カードを用いた秘匿グループ分けプロトコル

橋本　侑知[1,3]　品川　和雅[2,3]　縫田　光司[3]　稲村　勝樹[1]　花岡　悟一郎[3]

概要：あらかじめ人数を定めておいた二個のグループ A_1, A_2 に複数のメンバーを振り分けることを考える。ここで、グループ A_1 のメンバーは誰がグループ A_1 に所属しているかを知ることが出来るが、グループ A_2 のメンバーは他の人がどのグループに所属しているかを全く知ることができないようにしたい。例えば、このグループ分けはゲームマスターを仮定しない人狼ゲームに応用することができる。本論文では、上記を一般化した m グループ $A_1, A_2, \ldots, A_m$ への振り分けを、カードを用いて実現する。技術的には、置換を表すカード列を秘匿したまま置換同士の演算を行う新たなプロトコルを提案することで、グループ分けを実現している。

キーワード：秘匿グループ分け，置換，ナンバーカード

Secure Grouping Protocol Using Cards

Yuji Hashimoto[1,3]　Kazumasa Shinagawa[2,3]　Koji Nuida[3]　Masaki Inamura[1]　Goichiro Hanaoka[3]

Abstract: We consider the problem of classifying people into two groups A_1, A_2 where the numbers of group members are determined in advance. Additionally, we impose a condition such that all members of group A_1 knows who else is in group A_1, and all members of group A_2 knows nothing about the other members, i.e., the only thing they know is about the group they belong to. For an example, we can apply this sort of grouping to the Werewolf game without needing the Game Master. In this paper, we propose a secure card-based protocol and consider the generalized version of the above problem where we classify people into more than two groups. In our proposed protocol, we establish a new technique on secure card-based operations on permutations, which we believe to have applications elsewhere.

Keywords: secure grouping, permutation, number cards

1. はじめに

何人かのメンバーをランダムにグループ分けする場面を考える。この際、誰がどのグループに振り分けられたかという情報は秘密にしたい。このことを以後、秘匿グループ分けと呼ぶこととする。例えば、人数が 2 人, 3 人, 5 人のグループをそれぞれ A_1, A_2, A_3 グループとして、秘匿グループ分けをすることを考える。ここで、他の人のグループの所属情報を全て秘匿する場合には、1 から 10 までの番号が書かれた紙を用いてくじ引きを行えば十分である。一方、人狼ゲーム等で必要な秘匿グループ分けでは、このように単純に解決することができない。このゲームはいくつか役職が存在し、あらかじめ秘匿グループ分けを行ってからゲームを始める。このとき、人狼という役職に割り当てられた人は他に誰が人狼であるのかも知っていなければならない。このように、所属情報を全て秘匿するのではなく、同じグループのメンバー同士は知ることができるようにしたい場合、通常はこのようなシチュエーションで秘匿グループ分けをするとき、公平な第三者を仮定したり、コ

1 東京電機大学
Tokyo Denki University
2 筑波大学
University of Tsukuba
3 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology

ンピュータを用いたりするが、それをカードを用いて手軽に実装したい。

そこで本論文では、ランダムにグループのメンバーを選ぶという条件の下で、他の情報を互いに秘匿したまま各プレイヤーが自分の所属のチームとそのメンバー全員を知ることができるような秘匿グループ分けプロトコルをカードを用いて実現する。より詳細には、このプロトコルはあらかじめ人数を定めておいた m 個のグループ $A_1, A_2, \ldots, A_m$ にメンバーをランダムに振り分けるプロトコルである。さらに、このプロトコルは単に m 個のグループに分けられるというプロトコルではない。例えば A_1, A_2 のグループのメンバーはお互いに同じグループの仲間が誰であるかのみ知ることができるような柔軟なグループ分けも達成している。このようなプロトコルを構成するために我々は、置換を表すカード列を秘匿したまま置換同士の演算を行う従来にはない新たな手法を提案する。今回提案した方式は、ゲームマスターなし人狼など、多くの用途に応用できる事が期待される。

2. 準備

ここでは秘匿グループ分けプロトコルを構成する際に必要なカード、用語、シャッフルの準備をする。

2.1 プレイヤー番号

あらかじめ各プレイヤーに与えておく番号であり、プロトコルの始めに必ず定めて公開しておくものとする。

2.2 ナンバーカード

[4][6] などの通常のカードプロトコルでは、ブール値を扱うため、次のように符号化のルールを定義する。

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

本論文では n 文字の置換を直接扱うため、カードの表面に 1 から n までの番号が書いてあると考える方が議論の見通しが良い [*1]。そこで、下図のように $1, 2, \ldots, n$ が書かれたカードを用いる。これらのカードをナンバーカードと呼ぶことにする。

$$\boxed{1}\ \boxed{2}\ \ldots\ \boxed{n}$$

従来のカードと同様に、裏面はどのカードも同様で区別の付かないものとする。また、カードプロトコルではカードを裏にして、他のカードと区別の付かない状態のカード列をコミットメントと言う。またカードを表にすることを開示と言う。

[*1] 通常のブール値のためのカードを用いても、提案プロトコルは実行することができる。その場合、カード枚数は $2\log_2 n$ 倍に増える。

2.3 カードの操作に関する置換やシャッフル

n 枚のカード列 $(x_1, x_2, \ldots, x_n)$ の並べ替えは、n 次対称群 S_n の元と対応づけることができる。カード列 $(x_1, x_2, \ldots, x_n)$ に対して、置換 $\sigma \in S_n$ を適用した後のカード列 $\sigma(x_1, x_2, \ldots, x_n)$ を次のように定義する。

$$\sigma(x_1, x_2, \ldots, x_n) := (x_{\sigma^{-1}(1)}, x_{\sigma^{-1}(2)}, \ldots, x_{\sigma^{-1}(n)})$$

言い換えるとこの操作は、各 i について i 番目のカードを $\sigma(i)$ 番目へ移動させる操作である。また恒等置換 $\mathsf{id}_n \in S_n$ は、$(x_1, x_2, \ldots, x_n)$ を $(x_1, x_2, \ldots, x_n)$ に移す置換である。

次に、Pile-Scramble Shuffle の定義を述べる。このシャッフルは Ishikawa ら [7] によって初めて提案された。

定義 1 (Pile-Scramble Shuffle) n は任意の 2 以上の整数であるとする。n 次の *pile-scramble shuffle* は、カード列 $(x_1, x_2, \ldots, x_n)$ に対して、S_n の全ての置換をそれぞれ一様な確率で適用する操作である。

$$\left\| \boxed{?} \middle| \boxed{?} \middle| \ldots \middle| \boxed{?} \right\| (x) \ \rightarrow\ \boxed{?}\ \boxed{?}\ \ldots\ \boxed{?}\ \ (r(x))$$

ここで、$r \in S_n$ はランダムな置換である。また、各 x_i がカード単体ではなく複数のカードの束である場合にも同様の操作を考える。

2.4 置換についての定義, 定理

定義 2 (巡回置換の長さと巡回域) 相異なる数 $i_1, i_2, \ldots, i_r \in \mathbb{N}$ に対して、巡回的に $\tau(i_1) = i_2, \tau(i_2) = i_3, \ldots, \tau(i_r) = i_1$ と表されるものを長さ r の巡回置換と言い、$(i_1 i_2 \ldots i_r)$ と表す。ただし、上記のサイクルに現れない文字は動かさないものとする。また、集合 $\{i_1, i_2, \ldots, i_r\}$ をこの巡回置換 τ の巡回域と言う。

定義 3 (互いに素である置換) 二つの巡回置換 τ_α, τ_β について、それらの巡回域が共通部分を持たないとき、τ_α, τ_β は互いに素であると言う。

定理 1 (置換の表現) 任意の置換は互いに素である巡回置換の積で一意的に表される。

定義 4 (置換の型) 置換 τ を互いに素な巡回置換の積として表した時、$i = 1, 2, \ldots, k$ に対して、長さが r_i の巡回置換がちょうど m_i 個現れるならば、τ は型 $\langle r_1^{m_1}, r_2^{m_2}, \ldots, r_k^{m_k} \rangle$ を持つと定める。また、型 $\langle r_1^{m_1}, r_2^{m_2}, \ldots, r_k^{m_k} \rangle$ を持つ置換全体の集合のことも $\langle r_1^{m_1}, r_2^{m_2}, \ldots, r_k^{m_k} \rangle$ で表すことにする。

ここで、このような置換をカード列でどのように表現するか具体例を通して説明しておく。例えば、$\tau = (12)(345) \in \langle 2^1, 3^1 \rangle$ は

$$\boxed{2}\ \boxed{1}\ \boxed{5}\ \boxed{3}\ \boxed{4}$$

と表される。このカード列は、id_5 を表すカード列に 2.3 節の要領で置換 τ を利用してできるカード列に他ならない。

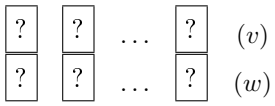
3. カードを用いた置換同士の演算

本章では、置換を表すカード列を用いて、秘匿グループ分けプロトコルで用いる置換同士の演算について説明する。さらに通常、対称群の元である置換をランダマイズしてしまうと置換の型が変わってしまい、思い通りのグループ分けができない。そこで、型を変えずに置換をランダマイズするプロトコルを提案する。数学的には、ある置換からそれと共役な置換へ移す写像を考えることにより、型を変えずにその型の中でランダマイズを達成している。この章が本論文での技術的な貢献部分である。

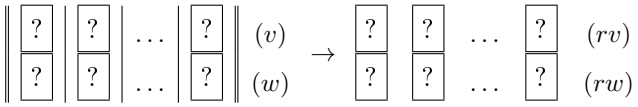
3.1 置換同士の演算

本節では、置換コミットメント v と w が与えられたときに、置換コミットメント $v^{-1}w$ を生成する方法を示す。

(1) 置換コミットメント v, w を下図のように揃えて並べる。以降の図の表示において、カード列の右側に、置換の値を明示すると約束する。この際、裏返しの状態の置換コミットメントには丸括弧を用い、開示された状態の置換コミットメントには括弧を用いないものとする。

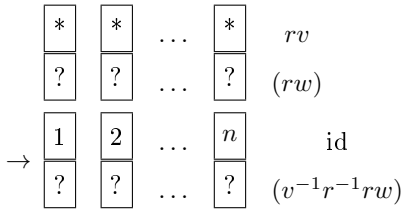


(2) Pile-Scramble Shuffle を適用する。

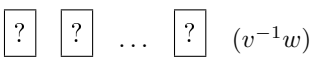


ここで、$r \in S_n$ はランダムな置換である。

(3) 上の行を開示し、rv を公開する。そして、上の行が id となるように、上下のカードをペアに並び替えを行う。下の行は $(rv)^{-1} = v^{-1}r^{-1}$ の並び替えを適用されたことになる。



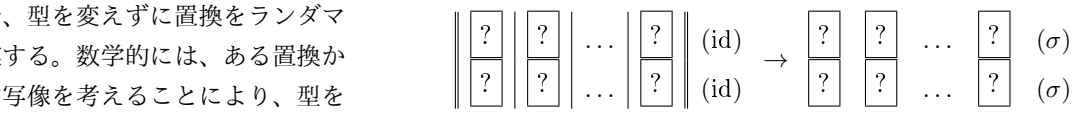
(4) 下の行を出力する。



3.2 置換のランダマイズ

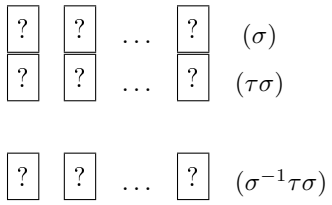
ここでは、秘匿グループ分けプロトコルを用いる際に必要となるグループの生成の手順を説明する。具体的には、置換 τ を入力として、τ と同じ型に属する置換 ρ を一様ランダムに選び、置換コミットメント ρ を出力する手順である。4 章でこの技法を用いた具体的な秘匿グループ分けプロトコルの構成について説明する。また、本論文では置換 τ は公開情報であることに注意しておく。

(1) 置換コミットメント id を二つ並べ、Pile-Scramble Shuffle を適用する。得られた置換を σ とする。



(2) 置換コミットメント σ に置換 τ を 2.3 節の要領で作用させて、置換コミットメント $\tau\sigma$ を作る。

(3) 3.1 節の手順に従って、二つの置換コミットメント σ と $\tau\sigma$ から置換コミットメント $\sigma^{-1}\tau\sigma$ を作る。このカード列を出力する。



この手順で生成した $\sigma^{-1}\tau\sigma$ を ρ とする。

4. 秘匿グループ分けプロトコル

本章では 3 章で紹介したプロトコルをどのように適用して秘匿グループ分けプロトコルを実現していくかを説明する。ここでは人狼ゲームを例に取って説明する。人狼ゲームとは基本的には村人、人狼のプレイヤーがおり、このゲームは、村人に成り済ましている人狼を見つけ出して処刑すれば村人の勝ち、できなければ人狼の勝ちとなる。実際は他にも特殊な役職が存在する事も多いが簡単のため、村人と人狼のみのグループ分けを解説し、そのあと一般のグループ分けについても解説する。

4.1 2 つのグループへの秘匿グループ分けプロトコル

人狼ゲームはゲームの設定上村人たちは自分自身が村人だという情報以外は何も知らず、人狼同士は自分自身が人狼であるということと同時に他に誰が狼なのか知る必要がある。ここでは例として A_1 を 2 人の狼グループ、A_2 を 3 人の村人グループとして解説する。

(1) あらかじめプレイヤー番号を決める。

(2) 2 人の狼グループである A_1、3 人の村人グループである A_2 に対応させた型 $\langle 1^3, 2^1 \rangle$ の置換を適当に 1 つ作り、それを置換 τ とする。例えば、$\tau = (1)(2)(3)(45)$ とすればよい。ここで、村人は村人同士の情報も知ることができないため、3 人の村人各々を単独のグループ (長さ 1 の巡回置換) とみなして型を設定している。

(3) 3.2 節の置換のランダマイズプロトコルに従って ρ を

生成する。

(4) プレイヤー i は最終的に生成されたカード列 ρ の左から i 番目のカードを取る。

(5) プレイヤー番号と同じ番号のカードを受け取ったプレイヤーは村人である。異なる番号を受け取ったプレイヤーは人狼であり、受け取ったカード番号が仲間の人狼に対応している。

この手順によって 2 つのグループ A_1, A_2 への秘匿グループ分けプロトコルが構成される。このプロトコルで例えばプレイヤー $1,3,4$ が村人になったとして、プレイヤー $2,5$ は人狼になったとしよう。この時 $1,3,4$ のプレイヤーは自分のプレイヤー番号と同じ番号のカードを得ることになり自分が村人であるという情報以外は何も分からない。一方、人狼であるプレイヤー $2,5$ は自分たちが人狼である事を知ることができると同時に他にどのプレイヤーが人狼であるのか知ることができる。より具体的に言えば、プレイヤー 2 は 5 のカードを、プレイヤー 5 は 2 のカードを得ることになるため、まず自分のプレイヤー番号と違うカードという事で人狼であると知ることができ、人狼のプレイヤーが得たカード番号と同じプレイヤー番号のプレイヤーが仲間の人狼であることがわかる。

4.2 一般的な秘匿グループ分けプロトコル

4.1 節での秘匿グループ分けプロトコルでは特殊な条件を課してグループ分けを達成した。しかし、例えば人狼が 3 人の設定の場合は 4.1 節での方法では人狼同士が完全に誰が人狼なのかを把握することは不可能である。また、人狼ゲームではないシチュエーションであるが、例えば A_1, A_2 のグループの人達のみ、お互いに同じグループの仲間が誰であるか知ることができるようにすることも 4.1 節での秘匿グループ分けプロトコルでは不可能である。このような自明でない 4.1 節でのプロトコルの拡張である一般的な秘匿グループ分けプロトコルを達成するためにまずは所属を表すナンバーカードを導入する。

4.2.1 所属を表すナンバーカード

4.1 節での秘匿グループ分けプロトコルでは単にプレイヤー番号と対応しているナンバーカードのみを用いていたが、それに加えて所属を表すナンバーカードを導入する。

所属を表すナンバーカードとは、プレイヤーがどの所属なのかを表すナンバーカードである。このカードはグループの数が m 個で、プレイヤーの人数を n 人とすると、プレイヤー番号と対応関係にある通常の 1 から n のナンバーカードに加えて、所属を表す $n+1$ から $n+m$ のナンバーカードを用いる。例えば、プレイヤーが 10 人でグループを A_1, A_2, A_3 の 3 個のグループに分けるとしよう。このとき、1 から 10 のナンバーカードに加えて、A_1, A_2, A_3 グループの所属を表すナンバーカード $11, 12, 13$ を用いる。この例では全グループがお互いに同じグループの仲間が誰であるか知ることができるようにすることを目標としていたが、一般的にはこのカードは、例えば A_1, A_2 のグループの人達のみ、お互いに同じグループの仲間が誰であるか知ることができるようにすることができる。この時に A_1, A_2 のグループの人達がたとえ同じグループの仲間を知る事ができたとしても A_i と A_j のどちらのグループに所属しているのか分からないということが生じないようにする役割を果たすカードが所属を表すナンバーカードである。

4.3 一般的な秘匿グループ分けプロトコルの手順

プレイヤーが n 人で m 個のグループ $A_1, A_2, \ldots, A_m$ にグループ分けすること考える。また、人数が r_i のグループが m_i 個あるとすると $n = \sum_{i=1}^{k} m_i r_i$ および $m = \sum_{i=1}^{k} m_i$ である。このとき、扱う置換は S_{n+m} の元である。

(1) あらかじめプレイヤー番号を決める。

(2) $\langle (r_1+1)^{m_1}, (r_2+1)^{m_2}, \ldots, (r_k+1)^{m_k} \rangle$ の型である置換に属する任意の τ を選び、置換コミットメント τ を生成する。ただし、この τ を巡回置換の積で表したときのその各々の因子には所属を表すナンバーを含んだサイクルにする必要がある。

(3) 3.2 節の置換のランダマイズプロトコルを用いて、ρ を生成することを考える。このとき、(1) の工程で、あらかじめ σ を $2r$ 個まとめて作っておく。ここで、$r = \max(r_1, \ldots, r_k)$ である。また、ランダマイズに用いる置換 σ は、$\sigma(n+i) = n+i$ が全ての $i = 1, 2, \ldots, m$ で成り立つという条件を満たすものとする。すなわち、σ を作る際に 1 枚目から n 枚目のカードに対してのみ Pile-Scramble Shuffle を適用する。同じ σ を用いることで、τ から ρ を生成した工程と同様な手順で τ^2 から ρ^2 を生成でき、このようにして $\rho, \cdots, \rho^r$ を生成する。

(4) (1) に従って、プレイヤー i は最終的に生成されたカード列 $\rho, \rho^2, \ldots, \rho^r$ の左から i 番目のカードをそれぞれ 1 枚ずつもらう。(カード列 $\rho, \rho^2, \ldots, \rho^r$ の全てにおいて $n+1$ 枚目から $n+m$ のカードのカードは伏せられたままである。)

(5) 各プレイヤーは少なくとも、所属を表すカード 1 枚と同じグループ全員のプレイヤーカードを 1 枚ずつ得ることとなり、その得たナンバーカードにより同じグループのメンバーを全員把握し、所属を表すナンバーカードにより所属グループを把握する。

この手順によって一般的な秘匿グループ分けプロトコルが達成される。ここでは、人数が 2 人, 2 人, 1 人のグループ A_1, A_2, A_3 のグループ分けを考える。ただし、A_1, A_2 はお互いに同じグループの仲間が誰であるか知ることができるように分けるとする。

(1) 各プレイヤーにプレイヤー番号を割り振る。

(2) A_1, A_2, A_3 の所属を表すナンバーカードを $6, 7, 8$ とす

る。置換の型が $\langle 3^2, 2\rangle$ である置換コミットメント τ を生成する。(手順通りであれば $\langle 3^2, 2\rangle$ の型を生成しなければならないが、実際この場合は、$\langle 3^2, 1\rangle$ を生成した方が賢い。なぜならば、1 人のグループが 1 つしかないため所属を表すナンバーカードがなくても自分のカードしか得られなかったプレイヤーが A_3 グループだと判定できるからである。)

(3) 3.2 節の置換のランダマイズプロトコルを用いて ρ を生成することを考える。このとき、3.1 の (1) の工程であらかじめ、σ を 4 個まとめて作っておく。ただし、3.1 節の置換の演算プロトコルにおいて、恒等置換コミットメント id の所属を表すナンバーカードの位置しない左から $\sum_{i=1}^{2} m_i r_i = (2 \times 2 + 1 \times 1) = 5$ 枚目までのカードに対して、Pile-Scramble Shuffle をする。$r = \max(2, 1) = 2$ であるから、ρ を生成したのと同様な手順で τ^2 から ρ^2 を生成する。

(4) プレイヤー i は最終的に生成されたカード列 ρ, ρ^2 の左から i 番のカードをそれぞれもらう。

(5) 各プレイヤーは少なくとも所属を表すナンバーカード 1 枚と同じグループ全員を表すナンバーカードを 1 枚ずつ得る。

4.4 正当性の証明

まず始めに、あらかじめ定められたある型 T の置換である入力 τ に対して、出力 $\sigma^{-1}\tau\sigma$ がまた T の置換であることを示す。

実際、これはある置換 τ からそれと共役な置換 $\sigma^{-1}\tau\sigma$ へ移す写像となっているので、入力と出力で置換の型は変わらない。

次に、置換のランダマイズによる出力が一様ランダムであることを示せばよい。すなわち、置換のランダマイズによって型は変わらないから、任意の $\tau \in T$ に対して、出力 $\sigma^{-1}\tau\sigma \in T$ が一様ランダムであることを言えればよい。ただし、$\sigma(q)(q = 1, 2, \cdots, n)$ であるランダムにシャッフルする部分を考える。

実際、τ の取り方に依存せず $\sigma^{-1}\tau\sigma \in T$ が決まり、σ が一様ランダムであれば出力 $\sigma^{-1}\tau\sigma \in T$ は一様ランダムである。なぜならば、置換 $\tau(a_i) = b_i (i = 1, 2, \cdots, n)$ のとき S_n に属する任意の置換 σ に対して、$\sigma^{-1}\tau\sigma(\sigma^{-1}(a_i)) = \sigma^{-1}(b_i)$ が成り立つからである。(これは直感的に言えば置換の中の文字の付け替えに相当する。) よって、任意の τ に対して、その型の任意の置換が生成でき、その個数に偏りがないことが示された。

参考文献

[1] T. Mizuki, M. Kumamoto, and H. Sone, "The five-card trick can be done with four cards," Proc. ASIACRYPT 2012, Lecture Notes in Computer Science, Springer-Verlag, vol. 7658, pp. 598-606, 2012.

[2] T. Mizuki and H. Shizuya, "A Formalization of Card-Based Cryptographic Protocols via Abstract Machine," International Journal of Information Security, Springer-Verlag, vol.13, no.1, pp.15-23, 2014.

[3] T. Mizuki and H. Shizuya, "Practical Card-Based Cryptography," Fun with Algorithms 2014, Lecture Notes in Computer Science, Springer-Verlag, vol.8496, pp.313-324, 2014.

[4] T. Mizuki and H. Sone, "Six-card secure AND and four-card secure XOR," Proc. Frontiers in Algorithmics (FAW 2009), Lecture Notes in Computer Science, vol. 5598, pp. 358-369, Springer-Verlag, 2009.

[5] Takuya Nishida, Yu-ichi Hayashi, Takaaki Mizuki, Hideaki Sone: Securely Computing Three-Input Functions with Eight Cards. IEICE Transactions 98-A(6): 1145-1152 (2015)

[6] T. Mizuki, M. Kumamoto, and H. Sone, "The five-card trick can be done with four cards," Proc. ASIACRYPT 2012, Lecture Notes in Computer Science, Springer-Verlag, vol. 7658, pp. 598-606, 2012.

[7] R. Ishikawa, E.chida, and T.Mizuki, "Efficient card-based protocols for generating a hidden random permutation without fixed points ," Unconventional Computation and Natural Computation, C.S. Calude and M.J. Denneen, eds., vol.9252, pp.215-226, Lect. Notes Comput. Sci., Springer International publishing, 2015.