Vol.2022-CSEC-97 No.12
Vol.2022-IOT-57 No.12
2022/5/19

AND拡張とXOR拡張の3値入力2者間カードベースプロトコルの関係について

須賀 祐治[1,a)]

概要：2 者間の AND 演算によるマッチングはカードベースプロトコルにおける一般的なアプリケーションであり，気まずくならない告白ができることが知られている．2 者間の秘密計算によって AND 演算出力が 0 である場合，入力が 0 だったのか 1 だったのかを秘匿できる意味で，相手に入力がバレないことから気まずくならないとされている．0, 1 という 2 択の入力を持つ通常の AND 演算を拡張し，0 でも 1 でもない第 3 の値「不定」を入力可能な拡張 AND プロトコルが SITA2021，SCIS2022 で提案されている．また XOR 演算に対する同様の拡張についても 84 回 IPSJ 全国大会 (2022 年 3 月) にて提案されており，それぞれ可換な半群の分類が与えられている．
本稿は上記の AND 拡張と XOR 拡張の可換な半群において，代数的に同値であることを見出すことで，カードベースプロトコルの実装として見るとそれぞれ互換性のある 2 つの事例について報告する．プロトコルの手順としては全く同じままで，エンコーディングルールのみを変更することにより，真偽表の異なるカードプロトコルを構成することができることを示す．

キーワード：Card-based protocols, Non-committed protocols, Five Card Trick, Commutative semigroups

Relationship between AND extension and XOR extension of 3-valued input with 2-party card-based protocols

Yuji Suga[1,a)]

Abstract: The matching situation with AND operations between two parties is a common application in card-based protocols, and it is known to provide a non-embarrassing confession of love. This means that the other party does not know whether the input was 0 or 1, which is said to avoid embarrassment. Extended AND protocols that extend the normal AND operation with two input choices of 0, 1 to allow a third "indeterminate" value that is neither 0 nor 1 has been proposed in SITA2021 and SCIS2022. Similar extensions to the XOR operation were also proposed at the 84th national Convention of IPSJ in March 2022, each of which is given commutative semigroup classifications.
By finding algebraic isomorphism in the commutative semigroups of AND and XOR extensions discussed above, this paper presents two examples of compatible implementations of the card-based protocols. We also show that it is possible to construct card protocols with different boolean tables by changing only the encoding rules while keeping the same procedures exactly.

[1] 株式会社インターネットイニシアティブ
Internet Initiative Japan Inc., Iidabashi Grand Bloom, 2-10-2 Fujimi, Chiyoda-ku, Tokyo 102-0071, Japan
[a)] suga@iij.ad.jp

Vol.2022-CSEC-97 No.12
Vol.2022-IOT-57 No.12
2022/5/19

1. はじめに

1.1 本稿で扱う対象

本稿では表面裏面ともに全く同じ絵柄であるカード（例えば名刺や麻雀牌）を用いることを考える（文献 [2] において水木らは絵柄の上下関係を生かしたメリットとデメリットについて考察されている）．このときカードの上下配置の違いを用いて，それぞれ（一般的なカードプロトコルで用いられる）異なるスートと対応づけることができる．つまり $\boxed{\downarrow}$ を $\boxed{\clubsuit}$ と，$\boxed{\uparrow}$ を $\boxed{\heartsuit}$ と同一視することを考える．スートを表現するメモ書きをしないでも，同一カードの束を用いてプロトコルを構成することができる点も一つのメリットである．

1.2 非コミットメント型プロトコル

本稿はカードベースプロトコルのうち非コミット型のプロトコルを扱う．一般的なカードベース暗号では 1 ビット入力を 2 種類 2 枚のカードが用いられる [1]．例えば，ユーザによる 1 ビット入力は以下の一般的なエンコーディングルールに従う：$\boxed{\clubsuit}\boxed{\heartsuit} = 0$, $\boxed{\heartsuit}\boxed{\clubsuit} = 1$.

出力がコミット型であるとは，プロトコル停止時に得られる結果が，入力のエンコーディングルールに基づいた形式であることを指す．一方で非コミット型であるとは，プロトコル停止時に利用されたカードを開示するなどして結果を得る方式である．

1.3 オリジナル Five-Card Trick

2 ユーザによる非コミットメント型として知られる Five-card trick [3] はハートとクラブ 2 種類のカードが用いられている．

Five-card trick は 2 ユーザ間で AND 演算を行うプロトコルである．2 入力を $a, b \in \{0,1\}$ としたとき $\boxed{?}\boxed{?}(=\overline{a})$ $\boxed{\heartsuit}$ $\boxed{?}\boxed{?}(=b)$ として 5 枚のカードを並べてランダムカット（巡回置換を c_5 としたとき，恒等置換 id と c_5, c_5^2, c_5^3, c_5^4 の 5 通りから等確率で選択してカード束に処理する操作）を行う．ここで $\boxed{?}$ は裏面にして入力したことを示しており $\overline{a}$ は a の否定 (negation) である．

ランダムカットを行う際には中央の $\boxed{\heartsuit}$ も $\boxed{?}$ とし，5 枚とも裏面に向けてシャッフルする．出力は 5 枚のカードをすべて開示することで得られる．3 枚の $\boxed{\heartsuit}$ が連続して並んで出力されたとき $a \wedge b = 1$，それ以外は $a \wedge b = 0$ となる．以下は 5 枚のカードの初期状態を示しており，これらの 5 枚のカードが巡回置換によりシャッフルされることから，出力時に 3 枚の $\boxed{\heartsuit}$ が連続して並んでいる場合のみが $a \wedge b = 1$ となることが分かる．さらに，$a \wedge b = 0$ となる 3 つのケースについてはすべて同一視されるため，出力だけを見ても入力 a, b がどのような値だったかについて認識できない点が本プロトコルのポイントである．

(a,b)	sequence
(0,0)	♡ ♣ ♡ ♣ ♡
(0,1)	♡ ♣ ♡ ♡ ♣
(1,0)	♣ ♡ ♡ ♣ ♡
(1,1)	♣ ♡ ♡ ♡ ♣

表 1 Five-Card Trick 初期入力状態

Five-Card Trick は，カード入力時の一般置換（このケースでは b を入力の際にカードを倒置して置いているため 5 枚のカード全体として $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 5 & 4 \end{pmatrix}$ の置換を行っていると考える）とランダムカットのみで構成されるシンプルなプロトコルである．

1.4 上下シャッフルの導入と Three Card Trick

Five Card Trick と同様に 2 者の AND 演算プロトコルを考える．一般的なカードプロトコルにおいては 2 種類のスート各 1 枚の計 2 枚が配布され，2 枚のカードで 1 ビットを表現する．そのため配布される最小枚数は 2 枚である．一方で，表面裏面ともに全く同じ絵柄であるカードを用いる場合には 1 枚のカードを上下関係，つまり 2 種類の方向 $\boxed{\downarrow}$ $\boxed{\uparrow}$ で入力可能なことから 1 ビットを 1 枚で表現可能である．そのため配布される最小枚数は 1 枚であり，1 枚のカードを配布した際の AND プロトコルが構成できれば，枚数としては optimal な方式であると言える．

例えば $\boxed{\downarrow} = 0$, $\boxed{\uparrow} = 1$ というエンコーディングルールを適用しようとする場合 5 Card Trick のような入力，つまり真ん中のエクストラカードを $\boxed{\uparrow}$ とし，両側から各ユーザが a, b 1 枚ずつの裏面入力を行うと初期状態として以下のような配置となる．仮にこれを Three Card Trick と呼ぶこととする．

(a,b)	sequence
(0,0)	↓ ↑ ↓
(0,1)	↓ ↑ ↑
(1,0)	↑ ↑ ↓
(1,1)	↑ ↑ ↑

表 2 Three-Card Trick 初期入力状態

Five Card Trick と同じように入力を撹乱するため 3 枚カードをランダムカットのカード処理を行うが，出力時には $a \wedge b = 1$ のときのみ エクストラカードを含めて $\boxed{\uparrow}$ が 3 枚並んでいることが分かる．一方で Five Card Trick において $a \wedge b = 0$ となる 3 つのケースはすべて同一視できていたが表 2 のように $\boxed{\uparrow}$ となるカードの枚数が異なることから同一視できず，このままでは入力 a, b を秘匿できない．

そのため次のテクニックを用いる．アイデアとしてはランダム 2 等分カット [5] を勘弁に実装する方式で用いられている方式によく似通っており，上下関係をランダムに入れ替えること（この操作を上下シャッフルと呼ぶ）を考える．ひとつの方法としてはランダムカット後のカード束にさらにエクストラカード 1 枚を用いて 3 枚のカードのうちの 1 枚めの表面を秘匿し，放り投げる等して上下関係を入れ替えた後にエクストラカードを抜き去るという方式が考えられる．ここは様々な実装方式があると考えるが，いずれにせよ上下シャッフルは $\boxed{\downarrow}$ と $\boxed{\uparrow}$ が入れ替わることを意味しており Three Card Trick では以下のように初期入力状態が変化することとなる．

(a,b)	sequence
(0,0)	$\boxed{\uparrow}\boxed{\downarrow}\boxed{\uparrow}$
(0,1)	$\boxed{\uparrow}\boxed{\downarrow}\boxed{\downarrow}$
(1,0)	$\boxed{\downarrow}\boxed{\downarrow}\boxed{\uparrow}$
(1,1)	$\boxed{\downarrow}\boxed{\downarrow}\boxed{\downarrow}$

表 3 Three-Card Trick にて上下シャッフル後の初期配置

表 2 と 表 3 の状態をすべて見比べたとき，ランダムカット→上下シャッフル後の状態は $\boxed{\uparrow}$ となるカードが 0,1,2,3 枚のいずれかとなり，$a \wedge b = 1$ のときは $\boxed{\uparrow}$ が 0,3 枚のいずれかである．さらに $a \wedge b = 0$ のときは $\boxed{\uparrow}$ が 1,2 枚であり，かつ 入力 a, b が秘匿された状態で出力を得ることができる．つまり $\boxed{\downarrow}\boxed{\downarrow}\boxed{\uparrow}$, $\boxed{\downarrow}\boxed{\uparrow}\boxed{\downarrow}$, $\boxed{\uparrow}\boxed{\downarrow}\boxed{\downarrow}$, $\boxed{\downarrow}\boxed{\uparrow}\boxed{\uparrow}$, $\boxed{\uparrow}\boxed{\downarrow}\boxed{\uparrow}$, $\boxed{\uparrow}\boxed{\uparrow}\boxed{\downarrow}$ の 6 通りはすべて同一視されることから本プロトコルの安全性（入力の秘匿性）を確保していることが分かる．また，Three Card Trick は入力カード枚数として optimal な AND 演算プロトコルを実現していることが分かる．

同一視できる 3 枚組に関して以下のタイプに分類することができる．

タイプ	同一視されるカード組
1	$\boxed{\downarrow}\boxed{\downarrow}\boxed{\uparrow}$, $\boxed{\downarrow}\boxed{\uparrow}\boxed{\downarrow}$, $\boxed{\uparrow}\boxed{\downarrow}\boxed{\downarrow}$, $\boxed{\downarrow}\boxed{\uparrow}\boxed{\uparrow}$, $\boxed{\uparrow}\boxed{\downarrow}\boxed{\uparrow}$, $\boxed{\uparrow}\boxed{\uparrow}\boxed{\downarrow}$
3	$\boxed{\uparrow}\boxed{\uparrow}\boxed{\uparrow}$, $\boxed{\downarrow}\boxed{\downarrow}\boxed{\downarrow}$

表 4 ランダムカット→上下シャッフル後の分類（3 枚組）

タイプ名は何枚 $\boxed{\uparrow}$ カードが連続するか最大の値を示す．例えばタイプ 3 にカテゴライズされる $\boxed{\uparrow}\boxed{\uparrow}\boxed{\uparrow}$ がそれに該当し，上下シャッフルにより $\boxed{\downarrow}\boxed{\downarrow}\boxed{\downarrow}$ と同一視されることも分かる．

1.5 本稿の貢献

上下シャッフルのテクニックを導入することにより，従来のカードプロトコルにおいて 2 枚で 1 ビットを表現する方式ではなく，第 3 の入力を入力することができることが示されている [12], [13]．またその際には 3 値入力時の真偽表を AND 真偽表と XOR 真偽表のいずれかから拡張し，かつプロトコルの連続性を考えた際に可換半群であること，つまり推移律を満たすように拡張するという制約を設けることで通常利用されないようなケースを排除し，過多な分類を避ける効果を狙っている．結果的には AND 拡張として 7 種類，XOR 拡張として 2 種類に集約できることが示されているが，すべてのケースにおいてカードプロトコルの実装が可能かどうかについては分かっていないケースも存在する．

そこで本稿では AND 拡張と XOR 拡張のケースから代数的に同値である相互関係のある事例を見い出し，実際にカードプロトコルとして実装可能であることを示す．具体的には XOR 拡張の 2 つの場合 XOR-(θ,θ,θ), XOR-$(0,\theta,1)$ のそれぞれについて，AND 拡張である AND-$(0,1,\theta)$, AND-$(\theta,0,\theta)$ と代数的に同型であることが分かった．これは AND 拡張のスキームが実装できれば XOR 拡張のスキームが実装可能であることを示しており，実際第 84 回 IPSJ 全国大会にて XOR-(θ,θ,θ) の実装が提示されていることから AND-$(0,1,\theta)$ が実装可能であることを示すこととなる．

本稿により相互性を示したことにより AND 拡張の 7 つの方式において，実装方法が示されていない AND-$(\theta,0,\theta)$, AND-$(0,0,\theta)$, AND-$(0,1,\theta)$, AND-$(0,\theta,0)$ の 4 方式が Open Problem として残されている．このうち XOR-$(0,\theta,1)$ の実装が示されれば AND-$(\theta,0,\theta)$ が実装可能であるため AND 拡張と XOR 拡張の両方面からの実装の検討が今後可能となる．

2. 3 値入力と 3 値論理

Five-Card Trick に限らずカードベースプロトコルの多くは 2 値 True(1), False(0) を入力することが想定されている．例えば 2 人による AND プロトコルは「気まずくならない告白」ができるとされており，実際プレーヤー A が True を入力して False という結果を得たとしても，相手のプレイヤー B には A が True を入力したことがバレることがない，という側面で気まずくならない点を保証している．

このようなシチュエーションを考えた場合，果たしてプレイヤーは 0 か 1 の 2 つの選択しか認めないのであろうか．相手に好意を寄せているのかどうかは 2 値ではなく，中間的な気持ちである「どちらでもない」「自分の気持ちが分からない」を排除してよいのか，という課題を考える [12], [13]．そこでカードプロトコルにおいて 0 でも 1 でもない第 3 の値を入力できるようにすることが可能にする．

2.1 3 値論理のバリエーション

第 3 の値を入力するとして 入力 a, b に対して $a \wedge b$ の真偽表としてどれを用いるかという問題を考える．従来の 2 値論理における AND 真偽表は以下となる．

$a\backslash b$	0	1
0	0	0
1	0	1

これに対して第 3 の値 θ を入れた際に様々な考え方が存在する．一例として，より代数的な考慮に基づいた Lukasiewicz の 3 値論理における AND 真偽表は以下となる．

$a\backslash b$	0	θ	1
0	0	0	0
θ	0	θ	θ
1	0	θ	1

0, 1 をそれぞれ体におけるゼロ元，単位元として考えたときのストレートフォワードな方式である．特に $\theta^2 = \theta$ を満たすように構成されている点に留意する．Kleene による 3 値論理においても AND 演算に関しては同じ真偽表を持つことが知られている．

次に Bochvar の 3 値論理における AND 真偽表を示す．

$a\backslash b$	0	θ	1
0	0	θ	0
θ	θ	θ	θ
1	0	θ	1

既に取り上げたように，この類のプロトコルでは「気まずくならない告白」ができるが，さらに「気持ちが揺らいでいる」ことも分かる，という観点で，ゲーム理論の側面としても面白い構造を持っていることが分かる．具体的には a または b のどちらかが θ を入力しただけで AND 演算の結果が θ となる点が面白い．しかし，これは秘密計算としては条件をバイオレーションしており，例えばプレイヤー A の入力が 0 または 1 のときには，マッチングがうまくいかない場合，プレイヤー B にその入力を秘匿することができるが，θ を入力してしまうと，入力が漏れてしまう．本稿ではこれをセキュリティ要件を満たさないという立場ではなく，このリスクを許容してプレイヤーは入力することを前提とする，という立ち位置で議論していく．

2.2 位数 3 の可換半群

ビルディングブロックとして 2 者間の拡張 AND 演算プロトコルまたは XOR 演算プロトコルを用い，複数のプロトコルを連続して実行することを想定すると，この拡張演算は推移律を満たす必要がある．さらに AND 演算，XOR 演算は可換であることから，位数 3 の可換半群がここで扱うべき対象となる．計算機による数え上げではなく手による証明を行い，自明な場合を取り除いた位数 3 の可換半群のうち AND の代数構造を持つ半群の真偽表は 7 種類となる [13] [15]．また XOR の代数構造を持つ半群の真偽表は 2 種類となる [14]

SCIS2022 にて AND-$(0, \theta, \theta)$, AND-$(\theta, 0, \theta)$, AND-$(0, 0, \theta)$ の 3 方式について optimal と考えられる方式について提示された．

2.3 AND-$(0, \theta, \theta)$ の実装

エンコーディングルールとして以下を適用する：$\boxed{\downarrow}\boxed{\uparrow} = 0$, $\boxed{\uparrow}\boxed{\downarrow} = \theta$, $\boxed{\downarrow}\boxed{\downarrow} = 1$. このとき真ん中にエクストラカード $\boxed{\uparrow}$を置いてその左側に a の negation, 右側に b を入力した場合，バリエーションとして表 5 の 9 パターンが得られる．ここで negation は左右を入れ替えたカードを入力することとする．Five Card Trick では 補数の入力を意味していたが，Five Card Trick においても negation は 2 枚のカードの左右を入れ替える操作と考えれば全く同じ操作であると考えることができる．

(a, b)	sequence
(0,0)	$\boxed{\uparrow}\boxed{\downarrow}\boxed{\uparrow}\boxed{\downarrow}\boxed{\uparrow}$
(0,1)	$\boxed{\uparrow}\boxed{\downarrow}\boxed{\uparrow}\boxed{\downarrow}\boxed{\downarrow}$
(1,0)	$\boxed{\downarrow}\boxed{\downarrow}\boxed{\uparrow}\boxed{\downarrow}\boxed{\uparrow}$
(1,1)	$\boxed{\downarrow}\boxed{\downarrow}\boxed{\uparrow}\boxed{\downarrow}\boxed{\downarrow}$
(0,θ)	$\boxed{\uparrow}\boxed{\downarrow}\boxed{\uparrow}\boxed{\uparrow}\boxed{\downarrow}$
(θ,0)	$\boxed{\downarrow}\boxed{\uparrow}\boxed{\uparrow}\boxed{\downarrow}\boxed{\uparrow}$
(1,θ)	$\boxed{\downarrow}\boxed{\downarrow}\boxed{\uparrow}\boxed{\uparrow}\boxed{\downarrow}$
(θ,1)	$\boxed{\downarrow}\boxed{\uparrow}\boxed{\uparrow}\boxed{\downarrow}\boxed{\downarrow}$
(θ,θ)	$\boxed{\downarrow}\boxed{\uparrow}\boxed{\uparrow}\boxed{\uparrow}\boxed{\downarrow}$

表 5 AND-$(0, \theta, \theta)$ 実装の初期状態

このときランダムカットと上下シャッフルと行うことにより $(a, b) = (1, 1)$ のときのみ $\boxed{\uparrow}$が 1 または 4 枚のパターンが現れる．また $(a, b) = (\theta, \theta), (\theta, 1), (1, \theta)$ の場合 $\boxed{\uparrow}$または $\boxed{\downarrow}$が 3 枚連続現れ，これらの 3 パターンは全て同一視される．さらにそれ以外の 5 パターンは例えば $\boxed{\downarrow}\boxed{\uparrow}\boxed{\downarrow}\boxed{\uparrow}\boxed{\uparrow}$等が現れ，この形式がランダムカットと上下シャッフルした状態のいずれかに一致するため同一視される．

このことから注意深く分類すると

$a\backslash b$	0	θ	1
0	0	0	0
θ	0	θ	θ
1	0	θ	1

という真偽表が得られることから Five Card Trick に上下シャッフル操作を追加するだけで AND-$(0,\theta,\theta)$ を実装できることが分かる．

表 4 と同様に，ランダムカット→上下シャッフル後の分類を行うと以下となる．

タイプ	同一視されるカード組
2	↑↓↑↓↓, ↓↑↓↓↑, ↑↓↓↑↓ ↓↓↑↓↑, ↓↑↓↑↓ ↓↑↓↑↑, ↑↓↑↑↓, ↓↑↑↓↑ ↑↑↓↑↓, ↑↓↑↓↑
3	↓↓↓↑↑, ↓↓↑↑↓, ↓↑↑↓↓ ↑↑↓↓↓, ↑↓↓↓↑ ↑↑↑↓↓, ↑↑↓↓↑, ↑↓↓↑↑ ↓↓↑↑↑, ↓↑↑↑↓
4	↓↓↓↓↑, ↓↓↓↓↓, ↓↓↑↓↓ ↓↑↓↓↓, ↑↓↓↓↓ ↑↑↑↑↓, ↑↑↑↓↑, ↑↑↓↑↑ ↑↓↑↑↑, ↓↑↑↑↑

表 6 ランダムカット→上下シャッフル後の分類（5 枚組）

タイプ 5 に分類される↑↑↑↑↑, ↓↓↓↓↓は出現せず，上記のように $2^5-2=30$ 通りが出現することとなる．

さきほどの真偽表にて上記タイプ (3,4,5 のいずれか) を当てはめると以下となる．ただし a の入力としては negation を表記している，つまり実際のカード入力であることに注意する．

$\bar{a}\backslash b$	↓↑	↑↓	↓↓
↑↓	Type-2	Type-2	Type-2
↓↑	Type-2	Type-3	Type-3
↓↓	Type-2	Type-3	Type-4

表 7 AND-$(0,\theta,\theta)$ 実装のタイプ分類

出現として Type-2 が 出力 0, Type-3 が出力 θ, Type-4 が出力 1 に該当することが分かる．

3. 代数的同型の適用

前章にて AND-$(0,\theta,\theta)$ の実装について具体的な方式を示した．本章ではまず AND-$(0,\theta,\theta)$ と同様に実装可能な AND-(θ,θ,θ) について触れる．

AND-$(0,\theta,\theta)$:

$a\backslash b$	0	θ	1
0	0	0	0
θ	0	θ	θ
1	0	θ	1

AND-(θ,θ,θ):

$a\backslash b$	0	θ	1
0	0	θ	0
θ	θ	θ	θ
1	0	θ	1

この 2 つの方式は真偽表の行列表現において第 1, 第 2 行と列を入れ替え，かつ 0 と *theta* を入れ替えることで同型であることが分かる．そこで AND-(θ,θ,θ) を実装するためにエンコーディングルールとして以下を適用する：↓↑$=\theta$, ↑↓$=0$, ↓↓$=1$. このとき真ん中にエクストラカード ↑を置いてその左側に a の negation, 右側に b を入力した場合，バリエーションとして表 8 の 9 パターンが得られる．ここで negation は左右を入れ替えたカードを入力することとする．

(a,b)	sequence
(0,0)	↓↑↑↑↓
(0,1)	↓↑↑↓↓
(1,0)	↓↓↑↑↓
(1,1)	↓↓↑↓↓
$(0,\theta)$	↓↑↑↓↑
$(\theta,0)$	↑↓↑↑↓
$(1,\theta)$	↓↓↑↓↑
$(\theta,1)$	↑↓↑↓↓
(θ,θ)	↑↓↑↓↑

表 8 AND-(θ,θ,θ) 実装の初期状態

このように行列を入れ替えることにより代数的に同型な 2 つのスキームをエンコーディングを変更すれば同じように実装できることが分かる．

3.1 XOR-(θ,θ,θ) と AND-$(0,1,\theta)$ の同型性

XOR-(θ,θ,θ):

$a\backslash b$	0	θ	1
0	0	θ	1
θ	θ	θ	θ
1	1	θ	0

AND-$(0,1,\theta)$:

$a\backslash b$	0	θ	1
0	0	0	0
θ	0	1	θ
1	0	θ	1

同様に第 1, 第 2 行列を入れ替え，XOR-(θ,θ,θ) における $0,\theta,1$ をそれぞれ $1,0,\theta$ に入れ替えることにより行列として同型になることが分かる．そのため 文献 [14] において XOR-(θ,θ,θ) の実装が提案されていることからエンコーディングルールを変更することにより AND-$(0,1,\theta)$ が実装可能であることを示せた．

4. まとめと今後について

AND 拡張と XOR 拡張の 3 値入力カードプロトコルにおいて，代数的に同値である相互関係のある事例を見い出し，実際にカードプロトコルとして実装可能であることを実証した．具体的には XOR 拡張の 2 つの場合 XOR-(θ,θ,θ), XOR-$(0,\theta,1)$ のそれぞれについて，AND 拡張である AND-$(0,1,\theta)$, AND-$(\theta,0,\theta)$ と代数的に同型であることを示した．これは AND 拡張のスキームが実装できれば XOR 拡張のスキームが実装可能であることを表しており，実際第 84 回 IPSJ 全国大会にて XOR-(θ,θ,θ) の実装が提示されていることから AND-$(0,1,\theta)$ が実装可能であることを示した．

本稿により相互性を示したことにより AND 拡張の 7 つの方式において，実装方法が示されていない AND-$(\theta,0,\theta)$, AND-$(0,0,\theta)$, AND-$(0,1,\theta)$, AND-$(0,\theta,0)$, の 4 方式が Open Problem として残されており，これらの実装を提案することを今後の課題とする．このうち XOR-$(0,\theta,1)$ の実装が示されれば AND-$(\theta,0,\theta)$ が実装可能であるため AND 拡張と XOR 拡張の両方面からの実装の検討が今後可能となる．

参考文献

[1] 水木, 電子情報通信学会 基礎・境界ソサイエティ Fundamentals Review, 2016 年 9 巻 3 号 pp.179-187, カード組を用いた秘密計算, https://www.jstage.jst.go.jp/article/essfr/9/3/9_179/_article/-char/ja
[2] T. Mizuki, H.Shizuya, Practical Card-Based Cryptography, FUN2014, pp.313-324, 2014.
[3] B. denBoer, More efficient match-making and satisability: the five card trick, EUROCRYPT'89, pp.208-217, 1989.
[4] T. Mizuki and H. Sone, Six-card secure AND and four-card secure XOR, International Workshop on Frontiers in Algorithmics, pp.358-369, 2009.
[5] T. Mizuki, M. Kumamoto and H. Sone, The Five-Card Trick Can Be Done with Four Cards, Asiacrypt2012.
[6] Y. Watanabe, Y. Kuroki, S. Suzuki, Y. Koga, M. Iwamoto, K. Ohta, Card-based majority voting protocols with three inputs using three cards, ISITA2018, pp.218–222, 2018.
[7] 須賀, 6 カード 3 入力 equality function (Six-Card Trick) における置換バリエーションの完全分類, 情報処理学会研究報告 Vol.2020-CSEC-91, No.34, 2020.
[8] 須賀, 怠惰なユーザのための非コミット型カードベース暗号, SCIS2021, 2F2-1, 2021.
[9] Y. Suga, Card-based Cryptography Meets Mahjong Tiles, Small-workshop on Communications between Academia and Industry for Security 2021, 2021.
[10] 須賀, 手の内だけで簡単に実行可能な Six Card Trick とカード入力後の置換に関する考察, 情報処理学会研究報告 Vol.2021-CSEC-92, No.7, 2021.
[11] 須賀, 三人寄ればチーズの知恵, マルチメディア, 分散協調とモバイルシンポジウム 2021(DOCOMO2021), pp.173-178, 2021.
[12] 須賀, 0,1, 不定の 3 値入力が可能な AND 演算カードベースプロトコルの初期検討, 第 44 回情報理論とその応用シンポジウム (SITA2021), 4-3-3, 2021.
[13] 須賀, 3 値入力可能な可換半群の条件を満たす非コミットメント型 AND 演算拡張カードベースプロトコルの構成, SCIS2022, 2F4-2, 2022.
[14] 須賀, 結合律を満たす XOR 拡張 3 値入力カードプロトコルの実装可能性, 情報処理学会 第 84 回全国大会講演論文集, 1E-05, 2022.
[15] 須賀, 局所的に AND 構造を持つ位数 3 の可換半群の分類証明とカードベースプロトコルへの適用, to be appeared, 2022.
[16] 須賀, 3 値入力可能な拡張 Five Card Trick における第 4 の未定義値の扱いについて, 情報処理学会研究報告 Vol.2022-CSEC-96, No.36, 2022.

AppendixE. AND 拡張 7 種類の分類一覧

AND-$(0,0,\theta)$:

$a\backslash b$	0	θ	1
0	0	0	0
θ	0	0	θ
1	0	θ	1

AND-$(\theta,0,\theta)$:

$a\backslash b$	0	θ	1
0	0	θ	0
θ	θ	0	θ
1	0	θ	1

AND-$(0,\theta,0)$:

$a\backslash b$	0	θ	1
0	0	0	0
θ	0	θ	0
1	0	0	1

AND-$(0,\theta,\theta)$:

$a\backslash b$	0	θ	1
0	0	0	0
θ	0	θ	θ
1	0	θ	1

AND-$(0,\theta,1)$:

$a\backslash b$	0	θ	1
0	0	0	0
θ	0	θ	1
1	0	1	1

AND-(θ,θ,θ):

$a\backslash b$	0	θ	1
0	0	θ	0
θ	θ	θ	θ
1	0	θ	1

AND-$(0,1,\theta)$:

$a\backslash b$	0	θ	1
0	0	0	0
θ	0	1	θ
1	0	θ	1

AppendixF. XOR 拡張 2 種類の分類一覧

XOR-(θ,θ,θ):

$a\backslash b$	0	θ	1
0	0	θ	1
θ	θ	θ	θ
1	1	θ	0

XOR-$(0,\theta,1)$:

$a\backslash b$	0	θ	1
0	0	0	1
θ	0	θ	1
1	1	1	0