電子指紋符号の結託攻撃パラメータ推定のための特徴ベクトル導出及びその次元削減

安井 達哉[1,a)]　栗林 稔[1,b)]　舩曳 信生[1,c)]

概要：電子指紋符号に対して結託攻撃を行った結託者を検出する最適な検出器が提案されている．しかし，この検出器では，結託攻撃戦略と結託者数の情報が必要となる．従来研究では，符号語のシンボルの偏りに注目することでパラメータを高精度で推定し，最適な検出器に近い性能を達成している．一方で，複数の結託者を検出するために符号語生成用パラメータの値が増加した場合，この推定器では推定のためのベクトル空間の次元が増加することが考えられる．本研究では，推定に必要な特徴ベクトルのドミナントな成分に着目し，ベクトル空間の次元の増加に依存せず一定の次元で推定が行えるように，推定のための高次元ベクトル空間の次元削減法を提案する．また，従来研究で考慮されていなかった，異なる符号語生成パラメータでの推定器を実装し，計算機シミュレーションによって検出性能の比較を行った．計算機シミュレーションの結果，次元削減を行った状態でも高い性能を達成し，最適な検出器に近い性能を有することが確認できた．

キーワード：電子指紋符号，最適な検出器，結託攻撃戦略，結託者数，推定器

Extraction of Characteristic Vectors and its Dimension Reduction for Estimating Collusion Strategy in Fingerprinting Codes

Tatsuya Yasui[1,a)]　Minoru Kuribayashi[1,b)]　Nobuo Funabiki[1,c)]

Abstract: The optimal detector against colluders in fingerprinting codes has been proposed. However, this detector requires information about the collusion attack strategy and the number of colluders. In the previous studies, the estimator using the bias of symbols of codewords made it possible to estimate these parameters with high accuracy and the performance of detection almost achieved the optimal detector. On the other hand, assuming multiple colluders attack and the parameter for generation of codewords is increased, it may be necessary to reduce dimensions for estimation because it depends on the parameters for generation. This study shows a detector for large colluders using low dimension estimator which uses dominant features and the computational simulation compared with previous studies. As a result of simulation, it reveal that our proposed detector reached the optimal detector and previous estimator without a large scale dimension.

Keywords: Fingerprinting code, Optimal detector, Collusion attack strategy, Colluders, Estimator

1. はじめに

電子指紋技術において，結託耐性を考慮して設計された符号 [1] として，Tardos 符号 [2] や Nuida 符号 [3], [4] がある．これらの符号は，符号語の各ビットをある確率分布に従って決定する方法で符号語を生成しており，符号長が理

[1] 岡山大学大学院自然科学研究科
Graduate School of Natural Science and Technology, Okayama University
[a)] yasui.tatsuya@s.okayama-u.ac.jp
[b)] kminoru@okayama-u.ac.jp
[c)] funabiki@okayama-u.ac.jp

論上最短オーダとなることが証明されている．従来の研究では，これらの符号に対して，適切なパラメータを設定することで符号長を短くする試みがあった．一方で，結託者を検出するための検出器においても結託者の検出性能を向上させる研究が行われている．Tardos 符号において提案されていた追跡アルゴリズム [2] では，ユーザの符号語と不正符号語との相関をスコアとして求めていた．このスコア計算に対して，Škorić らは符号生成の特徴を考慮して見直しを行った [5]．後に，Furon らにより最適な検出器が提案された [6]．この最適な検出器では，不正符号語の他に不正符号語を作るときに使用した攻撃戦略と結託者数を利用している．その結果，無実のユーザと不正者のスコアの分布を理論上最も分離できるようになっている．しかし，攻撃戦略と結託者数は結託者追跡時には未知の情報であるため，これらの情報を高精度に推定しなければ実装は困難であった．そこで，ある特定の守備戦略に基づいたスコア計算を行う研究 [7], [8], [9], [10]，シンボルの偏りから戦略の特徴を考慮しスコア計算における重みを調整した研究 [11] や未知の情報の推定を行う研究 [12], [13] が行われてきた．とりわけ最適な検出器のために攻撃戦略と結託攻撃者数の推定を行う研究では，精度の高い推定結果を得ており，最適な検出器に近い性能を実現している [12], [13].

本研究では，結託攻撃を行ったユーザをより多く検出することを目的とし，推定空間の次元を削除した推定器を提案する．従来研究 [12], [13] で提案した推定器では，符号語生成に用いるパラメータである最大結託者数に応じた次元で推定を行っていたが，より低次元で推定を行うことができれば，最大結託者数に依存しない検出器が実現可能である．計算機シミュレーションでは，推定した結果を最適な検出器に利用しスコアを求め，戦略ごとに検出者数を測定した．その結果，理論値である最適な検出器や従来研究の手法に近い性能を得ることができた．

2. 電子指紋符号

符号に対して結託耐性を持たせる研究として，符号理論に基づいた特殊な符号を設計する手法が提案されている [1]．この符号は，結託耐性を有することから，結託耐性符号と呼ばれいている．結託耐性符号として，1998 年に Boneh と Shaw により c-secure 符号が提案された [1]．c-secure 符号は，c 人の不正者による結託攻撃に対して，誤り確率 ϵ 以下で一人以上を検出可能な符号の総称である．これらの結託耐性符号は，主に電子指紋として埋め込みに用いられることを想定しているため電子指紋符号とも呼ばれている．

2.1 Tardos 符号

c-secure 符号 [1] を構成可能な手法の一つとして Tardos 符号がある． Tardos 符号 [2] は，理論上最短の符号長オーダを実現している．一人以上を検出可能な最大結託者数を c_{max} としたとき符号長 L の電子指紋符号は，以下の手順で構成される．

(1) ある連続型の確率密度関数 $f(P)$ に従って $p_i (1 \leq i \leq L)$ を独立に選び，確率系列 $\boldsymbol{P} = (p_1, \ldots, p_i, \ldots, p_L)$ を生成し秘密裏に保持しておく．ここで，p_i は以下を満たすものとする．

$$\begin{cases} t = 1/300 c_{max} \\ 0 < t' < \pi/4, \sin^2 t' = t, r_i \in [t', \pi/2 - t'] \\ p_i = \sin^2 r_i, t \leq p_i \leq 1 - t \end{cases} \quad (1)$$

また， $f(P)$ は以下のように表される．

$$f(P) = \frac{1}{2\sin^{-1}(1-2t)} \frac{1}{\sqrt{P(1-P)}} \quad (2)$$

(2) j 番目のユーザの符号語 $\boldsymbol{x_j} = (x_{j,1}, \ldots, x_{j,i}, \ldots, x_{j,L})$ は，$\Pr[x_{j,i} = 1] = p_i$ を満たすように $\{0,1\}$ から各ビットにおいて独立にかつランダムに選ぶ．

表 1　Nuida 符号における離散型の確率値

c_{max}	P	Q	c_{max}	P	Q
1,2	0.50000	1.00000	9,10	0.04691	0.19829
3,4	0.21132	0.50000		0.23077	0.20104
	0.78868	0.50000		0.50000	0.20134
5,6	0.11270	0.33201		0.76923	0.20104
	0.50000	0.33598		0.95309	0.19829
7,8	0.88730	0.33201	11,12	0.03377	0.16502
	0.06943	0.24833		0.16940	0.16733
	0.33001	0.25167		0.38069	0.16765
	0.66999	0.25167		0.61931	0.16765
	0.93057	0.24833		0.83060	0.16733
				0.96623	0.16502

2.2 Nuida 符号

Tardos 符号の性能向上を目的とし，Nuida らによって離散型バイアス分布を用いる符号が提案されている [3], [4]．この符号は Nuida 符号と呼ばれており，最大結託者数 c_{max} に対して，p_i の分布を離散的な分布にすることで符号長を減らし，検出性能を向上させている．例として，c_{max} が小さい場合の各パラメータを表 1 に示す．ただし，P は離散型の確率値であり，Q はその確率値の出現する確率である．

2.3 結託攻撃

各ユーザごとに別々の識別情報が埋め込まれているコンテンツを，複数のユーザが持ち寄って比較をした場合，異なる箇所を改ざん消去することができる．このように，複数のユーザの結託によって埋め込まれた識別情報からユーザの識別を困難にする攻撃を結託攻撃と呼ぶ．結託攻撃は，結託者の符号語から特定の攻撃戦略に基づいて，不正符号語 $\boldsymbol{y} = (y_1, \ldots, y_L)$ が作成される．また，攻撃の際に

用いられる攻撃戦略は，いずれもマーキング仮定を満たすと仮定する．

2.3.1 マーキング仮定

結託者が c 人の場合を考える．結託者の ID を $j_1, j_2, \ldots, j_c$ とすると，不正符号語の i 番目のシンボルは y_i は次の集合 $\{x_{j_1,i}, \ldots, x_{j_c,i}\}$ から選択されることを仮定している．もし，集合内の要素 $x_{id,i}(j_1 \leq id \leq j_c)$ がすべて同じであればこのシンボルを変更することができない．これは，結託攻撃を行う場合に互いのコンテンツから異なる箇所を抽出しその箇所を，結託者は識別情報として認識して攻撃を行う．つまり，あらかじめ埋め込まれている符号語のうち，結託者の符号の i 番目のシンボルが同じ場合には結託者が識別情報を観測し得ない．この条件はマーキング仮定と呼ばれ，一般的に結託耐性符号の結託耐性の評価において仮定される．

2.3.2 結託攻撃の種類とパラメータ表記

結託者は，上記のマーキング仮定の下で結託攻撃を行う．例えば，結託者数 $c = 6$ のときの結託者の i 番目のシンボルの集合が，$\{1, 1, 1, 0, 0, 1\}$ の場合，不正符号語の i 番目のシンボル y_i は多数決の戦略 (majority attack) の場合 $y_i = 1$ になり，少数決の戦略 (minority attack) の場合は $y_i = 0$ となる．このような戦略はパラメータ $\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}} = (\theta_0^{str}, \ldots, \theta_\lambda^{str}, \ldots, \theta_c^{str})$ を用いて表記される [6]．各成分 $\theta_\lambda^{str}, 0 \leq \lambda \leq c$ は以下の式によって与えられる．

$$\theta_\lambda^{str} = \Pr[y_i = 1 | \Phi = \lambda], (0 \leq \lambda \leq c) \tag{3}$$

ただし，Φ は次の通りである．

$$\Phi = \sum_{k=1}^{c} x_{j_k,i} \tag{4}$$

マーキング仮定によると $\Phi = 0, \Phi = c$ のときに $\theta_0^{str} = 0, \theta_c^{str} = 1$ が保証される．また，str は戦略名を表し本研究では，$str = \{maj, min, coin, all0, all1, int, WCA\}$ となる 7 戦略を想定する．

2.4 不正者の検出器

不正者を検出するためのアルゴリズムは追跡アルゴリズムと呼ばれ，電子指紋符号においては，その出力によって以下の三種類に分類できる．

Catch-All: 結託者をすべて出力

Catch-Many: 結託者をできる限り出力

Catch-One: 結託者の一人を出力

Tardos 符号や Nuida 符号のような確率的に符号語を生成する符号における最大結託者数 c_{max} は，実際の結託者数 $c = c_{max}$ のときに少なくとも一人を検出できるように設計される．つまり，Catch-One タイプにおいて最も符号長を短くする構築となっている．

また，追跡アルゴリズムでは以下に示す，誤って無実のユーザを検出する確率 ϵ_{FP} と結託者を一人も検出できない確率 ϵ_{FN} の 2 つの誤り確率を考慮する．ただし，Tr は追跡アルゴリズムであり結託者と見なしたユーザを出力する．また，$\boldsymbol{C}$ を結託者全員の集合とする．

- $\epsilon_{FP} = \Pr[Tr(\boldsymbol{y}) \notin \boldsymbol{C} | Tr(\boldsymbol{y}) \neq \emptyset]$
- $\epsilon_{FN} = \Pr[Tr(\boldsymbol{y}) \cap \boldsymbol{C} = \emptyset]$

Tardos が提案した検出器は，ユーザのスコア計算において結託攻撃の戦略 $\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}$ に関係なくスコアが計算される．一方で，スコア計算において，結託攻撃戦略 $\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}$ や結託者数 c を用いれば，理論上最も多くの結託者を検出できる検出器が設計可能であることがわかっている [6]．最適な検出器では，ユーザのスコア S_j がしきい値 Z を超えたユーザを不正者として検出する．最適なスコアを計算するために，攻撃戦略 $\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}$ と結託者数 c を用いて最大事後確率 (MAP: Maximum A Posteriori) を次の式に従って計算する．

$$S_j = \sum_{i=1}^{L} S_{i,j} = \sum_{i=1}^{L} \log \frac{\Pr[y_i | x_{j,i}, \boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}]}{\Pr[y_i | \boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}]} \tag{5}$$

このように計算することで，不正者と無実のユーザとの統計的な分布を最も離すことができ，不正者の最適な検出を行うことができる．これ以降，最適な検出器を MAP 検出器と呼ぶこととする．式 (5) の分母は以下のように計算する．

$$\begin{cases} \Pr[1 | \boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}] = \displaystyle\sum_{\rho=0}^{c} \theta_\rho^{str} \binom{c}{\rho} p_i^\rho (1 - p_i)^{c-\rho} \\ \Pr[0 | \boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}] = 1 - \Pr[1 | \boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}] \end{cases} \tag{6}$$

また，分子も同様に以下のように計算する．

$$\begin{cases} \Pr[1 | 1, \boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}] = \displaystyle\sum_{\rho=1}^{c} \theta_\rho^{str} \binom{c-1}{\rho-1} p_i^{\rho-1} (1 - p_i)^{c-\rho} \\ \Pr[0 | 1, \boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}] = 1 - \Pr[1 | 1, \boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}] \\ \Pr[1 | 0, \boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}] = \displaystyle\sum_{\rho=0}^{c-1} \theta_\rho^{str} \binom{c-1}{\rho} p_i^\rho (1 - p_i)^{c-\rho-1} \\ \Pr[0 | 0, \boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}] = 1 - \Pr[1 | 0, \boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}] \end{cases} \tag{7}$$

式 (5) におけるスコア計算は，攻撃戦略 $\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}$ と結託者数 c が既知の場合において最適であるが，不正符号語 $\boldsymbol{y}$ からこれらのパラメータを推定する必要がある．

2.4.1 しきい値の設定

しきい値 Z の設定では，Tardos の検出器においてスコアをガウス分布で近似して求める方法がある [14]．計算した結託者のスコアと無実のユーザのスコアは中心極限定理により符号長 L が増加するに伴って，それぞれ正規分布に近似可能であり，相補誤差関数を用いることでしきい値を求めることができる．極めて低い誤検出率 ϵ_{FP} を求めるためには，この正規分布右側のテール部分を正確に求める必要がある．しかし，符号長 L が有限長であるため，誤検出

率に応じたしきい値の計算に正規分布による近似は避けるべきであると報告されている [15]．本研究では，テール部分をより正確に求めることができる手法として Rare Event Simulator[6] を用いる．この手法では，条件付確率を用いることで，効率よく精度の高いしきい値を求めることが可能である．

3. 従来研究

この章では，検出側にとって未知のパラメータとされる MAP 検出器のための結託攻撃戦略 $\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}$ と結託者数 c を高精度に推定した従来手法 [12], [13] を紹介する．

3.1 戦略特徴ベクトル

Nuida 符号では，確率系列 $\boldsymbol{P} = (p_1, \ldots, p_i, \ldots, p_L)$ をもとに符号語を生成する．このとき，p_i は符号語の i 番目のビットが “1” になる確率である．これらの確率 p_i は表 1 で示した通り離散型の値であり，その数は有限個の $n_g = \lceil c_{max}/2 \rceil$ 種類となるため，生成された長さ L の符号語のシンボルは確率系列によって n_g 個のグループに分割が可能となる．各グループのシンボルの個数を $\ell_\xi, (1 \leq \xi \leq n_g)$ とすると，ℓ_ξ は $\sum \ell_\xi = L$ を満たす．さらに ℓ_ξ 個のシンボルの “0” と “1” の個数をそれぞれ $\ell_{\xi,0}, \ell_{\xi,1} (\ell_{\xi,0} + \ell_{\xi,1} = \ell_\xi)$ と表記する．マーキング仮定において特定の攻撃戦略に基づいて生成された不正符号語は，シンボルの “0” と “1” の数の割合が，ユーザに配布した符号語のものと異なっている．ユーザ符号語の各シンボルが “1” となる確率 $P_\xi, (1 \leq \xi \leq n_g)$ は，どのユーザ符号語においても，$\ell_{\xi,1}/\ell_\xi$ とほとんど等しい．したがって，以下のような式が成り立つ．

$$(P_1, \ldots, P_\xi, \ldots, P_{n_g}) \approx \left(\frac{\ell_{1,1}}{\ell_1}, \ldots, \frac{\ell_{\xi,1}}{\ell_\xi}, \ldots, \frac{\ell_{n_g,1}}{\ell_{n_g}} \right) \tag{8}$$

一方で，不正符号語における式（8）の右辺は攻撃時の戦略によって異なる．観測された攻撃の違いによる不正符号語の特徴を簡単のため $\boldsymbol{\Gamma} = (\gamma_1, \ldots, \gamma_\xi, \ldots, \gamma_{n_g})$ と表す．ただし，$\gamma_\xi = \ell_{\xi,1}/\ell_\xi$ である．従来手法 [12] では，$\boldsymbol{\Gamma}$ を攻撃戦略のパラメータ θ_c^{str} と結託者数 c を用いて，以下の式より理論的に導出している．

$$\gamma_{c,\xi}^{str} = \sum_{t=1}^{c} \binom{c}{t} P_\xi^t (1 - P_\xi)^{c-t} \theta_t^{str} \tag{9}$$

式（9）により理論的な計算によって導出されるベクトルを $\boldsymbol{\Gamma}_{\boldsymbol{c}}^{\boldsymbol{str}} = (\gamma_{c,1}^{str}, \ldots, \gamma_{c,\xi}^{str}, \ldots, \gamma_{c,n_g}^{str})$ とし，以下，戦略特徴ベクトル（CSCV: Collusion Strategy Characteristic Vector）と呼ぶこととする．攻撃戦略と結託者数の推定では，CSCV と観測値である $\boldsymbol{\Gamma}$ との距離から推定を行う．

3.2 MAP 検出器での検出方法

従来研究 [12], [13] では，MAP 検出器への適用を目的として，以下 3 種類の手法が提案されている．

3.2.1 Basic Method

最も単純な方法として，事前に想定し得るすべての CSCV を計算したうえで，観測した特徴ベクトル $\boldsymbol{\Gamma}$ と各 CSCV との距離を網羅的に計算し，その中から最も距離の短いパラメータを推定値として出力する手法である．各 CSCV と $\boldsymbol{\Gamma}$ との距離を $D^{str,c}$ とすると，以下の式で表せる．

$$\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{s\tilde{t}r}} = \arg\min_{str,c} D^{str,c} \tag{10}$$

本研究では，この方式を以下，Basic Method と呼ぶこととする．

3.2.2 Subset Method

Basic Method では，想定する戦略の種類や結託者数の増加に伴い，推定候補が増えることで推定精度が悪くなることが予想される．そこで，CSCV の空間を同じ結託者数ごとに分割し，推定を小規模で行う方法が提案された．この方式では，複数の推定候補が出力されるため，MAP 検出器を用いてスコア計算を行い，最もスコアが高くなる組み合わせを採用する方式である．この方式を，CSCV の全集合を部分集合（Subset）に分割することから，以下，Subset Method と呼ぶこととする．Subset Method による検出アルゴリズムは以下の通りである．ただし，c_{min} は想定最小結託者数を表し，c'_{max} は，想定最大結託者数を表す．

(1) $c = c_{min}$ で初期化する．
(2) c の Subset 内で式（10）を用いて $\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{s\tilde{t}r}}$ を推定する．
(3) $c \leftarrow c + 1$ に更新する．
(4) $c = c'_{max}$ であれば，ステップ 5 に進み，それ以外はステップ 2 に戻る．
(5) 以下の式で，$\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{s\tilde{t}r}}$ を用いてスコア $\tilde{S}_{j,i}^c$ を計算する．

$$\tilde{S}_{j,i}^c = \log \left(\frac{\Pr[y_i | x_{j,i}, \boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{s\tilde{t}r}}]}{\Pr[y_i | \boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{s\tilde{t}r}}]} \right) \tag{11}$$

(6) 以下の式により，スコアの最大値を選択しユーザのスコアを求める．

$$S_j = \max_{c_{min} \leq t \leq c'_{max}} \left(\sum_{i=1}^{L} \tilde{S}_{j,i}^t \right) \tag{12}$$

3.2.3 Dynamic Method

Subset Method では，想定する結託者数に比例して式（11）を繰り返すため，計算コストを抑える工夫が必要になる．そこで，動的に推定候補数が変わる手法が提案された．以下にその手法を示す．

(1) $t = 0$ に初期化し，半径 d を設定する．
(2) 距離 $D^{str,c}$ を計算する．
(3) 最小距離 $D^{\tilde{str},c}$ にある $\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{s\tilde{t}r}}$ より $S_{i,j}^t$ を計算する．

$$S_{i,j}^t = \left(\log \frac{\Pr[y_i | x_{j,i}, \boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{s\tilde{t}r}}]}{\Pr[y_i | \boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{s\tilde{t}r}}]} \right) \tag{13}$$

(4) $D^{\tilde{str},c} > d$ ならステップ 5 に進み，それ以外なら $t \leftarrow t+1$ に更新し, さらに $D^{\tilde{str},c}$ をすべての距離候補から取り除き，ステップ 3 に進む．

(5) 以下の式により，スコア S_j を計算する．

$$S_j = \max_t \left(\sum_{i=1}^{L} S_{i,j}^t \right) \tag{14}$$

この手法は，観測された不正符号語の特徴 $\mathbf{\Gamma}$ により動的に推定候補数が変化するため，以下，Dynamic Method と呼ぶこととする．

4. 低次元推定器

CSCV による推定では，最大結託者数 c_{max} に比例して推定空間の次元が増加する．この章では，最大結託者数に依存せず一定の次元で推定が行えるよう，CSCV の支配的な成分（ドミナントな成分）のみで推定を行う低次元推定器を提案する．

4.1 符号語生成パラメータと CSCV の関係

離散的な確率分布を用いて電子指紋符号語を生成する Nuida 符号語では，確率系列 $\boldsymbol{P}$ の各要素は，$n_g = \lceil c_{max}/2 \rceil$ 種類となるため推定のために用いる CSCV の次元は n_g 次元となる．そこで，従来研究 [13] では，符号語符号長 L の符号語を 2 元 L 次元空間 $(\mathbb{Z}_2^L)$ のただ一つの点から n_g 次元実空間 $(\mathbb{R}^{n_g})$ への次元削除のための写像であるとしている．しかし，多数の不正者による結託攻撃を想定する場合には，CSCV の次元が高くなり，各結託戦略と結託者数に応じた特徴空間がスパースになると考えられる．そこで，推定においてドミナントな成分のみに着目した推定を行う．

4.2 CSCV の分布とドミナントな成分

CSCV による推定で高精度に戦略を推定するためには，各攻撃戦略 $\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}$ から生成される CSCV 間の推定空間での距離が最も離れる分布になる必要がある．そこで CSCV を以下の 4 方式で比較して考察する．

(1) CSCV のうち 0 に近い半分を使用　$(\gamma_1^{str}, \ldots, \gamma_{ng/2}^{str})$

(2) CSCV のうち 1 に近い半分を使用　$(\gamma_{n_g/2}^{str}, \ldots, \gamma_{n_g}^{str})$

(3) CSCV のうち $1, n_g$ 番目の要素を使用　$(\gamma_1^{str}, \gamma_{n_g}^{str})$

(4) CSCV のうち $n_g/2$ 番目の要素の前後を使用　$(\gamma_{n_g/2-1}^{str}, \ldots, \gamma_{n_g/2+n_g\%2+1}^{str})$

方式 (1) は，CSCV の半分の情報を用いる方式である．推定に用いる CSCV $\mathbf{\Gamma}_{\boldsymbol{c}}^{\boldsymbol{str}}$ は，生成元の結託攻撃戦略 $\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}$ の成分が，$\theta_\lambda^{str} = 1 - \theta_{c-\lambda}^{str}$ となり，対称性を満たすとき，以下の性質を満たす．

$$\gamma_\xi^{str} = 1 - \gamma_{n_g-\xi}^{str} \tag{15}$$

一方で，$\theta_\lambda^{str} \neq 1 - \theta_{c-\lambda}^{str}$ となり成分が非対称となる戦略（all-0, all-1 等）攻撃では式 (15) を満たさない．したがって，方式（1）では，all-0, all-1 攻撃のような非対称な戦略において致命的な情報損失が発生する．方式（2）においても同様である．方式（3）の場合は，符号語生成パラメータ P がどちらも 0 や 1 に近い値となるため，その影響を強く受けることで生成される CSCV も 0 や 1 に近い値となってしまう．その結果，推定空間での CSCV の分布が偏り，誤推定の原因となる．上記理由により方式（1）（2）（3）は推定に不適と判断した．方式（4）においては，all-0, all-1 攻撃のような非対称な攻撃においても適切に推定ができると同時にその他の戦略についても全体に程よく分布する．したがって，推定において次元を削減するためには，これらの方式のうち，CSCV の分布により方式（4）が最も適した方式であると言える．表 2 に，方式（4）の次元削除を適応した CSCV の対応表を，各最大結託者数のパラメータに応じて示す．また，適切な CSCV の分布の選択によって高精度な推定が可能であるとき，この成分は推定においてドミナントな成分であると考えられる．

表 2　方式（4）を用いた CSCV の対応表

c_{max}	$\mathbf{\Gamma}_{\boldsymbol{c}}^{\boldsymbol{str}}$	方式 (4)
6	$(\gamma_1^{str}, \gamma_2^{str}, \gamma_3^{str})$	$(\gamma_1^{str}, \gamma_2^{str}, \gamma_3^{str})$
8	$(\gamma_1^{str}, \gamma_2^{str}, \gamma_3^{str}, \gamma_4^{str})$	$(*, \gamma_2^{str}, \gamma_3^{str}, *)$
10	$(\gamma_1^{str}, \gamma_2^{str}, \gamma_3^{str}, \gamma_4^{str}, \gamma_5^{str})$	$(*, \gamma_2^{str}, \gamma_3^{str}, \gamma_4^{str}, *)$
12	$(\gamma_1^{str}, \gamma_2^{str}, \gamma_3^{str}, \gamma_4^{str}, \gamma_5^{str}, \gamma_6^{str})$	$(*, *, \gamma_3^{str}, \gamma_4^{str}, *, *)$

5. シミュレーション結果

この章では，提案した戦略推定器の推定戦略数とその推定結果を用いた結託者の検出性能の評価のために計算機によるシミュレーションを行う．シミュレーションでは結託耐性符号として Nuida 符号を用いる．また，符号長 $L = 2048$，ユーザ数 $N = 10^6$，Nuida 符号生成のための最大結託者数 $c_{max} = \{6, 8, 10, 12\}$ 人，誤検出率 $\epsilon = 10^{-10}$，結託者数は $c = \{2, 3, \ldots, 10\}$ 人を想定する．また, 結託攻撃に用いる戦略は主要戦略である majority, minority, coin-flip, all-0, all-1, interleave, WCA attack の計 7 戦略を想定する．一回の試行における結託者の組み合わせは試行ごとに毎回異なるランダムな組み合わせを用いる．シミュレーション結果は，10^3 回の試行を行った平均とする．

5.1 戦略推定の精度評価

推定器の評価では，戦略推定のみの精度を評価するために結託者数を既知の情報として与える．各戦略で $c = \{2, 3, \ldots, 8\}$ 人で結託攻撃をして不正符号語が作られた場合の推定精度を最大結託者数 c_{max} について既存手法 [12] との比較を表 4 に示す．表 4 より，最大結託者数 c_{max} の増加に依存せずほとんどの戦略において高精度に推定が行われていることがわかる．しかし，結託者が coin-flip 攻撃や WCA 攻撃を行った場合に著しく推定精度が悪い場合がある．これは，戦略パラメータ $\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{coin}}$ と $\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{WCA}}$，およ

表 3　$c=4$ における coin-flip,WCA 攻撃の各ベクトル値の比較

str	$\gamma_{4,1}$	$\gamma_{4,2}$	$\gamma_{4,3}$	$\gamma_{4,4}$
majority	0.013792	0.254839	0.745161	0.986208
coin-flip	0.125068	0.405181	0.594819	0.874932
WCA	0.122175	0.401272	0.598728	0.877825

び，CSCV $\boldsymbol{\Gamma}_{\boldsymbol{c}}^{\boldsymbol{coin}}$ と $\boldsymbol{\Gamma}_{\boldsymbol{c}}^{\boldsymbol{WCA}}$ の類似性によるものである．表 3 に $c=4$ 人のときの majority 攻撃と coin-flip, WCA 攻撃の CSCV $\boldsymbol{\Gamma}_{\boldsymbol{c}}^{\boldsymbol{str}}$ の値を比較した表を示す．表 3 より，coin-flip, WCA 攻撃では，戦略特徴ベクトルの値がほとんど同じ値となっていることが分かる．予備実験により，coin-flip, WCA 攻撃のように CSCV が類似する戦略パラメータにおいては，これらの推定ミスは検出性能にほとんど影響がないことが分かっている．

5.2　検出者数の評価

低次元推定器を用いた検出性能を従来研究と比較する．また，従来研究 [12], [13] において検証が行われていなかった，最大結託者数を変化させた場合の検出性能の比較も行う．このとき，WCA defence とは，戦略推定を行わず守備戦略を WCA 攻撃に固定し，結託者数を 2 人から 10 人の間で変化させてスコアを計算させ，そのうちの最大値を用いる方式である．性能の比較では，検出器においてユーザのスコアを計算し，そのスコアがしきい値を超えたユーザを不正者として検出する．不正者として検出されたユーザが実際の不正者と一致している場合の人数を表 6 に示す．表 6 では，結託者数 c が 2 人から 10 人における合計検出数を示している．すなわち，合計検出数は最大 $\sum_{c=2}^{10} c = 54$ となる．また，最適な検出器である MAP 検出器では結託者数と攻撃戦略を既知としており，理論上この数値が上限値である． 表 6 より，提案手法や既存手法において検出者数平均が MAP 検出器の上限値を超えている．本研究では，しきい値の導出において Rare Event Simulator[15] を用いており，そのアルゴリズム内におけるランダム要素が原因の検出誤差であると考えられる．そのため，この誤差は試行回数をさらに増やすことで解決される．提案手法を従来の推定器を除いた他の手法と比較すると，ほとんどの戦略において大幅な性能向上を図ることができた．また，従来の推定器と比較しても次元削除による総検出者数への影響はほとんど無いことが分かった．

6. むすび

本研究では，結託攻撃を行ったユーザをより多く検出することを目的として，最適な検出器に必要とされるパラメータの推定で推定空間の次元の削減を行った．また，従来研究において検証されていなかった最大結託者数の増加に伴う推定器の拡張と，その精度についても同様に検証を行った．結果として，従来研究で提案された手法は最大結託者数に依存せず一定の推定精度を得ることができ，検出においても理論値に近い性能を得ることがわかった．また，推定器のドミナントな成分に着目した提案手法では，次元を削除したにも関わらず一定の推定精度と検出性能を得ることができた．今後の課題として，より高次元での次元削除と，高精度で推定可能な次元の理論的証明を得ることが考えられる．

謝辞　本研究は JSPS 科研費 JP19K22846 の助成を受けたものである．

参考文献

[1] D. Boneh and J. Shaw, “Collusion-secure fingerprinting for digital data,” IEEE Trans. Inform. Theory, vol.44, pp.1897–1905, 1998.

[2] G. Tardos, “Optimal probabilistic fingerprint codes,” Proc. STOC 2003, pp.116–225, 2003.

[3] K. Nuida, M. Hagiwara, H. Watanabe, and H. Imai, “Optimization of Tardos’s fingerprinting codes in a viewpoint of memory amount,” Proc. IH 2007, LNCS, vol.4567, pp.279–293, Springer, Heidelberg, 2008.

[4] K. Nuida, S. Fujitu, M. Hagiwara, T. Kitagawa, H. Watanabe, K. Ogawa, and H. Imai, “An improvement of discrete Tardos fingerprinting codes,” Designs, Codes and Cryptography, vol.52, no.3, pp.339–362, 2009.

[5] B. Škorić, S. Katzenbeisser, and M. Celik, “Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes,” Designs, Codes and Cryptography, vol.46, no.2, pp.137–166. 2008.

[6] T. Furon, L. P. Preire, A. Guyader, and F. Cérou, “Estimating the minimal length of Tardos code,” Proc. IH 2009, LNCS, vol.5806, pp.176–190, Springer, Heidelberg, 2009.

[7] P. Meerwald and T. Furon, “Towards practical joint decoding of binary Tardos fingerprinting codes,” IEEE Trans. Inform. Forensics and Security, vol.7, no.4, pp.1168–1180, 2012.

[8] M. Desoubeaux, C. Herzet, W. Puech, and G. L. Guelvouit, “Enhanced blind decoding of Tardos codes with new MAP-based functions,” Proc. MMSP, pp.283–288, 2013.

[9] J. J. Oosterwijk, B. Skorikc, and J. Doumen, “A capacity-achieving simple decoder for bias-based traitor tracing schemes,” IEEE Trans. Inform. Theory, vol.61, no.7, pp.3882–3900, 2015.

[10] T. Laarhoven, “Capacities and capacity-achieving decoders for various fingerprinting games,” Proc. IH&MMSec2014, pp.123–134, 2014.

[11] M. Kuribayashi and N. Funabiki “Universal scoring function based on bias equalizer for bias-based fingerprinting codes,” IEICE Trans. Fundamentals, vol.E101-A, no.1, pp.119–128, 2018.

[12] 安井達哉, 栗林稔, 舩曳信生, “電子指紋符号における結託攻撃の戦略推定,” 信学技報, EMM2017–80, vol.117, no.476, pp. 17–22, 2018.

[13] 安井達哉, 栗林稔, 舩曳信生, 越前功, “電子指紋符号における不正者検出のための動的戦略推定,” 信学技報, EMM2018–58, vol.118, no.224, pp. 65–70, 2018.

[14] M. Kuribayashi, “Tardos’ s fingerprinting code over AWGN channel,” Prof. IH2010, LNCS, vol.6387. Springer, Heidelberg, pp.103–117, 2010.

表 4　最大結託者数の違いによる推定精度の比較

(a) $c_{max} = 6$ 提案手法

$\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}$	number c of colluders						
	2	3	4	5	6	7	8
maj	100.0	100.0	100.0	100.0	100.0	100.0	100.0
min	100.0	100.0	100.0	100.0	100.0	100.0	100.0
coin	100.0	92.8	54.0	55.7	91.7	99.8	100.0
int	100.0	98.7	100.0	100.0	100.0	100.0	100.0
all0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
all1	100.0	100.0	100.0	100.0	100.0	100.0	100.0
WCA	100.0	95.0	57.6	59.2	95.0	99.8	100.0

(b) $c_{max} = 8$ 提案手法

$\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}$	number c of colluders						
	2	3	4	5	6	7	8
maj	100.0	99.7	99.9	100.0	100.0	100.0	100.0
min	100.0	99.3	100.0	100.0	100.0	100.0	100.0
coin	100.0	70.4	32.5	96.7	99.6	99.9	99.9
int	100.0	88.5	98.8	95.3	92.7	93.8	96.4
all0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
all1	100.0	100.0	100.0	100.0	100.0	100.0	100.0
WCA	100.0	86.2	74.1	89.5	91.7	92.8	95.6

(c) $c_{max} = 10$ 提案手法

$\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}$	number c of colluders						
	2	3	4	5	6	7	8
maj	100.0	100.0	100.0	100.0	100.0	100.0	100.0
min	100.0	99.7	100.0	100.0	100.0	100.0	100.0
coin	100.0	85.4	55.7	92.8	99.2	100.0	100.0
int	100.0	94.4	99.9	99.9	99.7	99.9	99.6
all0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
all1	100.0	100.0	100.0	100.0	100.0	100.0	100.0
WCA	100.0	90.6	52.9	92.7	98.4	99.4	98.8

(d) $c_{max} = 12$ 提案手法

$\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}$	number c of colluders						
	2	3	4	5	6	7	8
maj	100.0	93.6	94.1	99.5	99.5	100.0	99.9
min	100.0	91.8	99.9	100.0	100.0	100.0	100.0
coin	100.0	50.0	54.1	87.6	95.2	97.2	95.9
int	100.0	67.5	85.2	79.0	76.5	84.0	83.8
all0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
all1	100.0	100.0	100.0	100.0	100.0	100.0	100.0
WCA	100.0	64.1	42.7	64.1	69.6	66.1	79.7

(e) $c_{max} = 6$ 従来手法 [12]

$\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}$	number c of colluders						
	2	3	4	5	6	7	8
maj	100.0	100.0	100.0	100.0	100.0	100.0	100.0
min	100.0	100.0	100.0	100.0	100.0	100.0	100.0
coin	100.0	92.8	54.0	55.7	91.7	99.8	100.0
int	100.0	98.7	100.0	100.0	100.0	100.0	100.0
all0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
all1	100.0	100.0	100.0	100.0	100.0	100.0	100.0
WCA	100.0	95.0	57.6	59.2	95.0	99.8	100.0

(f) $c_{max} = 8$ 従来手法 [12]

$\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}$	number c of colluders						
	2	3	4	5	6	7	8
maj	100.0	100.0	100.0	100.0	100.0	100.0	100.0
min	100.0	99.8	100.0	100.0	100.0	100.0	100.0
coin	100.0	90.3	55.6	96.9	99.5	100.0	100.0
int	100.0	97.2	100.0	100.0	100.0	100.0	100.0
all0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
all1	100.0	100.0	100.0	100.0	100.0	100.0	100.0
WCA	100.0	91.3	57.4	95.9	99.7	100.0	100.0

(g) $c_{max} = 10$ 従来手法 [12]

$\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}$	number c of colluders						
	2	3	4	5	6	7	8
maj	100.0	100.0	100.0	100.0	100.0	100.0	100.0
min	100.0	100.0	100.0	100.0	100.0	100.0	100.0
coin	100.0	91.4	57.0	94.1	99.1	100.0	100.0
int	100.0	98.0	100.0	100.0	100.0	100.0	100.0
all0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
all1	100.0	100.0	100.0	100.0	100.0	100.0	100.0
WCA	100.0	92.3	55.0	92.7	99.4	100.0	100.0

(h) $c_{max} = 12$ 従来手法 [12]

$\boldsymbol{\theta}_{\boldsymbol{c}}^{\boldsymbol{str}}$	number c of colluders						
	2	3	4	5	6	7	8
maj	100.0	100.0	100.0	100.0	100.0	100.0	100.0
min	100.0	99.9	100.0	100.0	100.0	100.0	100.0
coin	100.0	88.5	56.3	93.0	99.7	99.8	99.9
int	100.0	97.0	100.0	100.0	100.0	100.0	100.0
all0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
all1	100.0	100.0	100.0	100.0	100.0	100.0	100.0
WCA	100.0	92.4	57.8	93.6	99.4	99.9	100.0

表 5　各戦略の違いにおける結託者検出者数の比較

	maj	min	coin	int	all0	all1	WCA	total
MAP (optimal)	45.31	54.00	23.14	21.87	53.85	53.84	17.98	269.98
Symmetric [5]	14.67	13.31	13.87	14.33	13.88	13.92	14.01	97.99
WCA defence	18.91	23.07	20.32	18.67	20.09	20.14	17.94	139.14
Bias Equalizer [11]	45.11	53.81	19.16	21.34	52.39	52.30	16.41	260.52
Basic [12]	45.29	54.00	22.73	21.74	53.85	53.84	17.86	269.31
Subset [12]	45.21	54.00	22.96	21.86	53.85	53.86	17.92	269.66
Dynamic [13]	45.27	54.00	23.00	21.84	53.85	53.86	17.89	269.71
Proposed	45.31	54.00	22.78	21.66	53.85	53.84	17.65	269.09

[15] A. Simone and B. Škorić, "Accusation probabilities in Tardos codes: beyond the Gaussian approximation," Designs, Codes and Cryptography, vol.63, no.3, pp.379–412, 2012.

表 6　各戦略と最大結託者数の違いにおける結託者検出者数の比較

(a) $c_{max} = 6$

	maj	min	coin	int	all0	all1	WCA	total
MAP(optimal)	47.44	54.00	22.10	22.73	53.99	53.99	16.44	270.69
Basic [12]	47.44	54.00	21.35	22.09	53.99	53.99	15.87	268.73
Subset [12]	47.45	54.00	21.80	22.76	53.99	53.99	16.39	270.38
Dynamic [13]	47.45	54.00	21.86	22.71	53.99	53.99	16.19	270.19
Proposed	47.44	54.00	21.35	22.09	53.99	53.99	15.87	268.73

(b) $c_{max} = 8$

	maj	min	coin	int	all0	all1	WCA	total
MAP(optimal)	45.31	54.00	23.14	21.87	53.85	53.84	17.97	269.98
Basic [12]	45.29	54.00	22.73	21.74	53.85	53.84	17.86	269.31
Subset [12]	45.21	54.00	22.96	21.86	53.85	53.86	17.92	269.66
Dynamic [13]	45.27	54.00	23.00	21.84	53.85	53.86	17.89	269.71
Proposed	45.31	54.00	22.78	21.66	53.85	53.84	17.65	269.09

(c) $c_{max} = 10$

	maj	min	coin	int	all0	all1	WCA	total
MAP(optimal)	44.15	54.00	22.11	21.21	53.57	53.54	18.24	266.82
Basic [12]	44.13	54.00	21.57	21.09	53.57	53.54	18.00	265.90
Subset [12]	44.03	54.00	21.91	21.18	53.57	53.51	18.18	266.38
Dynamic [12]	44.09	54.00	21.91	21.13	53.57	53.54	18.10	266.34
Proposed	44.13	54.00	21.28	21.06	53.57	53.53	17.70	265.27

(d) $c_{max} = 12$

	maj	min	coin	int	all0	all1	WCA	total
MAP(optimal)	43.65	54.00	21.53	20.84	53.30	53.23	17.84	264.39
Basic [12]	43.62	54.00	20.98	20.69	53.31	53.23	17.65	263.48
Subset [12]	43.57	54.00	21.30	20.90	53.28	53.27	17.84	264.16
Dynamic [13]	43.61	54.00	21.14	20.82	53.28	53.26	17.71	263.82
Proposed	43.61	54.00	20.60	20.17	53.30	53.22	16.88	261.78