3Dプリンタによるオープン装置や特殊カードケースの作成と対称関数の秘密計算への適用

伊藤 優樹[1,a)]　四方 隼人[1]　水木 敬明[2]　菅沼 拓夫[2]

概要：著者らは 2023 年 7 月の CSEC 研究会にて，カードベース暗号プロトコルの実装に役に立つカードケースや装置を 3D プリンタで作成したことを報告した．本稿では引き続き，3D プリンタを活用した取り組みを報告する．具体的にはまず，（通常の 2 色カードで動く）Five-card Trick の最終ステップで 5 枚のカードを同時にめくることのできるオープン装置を作成したことを報告する．次に，前述の CSEC 研究会で報告した複雑なシャッフルを実現できるカードケースについて，紙面の都合で省略していた機能の説明を与えるとともに，このカードケースによりコミットメントの加算が効率的に実現でき，対称関数の秘密計算に有用であることを示す．

キーワード：カードベース暗号，3D プリンタ，秘密計算

Creation of Card-Open Device and Special Card Cases Using 3D Printer and an Application to Secure Computations of Symmetric Functions

Yuki Ito[1,a)]　Hayato Shikata[1]　Takaaki Mizuki[2]　Takuo Suganuma[2]

Abstract: The authors reported at the CSEC meeting held in July 2023 that we created card cases and devices using a 3D printer, which are valuable for implementing card-based cryptographic protocols. In this paper, we continue to present our efforts involving 3D printing. Specifically, we report the development of a "card-open" device capable of flipping five cards simultaneously in the final step of the five-card trick, which works on a two-color deck of cards. Furthermore, we provide explanations for the functionalities that were omitted in the previous report at the CSEC meeting due to space constraints, regarding the card cases capable of achieving complex shuffles. In addition, we demonstrate how this card cases enable an efficient commitment addition and proves to be useful for secure computations of symmetric functions.

Keywords: Card-based cryptography, 3D Printer, Secure computation

1. はじめに

カードベース暗号は物理的なカード組を用いて秘密計算等の暗号機能を実現するものであり，基本的にカードベース暗号プロトコルは，人間の手によってカード組を操作することで暗号機能を実現することが想定される．

筆者らは 2023 年 7 月の CSEC 研究会にて，3D プリンタを活用し，カードベース暗号プロトコルの実装に役に立ついくつかのカードケースや装置の設計・作成について報告している [17]．本稿はその概要紹介から始める．

[1] 東北大学大学院情報科学研究科
Graduate School of Information Sciences, Tohoku University
[2] 東北大学サイバーサイエンスセンター
Cyberscience Center, Tohoku University
[a)] yuki.ito.q7@dc.tohoku.ac.jp

1.1 3D プリンタを活用した著者らの既存研究

3D プリンタで次の 3 つの装置を作成した [17]．

(i) 部分開示用ケース

トランプカードの部分開示 [5] について，その操作を実現するための道具として，図 1 に示す部分開示用ケースを設計し作成した．これを用いることで，部分

開示の操作を容易にし，ミスによる情報漏洩を防ぐ．

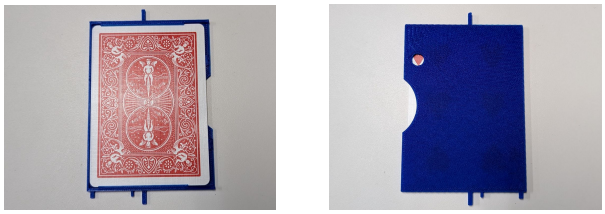


図 1: 部分開示用ケース

(ii) Five-Card Trick 用オープン装置

部分開示操作を活用すると，2 色カード組のプロトコルをトランプカードで実行することが可能となる．そこで，トランプカードで Five-Card Trick [1] を実行することを想定し，最終ステップで 5 枚をめくる際に利用できるオープン装置を設計・作成した（図 2 参照）．

図 2: Five-card Trick 用オープン装置

(iii) 特殊なシャッフル用ケース

複雑なシャッフルを実装するためのカードケースを，実際に設計し作成した（図 3 参照）．これを用いて，既存の 5 枚コピープロトコル [7] を実際に実行した．

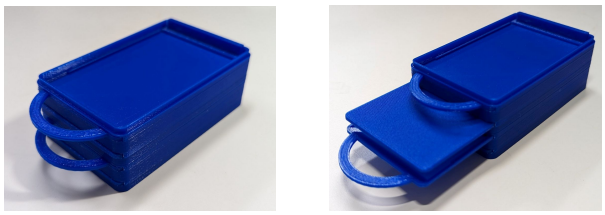


図 3: 特殊なシャッフル用ケース

1.2 本稿の貢献

1.1 節で紹介した著者らの既存研究 [17] において，(ii) の Five-card Trick 用オープン装置ではトランプカードを対象としている．一方，カードベース暗号では 2 色カード組が多く用いられている．

そこで本稿では，(ii) Five-card Trick 用オープン装置を拡張し，2 色カード組に対応したオープン装置を図 4 の通りに作成したことを報告する．Five-Card Trick はこれまで，大学でのオープンキャンパス等において高校生をはじめとする多くの一般市民によって 2 色カード組で実行されており [15]，この装置を活用することで Five-card Trick の魅力がさらに増すことが期待される．

また，(iii) の特殊なシャッフルケースについて，前回の

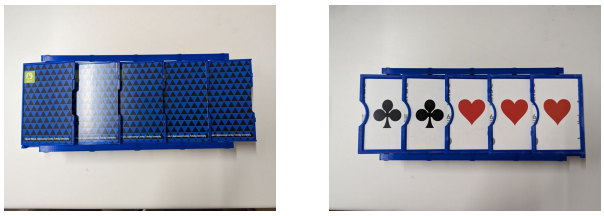


図 4: オープン装置

報告 [17] では紙面の都合によりいくつかの機能の説明等を省略していた．例えば，図 5 のように，2 つのケースをかきまぜた後，中の 2 つのカード束を 1 つの束に重ねる機構や補助装置の説明を省略していた．また，ケース上部にカードが収容可能である仕組みについては紹介に留まっており，この仕組みを利用した新しいカード操作やプロトコルの実現可能性・有用性については言及していない．

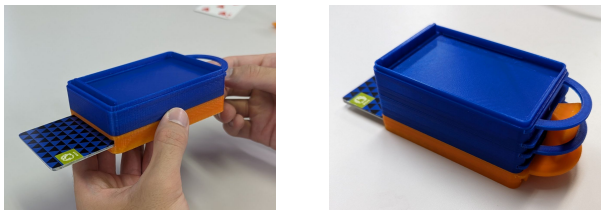


図 5: 取り出し装置を用いたカード束の取り出し

そこで本稿では，(iii) 特殊なシャッフルケースについて，紙面の都合で省略した機能を説明するとともに，このケースを活用することで，コミットメントの加算が効率的に実現でき対称関数の秘密計算に有用であることを報告する．

1.3 本稿の構成

本稿の残りの構成は次の通りである．まず 2 節において，準備として，使用するカードやシャッフル，既存のプロトコルの解説などを与える．次に 3 節において，通常の 2 色カード組で動く Five-card Trick のためのオープン装置の作成について説明する．次に 4 節において，特殊ケースの説明とそれを使った新しいカード操作である「ソート機能」について述べる．次に 5 節において，3D プリンタを使って特殊ケースを作成した様子を説明する．次に 6 節において，特殊ケースを使うことでコミットメントの加算が可能であることと，対称関数の秘密計算への応用を述べる．最後に 7 節で結論を述べる．

2. 準備

本節では，カードベース暗号で一般的に使用される操作や既存のプロトコルについて紹介する．

2.1 入力コミットメント

論理関数の秘密計算のためにはビットを扱える必要があり，カードベース暗号では多くの場合，1 ビットを ♣ と

♥ のペアを用いて

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

のように符号化する．

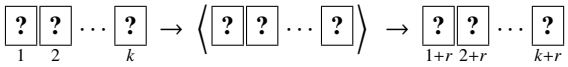
符号化ルール (1) の通り，1 ビットは 2 枚のカードで符号化されるため，各プレーヤー P_i は ♣ と ♥ を手に持ち，これらの 2 枚のカードを自分の入力 $x_i \in \{0,1\}$ に従って裏にした状態で置く（他のプレーヤーに並び順を秘密にして置く）．

これを**入力コミットメント**と呼ぶ．

2.2 ランダムカット

カードベース暗号において，もっとも基本的なシャッフル操作の一つである**ランダムカット**について紹介する．

ランダムカットとは，カード列に対してランダムに巡回的な並び替えを行う操作のことである．ランダムカットの適用を $\langle\rangle$ で表し，k 枚のカード列にランダムカットを適用すると，$r \in \{0,1,\ldots,k-1\}$ を乱数として，次のように遷移する．



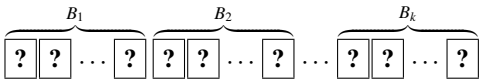
ただし，インデックスの数字が k を超えた場合は 1 に戻るものとする．

なお，ランダムカットは，「ヒンドゥーカット」と呼ばれる実装により現実世界で簡単に実現できる [14]．

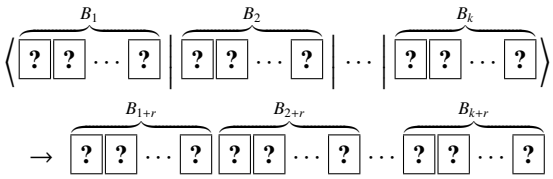
2.3 パイルシフティングシャッフル

本稿で提案するプロトコルでは使用しないが，（本稿のプロトコルと後ほど比較する）既存プロトコルで用いられる**パイルシフティングシャッフル** [6,11] を説明する．

このシャッフル操作では，カード列をいくつかの（同じサイズの）束に分けて，それらを束の単位でランダムカットする．いま，k 個の束 $B_1, B_2, \cdots, B_k$ があり，それぞれの束 B_i には同数のカードが含まれるとする．



これにパイルシフティングシャッフルを適用すると，乱数を $r \in \{0,1,\cdots,k-1\}$ として，以下のように遷移する．

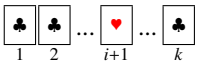


ただし，添え字が k を超えた場合は 1 に戻るものとする．

パイルシフティングシャッフルは，封筒や輪ゴム，あるいはスリーブを用いて束を固定し，それらに対してヒンドゥーカットを適用することで実装できる．

2.4 非負整数の表現

Ruangwises と Itoh [9,10] は，非負整数をカード列で表現するために以下のような符号化を考案した．$k \geq 2$ とし，$k-1$ 枚の ♥ カードと 1 枚の ♣ カードを用い，$i+1$ 番目に ♥ を挿入することで整数 i $(0 \leq i \leq k-1)$ を表現する．

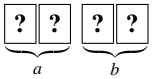


以降ではこのようなカード列が裏に置かれたものを $E_k^{\heartsuit}(i)$ で表す．また，色を逆転したものを $E_k^{\clubsuit}(i)$ と書く．

2.5 Five-card Trick

Five-card Trick [1] は，歴史上最初のカードベース暗号プロトコルであり，2 入力の AND の秘密計算を 5 枚のカードで実現する．

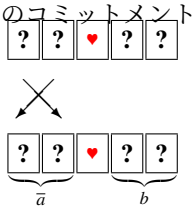
いま Alice と Bob が AND の秘密計算を行いたいとする．すなわち，それぞれ秘密のビット $a, b \in \{0,1\}$ を持っていて，論理積 $a \wedge b$ の値だけを知りたいとしよう．それぞれは符号化ルール (1) に従い，自身の秘密のビットのコミットメントを作成する．Alice と Bob のそれぞれの入力コミットメントを並べる．



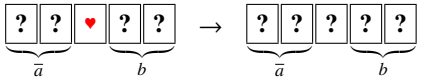
$a = 0$ のとき（$b = 0$ のとき）♣ ♥ の並びであり，$a = 1$ のとき（$b = 1$ のとき）♥ ♣ である．

これらのコミットメントの中央に赤いカードを 1 枚追加して，Five-card Trick は，次の手順で実行される．

(1) 先頭から 2 枚のカードを入れ替え，a のコミットメントをその否定 $\bar{a}$ のコミットメントに変換する．

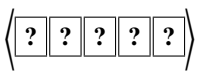


(2) 中央の ♥ カードを裏返す．



いま，$a \wedge b = 1$ のときに限り，中央は ♥ ♥ ♥ となることに注意しよう．

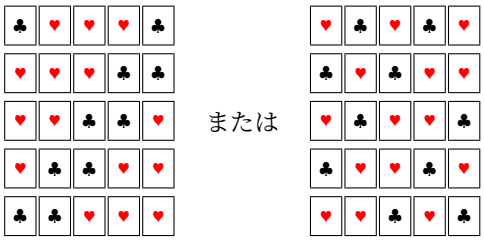
(3) 2.2 節で説明したランダムカットを適用する．



このとき，カード列はランダムな枚数だけシフトさ

れる．

(4) 5 枚のカード全てめくり，$a \wedge b$ の値を得る．



$a \wedge b = 1$　　$a \wedge b = 0$

この一連の操作によって，$a \wedge b$ の値のみを知ることができる．

2.6 非負整数の加算

Ruangwises と Itoh [9, 10] は，2.4 節で説明した，非負整数を表す裏に置かれたカード列が 2 つ与えられたとき，それらの加算を行う手法を次のように提案した（この手法のベースは Shinagawa らのアイデア [12] となっている）．

(1) 2 つの非負整数 a, b を表すカード列 $E_k^{\clubsuit}(a)$ と $E_k^{\heartsuit}(b)$ がある．説明のため，それぞれのカード列の各カードに次のように名称を付ける．

$$E_k^{\clubsuit}(a): \underset{x_0}{?}\,\underset{x_1}{?}\cdots\underset{x_{k-1}}{?} \qquad E_k^{\heartsuit}(b): \underset{y_0}{?}\,\underset{y_1}{?}\cdots\underset{y_{k-1}}{?}$$

(2) これらを以下のように並び替える．

$$\overset{y_{k-1}}{\underset{x_0}{?}}\,\overset{y_{k-1}}{\underset{x_0}{?}}\;\overset{y_{k-2}}{\underset{x_1}{?}}\,\overset{y_{k-2}}{\underset{x_1}{?}}\cdots\overset{y_0}{\underset{x_{k-1}}{?}}\,\overset{y_0}{\underset{x_{k-1}}{?}}$$

(3) このカード列に 2.3 節で説明したパイルシフティングシャッフルを適用する．カード列は r を乱数として次のように遷移する．

$$\left[\overset{y_{k-1}}{\underset{x_0}{?}}\,\overset{y_{k-1}}{\underset{x_0}{?}}\;\middle|\;\overset{y_{k-2}}{\underset{x_1}{?}}\,\overset{y_{k-2}}{\underset{x_1}{?}}\;\middle|\cdots\middle|\;\overset{y_0}{\underset{x_k-1}{?}}\,\overset{y_0}{\underset{x_k-1}{?}}\right]$$

$$\downarrow$$

$$\overset{y_{k-1-r}}{\underset{x_{0+r}}{?}}\,\overset{y_{k-1-r}}{\underset{x_{0+r}}{?}}\;\overset{y_{k-2-r}}{\underset{x_{1+r}}{?}}\,\overset{y_{k-2-r}}{\underset{x_{1+r}}{?}}\cdots\overset{y_{0-r}}{\underset{x_{k-1+r}}{?}}\,\overset{y_{0-r}}{\underset{x_{k-1+r}}{?}}$$

(4) これらを元のように並べ替え直す．

$$E_k^{\clubsuit}(a-r): \underset{x_{0+r}}{?}\,\underset{x_{1+r}}{?}\cdots\underset{x_{k-1+r}}{?}$$
$$E_k^{\heartsuit}(b+r): \underset{y_{0-r}}{?}\,\underset{y_{1-r}}{?}\cdots\underset{y_{k-1-r}}{?}$$

ここで，a にはランダムな値 r が減算され，b には r が加算されている．

(5) $E_k^{\heartsuit}(b+r)$ のカード列をめくり，その数値分（$s = b + r$）だけ $E_k^{\clubsuit}(a-r)$ のカード組を右に巡回的にシフトする（$a - r$ に s を加算する）．$E_k^{\heartsuit}(b+r)$ のカード列をめくったとき，b にはランダムな値 r が加算されているため，b の値は漏れない．

$$E_k^{\clubsuit}(a-r): \underset{x_{0+r}}{?}\,\underset{x_{1+r}}{?}\cdots\underset{x_{k-1+r}}{?}$$
$$\downarrow$$
$$E_k^{\clubsuit}(a+b): \underset{x_{0+r-s}}{?}\,\underset{x_{1+r-s}}{?}\cdots\underset{x_{k-1+r-s}}{?}$$

これにより，a と b の値を漏らすことなく $(a-r)+(b+r) = a+b$ を秘密計算できる．すなわち，$E_k^{\clubsuit}(a+b)$ が得られる．

$E_k^{\clubsuit}(a)$ と $E_k^{\heartsuit}(b)$ で加算の方法を説明したが，$E_k^{\heartsuit}(a)$ と $E_k^{\clubsuit}(b)$ など，他の色の組み合わせでも同様に実行可能である．

3. Five-card Trick 用オープン装置の作成

1.1 節で紹介した (ii) Five-card Trick 用オープン装置 [17] は，トランプカードの部分開示操作の活用によるプロトコルの実装を想定し設計・作成されている．本節では，（トランプカードではなく）2.5 節で説明したように 2 色カード組を用いて Five-card Trick を実行する際，5 枚同時にめくる操作に対応した装置を 3D プリンタで作成したことを報告する．

まず，3.1 節において，2 色カード組の Five-card Trick 用オープン装置の設計と作成を説明する．次に，3.2 節において，作成した装置を実際に使用する様子を述べる．

3.1 オープン装置の設計と作成

オープン装置の設計において，既存研究 [17] における設計思想と同様に，家庭用の 3D プリンタのみで装置一式を作成できることとしている．これにより誰でも容易に本装置を作成できることが期待される．

設計するケースは，操作中にケースからカードが外れないようにしつつカードの脱着がスムーズであるよう，ケース入り口における導入のための角度，カードの脱落防止のための突起，カードを取り出しやすくするために切り欠きを設けるなどの工夫を行っている．

設計には 3D CAD ソフトウェア（Autodesk 製 Fusion360）を用いて，図 6 のように 3D オブジェクトの設計データを作成した．また，印刷は家庭用 3D プリンタ（Flashforge 製 Adventure4）を用いて，図 7 のように出力した．完成した装置は 1.2 節の図 4 の写真に示したものである．

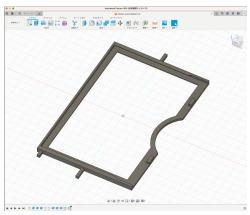


図 6: ケースの設計

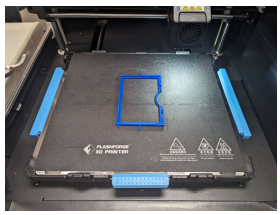


図 7: 出力の様子

3.2 オープン装置を用いた実装の様子

作成したオープン装置を用いて，2.5 節で紹介した AND

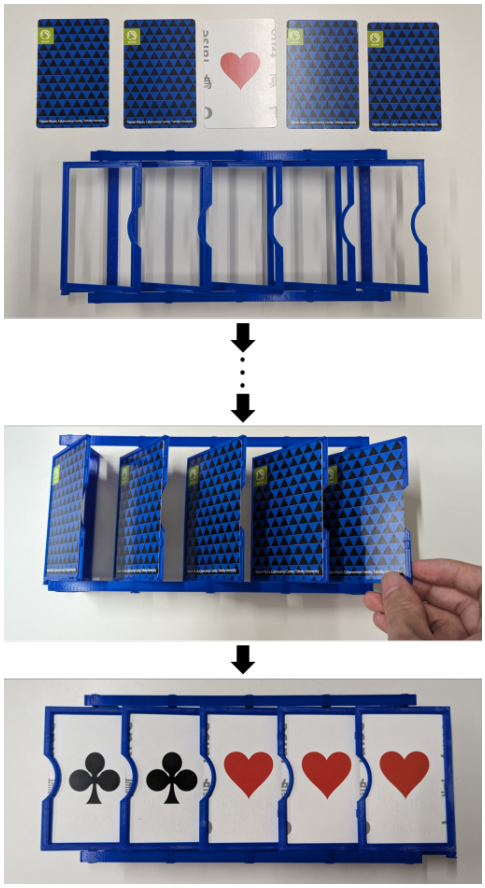


図 8: オープン装置を用いた Five-card Trick の実行

プロトコル，Five-card Trick を実際に実行し，最終ステップにおいて 5 枚を同時にめくった．その様子を図 8 に示す．

1 枚ずつカードを裏返して結果を得る場合と比較し，5 枚が 1 度に表向きになることで，実行時の楽しさとそれによる魅力の創出が期待できる．

4. 特殊ケースとソート機能

複雑なシャッフルを用いるとカード枚数やシャッフル回数の少ないプロトコルを構成できることが知られており [4,7,13]，特殊なカードケースを用いることで（現実世界で）実装できるであろうことが指摘されている [6,8]．本節では，そのような特殊ケースの仕様を確認し，さらにケースの上にカードを格納するという新しい機構を紹介し，その利用例を述べる．

4.1 特殊ケースの仕様

特殊ケースは，複雑な（閉じていない）シャッフル $(\mathsf{shuf}, \{\mathsf{id}, (1\,4\,2\,5\,3)\})$ の実装方法として初めて文献 [7] で言及され，図 9 のような形状をしている．

このカードケースは，図 10 のように重ねて，それぞれに入っていた中の（サイズの異なる 2 つの）カード束を 1 つの束に結合できる．そのため，次の条件を満たす必要がある．

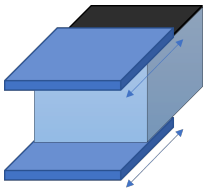


図 9: 特殊なカードケース

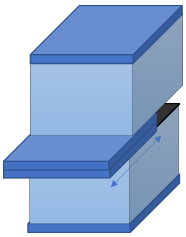


図 10: カード束の結合

1 上面と下面がそれぞれ独立してスライドし，中に格納したカードを取り出せる．
2 ケースはカードに比べて十分重く，カード枚数の差による重さの差は測定できない．

一方，ケース上部にカードを固定するという新しい手法に基づくプロトコルが文献 [16] において提案されている（図 11）．そこでは，輪ゴム等を用いてケース上部にカード固定する方法が示唆されている．

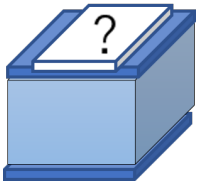
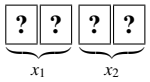


図 11: 特殊ケースの上にカードを置いた様子

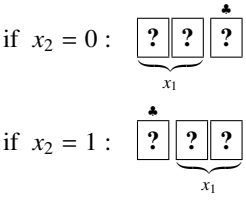
前述の通り，著者ら [17] は 3D プリンタでこのような特殊ケースを作成しており，今述べたケース上部にカードを固定するという機能にも対応している（詳しくは次節で説明する）．したがって，輪ゴム等で固定するよりも手軽に上述の手法が利用できることになる．

4.2 特殊ケースの利用の具体例

ここでは，特殊ケースを用いた操作の具体例として，$x_1, x_2 \in \{0,1\}$ のコミットメント



と  が与えられたとき，x_2 の値を秘密にしたまま，x_2 の値に応じて x_1 のコミットメントと裏向きの  を並べ替えられることを見る．すなわち，

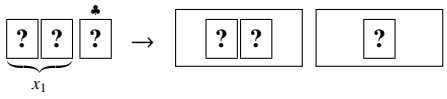


となるようにしたい．

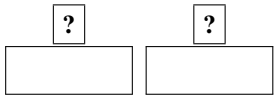
具体的な手順は次の通りである．

(1) 2 つの特殊ケースの上蓋を開け，x_1 のコミットメントと ♣ カードをそれぞれ裏向きにして中に置き，上蓋を

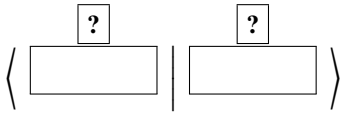
閉める．



(2) ケースの上部に x_2 のコミットメントから 1 枚ずつ裏向きでセットする．

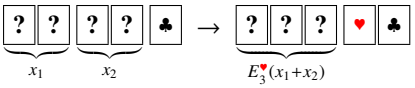


(3) 2 つの特殊ケースを（どちらがどちらか分からなくなるまで）かきまぜる．



(4) ケース上部のカードをめくり，♥ のケースが ♣ のケースの下になるように重ね，真ん中の仕切りを引き抜く．

この手順により，上述のことを実現できている．また，実はこの一連の結果，x_1 と x_2 を加算し，$E_3^{\heartsuit}(x_1+x_2)$ が得られている．



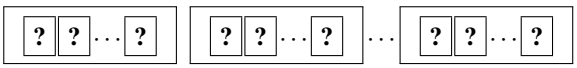
この手順は，$x_2=0$ の場合は ♣ カードをコミットメントの右側に，$x_2=1$ の場合は ♣ カードをコミットメントの左側にソートしていると言える．すなわち，特殊ケースを使用することによって，カード枚数の異なるカード列同士をこのように秘密裏にソートすることができる．

4.3 ソート機能

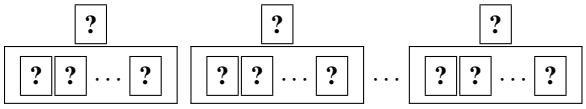
4.2 節の最後で示唆したように，特殊ケースを用いることでソートの機能を得ることができ，ここでは一般的に，1 から k までの番号カードを用いることで k 個の（サイズのふぞろいな）束に対して，ソートができること述べる．

いま，裏向きに置かれた k 枚の数字カードがあり，1 から k までの数字がなんらかの順序で並んでいるとする．加えて，サイズのふぞろいなカード束が k 個あるとする．

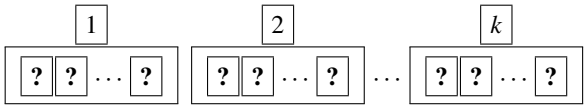
(1) 各特殊ケースにそれぞれカード束を入れる．



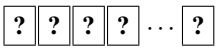
(2) 各特殊ケースの上部に，番号カードを裏向きでセットする．



(3) k 個特殊ケースを一様ランダムにかきまぜ，番号カードのみをめくって 1 から昇順になるように並べ替える．



(4) すべての特殊ケースを連結して仕切りを引き抜き，ソートされたカード列を得る．



サイズがすべて同じ束の列に対しては，文献 [3] のソート機能が使えるが，ふぞろいなサイズの束列には，本稿の手法が有効である．

5. 特殊ケースの作成

1.1 節で紹介した (iii) 特殊なシャッフル用ケースは，4.1 節に基づき設計が行われており，重ねた 2 つのケースの底と蓋のいずれかが開いたときにもう一方も連動して開くような構造としているほか，基本的な仕様に加え操作ミス等によるプロトコルの失敗を防止するための機構を設計においていくつか付与し 3D プリンタで作成している．前回の報告 [17] では紙面の都合によりいくつかの説明を省略したため，本節ではそれらについて報告する．

説明を省略した事項として，図 12 に示す，ケース中のカード束をまとめて安全に取り出す補助装置がある．ケース内で結合されたカード束を取り出す際，ケース上部の蓋を開けて取り出そうとするとカード束の一部が見えプロトコルの失敗の可能性がある．補助装置を利用することで，ケースを逆さまにせず安全にカードを取り出すことが可能である．また，操作回数の削減及び確実性を向上するためサポートも作成している．サポートを用いることでカード束の結合と取り出し操作を一度に行うことが可能であり操作を確実にしている．これらを用いてカード束の結合および取り出しを行う様子は図 5 に示した通りである．

また，文献 [16] において各ケースの上に裏向きカードを置く新しい操作が提案されているが，この操作の実現に対応するよう図 13 に示すようなケース上部にカードが収容可能な仕組みを設けている．これにより，シャッフル等の操作においてカードが外れる等のプロトコルの失敗が発生しないよう確実性を向上させている．

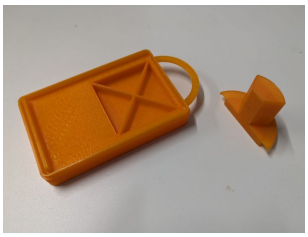


図 12: 補助装置とサポート

図 13: カード収容の機構

6. 特殊ケースによる対称関数の秘密計算

本節では，4 節で説明した特殊ケースを活用したプロトコルを提案する．すなわち，$2n+1$ 枚のカードと，2 つの特殊ケースを使用し，任意の多値出力の対称論理関数 $f:\{0,1\}^n \to R$ を秘密計算するプロトコルを与える．

6.1 対称関数の秘密計算

対称論理関数 $f:\{0,1\}^n \to \{0,1\}$ の出力値 $f(x_1,\ldots,x_n)$ は合計値 $\sum_{i=1}^n x_i$ のみに依存する．すなわち，ある関数 $g:\{0,1,\ldots,n\} \to \{0,1\}$ が存在して，$f(x_1,\ldots,x_n) = g\left(\sum_{i=1}^n x_i\right)$ となる．従って，対称関数 f を計算したい場合には合計値 $\sum_{i=1}^n x_i$ を求めれば良い．

このことから，カードベース暗号においては，入力コミットメント列から合計値を表すカード列 $E^{\heartsuit}_{n+1}(x_1+\cdots+x_n)$ を出力することが 1 つの典型であり，2.6 節で述べた非負整数の加算を用いると，追加カードが 2 枚あれば $E^{\heartsuit}_{n+1}(x_1+\cdots+x_n)$ を生成できる [10]．

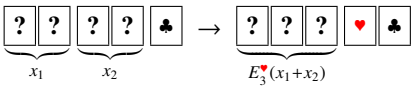
また，詳細は省略するが，合計値を表すカード列が得られた後は，任意の対称論理関数 f に対応する上述の関数 g に基づき，カード列を分割してシャッフルしてめくることで，$f(x_1,x_2,\ldots,x_n)$ の値を得ることができる [10]．

6.2 特殊ケースを用いたコミットメントの加算

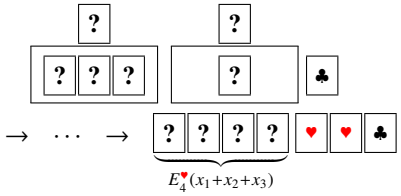
6.1 節で説明したように，本稿で提案するプロトコルでも，入力の合計値 $E^{\heartsuit}_{n+1}(x_1+x_2+\cdots+x_n)$ を秘密計算してから対称論理関数 f の値を得る．合計値から対称論理関数 f を出力する手順は上述の既存研究 [10] の方法と同じであるので，入力から合計値 $E^{\heartsuit}_{n+1}(x_1+x_2+\cdots+x_n)$ を生成する手順だけを与える．

n 個の入力コミットメントが与えられたとき，追加カード 1 枚と特殊ケース 2 個を用いて，次の手順を実行する．

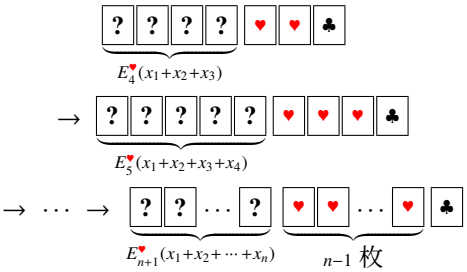
(1) 6 節で説明した手法で x_1 と x_2 のコミットメントを加算し，$E^{\heartsuit}_3(x_1+x_2)$ を得る．



(2) 左のケースに（1）で得られた $E^{\heartsuit}_3(x_1+x_2)$ を，右のケースに（1）で得られた ♣ カードをそれぞれ裏向きにして入れ，ケースの上部に x_3 のカードを 1 枚ずつ裏向きにしてセットし，6 節で説明した手法と同様にして加算を行う．



(3) 以降，同様の操作を繰り返し，x_n まで加算する．



既存研究 [10] では，追加カードを 2 枚要するが，本稿の手法では，2 つの特殊ケースを用いることでカード枚数が 1 枚削減されている．

また，既存手法 [10] では，カード束を作成する回数（束をスリーブや封筒に入れる回数）は $n^2/2+3n/2-2$ 回であるが，本手法の場合は，カード束を作成する回数（束を特殊ケースの中に入れる回数）は $2n-2$ 回であり，このような「束の作成」に要する手間・コストという観点において，本提案手法に利点がある．

なお，文献 [16] においても特殊ケースを用いた対称論理関数の秘密計算プロトコルを与えているが，本提案プロトコルの方がシンプルである．

6.3 5 入力多数決関数の実装

ここでは，提案プロトコルを実際に実行する様子を記述する．秘密計算する対称論理関数として，5 入力多数決関数を考えよう．

(1) 図 14 に示すように 5 個の入力コミットメントと 1 枚の ♣ カード，2 つの特殊ケースを用意する．

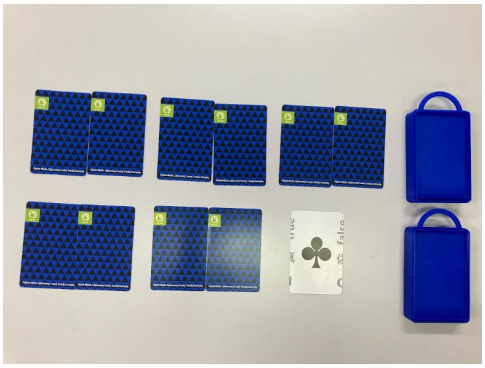


図 14: 5 入力多数決関数の計算に使用するセット

(2) 図 15 のように，2 つに特殊ケースの中にそれぞれ x_1 のコミットメントと ♣ カードを入れ，上部に x_2 のコミットメントのカードを 1 枚ずつ裏向きでセットし，x_1 と x_2 を加算する．

(3) 同様に加算を繰り返し，図 16 のように $E^{\heartsuit}_6(x_1+x_2+\cdots+x_5)$ を得る．

(4) 得られた合計値の左側 3 枚と右側 3 枚をそれぞれシャッフルし，めくる．例えば，右側 3 枚に ♥ カードがある

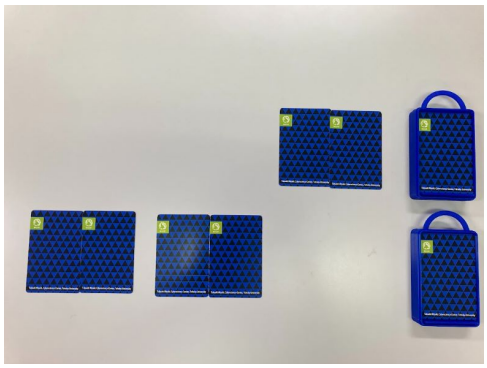


図 15: $x_1 + x_2$ の計算の様子

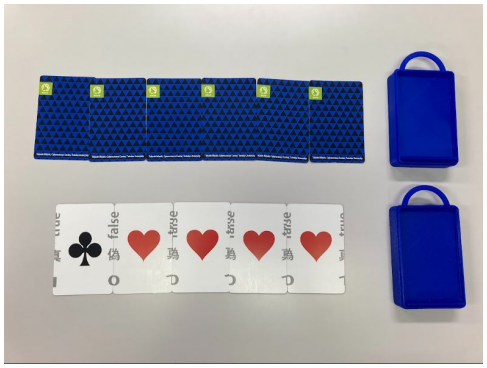


図 16: x_5 まで加算が終了した様子

場合は，合計値は少なくとも 3 以上であることが分かり，5 入力多数決関数の出力が 1 と決まる．

このように，特殊なカードケースを使用することによって，対称論理関数の秘密計算の現実世界での実装が可能となる．なお，一連の操作で特殊ケースの中にカードを入れる回数（束を作る回数）は 8 回である．既存手法 [10] の場合，束を作る回数は 18 回であり，提案プロトコルの方が手間がかからないと言える．

7. おわりに

本稿では，2 色カード組に対応したオープン装置の作成，および前回の報告 [17] で省略した特殊ケースの機能や実際の活用例を報告した．

今後の課題として，オープン装置の拡張によるシャッフル操作の自動化，特殊ケースの構造的な工夫が挙げられる．具体的には，現在手動で行っているランダムカット等のシャッフル操作について，オープン装置の拡張による自動化が考えられる．また，特殊ケースについて，前回の報告でも述べている通り，ケースを動かす際に中のカードと衝突し音が発生する点について構造的な工夫が望まれる．

最後に，[17] でも言及したように，秘密計算あるいは高機能暗号の社会への普及のためには，それらの暗号機能の意味するところや意義が幅広いステークホルダーに理解される必要があり [2, 15]，カードベース暗号とその実装がそのような一助となることを期待している．

謝辞 JP21K11881 と JP23H00479 の助成を受けている．

参考文献

[1] Den Boer, B.: More Efficient Match-Making and Satisfiability The Five Card Trick, *Advances in Cryptology—EUROCRYPT '89*, LNCS, Vol. 434, Berlin, Heidelberg, Springer, pp. 208–217 (1990).
[2] Hanaoka, G.: Towards User-Friendly Cryptography, *Paradigms in Cryptology–Mycrypt 2016. Malicious and Exploratory Cryptology*, LNCS, Vol. 10311, Cham, Springer, pp. 481–484 (2017).
[3] Koch, A. and Walzer, S.: Private Function Evaluation with Cards, *New Gener. Comput.*, Vol. 40, pp. 115–147 (2022).
[4] Koch, A., Walzer, S. and Härtel, K.: Card-Based Cryptographic Protocols Using a Minimal Number of Cards, *Advances in Cryptology—ASIACRYPT 2015*, LNCS, Vol. 9452, Berlin, Heidelberg, Springer, pp. 783–807 (2015).
[5] Miyahara, D. and Mizuki, T.: Secure Computations through Checking Suits of Playing Cards, *Frontiers in Algorithmics*, LNCS, Vol. 13461, Cham, Springer, pp. 110–128 (2022).
[6] Nishimura, A., Hayashi, Y., Mizuki, T. and Sone, H.: Pile-shifting scramble for card-based protocols, *IEICE Trans. Fundam.*, Vol. 101, No. 9, pp. 1494–1502 (2018).
[7] Nishimura, A., Nishida, T., Hayashi, Y., Mizuki, T. and Sone, H.: Five-Card Secure Computations Using Unequal Division Shuffle, *Theory and Practice of Natural Computing*, LNCS, Vol. 9477, Cham, Springer, pp. 109–120 (2015).
[8] Nishimura, A., Nishida, T., Hayashi, Y., Mizuki, T. and Sone, H.: Card-based protocols using unequal division shuffles, *Soft Comput.*, Vol. 22, pp. 361–371 (2018).
[9] Ruangwises, S. and Itoh, T.: Securely Computing the *n*-Variable Equality Function with 2*n* Cards, *Theory and Applications of Models of Computation*, LNCS, Vol. 12337, Cham, Springer, pp. 25–36 (2020).
[10] Ruangwises, S. and Itoh, T.: Securely computing the n-variable equality function with 2n cards, *Theor. Comput. Sci.*, Vol. 887, pp. 99–110 (2021).
[11] Shinagawa, K., Mizuki, T., Schuldt, J., Nuida, K., Kanayama, N., Nishide, T., Hanaoka, G. and Okamoto, E.: Card-Based Protocols Using Regular Polygon Cards, *IEICE Trans. Fundam.*, Vol. E100.A, No. 9, pp. 1900–1909 (2017).
[12] Shinagawa, K., Mizuki, T., Schuldt, J. C. N., Nuida, K., Kanayama, N., Nishide, T., Hanaoka, G. and Okamoto, E.: Multi-party Computation with Small Shuffle Complexity Using Regular Polygon Cards, *Provable Security*, LNCS, Vol. 9451, Cham, Springer, pp. 127–146 (2015).
[13] Shinagawa, K. and Nuida, K.: A single shuffle is enough for secure card-based computation of any Boolean circuit, *Discrete Applied Mathematics*, Vol. 289, pp. 248–261 (2021).
[14] Ueda, I., Nishimura, A., Hayashi, Y., Mizuki, T. and Sone, H.: How to Implement a Random Bisection Cut, *Theory and Practice of Natural Computing*, LNCS, Vol. 10071, Cham, Springer, pp. 58–69 (2016).
[15] 花岡悟一郎，岩本　貢，渡邉洋平，水木敬明，安部芳紀，品川和雅，新井美音，矢内直人：高機能暗号の社会展開を促進する物理・視覚暗号，電子情報通信学会論文誌 A, Vol. J106-A, No. 8, pp. 214–228 (2023).
[16] 四方隼人，豊田航大，宮原大輝，水木敬明：最小のカード枚数による対称関数の秘密計算について，2022 年暗号と情報セキュリティシンポジウム,1F4-3 (2022).
[17] 伊藤優樹，四方隼人，葛馬知紀，水木敬明，菅沼拓夫：3D プリンタのカードベース暗号実装への活用，情報処理学会研究報告コンピュータセキュリティ研究会（CSEC），Vol. 2023-CSEC-102, No. 10, pp. 1–8 (2023).