バックワード匿名シミュレーションを用いた匿名性の検証

Proving Anonymity with Backward Anonymous Simulations

河辺 義信[†]，真野 健[†]，櫻田 英樹[†]，塚田 恭章[†]
Yoshinobu KAWABE, Ken MANO, Hideki SAKURADA, Yasuyuki TSUKADA

[†] 日本電信電話株式会社 NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories, NTT Corporation
{kawabe, mano, sakurada, tsukada}@theory.brl.ntt.co.jp

匿名性の考え方は，寄付，投票，新聞や雑誌への投書，内部告発，論文の査読など，実世界のさまざまな場面に現れている．分散システムにおいても，インターネットを用いた投票システムなど，匿名性が保証されるべきシステムを考えることができる．本稿では，バックワード匿名シミュレーションと呼ばれる関係を用いた匿名性の検証方法を示す．さらに，藤岡・岡本・太田の電子投票プロトコル (FOO92 プロトコル) に対する匿名性の検証例を示す．この検証では，I/O-オートマトンに基づく仕様記述言語でプロトコルを記述し，定理証明器を用いて匿名性を証明する．

1 はじめに

匿名性の考え方は，寄付，投票，新聞や雑誌への投書，内部告発，論文の査読など，実世界のさまざまな場面に現れている．分散システムにおいても，たとえばインターネットを用いた電子投票システムなど，匿名性が保証されるべきシステムを考えることができる．しかし，そのようなシステムが匿名性を保証しているかをどのように確かめればよいかは，明らかではない．とくに，分散システムが持つ匿名性を厳密に証明する手法は，確立されていない．

ソフトウェア工学の分野では，プログラムや仕様を形式的に記述し，その正しさを計算機を使って検証する方法が知られている．同様の考え方に基づいて，匿名性などを形式的に記述し証明する研究も現れている (たとえば，[2, 5, 6])．しかし，定理証明器やモデル検査器など，計算機を用いた検証の例は少ない．とくに，モデル検査器による匿名性の検証は文献 [10] (プロセス代数 CSP で，匿名性の議論を行った論文) に見られるが，汎用の定理証明器を用いた検証は知られていない．定理証明器には，非有界システムを直接扱えるなどの特長がある．そこで，本稿では，分散アルゴリズムの形式的な記述・検証のための計算モデルで定理証明器による検証ツールを持つ I/O-オートマトン [9] を用いて匿名性を形式化・検証する方法を議論する．

本稿では，まず，I/O-オートマトンのトレース集合上で定義される匿名性 (トレース匿名性) を導入する．トレース匿名性は，ある通信者による注目した動作を他の任意の通信者も行いうることを表している．これは，文献 [10] の強匿名性 (strong anonymity) を拡張したものである．さらに我々は，トレース匿名性の証明手段として，バックワード匿名シミュレーションを導入する．これは，トレースの長さに関する帰納法によってトレース匿名性を証明する方法である．本稿では，バックワード匿名シミュレーションの存在がトレース匿名性の十分条件であることを証明することによって，バックワード匿名シミュレーションを用いた証明法が妥当なものであることを示す．さらに，本稿では，例として藤岡・岡本・太田の電子投票プロトコル (FOO92 プロトコル) に対する匿名性の検証を示す．この検証では，I/O-オートマトンに基づく仕様記述言語でプロトコルを記述し，定理証明器を用いて匿名性を証明する．

2 準備：I/O-オートマトン

I/O-オートマトン X は，アクションの集合 $sig(X)$，状態集合 $states(X)$，初期状態の集合 $start(X) \subset states(X)$ および遷移の集合 $trans(X) \subset states(X) \times sig(X) \times states(X)$ から成る．アクションには，入力，出力，内部の 3 種類があり，それぞれの集合を $in(X), out(X), int(X)$ と書く (これらは，互いに素な集合とする)．また，入力アクションと出力アクションを，外部アクションと呼ぶ．X の遷移 $(s, a, s') \in trans(X)$ を，$s \xrightarrow{a}_X s'$ とも書く．また，a が内部アクションのとき，$s \rightarrow_X s'$ とも書く．関係 $\twoheadrightarrow_X$ を，関係 $\rightarrow_X$ の反射的推移的閉包

とする．任意の $a \in sig(X)$ および $s, s' \in states(X)$ に関して，$s \stackrel{a}{\Longrightarrow}_X s'$ は以下を表す：(i) a が外部アクションのときは，ある $s_1, s_2 \in states(X)$ が存在して $s \twoheadrightarrow_X s_1 \stackrel{a}{\rightarrow}_X s_2 \twoheadrightarrow_X s'$．(ii) a が内部アクションのときは，$s \twoheadrightarrow_X s'$．任意の $s_0 \in start(X)$ および遷移列 $\alpha \equiv s_0 \stackrel{a_1}{\rightarrow}_X s_1 \stackrel{a_2}{\rightarrow}_X \cdots \stackrel{a_n}{\rightarrow}_X s_n$ に関して，α に現れるすべての外部アクションを順に並べたものを α のトレースと呼ぶ．X が持つすべてのトレースの集合を $traces(X)$ と書く．I/O-オートマトン X から I/O-オートマトン Y へのバックワードシミュレーション b が存在し，さらに b が像有限ならば，$traces(X) \subseteq traces(Y)$ が成り立つことが知られている ([9], 定理 3.17)．ここで，X から Y へのバックワードシミュレーションとは，次を満たす二項関係 $b \subset states(X) \times states(Y)$ である：(i) b は全域的である．すなわち，任意の $s \in states(X)$ に関して，$b(s, s')$ を満たす $s' \in states(Y)$ が存在する．(ii) 任意の $s \in start(X)$ および $s' \in states(Y)$ に関して，$b(s, s')$ ならば $s' \in start(Y)$ である．(iii) 任意の $s_1, s_2 \in states(X)$，$s_2' \in states(Y)$ および $a \in sig(X)$ に関して，$b(s_2, s_2')$ かつ $s_1 \stackrel{a}{\rightarrow}_X s_2$ ならば，ある $s_1' \in states(Y)$ が存在して $b(s_1, s_1')$ かつ $s_1' \stackrel{a}{\Longrightarrow}_Y s_2'$ である．また，関係 $R \subseteq A \times B$ が像有限であるとは，任意の $a \in A$ に関して集合 $\{b \,|\, (a, b) \in R\}$ が有限なことである．

3 匿名性の形式化と検証方法

3.1 基本的な考え方

次の例で，匿名性の基本的な考え方を説明する．

例 1 (匿名寄付) *Alice* と *Bob* のどちらか一方が，匿名で寄付をした．*Alice* は *5* ドル，*Bob* は *10* ドルを寄付するつもりだった．

図 1 は，上記を I/O-オートマトン D1 として記述したものである．ここで，$5 と$10 は出力アクションで，その額の募金行為をあらわす．また，$5 と$10 の発生後に，出力アクション I'm(Alice) と I'm(Bob) を加えておく．I'm(Alice) と I'm(Bob) は，$5 や$10 の行為者を明示するためのアクションで，匿名性を議論するために使う仮想的なものである．

問 1 第三者が募金箱をこっそり覗いたところ，5 ドル入っているのが見えた．寄付したのは誰か?

D1 では，$5 の発生後は I'm(Alice) のみが，$10 の発生後は I'm(Bob) のみが発生できる．したがって，

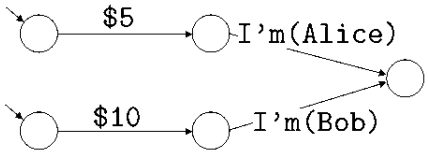

図 1: D1

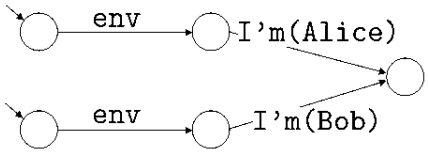

図 2: D2

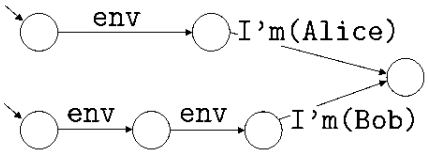

図 3: D3

問 1 の場合，$5 の発生から行為者が Alice である (I'm(Alice) が発生する) と特定されてしまう．よって，D1 は匿名的なシステムではない．

D1 が匿名的でない主な理由は，募金額が陽に現れることである．そこで，次のように問題を変形する．

問 2 募金箱に入れる前に，紙幣を封筒に入れることにした．この場合，匿名性は成り立つか?

図 2 は，上記を I/O-オートマトン D2 として図示したものである．ここで，出力アクション$5 と$10 は，別の出力アクション env で置き換えられる．これは，$5 と$10 を暗号化し，金額に関する情報を隠すことに対応する．問 2 の場合は，たとえアクション env の発生が盗聴されても，それが Alice によるものか Bob によるものかまではわからない．すなわち，D2 は匿名的なシステムである．

問 2 は，暗号化によって匿名性を確保できる場合があることを示していた．最後に，次の場合を考える．

問 3 Bob は，寄付するとき 5 ドル札 2 枚を使うことにした．それぞれのお札を封筒に入れるとき，匿名性は成り立つか?

図 3 は，上記を I/O-オートマトン D3 として図示したものである．問 3 の場合では，発生しうるすべて

の情報が暗号化されているが，アクション env の発生回数を数えることで，I'm(Alice) と I'm(Bob) のどちらが発生するか，特定できてしまう．すなわち，D3 は匿名的ではない．

以上のように，外部から観測されるアクション系列 (トレース) やその集合 (トレース集合) を考えることで，匿名性を議論することができる．次節では，トレース集合に基づく匿名性を，形式的に定義する．

3.2 匿名性の形式化

3.2.1 トレース匿名性

以下では，簡単化のため，外部アクションとして出力アクションのみを持つ (すなわち，$in(X) = \emptyset$ となるような) I/O-オートマトン X を考える．

定義 1 *I/O-*オートマトン X および出力アクションの集合族 A *(*ただし，$A', A'' \in A$ かつ $A' \neq A''$ ならば，A' と A'' は互いに素*)* を考える．*I/O-*オートマトン $anonym_A(X)$ を，次のように定める．

- $states(anonym_A(X)) = states(X)$,
- $start(anonym_A(X)) = start(X)$,
- $in(anonym_A(X)) = in(X)$,
- $out(anonym_A(X)) = out(X)$,
- $trans(anonym_A(X))$
$= \{(s_1, a, s_2) \mid (s_1, a, s_2) \in trans(X) \wedge a \notin \bigcup_{A' \in A} A'\}$
$\cup \{(s_1, a, s_2) \mid (s_1, a', s_2) \in trans(X) \wedge A' \in A \wedge a' \in A' \wedge a \in A'\}$.

また，$\bigcup_{A' \in A} A'$ の要素を，匿名アクションと呼ぶ．

定義 2 *I/O-*オートマトン X と出力アクションの集合族 A *(*ただし，$A', A'' \in A$ かつ $A' \neq A''$ ならば A' と A'' は互いに素*)* に関して，$traces(anonym_A(X)) = traces(X)$ のとき，X は A に関してトレース匿名的という．

直感的には，$anonym_A(X)$ は，次の意味で匿名的なシステムである．

> ある $someone \in A'$ (ただし $A' \in A$) に関して $s \xrightarrow{someone}_{anonym_A(X)} s'$ ならば，任意の $everyone \in A'$ に関して $s \xrightarrow{everyone}_{anonym_A(X)} s'$ である．

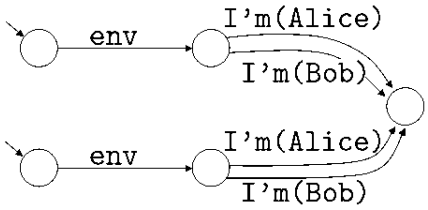

図 4: $anoym_A$(D2)

したがって，$traces(anonym_A(X)) = traces(X)$ ならば，$anonym_A(X)$ の匿名性から X の匿名性が言える．

例 2 第 *3.1* 節の D2 に関して，図 *4* に $anonym_A$(D2) *(*ただし，$A = \{\{$I'm(Alice), I'm(Bob)$\}\}$*)* を示す．I'm(Alice) と I'm(Bob) の一方が可能な状態では，必ずもう一方も可能である．

第 3.1 節の D1，D2 および D3 に関して，次が成り立つ．

- D1 の場合は，

$$traces(\text{D1}) = \{\text{\$5.I'm(Alice)}, \text{\$10.I'm(Bob)}\}$$
$$traces(anonym_A(\text{D1})) = \{\text{\$5.I'm(Alice)}, \text{\$10.I'm(Bob)}, \text{\$5.I'm(Bob)}, \text{\$10.I'm(Alice)}\}$$

より，$traces(\text{D1}) \neq traces(anonym_A(\text{D1}))$.

- D2 の場合は，

$$traces(\text{D2}) = traces(anonym_A(\text{D2})) = \{\text{env.I'm(Alice)}, \text{env.I'm(Bob)}\}$$

より，$traces(\text{D2}) = traces(anonym_A(\text{D2}))$.

- D3 の場合は，

$$traces(\text{D3}) = \{\text{env.I'm(Alice)}, \text{env.env.I'm(Bob)}\}$$
$$traces(anonym_A(\text{D3})) = \{\text{env.I'm(Alice)}, \text{env.env.I'm(Bob)}, \text{env.I'm(Bob)}, \text{env.env.I'm(Alice)}\}$$

より，$traces(\text{D3}) \neq traces(anonym_A(\text{D3}))$.

すなわち，D2 のみがトレース匿名性を満たす．

3.2.2 シール：暗号化されたデータの取扱い

第 3.1 節の問 2 では，D1 中のアクション$5 および$10 をアクション env に置き換えて D2 を生成した．この置き換えは，$5 と$10 を暗号化し，金額に関する情報を隠すことに対応する．以下では，暗号化されたデータの取扱いについて述べる．

I/O-オートマトンの理論では，一対一となるようなアクションの名前がえは，リネーミングと呼ばれる．これに対し，我々は，I/O-オートマトンに関するアクションの名前がえのうち，単射とは限らないものを擬リネーミングと呼ぶ．

定義 3 *I/O-オートマトン X を考える．写像 f が，$sig(X) \subseteq domain(f)$ を満たし，さらに*

- $in(f(X)) = \{f(a) \mid a \in in(X)\}$,
- $out(f(X)) = \{f(a) \mid a \in out(X)\}$,
- $int(f(X)) = \{f(a) \mid a \in int(X)\}$

に関して，$in(f(X))$，$out(f(X))$ および $int(f(X))$ が互いに素なとき，f を X の擬リネーミングと呼ぶ．

定義 4 *I/O-オートマトン X および X の擬リネーミング f に関して，I/O-オートマトン $f(X)$ を次で定める．*

- $states(f(X)) = states(X)$,
- $start(f(X)) = start(X)$,
- $sig(f(X)) = in(f(X)) \cup out(f(X)) \cup int(f(X))$,
- $trans(f(X))$
  $= \{(s_1, f(a), s_2) \mid (s_1, a, s_2) \in trans(X)\}$.

定義 5 *I/O-オートマトン X および X の出力アクションの集合族 A を考える．以下を満たす X の擬リネーミング f を，$sig(X) - \bigcup A$ へのシールと呼ぶ．*

- *ある $A' \in A$ に関して $a \in A'$ ならば $f(a) = a$.*
- *$a \notin \bigcup_{A' \in A} A'$ ならば $f(a) \notin \bigcup_{A' \in A} A'$.*

例 3 シール f を，

$$\begin{cases} f(x) = x \text{ if } x \in \{\texttt{I'm(Alice)}, \texttt{I'm(Bob)}\}, \\ f(\texttt{\$5}) = \texttt{env}, \\ f(\texttt{\$10}) = \texttt{env} \end{cases}$$

と定める．このとき，第 *3.1* 節の D1 および D2 に関して，D2 と f(D1) は同一である．

シールに関して，以下が成り立つ．

命題 1 *I/O-オートマトン X，出力アクションの集合族 A および $sig(X) - \bigcup A$ へのシール f に関して，*

$$traces(f(anonym_A(X))) = traces(anonym_A(f(X)))$$

である．

証明： $f(anonym_A(X))$ と $anonym_A(f(X))$ が同一であることを示すことで，命題は証明できる． □

写像 f をシールとする．このとき，写像 $traces \circ f$ は，シール f によって情報を隠蔽した後にトレース集合を求める操作を表している．この操作は，盗聴者が I/O-オートマトンからどのような情報を取り出せるかを指定する操作とみなせる．命題 1 より，I/O-オートマトン $f(X)$ がトレース匿名的なとき，

$$\begin{aligned} (traces \circ f)(X) &= traces(f(X)) \\ &= traces(anonym_A(f(X))) \\ &= traces(f(anonym_A(X))) \\ &= (traces \circ f)(anonym_A(X)) \end{aligned}$$

である．すなわち，$f(X)$ のトレース匿名性を示すことは，操作 $traces \circ f$ を用いて観測したときのシステム X と匿名的なシステム $anonym_A(X)$ の識別不可能性を示すことに相当する．

3.3 匿名性の検証方法

以下では，I/O-オートマトン $f(X)$ のトレース匿名性を示すための手法を導入する．この手法では，バックワード匿名シミュレーションと呼ばれる，状態集合上の二項関係を用いる．

定義 6 *I/O-オートマトン X，出力アクションの集合族 A (ただし，$A', A'' \in A$ かつ $A' \neq A''$ ならば，A' と A'' は互いに素)，および X の $sig(X) - \bigcup A$ へのシール f を考える．X の A および f に関するバックワード匿名シミュレーション $bas_{A,f}$ は，次を満たす $states(X)$ 上の二項関係である：*

1. *任意の状態 $s \in states(X)$ に関して，ある状態 $s' \in states(X)$ が存在して，$bas_{A,f}(s, s')$.*
2. *任意の $s \in start(X)$ および $s' \in states(X)$ に関して，$bas_{A,f}(s, s')$ ならば $s' \in start(X)$.*
3. *任意の $s_1, s_2, s_2' \in states(X)$ および $a \in sig(X)$ に関して，$bas_{A,f}(s_2, s_2')$ かつ $s_1 \xrightarrow{a}_X s_2$ ならば，*

- ある $A' \in A$ に関して $a \in A'$ ならば，すべての $a' \in A'$ に関してある状態 s'_1 が存在して，$bas_{A,f}(s_1, s'_1)$ かつ $s'_1 \stackrel{a'}{\Longrightarrow}_X s'_2$.
- $a \notin \bigcup_{A' \in A} A'$ ならば，ある状態 s'_1 およびアクション a' が存在して，$bas_{A,f}(s_1, s'_1)$，$s'_1 \stackrel{a'}{\Longrightarrow}_X s'_2$ かつ $f(a) = f(a')$.

バックワード匿名シミュレーションに関して，次が成り立つ.

定理 1 *I/O*-オートマトン X，出力アクションの集合族 A *(*ただし，$A', A'' \in A$ かつ $A' \neq A''$ ならば，A' と A'' は互いに素*)*，および X の $sig(X) - \bigcup A$ へのシール f を考える．X の A および f に関するバックワード匿名シミュレーション $bas_{A,f}$ が存在し，かつ $bas_{A,f}$ が像有限ならば，$f(X)$ は A に関してトレース匿名的である.

証明： $anonym_A(f(X))$ の定義から，

$$traces(anonym_A(f(X))) \supseteq traces(f(X))$$

は容易．第 2 節より，関係

$$bas_{A,f} \subset states(anonym_A(f(X))) \times states(f(X))$$

が $anonym_A(f(X))$ から $f(X)$ へのバックワードシミュレーションであることと像有限であることを示せば，逆向きの包含

$$traces(anonym_A(f(X))) \subseteq traces(f(X))$$

も得られる．以下では，

$$\begin{aligned} &start(X) = start(f(X)) = start(anonym_A(f(X))),\\ &states(X) = states(f(X))\\ &\qquad = states(anonym_A(f(X))),\\ &\text{および}\\ &sig(f(X)) = sig(anonym_A(f(X))) \end{aligned}$$

に注意する．まず，定義 6 の条件 1 より $bas_{A,f}$ は全域的である．次に，定義 6 の条件 2 より，$bas_{A,f}(s, s')$ を満たす任意の $s \in start(X)(= start(anonym_A(X)))$ および $s' \in states(X)$ に関して，$s' \in start(X)$ である．すなわち，任意の $s \in start(anonym_A(X))$ に関して，$\{s' \mid bas_{A,f}(s, s')\} \subseteq start(X)$ が成り立つ．したがって，バックワードシミュレーションのための初期状態の条件は満たされる．次に，$bas_{A,f}(s_2, s'_2)$ および $s_1 \stackrel{a}{\rightarrow}_{anonym_A(f(X))} s_2$ を満たす任意の状態 $s_1, s_2 \in states(anonym_A(f(X)))$，$s'_2 \in states(f(X))$ およびアクション $a \in sig(anonym_A(f(X)))$ を考える．$states(f(X)) = states(anonym_A(X))$ および $sig(f(X)) = sig(anonym_A(f(X)))$ より，$s_1, s_2 \in states(f(X))$ かつ $a \in sig(f(X))$ である．ここで，ある $A' \in A$ に関して $a \in A'$ か否かで分ける.

- ある $A' \in A$ に関して $a \in A'$ のときは，$anonym_A(f(X))$ の定義および $s_1 \stackrel{a}{\rightarrow}_{anonym_A(f(X))} s_2$ から，あるアクション $b \in A'$ が存在して $s_1 \stackrel{b}{\rightarrow}_{f(X)} s_2$ である．定義 5 より，任意の $x \in A'$ に関して $f(x) = x$ であり，任意の $y \notin \bigcup_{A'' \in A} A''$ に関して $f(y) \notin \bigcup_{A'' \in A} A''$ である．よって，$s_1 \stackrel{b}{\rightarrow}_X s_2$ である．定義 6 の条件 3 より，ある $s'_1 \in states(X)$ が存在して $bas_{A,f}(s_1, s'_1)$ かつ $s'_1 \stackrel{a}{\Longrightarrow}_X s'_2$ である．よって，$s'_1 \stackrel{f(a)}{\Longrightarrow}_{f(X)} s'_2$ である．さらに $f(a) = a$ であるから，$s'_1 \stackrel{a}{\Longrightarrow}_{f(X)} s'_2$ が成り立つ．したがって，$a \in A'$ の場合，バックワードシミュレーションのためのステップの対応が成り立つ.
- $a \notin \bigcup_{A' \in A} A'$ のときは，$anonym_A(f(X))$ の定義および $s_1 \stackrel{a}{\rightarrow}_{anonym_A(f(X))} s_2$ から，$s_1 \stackrel{a}{\rightarrow}_{f(X)} s_2$ である．I/O-オートマトン $f(X)$ の定義より，$f(a') = a$ を満たすあるアクション $a' \in sig(X) \backslash \bigcup_{A' \in A} A'$ が存在して，$s_1 \stackrel{a'}{\rightarrow}_X s_2$ である．$bas_{A,f}$ はバックワード匿名シミュレーションであり，$a' \notin \bigcup_{A' \in A} A'$ である．これより，定義 6 の条件 3 から，ある状態 $s'_1 \in states(X)$ およびアクション $a'' \in sig(X)$ が存在して，$bas_{A,f}(s_1, s'_1)$，$s'_1 \stackrel{a''}{\Longrightarrow}_X s'_2$ かつ $f(a') = f(a'') = a$ が成り立つ．I/O-オートマトン $f(X)$ の定義より，$s'_1 \stackrel{f(a'')}{\Longrightarrow}_{f(X)} s'_2$ である．すなわち，$s'_1 \stackrel{a}{\Longrightarrow}_{f(X)} s'_2$ である．よって，$a \notin \bigcup_{A' \in A} A'$ の場合も，ステップの対応が成り立つ.

これより，関係 $bas_{A,f}$ はバックワードシミュレーションである．さらに，$bas_{A,f}$ の像有限性と文献 [9] の定理 3.17(2) より，$traces(anonym_A(f(X))) \subseteq traces(f(X))$ である．以上より，$f(X)$ は A に関してトレース匿名的である． □

以上より，I/O-オートマトン $f(X)$ のトレース匿名性を示すには，X の出力アクション集合族 A およびシール f に関するバックワード匿名シミュレーション $bas_{A,f}$ を見つけ，それが像有限であることを示せばよい.

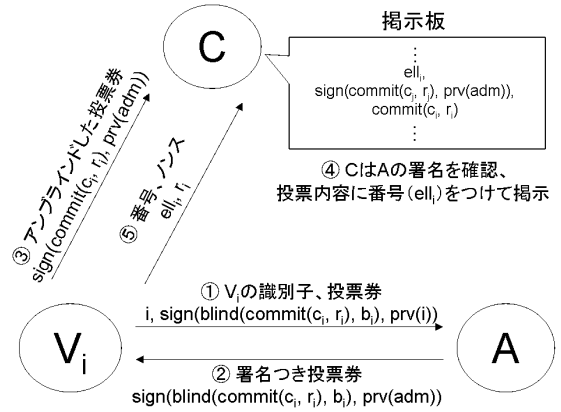

図 5: FOO92

4 検証例：FOO92 投票プロトコル

本節では，検証例として，藤岡・岡本・太田の投票プロトコル (以下，FOO92) のトレース匿名性を示す．

4.1 FOO92 の概要

FOO92 では，投票者 (V_i)，アドミニストレータ (A)，および集計者 (C) の間でやりとりを行う (図 5)．ただし，投票者数は有限とし，各投票者 V_i は，固有のノンス r_i および b_i (それぞれ，鍵およびブラインダーと呼ぶ) を持ち，候補者 c_i を支持する．また，`sign`，`blind`，`commit`，`prv`(i) および `prv(adm)` は，それぞれ，暗号化関数，ブラインド関数，コミットメント関数，V_i の秘密鍵および A の秘密鍵で，次を満たす．

```
open(commit(c, r), r) = c
checksign(sign(m, prv(i)), pub(i)) = m
unblind(sign(blind(m, b), prv(i)), b)
  = sign(m, prv(i)).
```

FOO92 は，次のフェーズから成る．

- フェーズ 1 (投票券の有効化)：各投票者 V_i は A に対し，自身の識別子と投票券の組

  i,`sign(blind(commit(`c_i`,`r_i`), `b_i`), prv(`i`))`

  を送り (図 5 中の 1)，署名つき投票券

  `sign(blind(commit(`c_i`,`r_i`), `b_i`), prv(adm))`

  を受け取る (図 5 中の 2)．

- フェーズ 2 (投票)：V_i は C に対し，フェーズ 1 で得た署名つき投票券をアンブラインドしたもの `sign(commit(`c_i`,`r_i`), prv(adm))` を匿名通信路で送る (図 5 中の 3)．C は A の署名を確認し，固有の番号 ℓ_i を生成して，結果 (ℓ_i, `sign(commit(`c_i`,`r_i`), prv(adm))`, `commit(`c_i`,`r_i`)`) を掲示する (図 5 中の 4)．

- フェーズ 3 (開票)：V_i は，匿名通信路を使って，C に番号と鍵の組 ℓ_i, r_i を送る (図 5 中の 5)．

- フェーズ 4 (結果発表)：集計者は，フェーズ 3 を締め切って，集計結果を公表する．

ただし，全ての投票者がフェーズ 3 までの処理を行うとは限らない．投票期限を過ぎた，などの理由で，途中で投票処理をやめてしまう投票者もありうる．

4.2 プロトコルの記述

図 6 は，FOO92 プロトコルの各フェーズを，IOA 言語 (I/O-オートマトンに基づく仕様記述言語) で書いたものである．IOA 言語は，

- `signature`-部 (アクション名とソートの宣言)，
- `states`-部 (変数と初期値の宣言)，および
- `transitions`-部 (アクションの本体)

から成る．各アクションの本体は，`pre`-部と `eff`-部の組から成る．各アクションは，`pre`-部の全ての条件が満たされたとき発火でき，発火時に `eff`-部の処理を行って次の状態に遷移する．

図 6 のプロトコル記述では，6 種類の変数

`phase`，配列 `r`，配列 `cand`，配列 `nthline`，`bbd` および `forOpen`

と 7 種類の出力アクション

`v2a_and_a2v(i, m, m')`，`timeoutPhase1`，
`v2c_and_checkBB(i, b, ell, m)`，
`timeoutPhase2`，`v2b(i, m)`，
`announceResult(b)` および
`disclose(ar, anl, ac)`

を使う．それぞれの変数とアクションの意味は，図 6 中にコメントとして記す．掲示板に示された番号を格納するための配列変数 `nthline[i]` は，投票者 `i` のプログラムカウンタを兼ねており，

- -1 のとき：フェーズ 1 の処理を行う前
- 0 のとき：フェーズ 1 の処理を完了し，フェーズ 2 の処理を行う前

```
output phase1(immls)
  pre phase = phase1
      /\ (\A i:ID (nthline[i] = -1))
      /\ bbd = {}
      /\ forOpen = {}
      /\ (\A m:MES
          ((m \in immls)
           => (\E i:ID
               (m = v2aMes(i, cand[i], r[i])))))
  eff phase := phase2;
      nthline := initNthline(immls)
```

ただし，

- v2aMes(i, c, r) = imm(α,β,γ)
  - α = i,
  - β = sign(blind(commit(c, r), blinder(i)), prv(i)),
  - γ = sign(checksign(β, pub(i)), prv(adm)) .
- 列 immls と候補者 i に関して，v2aMes(i,c,r) $\in$ immls となる c, r があるときは initNthline(immls)[i] = 0，ないときは initNthline(immls)[i] = -1 である．

とする．

図 7: 出力アクション phase1

- 1 以上のとき：フェーズ 2 の処理を完了し，フェーズ 3 の処理を行う前
- −2 以下のとき：フェーズ 3 の処理を完了

であることに対応する．また，アクション disclose は，FOO92 プロトコルの終了後に全投票者が，

1. r ：保持する鍵の値
2. nthline：終了時に，どのフェーズに到達したか
3. cand ：支持していた候補者名

を一斉に開示する行為に対応する．このアクションは仮想的な動作で，FOO92 プロトコル本体の動作は表していない．投票結果を変えない範囲のどのような開示の仕方も可能であるならば，FOO92 は匿名性を満たす投票プロトコルといえる．

4.3 トレース匿名性の証明

4.3.1 準備

まず，簡単化のため，プロトコル記述を変形する．FOO92 の v2a_and_a2v(i, m, m') と timeoutPhase1 を図 7 の出力アクション phase1(immls) に置き換えた結果を FOO92' と呼ぶ．FOO92' のアクション phase1(immls) は，FOO92 のフェーズ 1 を，1 ステップで行うアクションである．

次に，FOO92 および FOO92' のシール seal を，図 8 のように定める．シール seal は，

$$\mathtt{seal}(\mathtt{phase1}(immls)) = \mathtt{phase1}(\mathtt{sealDT}(immls))$$
$$\mathtt{seal}(\mathtt{v2a\_and\_a2v}(i,m,m')) = \mathtt{v2a\_and\_a2v}(i,\mathtt{sealDT}(m),\mathtt{sealDT}(m'))$$
$$\mathtt{seal}(\mathtt{timeoutPhase1}) = \mathtt{timeoutPhase1}$$
$$\mathtt{seal}(\mathtt{v2c\_and\_checkBB}(i,b,ell,m)) = \mathtt{v2c\_and\_checkBB}(b,ell,m)$$
$$\mathtt{seal}(\mathtt{timeoutPhase2}) = \mathtt{timeoutPhase2}$$
$$\mathtt{seal}(\mathtt{v2b}(i,m)) = \mathtt{v2b}(m)$$
$$\mathtt{seal}(\mathtt{announceResult}(b)) = \mathtt{announceResult}(b)$$
$$\mathtt{seal}(\mathtt{disclose}(ar,anl,ac)) = \mathtt{disclose}(ar,anl,ac)$$

$$\mathtt{sealDT}(\mathtt{empty}) = \mathtt{empty}$$
$$\mathtt{sealDT}(m\text{-}|ls) = \mathtt{sealDT}(m)\text{-}|\mathtt{sealDT}(ls)$$
$$\mathtt{sealDT}(\mathtt{v2a\_and\_a2v}(i,m,m')) = \mathtt{v2a\_and\_a2v}(i,\mathtt{sealDT}(m),\mathtt{sealDT}(m'))$$
$$\mathtt{sealDT}(\mathtt{blind}(m,\mathtt{blinder}(i))) = \mathtt{LookLikeNoise}$$
$$\mathtt{sealDT}(\mathtt{sign}(m,m')) = \mathtt{sign}(\mathtt{sealDT}(m),\mathtt{sealDT}(m'))$$
$$\mathtt{sealDT}(\mathtt{checksign}(m,m')) = \mathtt{checksign}(\mathtt{sealDT}(m),\mathtt{sealDT}(m'))$$

図 8: シール seal

- FOO92 と FOO92' のトレースに現れる v2a_and_a2v(i, m, m') のうち，m と m' に式 blind(m'',blinder(i)) が現れるならば，記号 LookLikeNoise で置き換える．これは，blind(m'',blinder(i)) から第三者が情報 m'' を取り出せない場合の匿名性を考えるためである (実際，ブラインダー blinder(i) は送受信されないので，第三者は blind(m'',blinder(i)) から情報 m'' を取り出すことはできない)．
- アクション v2c_and_checkBB(i, b, ell, m) と v2b(i, m) を，それぞれ，送信者情報 i を除いたもの v2c_and_checkBB(b, ell, m) と v2b(m) に置き換える (匿名通信路を使うため，送信者情報は隠されているとみなせる)．

を行うものである．

さらに，匿名アクションの集合族を，次で定める．

定義 7 出力アクション disclose に関する集合族 $A_{\mathtt{FOO92}}$ を，次で定める．任意の $A \in A_{\mathtt{FOO92}}$ に関して

$$\mathtt{disclose}(r,nl,c),\mathtt{disclose}(r',nl',c') \in A$$

であるのは，次を満たすときのみである：

1. 任意の投票者 i, j に関して，$i \neq j$ ならば $r[i] \neq r[j]$ かつ $r'[i] \neq r'[j]$,
2. 任意の投票者 i に関してある投票者 j がいて，$r[i] = r'[j]$ かつ $nl[i] = nl'[j]$ かつ $c[i] = c'[j]$,

3. 任意の投票者 j に関してある投票者 i がいて，$r[i] = r'[j]$ かつ $nl[i] = nl'[j]$ かつ $c[i] = c'[j]$，
4. 任意の投票者 i に関して，$nl[i] = -1$ のとき，かつそのときに限り，$nl'[i] = -1$.

定義 7 の条件 1 は，全ての投票者が異なる鍵を持つことを表す．条件 2 と 3 は，役割 (どの候補者を支持し，どの鍵を使い，どのフェーズまで処理を行うか) が，投票者間で交換可能であることを表している．ただし，条件 4 より，フェーズ 1 を行った投票者と行っていない投票者の間では，役割交換はできない．ある disclose アクションが，集合 $A, A' \in A_{\mathtt{FO092}}$ の両方の要素とすると，定義より，$A = A'$ である．すなわち，任意の $A, A' \in A_{\mathtt{FO092}}$ に関して $A \neq A'$ のとき，A と A' は互いに素である．

上記に関して，次が成り立つ (詳細は，付録 A).

命題 2 seal(FO092') が $A_{\mathtt{FO092}}$ に関してトレース匿名的ならば，seal(FO092) も $A_{\mathtt{FO092}}$ に関してトレース匿名的. □

4.3.2 バックワード匿名シミュレーションの証明

以下では，FO092' のトレース匿名性を証明する．バックワード匿名シミュレーションの候補 bas を，次に示す (s.var は，状態 s における変数 var の値).

```
bas(s, s') <=> (
    (s.phase = s'.phase)
 /\ (s.bbd = s'.bbd)
 /\ (s.forOpen = s'.forOpen)
 /\ (\A i:ID
      ((s.nthline[i] = -1)
        <=> (s'.nthline[i] = -1)))
 /\ ((s.phase ~= phase5)
     => (    (\A i:ID (\E j:ID
                (s.r[i] = s'.r[j]
                 /\ s.cand[i] = s'.cand[j]
                 /\ s.nthline[i] = s'.nthline[j])))
          /\ (\A j:ID (\E i:ID
                (s.r[i] = s'.r[j]
                 /\ s.cand[i] = s'.cand[j]
                 /\ s.nthline[i] = s'.nthline[j])))))
 /\ ((s.phase = phase5)
     => (\A i:ID
          (s.r[i] = s'.r[i]
           /\ s.cand[i] = s'.cand[i]
           /\ s.nthline[i] = s'.nthline[i])))
)
```

以下では，関係 bas がバックワード匿名シミュレーションであることを示す．全域性は，次を証明する．

```
% --- totality
prove (\A s:States[FO092'] (bas(s, s)))
```

初期状態の対応は，次を証明すればよい．

```
% --- start state condition
prove
 (\A s:States[FO092']
     (\A s':States[FO092']
         ((start(s:States[FO092'])
           /\ bas(s, s'))
          => start(s'))))
..
```

ステップ対応は，匿名アクションか否かに分けて示す．

```
% --- step correspondence ( 匿名アクションでない場合)
prove
  (bas(s2, s2')
   /\ enabled(s1, a)
   /\ effect(s1, a) = s2
   /\ ~anonymp(a)
   /\ ~input(a))
   => (\E s1':States[FO092']
          (\E a':Actions[FO092']
              (bas(s1, s1')
               /\ enabled(s1', a')
               /\ effect(s1', a') = s2'
               /\ seal(a) = seal(a'))))
..

% --- step correspondence ( 匿名アクションの場合)
prove
  (bas(s2, s2')
   /\ enabled(s1, a)
   /\ effect(s1, a) = s2
   /\ a = disclose(ar, anl, ac))
   => (\A a':Actions[FO092']
          ((a' = disclose(ar', anl', ac')
            /\ (   (\A i:ID (\A j:ID
                        ((i:ID ~= j)
                          => (ar[i] ~= ar[j]
                              /\ ar'[i] ~= ar'[j])))))
                /\ (\A i:ID (\E j:ID
                        (ar[i] = ar'[j]
                         /\ anl[i] = anl'[j]
                         /\ ac[i] = ac'[j])))
                /\ (\A j:ID (\E i:ID
                        (ar[i] = ar'[j]
                         /\ anl[i] = anl'[j]
                         /\ ac[i] = ac'[j])))
                /\ (\A i:ID
                        ((anl[i] = -1)
                         <=> (anl'[i] = -1)))))
             => (\E s1':States[FO092']
                    (bas(s1, s1')
                     /\ enabled(s1', a')
                     /\ effect(s1', a') = s2'))))
..
```

我々は，IOA 言語のための形式的検証ツール IOA-Toolkit[11] と定理証明器 LP[4] を使って，上記を証明した (詳細について，証明スクリプトと実行結果を別ファイルとして添付する)．以上より，次が成り立つ.

補題 1 関係 bas は，*I/O*-オートマトン FO092' の，$A_{\mathtt{FO092}}$ および seal に関するバックワード匿名シミュレーションである. □

最後に，関係 bas の像有限性を考える．状態 s と s' が bas(s, s') を満たすとき，

- 変数 phase，bbd，forOpen の値は，状態 s と s' では一致
- 配列 r，配列 cand，配列 nthline は，インデックス (投票者) に関する入れ換えの関係 (ただし，フェーズ 1 を行った (行わなかった) 投票者間のみで入れ換えてよい)

である．また，問題設定より，投票者数は有限である．以上より，任意の状態 s に関して，フェーズ 1 を行った後の投票者の数を n_1 人と，行っていない投票者の数を n_2 人とすると，$|\{\texttt{s'} \mid \texttt{bas(s, s')}\}| = n_1!\, n_2!$ である．これより，関係 bas は像有限である．以上より，次が成り立つ．

定理 2 seal(FOO92) は $A_{\texttt{FOO92}}$ に関してトレース匿名的である．

証明： 定理 1，補題 1，および関係 bas の像有限性より，seal(FOO92') は $A_{\texttt{FOO92}}$ に関してトレース匿名的．さらに命題 2 より，定理の主張が成り立つ． □

5 議論

5.1 フォワード匿名シミュレーションとバックワード匿名シミュレーション

文献 [7] で，我々は，フォワード匿名シミュレーション[1]と呼ぶ関係を導入した．

定義 8 *I/O*-オートマトン X，出力アクションの集合族 A *(*ただし，$A', A'' \in A$ かつ $A' \neq A''$ ならば，A' と A'' は互いに素*)*，および X の $sig(X) - \bigcup A$ へのシール f を考える．X の A および f に関するフォワード匿名シミュレーション $fas_{A,f}$ は，次を満たす $states(X)$ 上の二項関係である：

1. 任意の $s \in start(X)$ に関して，$fas_{A,f}(s, s)$.
2. 任意の $s_1, s_2, s_1' \in states(X)$ および $a \in sig(X)$ に関して，$fas_{A,f}(s_1, s_1')$ かつ $s_1 \xrightarrow{a}_X s_2$ ならば，
   - ある $A' \in A$ に関して $a \in A'$ ならば，すべての $a' \in A'$ に関して，ある状態 s_2' が存在して $fas_{A,f}(s_2, s_2')$ かつ $s_1' \overset{a'}{\Longrightarrow}_X s_2'$.

[1]文献 [7] 中では，単に匿名シミュレーションと呼んだ．

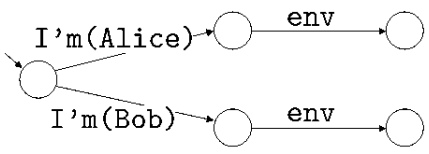

図 9: D2'

   - $a \notin \bigcup_{A' \in A} A'$ ならば，ある状態 s_2' およびアクション a' が存在して，$fas_{A,f}(s_2, s_2')$, $s_1' \overset{a'}{\Longrightarrow}_X s_2'$ かつ $f(a) = f(a')$.

次の定理より，フォワード匿名シミュレーションを見つけることで，トレース匿名性を証明できる．

定理 3 *I/O*-オートマトン X，出力アクションの集合族 A *(*ただし，$A', A'' \in A$ かつ $A' \neq A''$ ならば，A' と A'' は互いに素*)*，および X の $sig(X) - \bigcup A$ へのシール f を考える．X の A および f に関するフォワード匿名シミュレーション $fas_{A,f}$ が存在するならば，$f(X)$ は A に関してトレース匿名的である．

証明： 関係 $fas_{A,f}$ が，$anonym_A(f(X))$ から $f(X)$ へのフォワードシミュレーションであることを示すことによる[2] (フォワードシミュレーションの定義は，文献 [9] など)． □

フォワード匿名シミュレーションでトレース匿名性を証明する場合は，これから起こる事象を前もって宣言するために，匿名アクションを使う．たとえば，第 3.1 節の問 2 を図 9 の I/O-オートマトン D2' として形式化した場合，フォワード匿名シミュレーションが利用できる．ここで，出力アクション I'm(Alice) と I'm(Bob) は，その後に誰が募金を行うかをあらわす．図 9 の D2' に関してはフォワード匿名シミュレーションのみが，図 2 の D2 に関してはバックワード匿名シミュレーションのみが，それぞれ存在する．

フォワード匿名シミュレーションによる証明は，次の点において，バックワード匿名シミュレーションを使う場合よりも，証明の手間が少ないと考えられる．

- 全域性の証明が不要であり，初期状態の対応も簡単なもの (反射律のみ) を示せばよい．
- ステップ対応の証明において，到達可能な状態で成り立つ補題を使うことが容易 (なぜなら，遷

[2]文献 [7] では，シール f が恒等関数で，さらに匿名アクションの集合族 A に関して $|A| = 1$ の場合の証明が示されている．

移 $s_1 \stackrel{a}{\Longrightarrow}_X s_2$ に関して，s_1 が到達可能なとき s_2 も到達可能．しかし，逆は必ずしも成り立たないので，バックワード匿名シミュレーションの場合は，そのような補題を使うことが難しい)．

しかし，起こる事象を前もって宣言するために匿名アクションを使うのではなく，起こった事象が何だったかを事後に開示するために使う方が，形式化が容易な場合がある．たとえば，FOO92 では，どの投票者の投票内容から投票結果が決まるのか，前もって宣言するのは難しい．なぜなら，投票結果が定まるのはプロトコルの最後のフェーズにおいてであり，その時点でフェーズ 3 までの処理を完了した投票者の投票内容のみによって，投票結果が定まるからである．このような場合，バックワード匿名シミュレーションによる証明が適している．もし，フォワード匿名シミュレーションで FOO92 のトレース匿名性を示す場合，各投票者がどのフェーズまで処理を行うかを事前に宣言し，宣言した通りに振る舞うようにしなければならない．そのためには，投票プロトコル本体に対する変更が必要になる (具体的には，各フェーズのタイムアウトの条件や，どの投票者が動作してよいかに関する条件を，プロトコル記述に加える必要がある)．さらに，変更後のプロトコル記述に関するトレース匿名性から，変更前のプロトコル記述に関する匿名性が導けることを，別途証明する必要がある．バックワード匿名シミュレーションによる FOO92 の証明では，FOO92 の本体部分のプロトコル記述に `disclose` アクションを加えるだけでよく，フォワード匿名シミュレーションによる証明で必要になるプロトコル記述の変更やそれに伴う付加的な証明は不要であった．このことから，これら二種類の匿名シミュレーションの適用について，

- 計算の開始時点で，計算結果に関与する行為者が確定している分散システムの場合は，フォワード匿名シミュレーションによる証明が適している．
- 計算結果にどの行為者が関与するか，計算の開始時点や計算途中で決まらない分散システムに関しては，バックワード匿名シミュレーションが適している．

と考えられる．

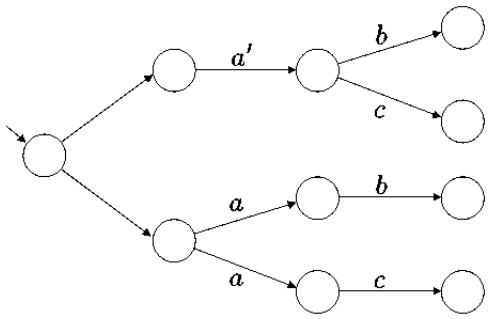

図 10: バックワード匿名シミュレーションもフォワード匿名シミュレーションもない場合

5.2 I/O-オートマトンの既存の証明法との関係

定理 1 の証明より，I/O-オートマトン $f(X)$ の匿名アクション集合族 A に関するトレース匿名性を示すことは，$anonym_A(f(X))$ から $f(X)$ へのバックワードシミュレーションの存在を証明することによって行える (同様に，定理 3 の証明より，$anonym_A(f(X))$ から $f(X)$ へのフォワードシミュレーションの存在を証明してもトレース匿名性が得られる)．しかし，この方法では，擬決定的とは限らない I/O-オートマトン $anonym_A(f(X))$ を生成しなければならない．I/O-オートマトンが擬決定的とは，任意の状態 s およびアクション a に関して，s の a に関する次状態が高々ひとつしか存在しないことをいう (本稿の検証例でも，擬決定的な I/O-オートマトンとして FOO92 を記述した)．擬決定的な I/O-オートマトンは，等式を使って形式化できることが知られている．そのため，LP などの等式推論の機能を持つ定理証明器で効率的に匿名性を検証できると考えられる．この点において，擬決定的なシステムのみで匿名性検証が可能な本稿の方法は，バックワードシミュレーションの存在を証明する方法に比べ，等式推論に基づく定理証明器を用いる上で都合がよい．

本稿で示したバックワード匿名シミュレーションやフォワード匿名シミュレーションの存在は，トレース匿名性のための必要十分条件ではない．たとえば，図 10 の I/O-オートマトン (匿名アクションの集合族として，$A = \{\{a, a'\}\}$ とする．また，ラベルのない矢印は，内部遷移を表す) は，トレース匿名的だがバックワード匿名シミュレーションもフォワード匿名シミュレーションも持たない．I/O-オートマトンの理論では，フォワード・バックワードシミュレーションやバックワード・フォワードシミュレーション

と呼ばれる関係が知られており，ある条件の下でトレース包含と同等であることが知られている [9]．そこで，I/O-オートマトン X および出力アクションの集合族 A に関して，$anonym_A(X)$ から X へのフォワード・バックワードシミュレーション (バックワード・フォワードシミュレーション) となるような二項関係に基づく匿名性の証明手法を設計できれば，本稿の手法で扱えない場合も証明できると考える．

5.3 関連研究

Schneider と Sidiropoulos は，プロセス代数 CSP 上で，強匿名性 (strong anonymity) を導入した [10]．強匿名性を定義するため，擬リネーミングの逆関係を適用した I/O-オートマトンを定義する．

定義 9 *I/O-*オートマトン X および X の擬リネーミング f を考える．ここで，*I/O-*オートマトン $f^{-1}(X)$ を次で定める．

- $states(f^{-1}(X)) = states(X)$,
- $start(f^{-1}(X)) = start(X)$,
- $in(f^{-1}(X)) = \{a \mid f(a) \in in(X)\}$,
- $out(f^{-1}(X)) = \{a \mid f(a) \in out(X)\}$,
- $int(f^{-1}(X)) = \{a \mid f(a) \in int(X)\}$,
- $sig(f^{-1}(X)) = in(f^{-1}(X)) \cup out(f^{-1}(X)) \cup int(f^{-1}(X))$,
- $trans(f^{-1}(X)) = \{(s_1, a', s_2) \mid (s_1, a, s_2) \in trans(X) \wedge f(a') = a\}$.

I/O-オートマトン X の状態 s でアクション a が発火可能なとき，I/O-オートマトン $f^{-1}(X)$ の状態 s では集合 $f^{-1}(a) = \{a' \mid f(a') = a\}$ 内の任意のアクションが発火できる．

定義 10 *I/O-*オートマトン X および出力アクションの集合族 $A = \{A_1, \ldots, A_n\}$ *(*ただし，$A', A'' \in A$ かつ $A' \neq A''$ ならば，A' と A'' は互いに素*)* に関して，擬リネーミング $f_A : sig(X) \to sig(X) \cup \{\alpha_1, \ldots, \alpha_n\}$ を次で定める．

$$\begin{cases} f_A(x) = \alpha_i & \text{if } x \in A_i \text{ for some } i \in \{1, \ldots, n\} \\ f_A(x) = x & \text{if } x \notin \bigcup_{1 \le i \le n} A_i, \end{cases}$$

ただし，$\alpha_1, \ldots, \alpha_n \notin sig(X)$ で，さらに $i \neq j$ ならば $\alpha_i \neq \alpha_i$ とする．$traces(f_A^{-1}(f_A(X))) = traces(X)$ のとき，X は A に関して強匿名という．

匿名アクションの集合族 A に関して $|A| = 1$ のとき，定義 10 は，文献 [10] の定義に一致する．

命題 3 任意の *I/O-*オートマトン X および出力アクションの集合族 $A = \{A_1, \ldots, A_n\}$ *(*ただし，$A', A'' \in A$ かつ $A' \neq A''$ ならば，A' と A'' は互いに素*)* に関して，$anonym_A(X)$ と $f_A^{-1}(f_A(X))$ は同一である．ただし，写像 f_A は，定義 *10* の擬リネーミング． □

命題 3 より，トレース匿名性は文献 [10] の定義の自然な拡張と言える．また，CSP などプロセス代数の枠組で，バックワード匿名シミュレーションやフォワード匿名シミュレーションの定義や結果を使うことも可能であると考える．

匿名性を扱える枠組として，applied π-計算 [1, 2] が知られている．この計算モデルでは，ふたつのプロセスの観測等価性を，labelled bisimulation と呼ばれる関係を用いて証明する．観測等価性は合同であり，攻撃者を含みうる任意の文脈において，情報が外部に漏洩しないことが保証される．このように，applied π-計算では，より能動的な攻撃者も扱える証明手法が考案されている．本稿の I/O-オートマトンと IOA 言語による方法でも，正当なプロセスからの入出力をあらわす出力アクションと同じ `eff`-部を持つ出力アクションを使って，攻撃者による入出力を形式化できると考える．すなわち，攻撃者による入出力をあらわす出力アクションを内部遷移に置き換えた I/O-オートマトンと正当なプロセスのみから成る I/O-オートマトンの間のトレース一致を示すことで，攻撃者がある場合の匿名性の議論ができると考えている．

FOO92 プロトコルの検証例としては，Kremer と Ryan の検証が挙げられる [8]．Kremer らは，applied π-計算で FOO92 の動作を記述し，ProVerif [3] で二人の投票者の入れ換え可能性に関する検証を行った．文献 [8] では，全ての投票者が開票フェーズまで行うようにモデル化されている．すなわち，タイムアウトによって途中で投票処理をやめてしまう投票者に関する匿名性は議論されていない．これに対し，我々は，これらの投票者もモデル化し検証を行った．

6 まとめと今後の課題

本稿では，トレース匿名性の証明手法として，バックワード匿名シミュレーションを導入した．また，像有限なバックワード匿名シミュレーションの存在が，トレース匿名性の十分条件であることを示した．さ

らに，トレース匿名性の検証例として，藤岡・岡本・太田の電子投票プロトコル (FOO92 プロトコル) に対する匿名性検証を行った．この検証では，I/O-オートマトンに基づく仕様記述言語でプロトコルを記述し，定理証明器を用いて匿名性を証明した．

今後は，電子商取引や電子現金のプロトコルに対する匿名性解析などにも，本稿の手法を適用したい．また，文献 [5] では，様相論理 (なかでも，epistemic logic) を用いた，いくつかの匿名性の条件が述べられている．さらに，それら条件のうちのひとつが，文献 [10] の強匿名性に対応することが示されている．文献 [5] の他の条件 (確率に基づく条件など) を，トレースに基づいて証明するよう，本稿の検証法を拡張することは，重要な課題として挙げられる．

謝辞

この研究の機会をいただいた，NTT コミュニケーション科学基礎研究所 情報基礎理論研究グループの白柳潔グループリーダに感謝いたします．また，多くの御討論・御助言をいただいた，東京大学大学院の萩谷昌己先生，NTT 情報流通プラットフォーム研究所の岡本龍明主席研究員，藤岡淳グループリーダに感謝いたします．

参考文献

[1] M. Abadi and C. Fournet. “Mobile values, new names, and secure communication”. In *POPL ’01*, pp.104–115, ACM Press, 2001.

[2] M. Abadi and C. Fournet. “Private authentication”. *TCS*, Vol. 322, pp.427–476, 2004.

[3] B. Blanchet. “An efficient crytographic protocol verifier based on Prolog rules”. In *CSFW ’01*, pp.82–96. IEEE CS Press, 2001.

[4] S. J. Garland et al. “An overview of Larch”. In LNCS 693, pp.329–348. Springer-Verlag, 1993.

[5] J. Y. Halpern and K. R. O’Neill. “Anonymity and information hiding in multiagent systems”. *J. of Computer Security*, to appear.

[6] D. Hughes and V. Shmatikov. “Information hiding, anonymity and privacy: a modular approach”. *J. of Computer Security*, Vol. 12, No.1, pp.3–36, 2004.

[7] 河辺 義信, 真野 健, 櫻田 英樹, 塚田 恭章. “無限状態システムの匿名性検証”. **電子情報通信学会 第 *18* 回回路とシステム軽井沢ワークショップ**, pages 293-298, 2005.

[8] S. Kremer and M. Ryan. “Analysis of an Electronic Voting Protocol in the Applied Pi Calculus”. In *ESOP ’05*, LNCS 3444, pp.186–200. Springer-Verlag, 2005.

[9] N. A. Lynch and F. Vaandrager. “Forward and backward simulations — part I: Untimed systems”. *Inform. and Comput.*, Vol. 121, No. 2, pp. 214-233, 1995.

[10] S. Schneider and A. Sidiropoulos, “CSP and anonymity”. In *ESORICS ’96*, LNCS 1146, pp.198–218. Springer-Verlag, 1996.

[11] J. F. Soegaard-Andersen, S. J. Garland, J. V. Guttag, N. A. Lynch, and A. Pogosyants. “Computer-assisted simulation proofs”. In *CAV ’93*, LNCS 697, pp. 305–319. Springer-Verlag, 1993.

A 命題 2 の証明

まず，FOO92’ のトレース集合から FOO92 のトレース集合を求めるための，以下の変換を導入する．

定義 11 トレース $t \in traces(anonym_{A_{\texttt{FOO92}}}(\texttt{FOO92'}))$ に現れるアクション $\texttt{phase1}(a_1 \cdots a_n)$ $(n \geq 0)$ を列 $a_1.\cdots.a_n.\texttt{timeoutPhase1}$ に置き換えた結果を，$[t]$ と書く．

定義 12 トレース $t \in traces(anonym_{A_{\texttt{FOO92}}}(\texttt{FOO92}))$，アクション $a_1, \ldots, a_n$ $(n \geq 0)$ および出力アクションの列 t' に関して

$$t = a_1.\cdots.a_n.\texttt{timeoutPhase1}.t'$$

のとき，$pref_{\texttt{phase1}}(t)$ を，列 $a_1.\cdots.a_n$ の全てのプレフィックスの集合とする．

定義 13 任意の $t \in traces(anonym_{A_{\texttt{FOO92}}}(\texttt{FOO92'}))$ および t の先頭のアクション $\texttt{phase1}(ls)$ に関して，記号

$$\texttt{v2a\_and\_a2v}(i, m_1, m_1')$$
$$\texttt{v2a\_and\_a2v}(i, m_2, m_2')$$

が列 ls の異なる位置に現れるとき，かつそのときのみ，$dup(t)$ と書く．

上記に関して，次の補題を証明できる．

補題 2 FOO92，FOO92’ および $A_{\texttt{FOO92}}$ に関して，

$$\begin{aligned}
&traces(\texttt{FOO92}) \\
&\quad = \{[t] \mid t \in traces(\texttt{FOO92'}) \wedge \neg dup(t)\} \\
&\qquad \cup \{t' \mid t \in traces(\texttt{FOO92'}) \wedge \neg dup(t) \\
&\qquad\qquad \wedge t' \in pref_{\texttt{phase1}}([t])\} \\
&traces(anonym_{A_{\texttt{FOO92}}}(\texttt{FOO92})) \\
&\quad = \{[t] \mid t \in traces(anonym_{A_{\texttt{FOO92}}}(\texttt{FOO92'})) \\
&\qquad\qquad \wedge \neg dup(t)\} \\
&\qquad \cup \{t' \mid t \in traces(anonym_{A_{\texttt{FOO92}}}(\texttt{FOO92'})) \\
&\qquad\qquad \wedge \neg dup(t) \wedge t' \in pref_{\texttt{phase1}}([t])\}
\end{aligned}$$

である． □

ここで，次を定める．

定義 14 任意の $t \in traces(anonym_{A_{\mathtt{F0092}}}(\mathtt{F0092'}))$ に関して，$\mathtt{seal}(t)$ を次の条件のみで定める．

- t が空列のときは，$\mathtt{seal}(t) = t$,
- $t = a.t'$ *(*a はアクション，t' は出力アクションの列*)* のときは，$\mathtt{seal}(t) = \mathtt{seal}(a).\mathtt{seal}(t')$.

$pref_{\mathtt{phase1}}$，$\mathtt{seal}$，$[\cdot]$ および dup に関して，次のみっつの命題が成り立つ．

命題 4 F0092，F0092' および $A_{\mathtt{F0092}}$ に関して，

$$\begin{aligned}&\{t' \mid t \in traces(anonym_{A_{\mathtt{F0092}}}(\mathtt{F0092'}))\\ &\qquad\qquad \wedge\neg dup(t) \wedge t' \in pref_{\mathtt{phase1}}([t])\}\\ &= \{t' \mid t \in traces(\mathtt{F0092'})\\ &\qquad\qquad \wedge\neg dup(t) \wedge t' \in pref_{\mathtt{phase1}}([t])\}\end{aligned}$$

である． □

命題 5 任意の $t \in traces(anonym_{A_{\mathtt{F0092}}}(\mathtt{F0092'}))$ に関して，$\mathtt{seal}([t]) = [\mathtt{seal}(t)]$ である． □

命題 6 任意の $t \in traces(anonym_{A_{\mathtt{F0092}}}(\mathtt{F0092'}))$ に関して，$\neg dup(t)$ ならば $\neg dup(\mathtt{seal}(t))$． □

以上より，命題 2 が示せる．

命題 7 (命題 2) $\mathtt{seal}(\mathtt{F0092'})$ が $A_{\mathtt{F0092}}$ に関してトレース匿名的ならば，$\mathtt{seal}(\mathtt{F0092})$ も $A_{\mathtt{F0092}}$ に関してトレース匿名的．

証明： $\mathtt{seal}(\mathtt{F0092'})$ が $A_{\mathtt{F0092}}$ に関してトレース匿名的なとき，

$$\begin{aligned}&traces(\mathtt{seal}(\mathtt{F0092'}))\\ &\quad = traces(anonym_{A_{\mathtt{F0092}}}(\mathtt{seal}(\mathtt{F0092'})))\end{aligned}$$

である．ここで，

$$\begin{aligned}&traces(\mathtt{seal}(\mathtt{F0092}))\\ &\quad = \{[t] \mid t \in traces(\mathtt{seal}(\mathtt{F0092'})) \wedge \neg dup(t)\}\\ &\qquad \cup\{\mathtt{seal}(t') \mid t \in traces(\mathtt{F0092'})\\ &\qquad\qquad \wedge\neg dup(t) \wedge t' \in pref_{\mathtt{phase1}}([t])\}\\ &traces(anonym_{A_{\mathtt{F0092}}}(\mathtt{seal}(\mathtt{F0092})))\\ &\quad = \{[t] \mid t \in traces(anonym_{A_{\mathtt{F0092}}}(\mathtt{seal}(\mathtt{F0092'})))\\ &\qquad\qquad \wedge\neg dup(t)\}\\ &\qquad \cup\{\mathtt{seal}(t') \mid t \in traces(\mathtt{F0092'})\\ &\qquad\qquad \wedge\neg dup(t) \wedge t' \in pref_{\mathtt{phase1}}([t])\}\end{aligned}$$

を示せば，

$$\begin{aligned}&traces(\mathtt{seal}(\mathtt{F0092}))\\ &\quad = traces(anonym_{A_{\mathtt{F0092}}}(\mathtt{seal}(\mathtt{F0092})))\end{aligned}$$

である．まず，$traces(\mathtt{seal}(\mathtt{F0092}))$ に関する等式は，

$$\begin{aligned}&traces(\mathtt{seal}(\mathtt{F0092}))\\ &\quad = \{\mathtt{seal}([t]) \mid t \in traces(\mathtt{F0092'}) \wedge \neg dup(t)\}\\ &\qquad \cup\{\mathtt{seal}(t') \mid t \in traces(\mathtt{F0092'}) \wedge \neg dup(t)\\ &\qquad\qquad \wedge t' \in pref_{\mathtt{phase1}}([t])\} \quad \text{(補題 2 を使う)}\\ &\quad = \{[\mathtt{seal}(t)] \mid t \in traces(\mathtt{F0092'}) \wedge \neg dup(t)\}\\ &\qquad \cup\{\mathtt{seal}(t') \mid t \in traces(\mathtt{F0092'}) \wedge \neg dup(t)\\ &\qquad\qquad \wedge t' \in pref_{\mathtt{phase1}}([t])\} \quad \text{(命題 5 を使う)}\\ &\quad = \{[t] \mid t \in traces(\mathtt{seal}(\mathtt{F0092'})) \wedge \neg dup(t)\}\\ &\qquad \cup\{\mathtt{seal}(t') \mid t \in traces(\mathtt{F0092'}) \wedge \neg dup(t)\\ &\qquad\qquad \wedge t' \in pref_{\mathtt{phase1}}([t])\} \quad \text{(命題 6 を使う)}\end{aligned}$$

である．一方，$traces(anonym_{A_{\mathtt{F0092}}}(\mathtt{seal}(\mathtt{F0092})))$ に関する等式は，

$$\begin{aligned}&traces(anonym_{A_{\mathtt{F0092}}}(\mathtt{seal}(\mathtt{F0092})))\\ &\quad = traces(\mathtt{seal}(anonym_{A_{\mathtt{F0092}}}(\mathtt{F0092})))\\ &\qquad\qquad \text{(命題 1 を使う)}\\ &\quad = \{\mathtt{seal}([t]) \mid t \in traces(anonym_{A_{\mathtt{F0092}}}(\mathtt{F0092'}))\\ &\qquad\qquad \wedge\neg dup(t)\}\\ &\qquad \cup\{\mathtt{seal}(t') \mid t \in traces(anonym_{A_{\mathtt{F0092}}}(\mathtt{F0092'}))\\ &\qquad\qquad \wedge\neg dup(t) \wedge t' \in pref_{\mathtt{phase1}}([t])\}\\ &\qquad\qquad \text{(補題 2 を使う)}\\ &\quad = \{\mathtt{seal}([t]) \mid t \in traces(anonym_{A_{\mathtt{F0092}}}(\mathtt{F0092'}))\}\\ &\qquad\qquad \wedge\neg dup(t)\}\\ &\qquad \cup\{\mathtt{seal}(t') \mid t \in traces(\mathtt{F0092'}) \wedge \neg dup(t)\\ &\qquad\qquad \wedge t' \in pref_{\mathtt{phase1}}([t])\} \quad \text{(命題 4 を使う)}\\ &\quad = \{[\mathtt{seal}(t)] \mid t \in traces(anonym_{A_{\mathtt{F0092}}}(\mathtt{F0092'}))\}\\ &\qquad\qquad \wedge\neg dup(t)\}\\ &\qquad \cup\{\mathtt{seal}(t') \mid t \in traces(\mathtt{F0092'}) \wedge \neg dup(t)\\ &\qquad\qquad \wedge t' \in pref_{\mathtt{phase1}}([t])\} \quad \text{(命題 5 を使う)}\\ &\quad = \{[t] \mid t \in traces(\mathtt{seal}(anonym_{A_{\mathtt{F0092}}}(\mathtt{F0092'})))\}\\ &\qquad\qquad \wedge\neg dup(t)\}\\ &\qquad \cup\{\mathtt{seal}(t') \mid t \in traces(\mathtt{F0092'}) \wedge \neg dup(t)\\ &\qquad\qquad \wedge t' \in pref_{\mathtt{phase1}}([t])\} \quad \text{(命題 6 を使う)}\\ &\quad = \{[t] \mid t \in traces(anonym_{A_{\mathtt{F0092}}}(\mathtt{seal}(\mathtt{F0092'})))\}\\ &\qquad\qquad \wedge\neg dup(t)\}\\ &\qquad \cup\{\mathtt{seal}(t') \mid t \in traces(\mathtt{F0092'}) \wedge \neg dup(t)\\ &\qquad\qquad \wedge t' \in pref_{\mathtt{phase1}}([t])\} \quad \text{(命題 1 を使う)}\end{aligned}$$

である．以上より，$\mathtt{seal}(\mathtt{F0092})$ は，$A_{\mathtt{F0092}}$ に関してトレース匿名的である． □

```
automaton FO092
  signature
    output   v2a_and_a2v(i:ID, m:MES, m':MES), timeoutPhase1,
             v2c_and_checkBB(i:ID, b:Set[MES], ell:Int, m:MES), timeoutPhase2,
             v2b(i:ID, m:MES), announceResult(b:Set[MES]),
             disclose(ar:Array[ID, Key], anl:Array[ID, Int], ac:Array[ID, CAND])
  states
    phase:       Phase     := phase1,      % 現在のフェーズ名
    r:           Array[ID, Key],           % r[i] ：投票者 i が使う鍵
    cand:        Array[ID, CAND],          % cand[i] ：投票者 i が支持する候補者
    nthline:     Array[ID, Int],           % nthline[i] ：掲示板に示された番号の格納用 (兼プログラムカウンタ)
    bbd:         Set[MES]  := {},          % 掲示板の内容
    forOpen:     Set[MES]  := {}           % 集計者の鍵の保存用
      so that    (\A i:ID (\A j:ID (i ~= j => r[i] ~= r[j])))    % 注：so that ... は，初期状態に関する宣言
              /\ (\A i:ID (nthline[i] = -1))

  transitions
    output v2a_and_a2v(i, m, m')           % フェーズ 1 において，投票者 i はアドミニストレータに投票券 m を送る．
      pre phase = phase1                   % さらに署名された投票券 m' を受け取る．
          /\ nthline[i] = -1
          /\ m = sign(blind(commit(cand[i], r[i]), blinder(i)), prv(i))
          /\ m' = sign(checksign(m, pub(i)), prv(adm))
      eff nthline[i] := 0

    output timeoutPhase1                   % フェーズ 1 のタイムアウト．
      pre phase = phase1
      eff phase := phase2

    output v2c_and_checkBB(i, b, ell, m)   % フェーズ 2 において，投票者 i は集計者に対し，署名された投票券のアン
      pre phase = phase2                   % ブラインド化 m を送る．さらに (ell, m, checksign(m, pub(adm))) が
          /\ nthline[i] = 0                % 掲示板 b に加わったことを確認する (ell は，投票 m に対応する番号)．
          /\ m = unblind(                  % (注：m = sign(commit(cand[i], r[i]), prv(adm)))
                   sign(checksign(
                     sign(blind(commit(cand[i], r[i]), blinder(i)), prv(i)),
                     pub(i)), prv(adm)), blinder(i))
          /\ b = bbd
          /\ ell = size(b)+1
      eff nthline[i] := ell;
          bbd := insert(entry(ell, m, checksign(m, pub(adm))), bbd)

    output timeoutPhase2                   % フェーズ 2 のタイムアウト．
      pre phase = phase2
      eff phase := phase3

    output v2b(i, m)                       % フェーズ 3 において，投票者 i は，開票に必要な情報 m (番号と鍵の
      pre phase = phase3                   % 組) を集計者に送る．
          /\ nthline[i] > 0
          /\ m = ellrnd(nthline[i], r[i])
      eff nthline[i] := -(nthline[i]+1);
          forOpen := insert(m, forOpen)

    output announceResult(b)               % 集計者は，フェーズ 3 を締め切って，開票結果 b を公表する．
      pre phase = phase3
          /\ b = result(bbd, forOpen)
      eff phase := phase4

% ---- 上記までが，FO092 プロトコルの本体．次の disclose は，匿名性を議論するために付加するアクション．

    output disclose(ar, anl, ac)           % 配列変数 r, nthline, cand の値を，外部に開示する．
      pre phase = phase4
          /\ ar  = r
          /\ anl = nthline
          /\ ac  = cand
      eff phase := phase5;
          r := rSort(ar);
          cand := constant(cand0);
          nthline := constant(-1)
```

図 6: FO092：藤岡・岡本・太田の電子投票プロトコル