秘密のリンク構造を持つグラフのリンク解析

Link Analysis for Privately Weighted Graphs

佐久間 淳[*1] 小林 重信[*2]
Jun Sakuma Shigenobu Kobayashi

[*1]筑波大学 システム情報工学研究科
Graduate School of Systems and Information Engineering

[*2]東京工業大学 総合理工学研究科
Graduate School of Intediciplinary Science and Engineering

Link analysis methods have been used successfully for learning valuable information from the link structure of mutually linking/citing entities. Existing link analysis methods have been inherently designed based on the fact that the entire link structure of the target graph is observable such as public web documents; however, link information in graphs in the real world, such as human relationship or economic activities, is rarely open to public. If link analysis can be performed using graphs with private links in a privacy-preserving way, it enables us to rank entities connected with private ties, such as people, organizations, or business transactions. In this paper, we present a secure link analysis for graphs with private links by means of cryptographic protocols.

1. はじめに

リンク解析とは互いにリンクされたエンティティのリンク構造から有用情報の抽出を目指すアルゴリズムである. 特に PageRank [4] はハイパーリンクを持つ web 文書のランキングに実際に利用され, 有用性が知られる. リンク解析は web 文書のみならず相互引用関係を持つ学術論文や蛋白質間相互作用などの解析にも利用される. 既存のリンク解析はリンク構造全体が解析者に visible であることを前提とする. しかし人間関係や経済活動など, 現実のネットワークが持つリンク構造が第三者に対して公であることは稀である. 本稿では, このような秘密のリンク関係を持つエンティティが構成するグラフに対する安全なリンク解析法を提案する.

重みつき有向グラフ $G = (V, E, W)$ を考える. V は頂点集合, E は枝集合, W は重み行列である. 本稿では, 個々の頂点として, 計算能力が多項式オーダーに制限された独立に動作する計算ノード (PC やモバイル機器など) を想定する. 枝は計算ノード間のリンク, 重みはそのリンクに与えられた量である. またリンクは必ずしも物理的な通信関係に制約されるものではない. 例えば"i さんは j さんに興味がある"といったような物理的通信を伴わない関係を含む. 一方任意の 2 ノード間の通信は, internet routing が実現するように, G で定義されるリンク関係とは無関係に可能であるとする.

本稿ではノード i からノード j に張られたリンクを $e_{ij} \in E$, その重みを w_{ij} としたとき, e_{ij} の存在および w_{ij} の値は, i, j 以外のノードには知られたくない秘密情報であると想定する. このような例として, 企業間の取引関係がある. 企業 i は企業 j から何かの製品を購入しているおり, 取引がリンク e_{ij}, 取引額が重み w_{ij} に相当する. 取引および取引額は秘密情報であり, この場合, i および j は両者を張るリンクの存在およびその重みを共有するが, i, j 以外の企業に取引の存在と取引額を秘密にしている. 類似の性質をもつグラフの例として, 電話やメールによる通信関係 (通信がリンク, 通信頻度が重み) や評価する人・評価される人を匿名化するダブルブラインドレビュー (評価関係がリンク, 評価値が重み) などがある.

連絡先: 佐久間 淳, 筑波大学大学院システム情報工学研究科 コンピュータサイエンス専攻, 茨城県つくば市天王台 1-1-1, 029-853-5542, jun@cs.tsukuba.ac.jp

このようなグラフを *privately weighted graph* (PWG) と呼ぶ。一般に, "誰が誰から買った", "誰が誰に電話 (メール) をした", "誰が誰を評価した", などのリンクや社会的紐帯は, 機密性やプライバシーの問題のため, 公にされることが好まれない場合が多い. もし秘密のリンクを含むネットワークを対象として, その秘匿性を損なうことなく安全にリンク解析を適用できれば, これまで解析対処とならなかった現実の多様なネットワークからの情報抽出を可能にする. 本稿の目的は, PWG のための安全なリンク解析アルゴリズムを提案することにある.

2. リンク解析

Notation. 頂点 (以下, ノード) 集合 $V = \{1, ..., n\}$, 枝 (以下, リンク) 集合 $E = \{e_{ij}\}$, および重み行列 $W = (w_{ij})$ からなる重み付き有効グラフ $G = (V, E, W)$ を考える. $w_{ij} = 0$ は ij 間にリンクがないことを意味する. ノード i の**度数**は $d_i = \sum_{j \in V} w_{ij}$, G の**最大度数**は $\Delta = \max_i d_i$ と定義される. $D = \mathrm{diag}(d_1, ..., d_n)$ を**度数行列**と呼ぶ. 隣接行列 $A = (a_{ij})$ は下式によって定義される:

$$a_{ij} = \begin{cases} 1 \text{ if } e_{ij} \in E \\ 0 \text{ o.w.} \end{cases} . \tag{1}$$

また, ノード i からリンクされるノード集合を $N_{out}(i) = \{j | j \in V, e_{ij} \in E\}$, ノード i にリンクするノード集合を $N_{in}(i) = \{j | j \in V, e_{ji} \in E\}$ とする.

Spectral ranking. リンク解析は, 与えられたグラフ持つリンク構造の特徴を考慮して各ノードに何らかのスコアを与えるアルゴリズムである. グラフ上のマルコフランダムウォークにおける定常分布密度によりノードをスコアリングする方法を spectral ranking と呼ぶ.

ノード i からノード j に, 確率 p_{ij} で遷移するマルコフ連鎖を考える. ただし状態遷移確率行列 $P = (p_{ij})$ を $P = D^{-1}W$ として定義する. 定常分布 $\boldsymbol{x} = (x_1, ..., x_n)^T$ は遷移後もその分布を変えないことから以下を満たす:

$$\boldsymbol{x}^T = \boldsymbol{x}^T P. \tag{2}$$

ただし $\sum_i x_i = 1$ である. この定常分布は, P の最大の固有値 (= 1) と対になる固有ベクター (principal eigenvector) に t 対応することが知られている.

PageRank. PageRank は状態遷移確率に下式を用いた spectral ranking の一種である.

$$P = (1-\epsilon)D^{-1}A + \frac{\epsilon}{n}\mathbf{1}\mathbf{1}^T \tag{3}$$

ここで, $\mathbf{1}$ は全要素が 1 であるようなベクターである. この状態遷移確率行列は, web 文書において, 確率 $1-\epsilon$ で現在の文書に含まれるハイパーリンクからランダムに選択して遷移し, 確率 ϵ で全文書集合からランダムに選択された文書に遷移するようなユーザの行動をモデル化したものである.

Power Iteration. Principal eigenvector の計算には *Power iteration* がしばしば用いられる. 初期値として $\sum_i x_i^{(0)} = 1$ なるベクターを与え, 以下の更新式を繰り返す:

$$(\boldsymbol{x}^{(t)})^T \leftarrow P(\vec{\boldsymbol{x}}^{(t-1)})^T, \quad \vec{\boldsymbol{x}}^{(t)} \leftarrow \frac{\boldsymbol{x}^{(t)}}{\|\boldsymbol{x}^{(t)}\|}. \tag{4}$$

P が確率行列の場合は, 正規化ステップは省略可能である. Power iteration の収束性は以下に示される (証明略):

補題 1 *$\boldsymbol{x}$ を P の principal eigenvector とする. $\boldsymbol{x}^{(t)}$ を power iteration によって t 回更新した後に得られたベクターとする. このとき*

$$\|\boldsymbol{x}^{(t)} - \boldsymbol{x}\| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^t\right), \tag{5}$$

が成立する. ただし, λ_1, λ_2 はそれぞれ, 最大, および二番目に大きい P の固有値である.

3. 問題の定式化と接近法

$n \times n$ 行列 $M = (m_{ij})$ の i 番目の行ベクターを $\boldsymbol{m}_{i*}$, i 番目の列ベクターを $\boldsymbol{m}_{*i}$ とする. まず n 個のノードが M をシェアするときの典型的な情報分割モデルを二つ定義する.

定義 1 *(Row private) 全ての i において, i 番目のノードが行ベクター $\boldsymbol{m}_{i*}$ を知っているが, 他の行ベクター $\boldsymbol{m}_{p*}(p \neq i)$ を知らないならば, M は* row private *である.*

定義 2 *(Symmetrically private) 全ての i において, i 番目のノードが行ベクター $\boldsymbol{m}_{i*}$ および列ベクター $\boldsymbol{m}_{*i}$ を知っているが, その他の要素 m_{pq} $(p, q \neq i)$ を知らないならば, M は* symmetrically private *である.*

この分割行列モデルに基づき, 三種のグラフにおけるプライバシーモデルを定義する.

定義 3 *(Weight-aware PWG) V が計算ノードの集合であるようなグラフ $G = (V, E, W)$ を考える. 枝集合 E に基づく隣接行列 A と重み行列 W が symmetrically private ならば, G は weight-aware privately weighted graph である.*

定義 4 *(Link-aware PWG) 定義 3 において, 隣接行列 A が symmetrically private で, 重み行列 W が row private ならば, G は link-aware privately weighted graph である.*

隣接行列 A が symmetrically private とは, ノード i は $N_{in}(i)$ および $N_{out}(i)$ を知っていることを意味する. つまり, ノード i は自分にリンクしているのは誰かを知っており, その重みも知っている. 一方, link-aware PWG では, 自分が誰からリンクされているかは知っているが, その重みは知らないことを意味する.

定義 5 *(Link-unaware PWG) 定義 3 において, 隣接行列 A と重み行列 W が row private ならば, G は link-unaware PWG である.*

接続行列 A が row private とは, ノード i は自分にリンクしているのは誰かすら知らないケースである. link-aware PWG は weight-aware PWG の一般化として考えられるため, 本稿では link-aware PWG のみ取り扱う. link-unaware PWG の扱いは本稿では省略する. 詳細は [5] を参照されたい.

続いて, PWG 上でのリンク解析を定義する. $f : \mathbb{R}^{n\times n} \mapsto \mathbb{R}^{n\times d}$ をリンク解析のためのスコアリング関数とする. f は重み行列 $W \in \mathbb{R}^{n\times n}$ を入力として, スコア行列 $X \in \mathbb{R}^{n\times d}$ を出力する. このとき secure link analysis は以下のように定義される.

定義 6 *(Secure link analysis for link-aware PWGs) $G = (V, E, W)$ を link-aware PWG とする. secure link analysis の実行後, $f(W) \to X$ は正しく評価され, X はノード間で row private になるように分配されるが, それ以外の知識は得ない.*

直感的にいえば, リンク解析が定義 6 の意味で secure であるとは, プロトコル実行後にノード i が知ることができるのはスコア行列 X の i 番の行ベクターのみであり, プロトコル実行後にも実行前と同様に, W/A は row/symmetrically private に保たれていることを意味する. 4 章では, 定義 6 において secure な spectral ranking を提案し, その PageRank への拡張も示す.

以下に提案プロトコルの概要を示す. row private な W からは row private な $P(= D^{-1}W)$ が計算可能である. 初期定常分布 $\boldsymbol{x}^{(0)}$ を row private なベクターとして与えたとき , power iteration の更新は, $j \notin N_{in}(i)$ において $p_{ji} = 0$ ゆえ, ノード j がノード i に, x_j, p_{ji} を送信することで以下のように計算できる.

$$x_i \leftarrow \sum_{j \in N_{in}(i)} x_j p_{ji}. \tag{6}$$

ここで問題となるのは, ノード j に保持される x_j と p_{ji} はノード j の秘密情報であり, ノード i には開示できない点である. これらの値をノード i に知られずに更新をするために, 準同型性公開鍵暗号を導入する. これは, 暗号化された値を解読することなしに加算することを可能にする暗号系である. x_j と p_{ji} が, ノード i には解読不可能であるが加算可能であるように暗号化することによって, eq. 6 の更新は, $j(\in N_{in}(i))$ の情報を i に見せずに行うことができる.

4. Secure Link Analysis

本章では前章に示した定義に基づく secure spectral ranking (SSR) を提案し, その PageRank への拡張を示す. はじめに, 準同型性公開鍵暗号を導入する.

4.1 準同型性公開鍵暗号

公開鍵暗号系において, 暗号化は公にされた**公開鍵** pk を, 解読にはメッセージの受信者のみが保持する公開鍵に対応した**秘密鍵** sk を用いる. 平文 m について, $c = \mathrm{Enc}_{\mathsf{pk}}(m; \ell)$ は m の確率暗号による暗号化を, $m = \mathrm{Dec}_{\mathsf{sk}}(c)$ はその解読をあらわす. ℓ が $\mathbb{Z}_N(= \{0, 1, ..., N-1\})$ 上で一様ランダムに選ばれたならば, 暗文 c も同様に $\mathbb{Z}_N$ で一様ランダムに分布する. **加法的準同型公開鍵暗号**は, 秘密鍵の知識なしに, 暗文同士の加算 (演算子"$\cdot$" を暗文の加算とする)

$$\mathrm{Enc}_{\mathsf{pk}}(m_1 + m_2; \ell) = \mathrm{Enc}_{\mathsf{pk}}(m_1; \ell_1) \cdot \mathrm{Enc}_{\mathsf{pk}}(m_2; \ell_2), \tag{7}$$

が可能である. ここで, ℓ は ℓ_1 か ℓ_2 の少なくともどちらかひとつが $\mathbb{Z}_N$ 上で一様ランダムならば, 同様に一様ランダムである.

この性質に基づき, 定数 k と暗文 $\mathrm{Enc}_{\mathsf{pk}}(m_1;\ell)$ の乗算が, $\cdot$ の繰り返し (下記のように記述する) により実現される.

$$\mathrm{Enc}_{\mathsf{pk}}(km;\ell) \quad = \quad \prod_{i=1}^{k} \mathrm{Enc}_{\mathsf{pk}}(m;\ell_k) = \mathrm{Enc}_{\mathsf{pk}}(m)^k$$

ℓ は ℓ_1,...,ℓ_k の少なくともどちらかひとつが $\mathbb{Z}_N$ 上で一様ランダムならば, 同様に一様ランダムである. 以降は, 簡単のために乱数 ℓ は表示しない.

(m,t)-**閾値暗号系**では, m ノードが共通の公開鍵 pk を保持し, 各ノードはそれぞれ異なる秘密鍵 $\mathsf{sk}^1,...,\mathsf{sk}^n$ を保持している. 各ノードは共通の公開鍵により任意のメッセージを暗号化可能である. 一方, 解読には, 少なくとも t 以上のノードのグループが協力し, 公開鍵とそれぞれのノードが持つ *decryption shares* $\mathrm{Dec}_{\mathsf{sk}^1}(c),...,\mathrm{Dec}_{\mathsf{sk}^n}(c)$ を引数にとる **recovery アルゴリズム**を実行する必要がある. 本稿で示すプロトコルは, 加法的準同型性を持つ閾値暗号系を用いる. 後の実験では, これらの性質を満たす generalized Paillier 暗号系を用いる [1].

4.2 Link-aware モデルにおける Spectral Ranking

Secure spectral ranking では, eq. 6 の計算を, 準同型性公開鍵暗号上で行う. 閾値準同型性公開鍵暗号系の鍵集合を $\mathcal{K} = \{\mathsf{pk}, \mathsf{sk}^i, ..., \mathsf{sk}^n\}$ とする. sk^i はノード i のみが保持し pk は全ノードが共有する.

まず step 1 において, 確率行列 $P = D^{-1}W$ を準備する. $p_{ij} = w_{ij}/d_i$ は W が row private であっても各ノードが独立に計算可能である. 典型的な暗号系は整数のみを引数に取るため, p_{ij} は十分に大きい定数 L に基づいて, $b_{ij}(= Lp_{ij}) \in \mathbb{Z}_N$ となるように拡大される. 同様の理由により, 初期定常分布 $\boldsymbol{q}$ も十分に大きい定数 K を用いて, $\sum_i q_i = K$ となるように初期化する.

step 3 では, power iteration を実行する. 更新において, ノード i は $\boldsymbol{q}$ の第 i 要素を担当する. 以下の補題により, プロトコルが正しく P の定常分布を求めことを示す.

補題 2 $\boldsymbol{\pi}^*$ を P によるマルコフランダムウォークの定常分布とし, *secure spectral ranking* の更新回数を t とする. このとき $\boldsymbol{\pi}^{(t)}$ は, 以下を満足する確率ベクターである.

$$\|\boldsymbol{\pi}^* - \boldsymbol{\pi}^{(t)}\| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^t\right). \tag{8}$$

ここで λ_1 および λ_2 は最大および二番目に大きい P の固有値である.

証明の概略を示す. step 3(b) の更新は, 暗号系の準同型性を考慮すると,

$$\mathrm{Enc}_{\mathsf{pk}}(q_i^{(t)}) \leftarrow \prod_{j\in N_{in}(i)} \left(\mathrm{Enc}_{\mathsf{pk}}(q_j^{(t-1)})^{b_{ji}}\right) = \mathrm{Enc}_{\mathsf{pk}}\left(\sum_{j\in N_{in}(i)} b_{ji} q_j^{(t-1)}\right). \tag{9}$$

と整理される. もしこの両辺が解読されれば,

$$q_i^{(t)} \leftarrow \sum_{j\in N_{in}(i)} \left(b_{ji} q_j^{(t-1)}\right). \tag{10}$$

を得る. この更新式が P の最大固有値に対応する固有ベクターと平行なベクターに漸近することを示すことで, 補題が証明される.

続いてこのプロトコルのセキュリティについて考察する. 全てのノードが *semi-honest* に振舞う[*1]ことを想定する . このとき, このプロトコルのセキュリティは以下のように示される.

[*1] 各ノードは定められたプロトコルを逸脱しないが, 実行途中で受け取った全ての情報から他ノードの情報を推測しようとする振る舞い

Procedure LinkAwareSecureSpectralRanking (W, K, L)

- Public input: $K \in \mathbb{Z}_N$, $L \in \mathbb{Z}_N$ s.t. $Lp_{ij} \in \mathbb{Z}_N$ for all i, j
- Private input of node i: $\boldsymbol{w}_{i*}$ (row private W)
- Key setup: All nodes jointly generate a key set $\mathcal{K} = \{\mathsf{pk}, \mathsf{sk}^i, ..., \mathsf{sk}^n\}$ in a distributed way so that pk is commonly known to all nodes and sk^i is possessed only by the ith node.

1. (Setup of P) Agent i computes:
   (a) $d_i \leftarrow \sum_{j\in N_{out}(i)} w_{ij}$
   (b) for all j, $p_{ij} \leftarrow w_{ij}/d_i$, $b_{ij} \leftarrow Lp_{ij}$
2. (Initialization) Agent i updates $q_i^{(0)} \leftarrow K_i$ s.t. $\sum_i K_i = K$ and $t \leftarrow 1$.
3. (Power iteration) Node i computes the following steps until convergence:
   (a) For all $k \in N_{out}(i)$, node i computes $c_{ik,\mathsf{pk}} \leftarrow (\mathrm{Enc}_{\mathsf{pk}}(q_i^{(t-1)}))^{b_{ik}}$ and sends $c_{ik,\mathsf{pk}}$ to node k
   (b) For all $j \in N_{in}(i)$, node i receives $c_{ji,\mathsf{pk}}$ and computes $\mathrm{Enc}_{\mathsf{pk}}(q_i^{(t)}) \leftarrow \prod_{j\in N_{in}(i)} c_{ji,\mathsf{pk}}$, $t \leftarrow t+1$
   (c) Node i and $j \in_r N(i)$ performs convergence detection protocol. If convergence is not detected, broadcast "no convergence". If no broadcast message received, jump to step 4. Else, jump to step 3(a).
4. (Decryption) Node i and $N_{out}(i)$ run the recover scheme and obtain $q_i^{(t)}$. Then, output $\pi_i^{(t)} = q_i^{(t)}/KL^{t-1}$

図 1: Secure Spectral Ranking in the Link-aware Model

補題 3 T を更新回数とする. 全てのノードが *semi-honest* に振舞うならば, *link-aware PWG* における *secure spectral rank* プロトコル実行後, 全ての i において, ノード i は $\pi_i^{(T)}$ のみを知り, そのほかの情報は何も得ない.

証明は省略する. これらの補題から, 以下の定理が導かれる.

定理 1 全てのノードが *semi-honest* に振舞うならば, *secure spectral ranking* は正しくかつ定義 *6* の意味で安全に *link-aware PWG* の定常分布を計算する.

証明は各補題から直ちに導かれる.

4.3 PageRank への拡張: PrivateRank

Secure spectral ranking の PageRank への拡張, PrivateRank を示す. PageRank ではノード i からノード j への遷移確率は

$$p_{ij} \leftarrow (1-\epsilon)\frac{a_{ij}}{d_i} + \epsilon\frac{1}{n}. \tag{11}$$

と定義される. 前節と同様に, b_{ij} は L によって $Lp_{ij} \in \mathbb{Z}_N$ となるように拡大される. ただしここでは L は, $L\epsilon/n \in \mathbb{Z}_N$ なる整

数である. この遷移確率の定義を考慮すると, 更新式は

$$q_i^{(t)} = \sum_{j=1}^{n} q_j^{(t)} b_{ji} = \sum_{j\in N_{in}(i)} q_j^{(t)} b_{ji} + \sum_{j\notin N_{in}(i)} q_j^{(t)} \frac{L\epsilon}{n}$$
$$= \sum_{j\in N_{in}(i)} q_j^{(t)} b_{ji} + \frac{L\epsilon}{n}\left(LK^{t-1} - \sum_{j\in N_{in}(i)} q_j^{(t)}\right) \quad (12)$$

と整理される. ここで変形には $\sum_{i=1}^{n} q_j^{(t)} = LK^{t-1}$ であることを利用している. 上式の両辺を暗号化することによって

$$\mathrm{Enc}_{\mathrm{pk}}(q_i^{(t)}) \leftarrow \mathrm{Enc}_{\mathrm{pk}}\left(\sum_{j\in N_{in}(i)} q_j^{(t)} b_{ji} + \frac{L\epsilon}{n}\left(LK^{t-1} - \sum_{j\in N_{in}(i)} q_j^{(t)}\right)\right)$$
$$= \left\{\prod_{j\in N_{in}(i)} \mathrm{Enc}_{\mathrm{pk}}(q_j^{(t-1)})^{b_{ji}}\right\} \cdot \left[\mathrm{Enc}_{\mathrm{pk}}(LK^{t-1}) \prod_{j\in N_{in}(i)} \left\{\mathrm{Enc}_{\mathrm{pk}}(q_j^{(t-1)})^{-1}\right\}\right]^{L\epsilon/n} \quad (13)$$

を得る. ここで eq. 13 は L, K, t, n, ϵ および $j \in N_{in}(i)$ について $\mathrm{Enc}_{\mathrm{pk}}(q_j^{t-1})$ のみを用いて構成されることに注意されたい. これらは全てノード i が取得可能な情報であり, 独立に更新可能である. ゆえに, 図 1 の step 3(b) において, eq. 13 を更新式として用いることによって PrivateRank が実現する.

5. 実験

プロトコルの有効性を検証するための実験を行った. 提案プロトコルである PriavteRank は, プライバシが考慮されない場合の PageRank と同様の結果を返すことが保障されているため, プロトコルがどのような結果を得るか, ではなく, プロトコルの計算効率性およびプロトコル実行の結果として開示される情報を検証の対象とした. 実験は実際のネットワーク上ではなく単一の計算機上でシミュレートされたため, 通信時間は含まれていないことに注意されたい.

PrivateRank (PR) の実験結果は, 任意の分散計算を安全に実行可能な secure function evaluation (SFE) [3], decentralized spectral analysis (DSA) [2] と比較した. これらの方法の更新毎の時間計算量は最大度数 Δ にのみ依存するため, 図 2 では Δ に対する更新毎の計算時間の変化を示した. 実験は最大度数が Δ になるようにランダムに生成されたグラフを用いて行われた. 表 1 は計算時間と開示される情報のサマリーである.

DSA は本来 peer-to-peer ネットワークのために設計された spectral analysis である. 計算は各ノードに等しく分散化されており, 更新毎の時間計算量は $O(\Delta)$ である. DSA は暗号化処理を含まないため, 計算時間はきわめて小さい. また DSA はメッセージ交換において他ノードに重みを直接明かすことはないため, プライバシ保護がなされているように見えるが, 交換されたメッセージから重みが推測されるケースがありえるため, 理論的/統計的なプライバシ保護の保障はなされない. SFE は本来定義 6 に示されるような完全なプライバシ保護が達成可能である. SFE の計算時間は入力サイズに対して多項式で抑えられるが, 厳密に実装した場合, 計算時間は非現実的なほど多く実験は困難である (SFE strict). そこで, 単一の更新のみを SFE で行い, その結果を各ノードに返すような設計で SFE を構成した (SFE relaxed). この設計はノードに中間情報を漏らす可能性があるものの, 計算時間は大幅に短縮され, 厳密に実装した場合 (SFE strict) の計算時間の下限として考えることができる. 図 2 にあるように, SFE の計算時間は, セキュリティを緩和したバージョンであってさえもその他の方法に比べてかなり長い. PrivateRank は, 全ノードで共通の鍵集合を利用する設定と, 鍵集合の有効範囲を局所化し結託耐性を考慮した設定 (collusion-resistant, 詳細は [5] 参照) の二つを比較した. PrivateRank は DSA の暗号学的拡張と解釈できる. 計算は全ノードに等価に分散可能であり, ステップ毎の時間計算量は $O(\Delta)$ とスケーラブルである. PrivateRank の計算時間は暗号化操作のために DSA よりも長いが SFE relax よりも短く, また SFE strict と同等同じプライバシ保護を達成している. 本稿では触れなかった Link-unaware PWGs における実験は [5] を参照されたい.

表 1: Comparison of the computation time and the information disclosed to node i in the link-aware model, $\Delta = 100$

	computation time (msec)	disclosed information
DSA	0.015	$x_j^{(t)} p_{ji}$ for all t, j
PR (common key set)	**679**	**none**
PR (collusion-resistant)	**1740**	**none**
SFE (relaxed)	63400	$x_i^{(t)}$ for all i
SFE (strict)	>63400	none

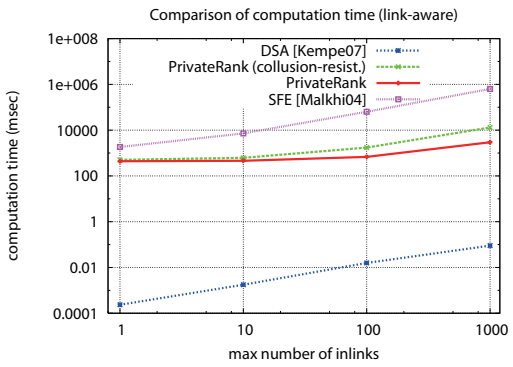

図 2: Max degree v.s. computational time in the link-aware model.

6. 終わりに

本稿では, privately weighted graph と呼ばれる計算モデルを導入し, それに対応した secure link analysis を提案した. privately weighted graph 上の secure な計算は, 秘密のリンクを持つ多様なエンティティからの知識獲得の新しい方法を提供する. ノードクラスタリング, リンク予測, 頻出構造の発見など, グラフマイニングにおける様々な問題を privately weighted graph 上で解決することが今後の課題である.

参考文献

[1] I. Dåmgard and M. Jurik. A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System. In *Public Key Cryptography 2001*. Springer, 2001.

[2] D. Kempe and F. McSherry. A decentralized algorithm for spectral analysis. *Journal of Computer and System Sciences*, 74(1):70–83, 2008.

[3] D. Malkhi, N. Nisan, B. Pinkas, and Y. Sella. Fairplay: a secure two-party computation system. In *Proceedings of the 13th USENIX Security Symposium*, pages 287–302, 2004.

[4] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web, 1998.

[5] J. Sakuma and S. Kobayashi. Link analysis for private weighted graphs. In *Proceedings of the 32nd annual international ACM SIGIR conference on Research and development in information retrieval, to appear*, 2009.