!\{\bullet,\circ\}\!\};) \quad (\tfrac{p_{000}}{3},\tfrac{p_{001}}{3},\tfrac{p_{010}}{3},\tfrac{p_{100}}{3},\tfrac{p_{011}}{3},\tfrac{p_{101}}{3},\tfrac{p_{110}}{3},0),\\
&(;\{\!\{\bullet\}\!\},\{\!\{\circ,\circ\}\!\},\{\!\{\circ,\circ\}\!\},\{\!\{\circ,\circ\}\!\};) \quad (0,0,0,0,0,0,0,p_{111}).
\end{aligned} \tag{8}$$

このコンフィグレーションは式 (7) のものと “同様” である．すなわち，$x_1=x_2=x_3=1$ のときだけ（最初の）袋 B_1 に $\bullet$ が含まれている．したがって，P_3 と同様に P_4 がボールを秘密に入れることで，4 入力 AND プロトコルを構成できる．

同様に拡張していくことで，n 入力 AND プロトコルを構成できる．

5.2 記述

5.1 節で説明したアイデアに基づき，$n\,(\geq 3)$ 入力 AND プロトコルの形式的な記述を与える．ここで，

$$\begin{aligned}
\mathbf{I}_0^1 &= \big([\circ],[\bullet],[\circ],[],\ldots,[]\big)\\
\mathbf{I}_0^2 &= \big([\circ],[\circ],[\bullet],[],\ldots,[]\big)\\
\mathbf{I}_1^1 = \mathbf{I}_1^2 &= \big([\bullet],[\circ],[\circ],[],\ldots,[]\big)
\end{aligned}$$

とし，$i\in\{3,4,\ldots,n\}$ に対して

$$\begin{aligned}
\mathbf{I}_0^i &= \big(\overset{1}{[\circ]},\overset{2}{[]},\ldots,\overset{i-1}{[]},\overset{i}{[\bullet,\circ]},\overset{i+1}{[]},\ldots,\overset{k}{[]}\big)\\
\mathbf{I}_1^i &= \big([\bullet],[],\ldots,[],[\circ,\circ],[],\ldots,[]\big)
\end{aligned}$$

とする．

$n\,(\geq 3)$ 入力 AND プロトコル

$([\bullet,\bullet,\circ,\circ,\ldots,\circ],n+1,n,\{0,1\}^n,Q,A)$

Steps:

(1) $(\mathsf{PrivatePut},1,\mathbf{I}_0^1,\mathbf{I}_1^1)$
(2) $(\mathsf{PrivatePut},2,\mathbf{I}_0^2,\mathbf{I}_1^2)$
(3) **for** $i\leftarrow 3$ to $n-1$ **do**
(4) **while**(1)
(5) $(\mathsf{Shuf},\{1,2,\ldots,i\})$
(6) $(\mathsf{Take},1)$
(7) **if** visible seq. $=([\bullet])$ **then**
(8) $(\mathsf{Move},\bullet,1)$
(9) $(\mathsf{PrivatePut},i,\mathbf{I}_0^i,\mathbf{I}_1^i)$
(10) **break**
(11) **else if** visible seq. $=([\circ])$ **then**
(12) $(\mathsf{Move},\circ,1)$
(13) $(\mathsf{PublicPut},\circ,1)$
(14) $(\mathsf{PrivatePut},n,\mathbf{I}_0^n,\mathbf{I}_1^n)$
(15) $(\mathsf{Shuf},\{1,2,\ldots,n+1\})$
(16) $(\mathsf{TakeAll})$
(17) **if** visible seq. includes $[\bullet,\bullet]$ **then**
(18) $(\mathsf{Result},\text{“}x_1\wedge\cdots\wedge x_n=1\text{”})$
(19) **else if** visible seq. does not include $[\bullet,\bullet]$ **then**
(20) $(\mathsf{Result},\text{“}x_1\wedge\cdots\wedge x_n=0\text{”})$

表 1: AND プロトコルの効率性．

	ボールの数	袋の数	実行時間	
2 入力 AND	6	3	有限	4 節
n 入力 AND	$2n+2$	$n+1$	非有限	5 節

5.3 正当性と安全性

5.2 節で記述した n AND プロトコルの，$n=3$ の場合のダイアグラムの 1 部分を図 3 に示す．この図から，$\{\!\{\bullet,\bullet\}\!\}$ の袋がある場合は論理積の値が 1 であり，そうでない場合は 0 であることが分かるため，正当性を満たす．その値が 0 の場合，入力がどの場合であっても，4 個の袋は $\{\!\{\bullet,\circ\}\!\},\{\!\{\bullet,\circ\}\!\},\{\!\{\circ,\circ\}\!\},\{\!\{\circ,\circ\}\!\}$ を並び替えたものの 1 つとなっており，それらは等確率で分布しているため，入力の情報が漏れることはない．

6. 議論

本節では，4 節と 5 節で示した 2 つの AND プロトコルの効率性と open problem を議論する．表 1 は提案プロトコルの効率性をまとめたものであり，3 つの指標（用いるボールの数・袋の数・実行時間）で評価した．

6.1 ボールと袋の数

4 節で見た通り，2 入力 AND プロトコルでは，6 個のボール（赤 2 個と白 4 個）と 3 個の袋が用いられている．4.1 節で述べた通り，2 入力 AND の安全な計算を行うのに，2 個の袋だけでは不可能のように思える．一方，n 入力 AND プロトコルでは，$2n+2$ 個のボール（赤 2 個と白 $2n$ 個）と $n+1$ 個の袋が用いられている．用いるボールと袋の数がコンスタントの n 入力 AND プロトコルを構成することは興味深い open problem である．

6.2 実行時間

4 節で見た通り，2 入力 AND プロトコルは固定のステッ

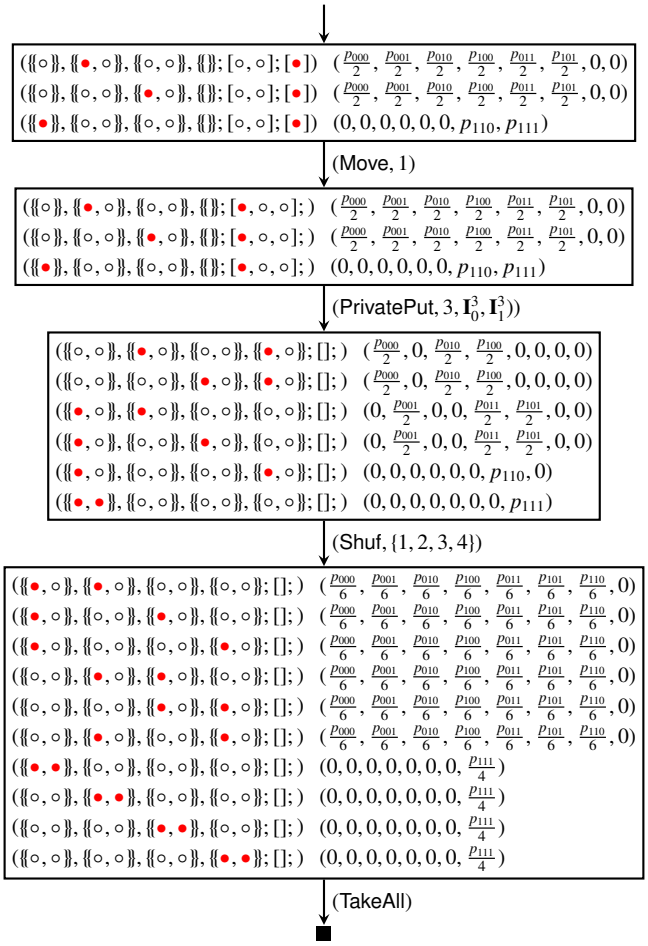


図 3: 3 入力 $x_1, x_2, x_3 \in \{0,1\}^3$ AND プロトコルのダイアグラムの 1 部分.

プ数で終了する．しかしながら，$n \geq 3$ なる n 入力 AND プロトコルは Las Vegas アルゴリズムであり，ステップ数は期待値で表される．以下では，プロトコルが終了するまでに必要なシャッフル操作の数を計算する．[*3]

まずは，最もシンプルな例として，$n = 3$ の場合を計算しよう．式 (3) から，赤のボールが取り出されるまで (Shuf, {1, 2, 3}) と (Take, 1) (と (PublicPut, ●, 1)) の操作を繰り返していたことを思い出そう．赤のボールが取り出される確率は常に 1/3 であるため，赤のボールが取り出されるまでに必要なシャッフル操作の期待値は 3 である．したがって，必要なシャッフル操作の総数の期待値は $3 + 1 = 4$ である．続いて，$n = 4$ の場合を計算しよう．この場合は，$n = 3$ の場合の操作に加えて，8 個のボール（赤 2 個と白 6 個）が入った 4 個の袋から赤のボールが取り出されるまで，(Shuf, {1, 2, 3, 4}) と (Take, 1) の操作を繰り返す必要がある．したがって，シャフル操作の期待値は $3 + 4 + 1 = 8$ である．

以上を踏まえ，n 入力の場合の期待値は，次のように表される．

$$\sum_{k=3}^{n} k + 1 = \frac{(n-2)(3+n)}{2} + 1$$

すなわち，シャッフル操作の期待値は $O(n^2)$ である．この次数を下げることや，固定のステップ数で終了する n 入力 AND プロトコルの構成は open problem である．

7. 終わりに

本稿では，ボールと袋を用いた AND の秘密計算を提案した．加えて，正当性と安全性を容易に検証するためのダイアグラムを導入した．今後の課題として，ボールと袋を用いて任意のブール関数を秘密計算するプロトコルを構成したい．

参考文献

[1] den Boer, B.: More Efficient Match-Making and Satisfiability The Five Card Trick, *Advances in Cryptology — EUROCRYPT '89* (Quisquater, J.-J. and Vandewalle, J., eds.), Lecture Notes in Computer Science, Vol. 434, Berlin, Heidelberg, Springer, pp. 208–217 (1990).

[2] Crépeau, C. and Kilian, J.: Discreet Solitary Games, *Advances in Cryptology — CRYPTO' 93* (Stinson, D. R., ed.), Lecture Notes in Computer Science, Vol. 773, Berlin, Heidelberg, Springer, pp. 319–330 (1994).

[*3] 定義 2 で示したプロトコルの操作を実際に行うことを考えると，(Shuf, R) が最も時間のかかる操作であることが分かる．

[3] Mizuki, T., Kumamoto, M. and Sone, H.: The Five-Card Trick Can Be Done with Four Cards, *Advances in Cryptology – ASIACRYPT 2012* (Wang, X. and Sako, K., eds.), Lecture Notes in Computer Science, Vol. 7658, Berlin, Heidelberg, Springer, pp. 598–606 (2012).

[4] Komano, Y. and Mizuki, T.: Multi-party Computation Based on Physical Coins, *Theory and Practice of Natural Computing* (Fagan, D., Martín-Vide, C., O'Neill, M. and Vega-Rodríguez, M. A., eds.), Lecture Notes in Computer Science, Vol. 11324, Cham, Springer, pp. 87–98 (2018).

[5] Fagin, R., Naor, M. and Winkler, P.: Comparing Information Without Leaking It, *Commun. ACM*, Vol. 39, No. 5, pp. 77–85 (online), DOI: 10.1145/229459.229469 (1996).

[6] Gradwohl, R., Naor, M., Pinkas, B. and Rothblum, G. N.: Cryptographic and Physical Zero-Knowledge Proof Systems for Solutions of Sudoku Puzzles, *Theory of Computing Systems*, Vol. 44, No. 2, pp. 245–268 (online), DOI: 10.1007/s00224-008-9119-9 (2009).

[7] Moran, T. and Naor, M.: Polling with Physical Envelopes: A Rigorous Analysis of a Human-Centric Protocol, *Advances in Cryptology - EUROCRYPT 2006* (Vaudenay, S., ed.), Lecture Notes in Computer Science, Vol. 4004, Berlin, Heidelberg, Springer, pp. 88–108 (2006).

[8] Naor, M. and Shamir, A.: Visual cryptography, *Advances in Cryptology — EUROCRYPT'94* (De Santis, A., ed.), Lecture Notes in Computer Science, Vol. 950, Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 1–12 (1995).

[9] Yao, A. C.: Protocols for Secure Computations, *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, FOCS '82, Washington, DC, USA, IEEE Computer Society, pp. 160–164 (online), DOI: 10.1109/SFCS.1982.88 (1982).

[10] Araki, T., Furukawa, J., Lindell, Y., Nof, A. and Ohara, K.: High-Throughput Semi-Honest Secure Three-Party Computation with an Honest Majority, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, New York, NY, USA, ACM, pp. 805–817 (online), DOI: 10.1145/2976749.2978331 (2016).

[11] Balogh, J., Csirik, J. A., Ishai, Y. and Kushilevitz, E.: Private computation using a PEZ dispenser, *Theoretical Computer Science*, Vol. 306, No. 1, pp. 69–84 (online), DOI: https://doi.org/10.1016/S0304-3975(03)00210-X (2003).

[12] Moran, T. and Naor, M.: Basing cryptographic protocols on tamper-evident seals, *Theoretical Computer Science*, Vol. 411, No. 10, pp. 1283 – 1310 (online), DOI: https://doi.org/10.1016/j.tcs.2009.10.023 (2010).

[13] D'Arco, P. and Prisco, R. D.: Secure computation without computers, *Theoretical Computer Science*, Vol. 651, pp. 11 – 36 (online), DOI: https://doi.org/10.1016/j.tcs.2016.08.003 (2016).

[14] Fellows, M. and Koblitz, N.: Kid Krypto, *Advances in Cryptology — CRYPTO' 92* (Brickell, E. F., ed.), Lecture Notes in Computer Science, Vol. 740, Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 371–389 (1993).

[15] 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).

[16] Mizuki, T. and Komano, Y.: Analysis of Information Leakage Due to Operative Errors in Card-Based Protocols, *Combinatorial Algorithms* (Iliopoulos, C., Leong, H. W. and Sung, W.-K., eds.), Lecture Notes in Computer Science, Vol. 10979, Cham, Springer International Publishing, pp. 250–262 (2018).