3K4-OS-20b-4

クラウドソーシングにおけるプライバシ保護タスク割り当て

梶野 洸[*1] 荒井 ひろみ[*2] 佐久間 淳[*3] 鹿島 久嗣[*4]
Hiroshi Kajino Hiromi Arai Jun Sakuma Hisashi Kashima

[*1]東京大学大学院情報理工学系研究科数理情報学専攻
Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo

[*2]東京大学情報基盤センター
Information Technology Center, The University of Tokyo

[*3]筑波大学大学院システム情報工学研究科
Graduate School of Systems and Information Engineering, Tsukuba University

[*4]京都大学大学院情報学研究科知能情報学専攻
Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University

Task assignment in crowdsourcing is an important technique to improve the task completion rate, especially when crowdsourcing complex tasks that require workers to have special properties. For example, an English-French translation task cannot be completed by a worker who does not understand both English and French, and a spatial crowdsourcing task that asks workers to collect information at a specific location should be assigned to a worker who is close to the location. However, we notice that task assignment based on skills and properties can invade both workers' and requesters' privacy. In this paper, we present a privacy-preserving task assignment protocol to address the privacy issue. Our protocol allows a crowdsourcing platform to learn an optimal task assignment without leaking any other information to all the parties.

1. 序論

クラウドソーシングはウェブ経由で柔軟な雇用を実現する仕組みである．仕事（タスク）を依頼する依頼者と仕事を処理するワーカーからなる．既存のシステムの多くでは誰でも処理できるような単純なタスクを取り扱っているが，処理に特定の特性を要求するような高度なタスクを依頼する試みに注目が集まっている．例えば英仏翻訳タスクではワーカーは英語と仏語両方の能力が求められる．また災害後に特定地点で破損箇所の有無を確認するタスクを考えると，ワーカーはその地点に近い位置にいる必要がある．このように多くの実用的なタスクは高度なタスクに分類されるため，高度なタスクのクラウドソーシングを実現する研究がますます重要視されてきている．

高度なタスクを依頼する際には，タスクの処理量を向上させることが重要となる．特に，従来のようにワーカーがタスクを選択する方式では処理量が最大化されないため，ワーカーやタスクの特性を考慮してクラウドソーシング運営会社が最適なタスク割り当てを計算してタスクの処理量を最大化する手法が重要となる．本論文の問題意識は，このようなタスク割り当ての最適化を行う際にプライバシ問題が生じることである．ワーカーは自身の能力だけではなく，所在地，希望賃金，希望勤務時間などの属性情報，場合によってはさらに個人的な情報をを運営会社へ伝える必要がある．このような情報を用いると，ワーカー個人を特定したり個人情報を推定するだけではなく，位置情報を用いて現実に危険が生じる可能性がある．同様に依頼者は自身のタスクの必要とする能力や属性情報を伝える必要があり，そのような情報から依頼者が特定されたり，タスクの内容が推定される危険性がある．

本論文ではプライバシを保護しつつタスク割り当てを計算するプロトコルを提案する．提案プロトコルでは，各ワーカー各依頼者が自身の特性を秘密に持ったまま計算を進め，最終的に運営者が最適なタスク割り当てのみを得る．タスク割り当てが最大流問題で定式化できるため，提案プロトコルではまず Paillier 暗号 [Paillier 99] を用いて最大流問題のインスタンスを構築し，次に Paillier 暗号で秘匿化したプッシュリラベル法 [Goldberg 88] を適用することで，プライバシを侵害せずに最適なタスク割り当てを計算する．クラウドソーシングを用いて計算分担者を集めるという手法を用いることで，クラウドソーシングの文脈で実用的なプロトコルを構成する．

連絡先: 梶野 洸，hiroshi_kajino@mist.i.u-tokyo.ac.jp

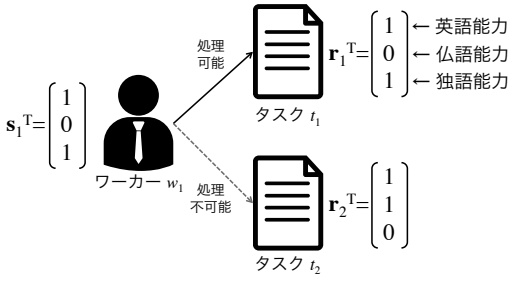

図 1: 処理可能なタスクと処理不可能なタスクの例．ワーカー 1 は英語および独語の能力を持つため，タスク 1 は処理できるが仏語能力を要求するタスク 2 は処理できない．

2. プライバシ保護タスク割り当て問題

本章ではクラウドソーシングのモデルを定義し，プライバシ保護タスク割当問題を定義する．

2.1 クラウドソーシングモデル

本モデルは，依頼者，ワーカー，運営会社の三種類からなる．

依頼者は，仕事の指示書と複数のインスタンス（処理対象物）からなるタスクをクラウドソーシングに依頼する．簡単のため，各依頼者は一種類のタスクのみを依頼するとし，以降依頼者とタ

スクを同一視して扱う．各依頼者のタスクを t_i とし，全依頼者によるタスクの集合を $\mathcal{T} = \{t_0, \ldots, t_{I-1}\}$ $(I \geq 1)$ とする．図 1 では，タスク 1, 2 はそれぞれ英独・英仏翻訳で，インスタンスは英文となる．さらに各タスク t_i には二つのパラメタ $(\mathbf{r}_i, M_{t_i})$ が付いているとする．要件ベクトル $\mathbf{r}_i \in \{0,1\}^D$ $(D \geq 1)$ は，タスク t_i を処理するのに必要な特性を表す．タスク t_i の処理に特性 d が必要ならば $r_{i,d} = 1$，必要でないならば $r_{i,d} = 0$ と定義する．図 1 では，$d = 1, 2, 3$ はそれぞれ英語・仏語・独語能力を表している．容量 M_{t_i} はタスク t_i のインスタンス数を表す．各依頼者は自分のタスクのパラメタを秘匿する．

ワーカーはクラウドソーシングでタスクを処理して報酬を得る．各ワーカーを w_j と表し，ワーカーの集合を $\mathcal{W} = \{w_0, \ldots, w_{J-1}\}$ $(J \geq 1)$ とする．各ワーカー w_j には二つのパラメタ $(\mathbf{s}_j, M_{w_j})$ が付いているとする．特性ベクトル $\mathbf{s}_j \in \{0,1\}^D$ は，ワーカー w_j が持つ特性を表す．ワーカー w_j が特性 d を持つならば $s_{j,d} = 1$，持たないならば $s_{i,d} = 0$ と定義する．図 1 では，ワーカー 1 は英語・独語が堪能であるが，仏語ができない．ここで，特性ベクトルと要件ベクトルの各次元の意味は共通であるとする．容量 M_{w_j} はワーカー w_j の処理可能なインスタンス数の最大値を表す．各ワーカーは自分のパラメタを秘匿する．

タスクが処理可能かどうかはタスクのパラメタと，タスクが割り当てられたワーカーのパラメタによって決まる．処理可能なタスクの割り当てを定義 1 に示す．

定義 1. $\mathbb{Z}_K := \{0, \ldots, K-1\}$ とする．大きさ K のタスク割り当てを多重集合 $\mathcal{A} := \{(w_{j_k}, t_{i_k}) \mid k \in \mathbb{Z}_K, i_k \in \mathbb{Z}_I, j_k \in \mathbb{Z}_J\}$ で表す．$\mathcal{A}$ が以下を満たすとき，これを処理可能なタスク割り当てと定義する*:*

1. 各 $i \in \mathbb{Z}_I$ に対し $|\{k \mid i_k = i\}| \leq M_{t_i}$,
2. 各 $j \in \mathbb{Z}_J$ に対し $|\{k \mid j_k = j\}| \leq M_{w_j}$,
3. 各 $k \in \mathbb{Z}_K$ に対し $\mathbf{s}_{j_k} \geq \mathbf{r}_{i_k}$.

運営会社はクラウドソーシングサービスを運営する．本モデルでは通信のハブとしての機能も果たす．どの二人も運営会社を経由しなければ通信ができないと仮定する．また通信路は暗号化することができ，運営会社は通信の中身を見ることができないと仮定する．

2.2 問題設定

プライバシ保護タスク割り当て問題は，依頼者やワーカーのプライバシを侵害せずに，大きさ最大の処理可能タスク割り当てを求める問題である．本論文の目的はこの問題を解決するプロトコルを提案することである．ここで単にタスク割り当て問題と書いた場合はプライバシ保護の要請がない問題を指す．

2.3 プライバシに関する仮定

すべての人は semi-honest モデルに従い，結託をしないと仮定する．つまりプロトコルに従い各々が持つ情報は共有しないが，自分がプロトコル実行中に得た情報を用いて他の人の秘密情報を推測する．プロトコルの参加者がプロトコルを守らなかった場合，得られるタスク割り当てが処理可能でない可能性があり参加者自体も不利益を被るため，semi-honest モデルの仮定は正当である．また結託をしない仮定は，運営会社がなりすましがないことを保証した元で成り立つと考えられる．参加者間の通信は運営会社によって制限されており直接の通信路も用意されていないため，参加者同士は結託しないと仮定できる．さらに，運営会社は結託が失敗した場合のリスクが大きいため，他の参加者と結託しないと仮定できる．

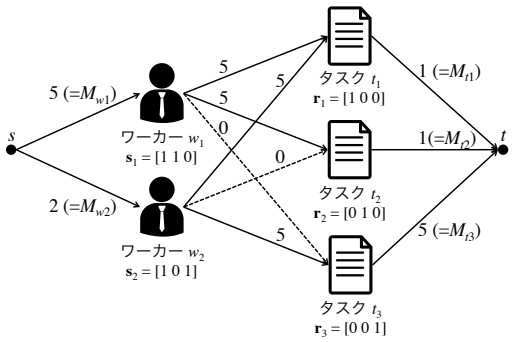

図 2: 最大流問題を用いたタスク割り当て問題の定式化.

3. 定式化

タスク割り当て問題を最大流問題を用いて定式化する．

3.1 最大流問題

V を始点 s 終点 t を含む頂点集合，E を頂点間の有向枝の集合，C を枝容量の集合とする．$N = (V, E, C)$ をネットワークとよぶ．枝容量 $c_{u,v} \in \mathbb{Z}_+$ は，$u \in V$ から $v \in V$ へ流せる流量の上限である．$(u, v) \notin E$ の場合は $c_{u,v} = 0$ と定義する．ネットワーク上のフローは定義 2 のように与えられる．本論文では整数枝容量のみを取り扱うため整数フローのみを考える．

定義 2. ネットワーク N 上のフロー F は，以下の条件を満たす整数集合 $\{f_{u,v} \in \mathbb{Z}_+\}_{(u,v) \in V \times V}$ である*:*

1. $f_{u,v} \leq c_{u,v}$, $\forall (u, v) \in V \times V$,
2. $f_{u,v} = -f_{v,u}$, $\forall (u, v) \in V \times V$,
3. $\sum_{u:(u,v) \in E} f_{u,v} = \sum_{u:(v,u) \in E} f_{v,u}$, $\forall v \in V \backslash \{s, t\}$.

フロー F の流量を $|F| = \sum_{v:(s,v) \in E} f_{s,v}$ と定義する．

最大流問題はネットワーク N が与えられた元で流量最大のフローを求める問題である．

3.2 最大流問題を用いた定式化

割り当てネットワーク $N_a = (V_a, E_a, C_a)$ を構成し，その上で最大流問題を解くことでタスク割り当て問題を解く．割り当てネットワークの構成法を提案する．頂点集合を $V_a := \{s, t\} \cup \mathcal{T} \cup \mathcal{W}$，枝集合を $E_a := \{(s, w_j) \mid w_j \in \mathcal{W}\} \cup \{(w_j, t_i) \mid w_j \in \mathcal{W},\ t_i \in \mathcal{T}\} \cup \{(t_i, t) \mid t_i \in \mathcal{T}\}$ と定義する．枝 (s, w_j) $(\forall w_j \in \mathcal{W})$ の枝容量を $c_{s,w_j} := M_{w_j}$，枝 (t_i, t) $(\forall t_i \in \mathcal{T})$ の枝容量を $c_{t_i,t} := M_{t_i}$，枝 (w_j, t_i) $(\forall w_j \in \mathcal{W}, \forall t_i \in \mathcal{T})$ の枝容量を

$$c_{w_j,t_i} := \begin{cases} M_\top & (\mathbf{s}_j \geq \mathbf{r}_i), \\ 0 & (\mathbf{s}_j < \mathbf{r}_i), \end{cases}$$

と定義する．ここで $M_\top = \max_{k \in \mathcal{T} \cup \mathcal{W}} M_k$ とする．例を図 2 に示す．割り当てネットワーク上の最大フロー $F^\star$ が得られたとき，ワーカー w_j にタスク t_i のインスタンスを $f^\star_{w_j,t_i}$ 個割り当てることで大きさ最大の処理可能タスク割り当てが得られる．

4. 暗号学的な準備

プロトコルを構成するために必要な暗号学的な準備を行う．まず提案手法で用いる暗号系の導入をし，プロトコルで用いるデータ構造を定義する．最後にプロトコル実装に必要な暗号学的な関数を提案する．

4.1 暗号系

提案プロトコルでは暗号系として Paillier 暗号 [Paillier 99] を用いる．参加者に公開されている公開鍵を用いて暗号化が可能な一方で，秘密鍵を持つ復号者のみが暗号文を復号可能である．平文を $m \in \mathbb{Z}_N$，対応する暗号文を $c = \mathsf{E}(m;r) \in \mathbb{Z}^*_{N^2}$ $(r \in \mathbb{Z}^*_N)$[*1]とおく．r を一様乱数に選ぶことで，平文の内容に依存せず暗号文も一様乱数となる．また秘密鍵を用いずに暗号文を解読することは計算量的に難しいと予想されている．ゆえに復号者以外の参加者は暗号文から何も情報を引き出すことはできない．さらに Paillier 暗号は公開鍵暗号の性質に加え準同型性を持つため，暗号文を復号せずに平文の加算を実行できる．つまり二つの暗号文 $\mathsf{E}(m_1;r_1), \mathsf{E}(m_2;r_2)$ が与えられた時，$\mathsf{E}(m_1+m_2;r)$ を $\mathsf{E}(m_1+m_2;r) = \mathsf{E}(m_1;r_1)\cdot\mathsf{E}(m_2;r_2)$ のように計算できる．ここで r_1, r_2 いずれかが一様乱数であれば r も一様乱数である．平文空間を $N/2$ シフトすることで引き算も実行可能となる．以降，平文 m の暗号文では確率変数 r を省略して $\mathsf{E}(m)$ と書く．

4.2 データ構造

以下に定義する暗号化ネットワーク (定義 3) を用いてネットワークとフローの情報を保存する．ネットワークの構造（枝の有無）を枝容量を用いて表すことで，枝容量だけでなくネットワークの構造を秘匿する．頂点数は公開情報とする．

定義 3. 暗号化ネットワークを $\boldsymbol{E}(N) := (\boldsymbol{E}(\mathbf{C}), \boldsymbol{E}(\mathbf{F}))$ と定義する．ここで $\boldsymbol{E}(\mathbf{C}) := \left[\boldsymbol{E}(c_{u,v})\right]_{u,v\in V}$ と $\boldsymbol{E}(\mathbf{F}) := \left[\boldsymbol{E}(f_{u,v})\right]_{u,v\in V}$ は，各頂点対の枝容量とフロー値を暗号化し行列化したものである．

4.3 再暗号化集合順序交替

暗号文からなる順序付き集合が与えられたとき，再暗号化集合順序交替 (re-encryption mix network) [Neff 01] は，各暗号文を再暗号化し，それらの順番をランダムに入れ替えたものを返す．このプロトコルの参加者は，計算者と補助者であり，いずれも秘密鍵は所持していないとする．また計算者は暗号文からなる順序付き集合を所持している．はじめに，計算者は暗号文の集合を補助者に送る．補助者は各暗号文を再暗号化[*2]し，集合の順序をランダムに入れ替えて計算者へ送り返す．その結果，計算者は元の集合の各要素と新しい集合の各要素との対応付けができなくなる．

4.4 プライバシ保護条件分岐

提案プロトコルの重要なサブプロトコルとして，プライバシ保護条件分岐を導入する．二つの暗号文 $\mathsf{E}(m_1), \mathsf{E}(m_2)$ と暗号化された条件値 $\mathsf{E}(c)$ を受け取って，条件値に応じて二つの暗号文のうちの一つを次のように出力する．

$$\mathsf{COND}(\mathsf{E}(m_1), \mathsf{E}(m_2), \mathsf{E}(c)) = \begin{cases} \mathsf{E}(m_1) & (c > 0), \\ \mathsf{E}(m_2) & (c \leq 0). \end{cases}$$

参加者は計算者・復号者・補助者の三人で，計算者は上記の暗号文をすべて所持する．どの参加者もこのプロトコル実行後には何も情報を得られない．このプロトコルは，再暗号化集合順序交替とプライバシ保護不等式評価 [Golle 06] を用いて実装できる．またプライバシ保護条件分岐を用いることで，最小値・最大値プロトコル $\mathsf{MIN}, \mathsf{MAX}$ を構成できる．それぞれ複数の暗号文を入力とし，平文が最小・最大の暗号文を出力する．

*1 ここで p と q を十分大きな素数とし，$N = pq$ とおいた．また $\mathbb{Z}^*_{N^2} := \{z \in \mathbb{Z}_{N^2} \mid \gcd(z, N^2) = 1\}$ と定義する．

*2 $\mathsf{E}(m)$ は $\mathsf{E}(m;r)\cdot\mathsf{E}(0;r')$ を計算することで再暗号化できる

5. プライバシ保護タスク割り当てプロトコル

プライバシ保護タスク割り当て問題を解く手法として，プライバシ保護タスク割り当てプロトコルを提案する．提案プロトコルは，初期化，割り当てネットワーク構成，プライバシ保護プッシュリラベル法，後処理の 4 つの部分から構成される．提案プロトコル実行後，運営会社がタスク割り当てのみを得て，他の参加者は何も情報を獲得しないため，ワーカー・依頼者のプライバシは保護される．

提案プロトコルの一番の特長は，クラウドソーシングの文脈で実用的な点である．大多数のワーカーや依頼者がプロトコルに長時間参加することは難しいため，復号者，計算者，補助者の三人のみが主要なプロトコルを実行するように工夫をした．これにより三人以外は計算中にオンラインでいる必要がなくなるという利点がある．計算を担当する三人を決める手法を提供することで，実際に応用しやすいプロトコルとなるような工夫を行った．提案手法では，運営者が復号者を担当し，残りの役割に関してはクラウドソーシングを用いて分担を決定する．

5.1 初期化

運営会社は秘密鍵と公開鍵の組を生成し，復号を担当する．さらにクラウドソーシングを用いて他に計算を担当する二人を確保する．計算者はプロトコル中の計算を主に実行し，補助者はプライバシ保護条件分岐を実行する際に計算者を補助する．計算者および補助者を確保したのち，各ワーカー・依頼者は公開鍵を用いて自身のパラメタを暗号化し，計算者は割り当てネットワークを $\left(\left[\mathsf{E}(0)\right]_{u,v\in V}, \left[\mathsf{E}(0)\right]_{u,v\in V}\right)$ と初期化する．

5.2 タスク割り当てネットワークの構築

次に割り当てネットワークを構築する．まず始点 s からワーカー w_j への枝を張る．各ワーカー w_j が $\mathsf{E}(M_{w_j})$ を計算者へ送り，計算者が $\mathsf{E}(c_{s,w_j}) \leftarrow \mathsf{E}(M_{w_j})$ と設定すればよい．次にタスク t_i から終点 t への枝を張る．タスク t_i をもつ各依頼者が $\mathsf{E}(M_{t_i})$ を計算者へ送り，計算者が $\mathsf{E}(c_{t_i,t}) \leftarrow \mathsf{E}(M_{t_i})$ と設定すればよい．最後にワーカー w_j からタスク t_i へ枝を張る．このとき，タスク t_i が要求している特性のうちワーカー w_j が持たないものの数が $\|\mathbf{r}_i\|_2^2 - \mathbf{s}_j\cdot\mathbf{r}_i$ と書けることを利用する．まず参加者全員で MAX を用いて $\mathsf{E}(M_\top)$ を計算する．次に各ワーカーは $\mathsf{E}(\mathbf{s}_j)$ をすべての依頼者へ送る．各依頼者は $\{\mathsf{E}(\mathbf{s}_j)\}_{j\in\mathbb{Z}_J}$ を受け取り，各 $j \in \mathbb{Z}_J$ に対して $\mathsf{E}(\|\mathbf{r}_i\|_2^2 - \mathbf{s}_j\cdot\mathbf{r}_i) = \prod_{d=1}^D \mathsf{E}(r_{i,d}^2)\cdot\prod_{d=1}^D \mathsf{E}(s_{i,d})^{-r_{i,d}}$ を計算して，結果を計算者へ送る．最後に計算者は，運営会社と補助者とともに $\mathsf{E}(c_{w_j,t_i}) \leftarrow \mathsf{COND}(\mathsf{E}(0), \mathsf{E}(M_\top), \mathsf{E}(\|\mathbf{r}_i\|_2^2 - \mathbf{s}_j\cdot\mathbf{r}_i))$ を計算する．

5.3 プライバシ保護プッシュリラベル法

プライバシ保護プッシュリラベル法は，プッシュリラベル法 [Goldberg 88] を秘匿化して行うことで，割り当てネットワーク上の最大フローを計算するプロトコルである．

5.3.1 プッシュリラベル法の準備

プッシュリラベル法では，プリフローと呼ばれる緩和されたフローを更新していく．更新の適用の可否は各頂点に定められている高さに依る．この二つの変数を紹介する．

プリフローは，ある頂点への流入量が流出量を超えることを許すフローである．頂点 v の超過量を $e_v := \sum_{u\in V\setminus\{v\}} f_{u,v}$ と定義し，超過量の集合を $\mathbf{e} \in \mathbb{R}^{|V|}$ と定義する．頂点 $v \in V$ の高さ h_v は，非負整数のラベルであり，高さの集合を $\mathbf{h} \in \mathbb{Z}_+^{|V|}$ とおく．高さの集合は次の 3 つの条件を満たす必要がある: (i)

プロトコル 1 $\textsc{Push}(v, \mathsf{E}(N), \mathsf{E}(\mathbf{e}), \mathsf{E}(\mathbf{h}))$

参加者: 計算者，運営者，補助者．
秘密入力:

- **計算者:** $v \in V$，$\mathsf{E}(N)$，$\mathsf{E}(\mathbf{e})$，$\mathsf{E}(\mathbf{h})$．
- **運営者:** 秘密鍵．

1: **for** each $w \in V \backslash v$ **do**
2: 参加者全員で $\mathsf{E}(f')$ を次のように計算する:

$$\mathsf{COND}(\mathsf{E}(c_{v,w}), \mathsf{E}(f_{v,w} + e_v), \mathsf{E}(f_{v,w} + e_v - c_{v,w})).$$

3: 参加者全員で以下の計算を行う:

$$\mathsf{E}(f'') \leftarrow \mathsf{COND}(\mathsf{E}(f_{v,w}), \mathsf{E}(f'), \mathsf{E}(h_v - h_w - 1)),$$
$$\mathsf{E}(f_{v,w}) \leftarrow \mathsf{COND}(\mathsf{E}(f_{v,w}), \mathsf{E}(f''), \mathsf{E}(h_w - h_v + 1)).$$

4: 計算者は次のように内部変数を更新する:

$$\mathsf{E}(e_v) \leftarrow \mathsf{E}(e_v - f_{v,w} + f'),$$
$$\mathsf{E}(e_w) \leftarrow \mathsf{E}(e_w + f_{v,w} - f'),$$
$$\mathsf{E}(f_{w,v}) \leftarrow \mathsf{E}(-f_{v,w}).$$

5: **end for**

$c_{u,v} - f_{u,v} > 0$ となるすべての $(u, v) \in V \times V$ に対して $h_u \leq h_v + 1$，(ii) $h_s = |V|$，(iii) $h_t = 0$.

5.3.2 主プロトコル

主プロトコルは，後述するプッシュ操作とリラベル操作を繰り返す．プロトコルの参加者は，運営者，計算者，補助者の三人である．はじめに，計算者は頂点集合 V の順序を任意に定め，以下のように内部変数を初期化する: 各 $v \in V$ に対して，$\mathsf{E}(f_{s,v}) \leftarrow \mathsf{E}(c_{s,v}), \mathsf{E}(e_v) \leftarrow \mathsf{E}(c_{s,v})$ とし，各 $v \in V \backslash \{s\}$ に対して，$\mathsf{E}(h_s) \leftarrow \mathsf{E}(|V|), \mathsf{E}(h_v) \leftarrow \mathsf{E}(0)$ とする．次に，参加者全員で，頂点集合 V の順序に従って各頂点 v に対してリラベル操作，プッシュ操作を順次適用する．頂点集合のリストを $(2|V| - 1)(|V| - 2) + 2|V||E| + 4|V|^2|E|$ 回走査したのちプロトコルを終了する．

5.3.3 プッシュ操作・リラベル操作

主プロトコルで用いたプッシュ操作，リラベル操作を紹介する．ここで提案するプッシュ操作，リラベル操作は，プライバシを保護したまま実行ができるという点で Goldberg らのアルゴリズムと異なる．暗号化されたネットワークに対して操作が適用可能であり，さらに操作の適用過程から情報は一切漏洩しない．プッシュ操作（プロトコル 1）は，頂点 $v \in V$ の超過量 e_v を，条件を満たす隣接する頂点へ分配する操作である．リラベル操作（プロトコル 2）は，条件を満たした時に頂点 $v \in V$ の高さを更新する操作である．それぞれの操作で適用条件が成立しない場合は空の操作を行うことで，適用条件の成立に関する情報を漏洩しないようにしている．

5.4 後処理

最後に，計算者は $\mathsf{E}(\mathbf{F})$ を運営会社へ送り，運営会社はタスク割り当てを抽出する．各枝 (w_j, t_i) に対して $\mathsf{E}(f_{w_j,t_i})$ を復号して f_{w_j,t_i} を得て，タスク t_i のインスタンスのうち f_{w_j,t_i} 個をワーカー w_j へ割り当てる．この処理の実行後，運営会社はタスク割り当てのみを得て，他の参加者は何も情報を得ない．

プロトコル 2 $\textsc{Relabel}(v, \mathsf{E}(N), \mathsf{E}(\mathbf{e}), \mathsf{E}(\mathbf{h}))$

参加者: 計算者，運営者，補助者．
秘密入力:

- **計算者:** $v \in V$，$\mathsf{E}(N)$，$\mathsf{E}(\mathbf{e})$，$\mathsf{E}(\mathbf{h})$．
- **運営者:** 秘密鍵．

1: 計算者が $S_v \leftarrow \emptyset$ と初期化する．
2: **for** each $w \in V \backslash v$ **do**
3: 参加者全員で $\mathsf{E}(h')$ を次のように計算する:

$$\mathsf{COND}(\mathsf{E}(h_w + 1), \mathsf{E}(h_w + 2|V| + 1), \mathsf{E}(c_{v,w} - f_{v,w})).$$

4: 計算者は $S_v \leftarrow S_v \cup \{\mathsf{E}(h')\}$ と更新する．
5: **end for**
6: 参加者全員で $\mathsf{E}(h_v^\star) \leftarrow \mathsf{MIN}(S_v)$ を計算する．
7: 参加者全員で $\mathsf{E}(h_v)$ を次のように計算する:

$$\mathsf{E}(h_v^{\star\star}) \leftarrow \mathsf{COND}(\mathsf{E}(h_v^\star), \mathsf{E}(h_v), \mathsf{E}(e_v)),$$
$$\mathsf{E}(h_v) \leftarrow \mathsf{COND}(\mathsf{E}(h_v), \mathsf{E}(h_v^{\star\star}), \mathsf{E}(h_v - h_v^{\star\star} + 1)).$$

6. 結論

本論文では，クラウドソーシングにおけるタスク割り当て時に，ワーカーと依頼者のプライバシが侵害される可能性を指摘した．タスク割り当てのためにはワーカーやタスクの特性を公開する必要があり，そこからプライバシ侵害に繋がる可能性がある．タスク割り当て問題が最大流問題に帰着できることに注目し，本論文では，プライバシを保護しつつ最大流問題のインスタンスを構成し最大流問題を解くプロトコルを提案した．計算分担者をクラウドソーシングで決めることにより，実用的なプロトコルを達成したことが主な貢献となっている．

参考文献

[Goldberg 88] Goldberg, A. V. and Tarjan, R. E.: A new approach to the maximum-flow problem, *Journal of the ACM*, Vol. 35, No. 4, pp. 921–940 (1988)

[Golle 06] Golle, P.: A private stable matching algorithm, in *Proceeding of Financial Cryptography and Data Security*, pp. 65–80 (2006)

[Neff 01] Neff, C. A.: A verifiable secret shuffle and its application to e-voting, in *Proceedings of the 8th ACM conference on Computer and Communications Security (CCS '01)*, pp. 116–125 (2001)

[Paillier 99] Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes, Advances in Cryptology – EUROCRYPT '99, pp. 223–238 (1999)