一様巡回群分解に基づく
一様閉シャッフルの実現方法とその応用

2024年5月22日

産学連携と数理・暗号分野連携によるカードベース暗号の深化と新境地

品川 和雅（茨城大）

本研究は金井和貴氏 (呉高専)、宮本賢伍氏 (茨城大)、縫田光司氏 (九州大) との共同研究

本講演の概要

- 一様巡回群分解（有限群の分解の一種）という概念の提案
- 一様巡回群分解のカードベース暗号への応用
- 論文情報
  - Uniform cyclic group factorizations of finite groups (Commu. Alg. 2023)
    - Kazuki Kanai, Kengo Miyamoto, Koji Nuida, Kazumasa Shinagawa
    - 概要：「一様巡回群分解」の定義と基本性質の証明
  - How to Covertly and Uniformly Scramble the 15 Puzzle and Rubik's Cube (FUN 2024)
    - Kazumasa Shinagawa, Kazuki Kanai, Kengo Miyamoto and Koji Nuida
    - 概要：「一様巡回群分解」のカードベース暗号への応用

目次

- イントロダクション
- 一様巡回群分解（UCF）について(Comm. Alg.の結果)
- 応用：一様閉シャッフルの実装(Comm. Alg.の結果 + FUNの結果)
- 応用：パズルの秘匿ランダムインスタンス生成(FUNの結果)

Mizuki-Shizuyaモデルにおける操作

- (perm, π)：$\pi \in S_n$
  - 置換πで並び替える操作
- (turn, T)：$T \subseteq \{1,2,\ldots,n\}$
  - カードをめくる操作
- $(\mathrm{shuffle}, \Pi, \mathcal{F})$：$\mathcal{F}$は$\Pi(\subseteq S_n)$上の確率分布
  - 置換$\pi \leftarrow \mathcal{F}$で並び替える操作
  - **どのπが選ばれたかは秘匿される必要がある**

シャッフルを安全に実装する方法は自明ではない

Practicalなシャッフルたち

- **ランダムカット (RC)**：ランダム巡回シフト
  - 実装方法：ヒンズーカット、コマ
- **ランダム二等分割カット(RBC)**：二束のランダム巡回シフト
  - 実装方法：封筒＋ランダム交換、コマ
- **パイルシフティングシャッフル(PShift)**：k束のランダム巡回シフト
  - 実装方法：封筒＋ヒンズーカット、コマ
- **完全シャッフル(CS)**：ランダム置換
  - 実装方法：かき混ぜる
- **パイルスクランブルシャッフル(PScramble)**：k束のランダム置換
  - 実装方法：封筒＋かき混ぜる

- これらの共通点：**ランダム巡回シフト**または**ランダム置換**

本研究の出発点

- **巡回群シャッフル**
  - 置換πを一様ランダムな回数だけ適用するシャッフル
  - すなわち、一定の操作を繰り返すことによって実装されるシャッフル

- RC/RBC/PShiftは巡回群シャッフルの一種

- どのようなシャッフルたちが巡回群シャッフルから実現されるか？

本研究の成果

- **一様巡回群分解**（Uniform Cyclic group Factorization: **UCF**）という新しい種類の有限群の分解を提案
  - この概念自体は、純粋に有限群論的な概念

- 応用①：一様閉シャッフルの巡回群シャッフルによる実装
  - 一様閉シャッフル： Πが群で、$\mathcal{F}$が一様分布のシャッフル
  - 巡回群シャッフルは一様閉シャッフルの特別な場合

- 応用②：パズルの秘匿ランダムインスタンス生成
  - ルービックキューブ＆15パズルに対して適用

目次

- イントロダクション
- **一様巡回群分解（UCF）について(Comm. Alg.の結果)**
- 応用：一様閉シャッフルの実装(Comm. Alg.の結果 + FUNの結果)
- 応用：パズルの秘匿ランダムインスタンス生成(FUNの結果)

一様巡回群分解 (UCF)の定義

- 群Gの部分群の列$\vec{H} = (H_1, H_2, \ldots, H_k)$が$G$の**一様群分解**であるとは、

$$H_1 H_2 \cdots H_k := \{h_1 h_2 \cdots h_k \mid h_i \in H_i\}$$

を多重集合（重複を許す集合）としたとき、以下を満たすこと

  - $H_1 H_2 \cdots H_k$にGのすべての元が現れること
  - その重複度が一定であること（i.e. すべての元が同じ個数出現）

- H_iたちが**巡回群**のとき**一様巡回群分解 (UCF)**という

UCFの例

- 対称群S_3のUCFの例(H_1, H_2)
  - $H_1 = \langle (1,2) \rangle = \{e, (1,2)\}$
  - $H_2 = \langle (1,2,3) \rangle = \{e, (1,2,3), (1,3,2)\}$
  - $H_1 H_2 = \{e, (1,2), (1,2,3), (1,2)(1,2,3), (1,3,2), (1,2)(1,3,2)\}$
    $= \{e, (1,2), (1,2,3), (2,3), (1,3,2), (1,3)\} = S_3$

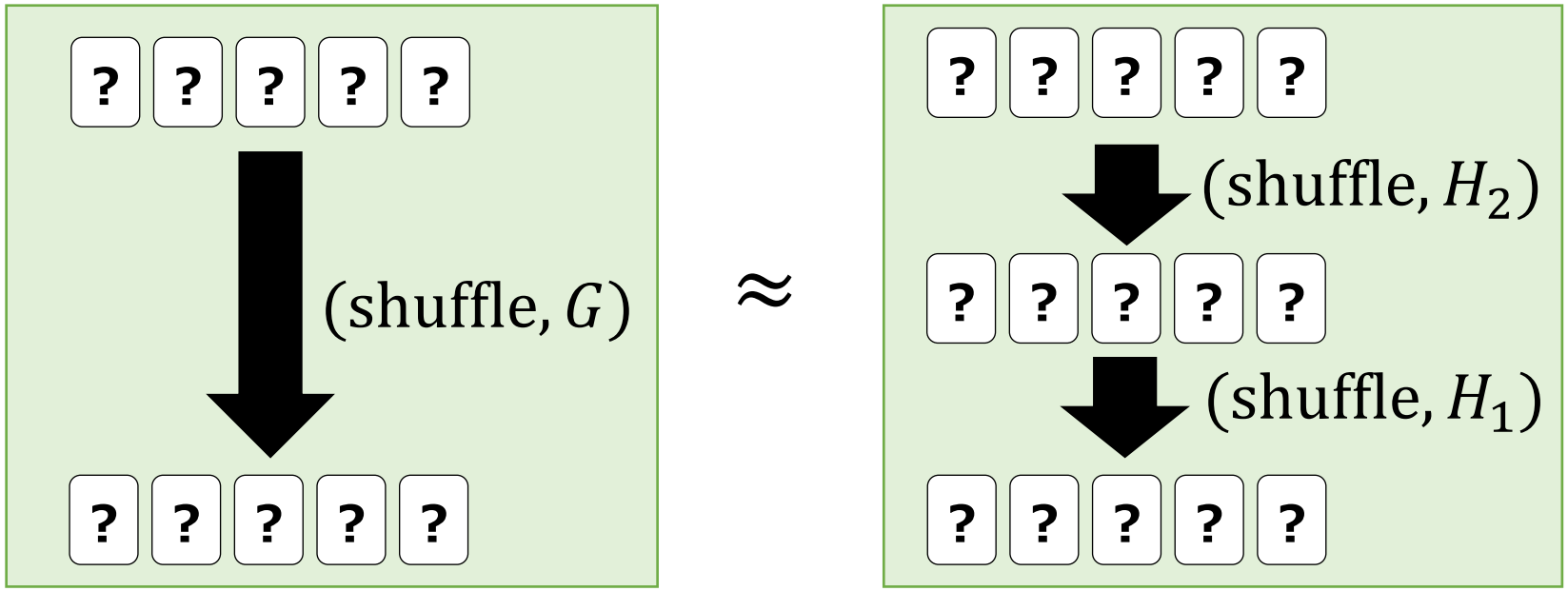
- 交代群A_4のUCFの例(H_1, H_2, H_3)
  - $H_1 = \langle (1,2)(3,4) \rangle = \{e, (1,2)(3,4)\}$
  - $H_2 = \langle (1,3)(2,4) \rangle = \{e, (1,3)(2,4)\}$
  - $H_3 = \langle (2,4,3) \rangle = \{e, (2,4,3), (2,3,4)\}$

目次

- イントロダクション
- 一様巡回群分解（UCF）について(Comm. Alg.の結果)
- **応用：一様閉シャッフルの実装(Comm. Alg.の結果 + FUNの結果)**
- 応用：パズルの秘匿ランダムインスタンス生成(FUNの結果)

一様閉シャッフルの分解

- 定理：群Gが長さkのUCF$(H_1, H_2, \dots, H_k)$を持つとする。
  このとき、一様閉シャッフル$(\mathrm{shuffle}, G)$はk個の巡回群シャッフル
  $$(\mathrm{shuffle}, H_k), \dots, (\mathrm{shuffle}, H_2), (\mathrm{shuffle}, H_1)$$
  に分解できる（逆順の理由は、置換の作用を$(h_1 h_2)(x) \coloneqq h_1(h_2(x))$としているため）



例：完全シャッフルの分解

- 対称群S_nはUCF$(H_n, H_{n-1}, \ldots, H_2)$を持つ
  - $H_n \coloneqq \langle(1\ 2\ 3\ \cdots\ n)\rangle$
  - $H_{n-1} \coloneqq \langle(1\ 2\ 3\ \cdots n-1)\rangle$
  - $H_2 \coloneqq \langle(1\ 2)\rangle$

- 完全シャッフル$(\mathrm{shuffle}, S_n)$は$n-1$回のRCに分解できる

どのような群がUCFを持つか？

- 現状わかっていること
  - 任意の可解群は一様群分解を持つ
    - 可解群の例：アーベル群 (可換群)、対称群S_3, S_4、奇数位数の群
  - もし任意の単純群が一様群分解を持てば、任意の群はUCFを持つ

- **UCF予想**：任意の群はUCFを持つ
  - YES $\Longrightarrow$ 任意の一様閉シャッフルは巡回群シャッフルで実装可能

目次

- イントロダクション
- 一様巡回群分解（UCF）について(Comm. Alg.の結果)
- 応用：一様閉シャッフルの実装(Comm. Alg.の結果 + FUNの結果)
- **応用：パズルの秘匿ランダムインスタンス生成(FUNの結果)**

シナリオ

- パズルの早解き競争は世界的に人気の競技である
  - スピードキュービング
  - 世界パズル選手権
  - 世界数独選手権

- スピードキュービングのインスタンス生成方法
  1. ソフトウェアを用いてスクランブル手順を生成（PCが実行）
  2. **スクランブラ**という役割の人が手順を実行する（人間が実行）

- 課題：PCと人間への信頼/負担を軽減できないか？

インスタンス生成に関する課題

- スクランブラに対してもインスタンスを秘匿できないか？
  - 現状、スクランブラはインスタンスを事前に知ってしまう
  - そのためスクランブラは競技に参加できない

- コンピュータを用いずにインスタンス生成はできないか？
  - 現状、スクランブル手順はコンピュータによって生成する
  - しかしコンピュータの動作の正しさの検証は必ずしも容易ではない

秘匿かつ一様ランダムなインスタンスを物理的に生成するには？

我々のアプローチ：UCFに基づくインスタンス生成

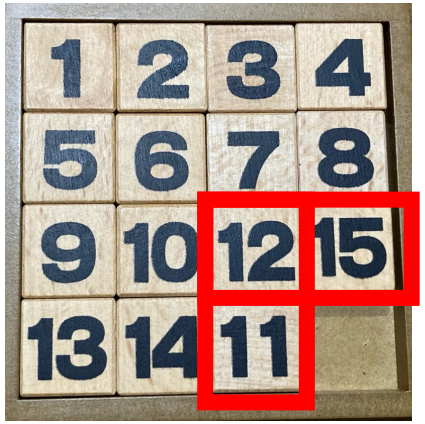
15パズルの場合

自明な方法とその問題点

- 自明な方法
  - 裏向きに伏せて一様ランダムに混ぜる
  - 競技開始と同時にオモテにする

- 問題点
  - 50%の確率で解が存在しない

解なし：奇置換(14,15)



解あり：偶置換(11,15,12)

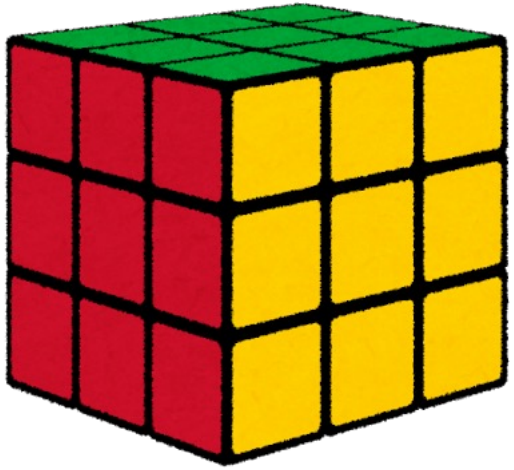
交代群A_{15}の一様閉シャッフルを実装できればよい

交代群シャッフルの実現方法

- 定理：A_nシャッフルは$n = 2m$のときは$3m - 3$回、 $n = 2m + 1$のときは$3m - 2$回のRC/RBC/PShiftに分解できる

1. RC：$\langle(1\ 2\ 3)\rangle$
2. RBC：$\langle(1\ 3)(2\ 4)\rangle$
3. RBC：$\langle(1\ 2)(3\ 4)\rangle$
4. RC：$\langle(1\ 2\ 3\ 4\ 5)\rangle$
5. RBC：$\langle(1\ 4)(3\ 6)\rangle$
6. PShift：$\langle(1\ 2\ 3)(4\ 5\ 6)\rangle$
7. RC：$\langle(1\ 2\ 3\ 4\ 5\ 6\ 7)\rangle$
8. RBC：$\langle(1\ 5)(4\ 8)\rangle$
9. PShift：$\langle(1\ 2\ 3\ 4)(5\ 6\ 7\ 8)\rangle$
10. RC：$\langle(1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10)\rangle$
11. RBC：$\langle(1\ 6)(5\ 10)\rangle$
12. PShift：$\langle(1\ 2\ 3\ 4\ 5)(6\ 7\ 8\ 9\ 10)\rangle$
13. RC：$\langle(1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11)\rangle$
14. RBC：$\langle(1\ 7)(6\ 12)\rangle$
15. PShift：$\langle(1\ 2\ 3\ 4\ 5\ 6)(7\ 8\ 9\ 10\ 11)\rangle$
16. RC：$\langle(1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12\ 13)\rangle$
17. RBC：$\langle(1\ 8)(7\ 14)\rangle$
18. PShift：$\langle(1\ 2\ 3\ 4\ 5\ 6\ 7)(8\ 9\ 10\ 11\ 12\ 13\ 14)\rangle$
19. RC：$\langle(1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12\ 13\ 14\ 15)\rangle$

ルービックキューブの場合



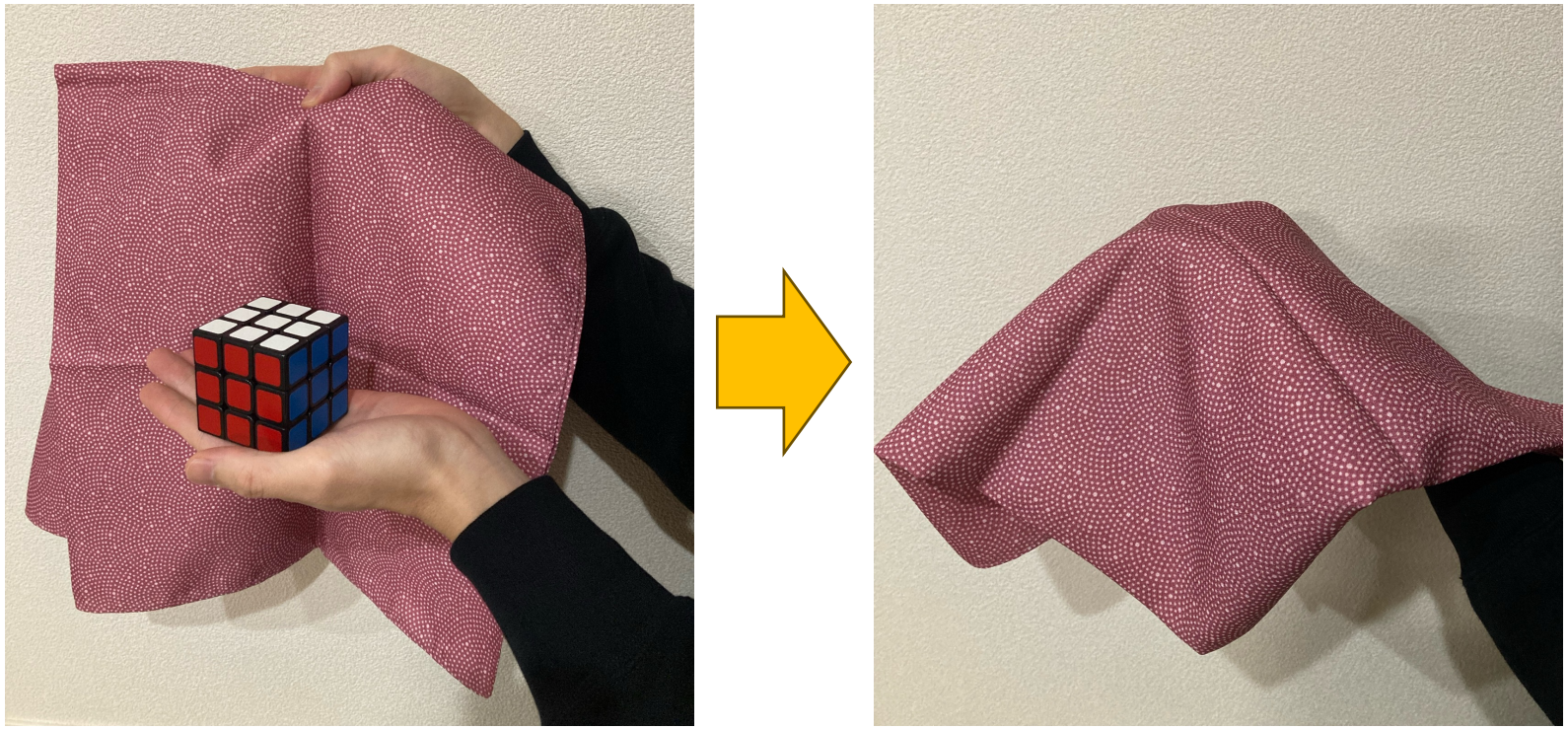
ルービックキューブのインスタンス生成

- 自明な方法：分解して、バラバラなピースをシャッフル
  - デタラメに並び替えると、解無しの場合もある（15パズルと同様）
  - 壊れる/傷がつく可能性もある

- ピースを外さない実装方法
  - 方法１：ルービックキューブの操作を用いてランダマイズする
  - 方法２：色シールをシャッフルして、最後に貼り付ける

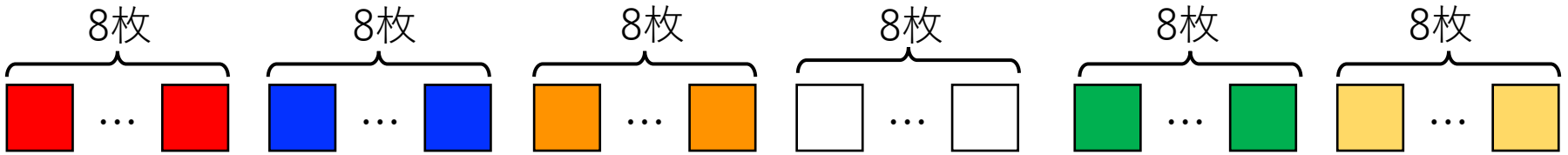
方法１：キューブ操作を用いる

1. 揃った状態のキューブを用意し、布でキューブを隠す
2. 布の中で手操作でランダマイズする

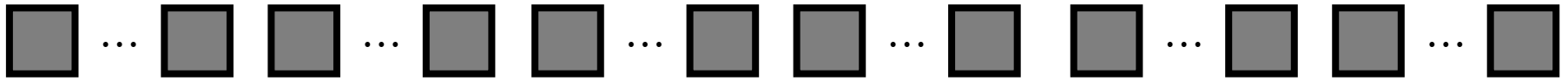


方法２：色シールをシャッフルする

1. 8枚×6色の色シールを用意する



2. 無色のカバーシールを貼る



3. ルービックキューブ群シャッフルを適用する

4. キューブにシールを貼る

ルービックキューブ群シャッフル（1/2）

- ルービックキューブ群の群構造
  - $G = (\mathbb{Z}_3^7 \times \mathbb{Z}_2^{11}) \rtimes ((A_8 \times A_{12}) \rtimes \mathbb{Z}_2)$（$\rtimes$は半直積）
  - 一様巡回群分解：$G = C_1 C_2 C_3 \cdots C_{43}$

- 方法2（色シール）の場合：43回のRBC/PShift
  1. RBC：$\langle (2\ 34)(4\ 10) \rangle$
  2. RBC：$\langle (2\ 34)(5\ 26) \rangle$

     ⋮

  43. RBC：$\langle (2\ 4)(34\ 10)(1\ 3)(35\ 27)(9\ 33) \rangle$

- 方法1（キューブ操作）の場合：繰り返し操作によるシャッフル

ルービックキューブ群シャッフル（2/2）

- 表記法：R, U, B等は右面/上面/背面の回転、 R', U', B'等は逆操作

- 「操作列の繰り返しによるシャッフル」を43回適用する

1. $\langle RLFU^2F'RL'UB^2U'F^2L^2U'F^2R^2B^2D' \rangle$
2. $\langle RUR'U'R'U'RURB'U'R^2URB \rangle$

⋮

43. $\langle U'R^2U'R^2DR^2D'F^2UF^2R^2 \rangle$

- 操作列は計算機代数システムGAPで求めた

まとめ

- 一様閉シャッフルは巡回群シャッフルから実装可能
  - 位数の小さな群はすでにUCFの構成が得られているものが多い
  - 一方、位数の大きな単純群（例：モンスター群）のシャッフルが将来用いられる可能性もあるが、その巡回群シャッフルへの分解は未解決

- 秘匿ランダムインスタンス生成問題は新しい研究分野
  - 興味のある方はぜひ参入していただけると嬉しいです

- 未解決問題
  - UCF予想は成り立つか？
  - 他のパズルに対する秘匿ランダムインスタンス生成