平成 25 年度卒業論文

論文題目

マイナンバー制度におけるアクセス制御

神奈川大学 工学部 電子情報フロンティア学科
学籍番号　201002666
伊藤　圭祐

指導担当者　木下宏揚　教授

目 次

図 目 次

第1章　序章

1.1　背景

近年, 企業や個人が扱う情報量は増加しており, その情報が漏えいする可能性もまた増加している. 情報漏えいを防止するために様々な研究が進められているが, 個人情報漏えいが後を絶たない. 情報漏えいを防ぐことの主な目的は, 機密情報すべてが漏えいしないようにすることである. そのために研究されている技術の一つがアクセス制御である. アクセス制御はファイルやデータにあらかじめユーザに対するアクセス権限を設定する方法である. その中でもロールベースアクセス制御 (以下 RBAC) は最近開発され, 企業などの組織体でのアクセス制御に活用されている.RBAC は役割によってデータやファイルへのアクセス権が違うので, 複数の役割がある企業でのアクセス制御に向いている. しかし, このような技術が発展しているにも関わらず, 情報漏えいが後を絶たないのが現状である. その理由は推論にあるのではないかとされている [1]. 一つ一つの秘密情報として扱われていない情報が複数集まってしまうと, ある条件下におけるとき, 推論によって本来は得ることができない秘密情報として外部に流出してしまう可能性が出てきたのである. これにより, 複数の情報が集まることにより情報漏えいが起きやすくなってきている.

また,2016 年 1 月より「マイナンバー制度」の運用が開始される. 国民一人一人に 12 桁の番号が割り当てられ, 氏名, 住所, 生年月日などの個人情報を番号で一元管理する制度である. しかしこの制度による個人情報の漏えいが懸念されている. マイナンバー制度において市民一人一人に個人番号カードという IC チップ付きのカードが交付される [2]. この IC チップに記録されるのは氏名, 住所, 生年月日, 個人番号, 本人写真などでプライバシー性の高い情報は記録されない. つまり, 個人番号カードからの情報漏えいはほぼ問題ないと考えられる. しかし市役所などの行政内部に悪意を持って情報を流出させる職員などがいないとも限らない. 職員による情報漏えいを防ぐには前述の RBAC を用いれば, 職員ごとにアクセス権を変えられるので情報漏えいを防ぐことができるのではないかと考えた. 本研究ではマイナンバー制度という制度のもと, 効率的なアクセス制御モデルを提案する.

1.2 問題点

一般に行政が扱うデータは膨大で, 更にマイナンバー制度では市役所と市役所, または部署と部署の間に膨大なデータが行き来することとなる. 一つの部署で扱っていたデータが複数の部署間で扱われることとなると, 少量だったデータが膨大になりデータ移行の際に改ざんや情報漏えいが起こる可能性がある. また, 利用者には次のような懸念があるとされる [3].

- 情報を取られたくない
- 追跡されたくない
- 都合の悪い情報は忘れてほしい

つまり個人情報の流出を恐れて情報を残したくないのである. さらに集積・集約された個人情報をもとに推論によって, 流出元の知人である特定の個人の個人情報が暴かれてしまうといった懸念がある. さらに複数のデータが一つに集まった際の処理が複雑になることも考えられる. マイナンバー制度ではマイナンバーに所得情報, 納税実績, 社会保障などの情報を紐づけて手続きを簡略化することを目的としている. 紐づけることによってマイナンバーが漏えいした時にそのものの様々な個人情報の漏えいにつながる危険性は否定できない [9]. 具体的には,

- 誰がどのように利用するのかわからない
- 目的不明確な名称により軽々しい扱いを助長してしまう
- 高度な情報収集により情報への勝手な意味づけがされてしまう

などの危険性がある.

1.3 外部の研究

1.3.1 オントロジーを用いた RBAC

この研究では大学という限られた領域内でのアクセス制御をオントロジーによって可能にした [4]. 個々のユーザーのために特権を更新することなく役割を変更できるので, 特権の管理が容易になったとされる. この研究で提案されたモデルではユーザーログイン, また役割の割り当ての際に知識ベースと呼ばれる事実や常識, 経験などの知識をコンピューターが解読できる形にしてデータベースにしたものである. この知識をコンピューターに理解させるのにオントロジーが使われている。知識ベースの中にユーザー ID やパスワード, また RoleSet と呼ばれる訳の集合を格納してあり認証のたびに知識ベースから呼び出される。この研究では大学という限られた組織の中でアクセス制御を行っているので、本紙で研究するマイナン

バー制度における RBAC も市役所という限られた組織でのアクセス制御モデルを提案することを目的としているので, 領域的に類似していると考えられる.

第2章　基礎知識

2.1　アクセス行列

アクセス行列とは主体 (subject) と客体 (object) の関係を表した行列のことで主体と客体の関係にはR(Read:読み書き可能),W(Write:書き込み可能),RW(Read+Write:読み書き可能), ϕ (Phi:読み書き不可) の 4 種類の権限がある [4].

	S1	S2
O1	ϕ	W
O2	R	R

図 2.1: アクセス行列

2.1.1　主体 (subject)

主体とはネットワークやデータベース内で管理されている客体にアクセスする行為者でありユーザに相当する.

- 名前…主体の名前
- 競合…管理している主体のコミュニティの情報
- 階層…コミュニティで指定されたセキュリティレベル
- 役割…客体の権限を決定する役割

2.1.2　客体 (object)

客体はネットワークやデータベース内で管理されている情報であり, ファイルに相当する.

- 名前…客体の名前
- 競合…管理している主体のコミュニティの情報
- 階層…コミュニティで指定されたセキュリティレベル
- 所有…管理している主体の情報
- プライベート…管理している主体の情報

2.1.3 コミュニティ(Community)

コミュニティとはコミュニティの属性, コミュニティに属する主体, およびコミュニティが管理する客体とその属性の集まりからなる社会システムに相当する. コミュニティ同士には利害関係があり. 管理している主体には組織的に階層レベルや役割を割り振られる. コミュニティにも様々な種類があるが, インターネット上のコミュニティを考えてみると主体の役割や利害関係, あるいはプライベートな情報(個人情報) が複雑に絡み合っている. 現在求められているのはこのように複雑に絡み合ったコミュニティにおいて実現するセキュリティモデルである. それが実現されたのが Community Based Access Control Model である.

2.2 Covert Channel

2.2.1 間接情報フロー

Covert Channel とはアクセス行列において, 本来客体 (Object:データやそれを含む情報) に直接アクセスする権限 (Permission:アクセス権) がない主体 (Subject:利用者, ユーザ) なのにもかかわらず, アクセス権限を持つ第三者の力を借りて間接的にその客体にアクセスできるようになってしまう. その時に発生する客体に対しての主体へのアクセス権限が矛盾した不正な経路を Covert Channel という. またこれを関節情報フローと呼ぶ. 以下の流れが例である. 初期状態 S2 は直接 O1 の情報を読み込むことができないが以下の流れで読むことができてしまう.

1. S1(Subject) が O1(Object) を読み込む.
2. S1 が O1 で読み込んだ情報を O2(Object) に書き込む.
3. S2(Subject) が O2 を読み込む.
4. 発生した Covert Channel より読めないはずの O1 の情報を S2 が読める.

このような流れで不正な情報流出が発生してしまうためアクセス制御を行う推論エンジンとしてはできる限りこれが発生するのを府伊勢具検出と訂正を的確に行えるようにするのが情報フィルタに必要とされる機能である.

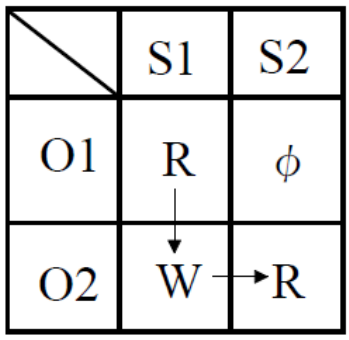


図 2.2: Covert Channel の例

2.2.2 実際に発生する Covert Channel

不正な情報経路である Covert channel をすべて塞いでしまえば安全なシステムを構築することができるように見えるが, 単独では隠れチャンネル (Covert Channel) が存在しないようなコンピュータでもネットワークに接続されたコンピュータ群が協調することによって, 隠れチャンネルを構成できてしまう. つまり, 単独では安全なコンピュータでも, それがネットワークを構成すると安全ではなくなるような状況が簡単に存在し得るのである. このようなネットワーク構成機能の問題点が Covert Channel で利用される. WWW など不特定大多数が利用するネットワークでは意図しなくても covert Channel が発生してしまう恐れがあるのでそういった情報網では比較的安易に情報漏洩が起こりうる. このように Covert Channel は今のネットワーク社会にとって情報を安易に流出させてしまう存在なのである.

2.2.3 セキュリティモデル

セキュリティモデルはアクセス制御システムを構築する上で, セキュリティポリシーを具体的な論理的形式で表現したものである. そこには制御したいサービスや組織構造が反映される. もっとも単純な型では,permission(read,write,$\neg read, \neg write$) であり, $subject$(主体), $Object$(客体) を含めた 3 つでアクセストリプルと呼び, それをシステムで如何に扱うかによってアクセス制御が行われる.

2.2.4 情報フィルタ

情報フィルタとは Covert Channel 検出時にその Covert Channel がなくなるように特定の権限を変更することである. 情報フィルタには 4 種類の方法があり, それぞれ一長一短である.3 種類はフロー経路の権限を禁止して遮断するのに対し,Read 権限を許可する方法は情報共有の拡大の意味を持つ. 以前は読めなかった客体が修

正により普通に読めるようになれば, 不正経路ではなくなるので Covert Channel 自体はなくすことができる. 情報フィルタの愚弟的な処理を以下にまとめていく. 図のように Covert Channel が発生して検出された場合, 以下, 図 2.3 の (a)(b)(c)(d) のいずれかを適用すれば Covert Channel が解消される.

1. (S1,O1) の READ 権限を削除
2. (S1,O2) の WRITE 権限を削除
3. (S2,O1) に READ 権限を添付
4. (S2,O2) の READ を削除

上記のどの情報フィルタを選択するかは各コミュニティのセキュリティポリシーや主体のアクセス履歴, ユーザがどういう方針で処理するか定めるユーザーポリシーを考慮して決定するが,(a) から (d) のどの場合でも Covert Channel は訂正できる.

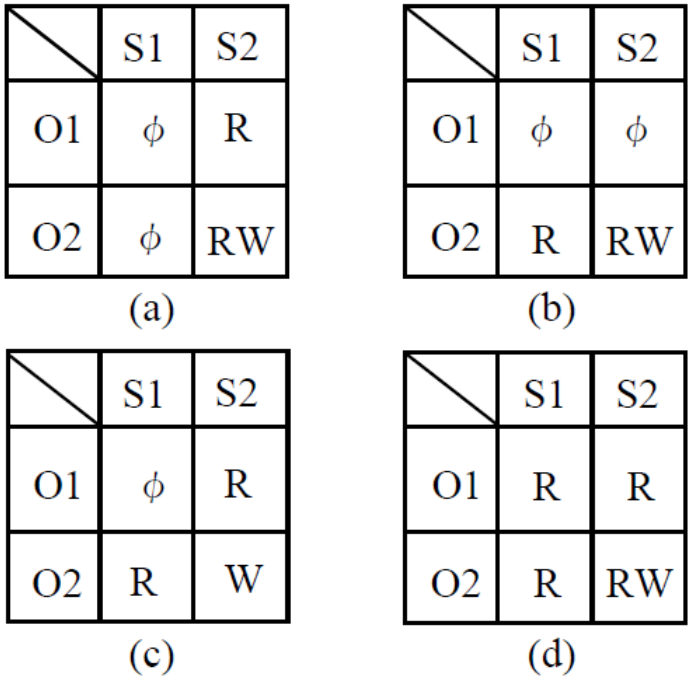


図 2.3: 情報フィルタ

2.3 推論による情報漏洩

一つ一つの情報それ自体は秘密情報でなかったとしても, それらが複数集まり何らかの推論を施すことによって, 秘密情報を抽出できてしまうことがある. そのよ

うな攻撃を推論攻撃と呼ぶ. データベースに蓄積されているデータは, それぞれが無関係に独立に存在しているわけではなく統計的あるいは意味的に関連している場合が多い. たとえば, ある SNS である発言者が同じ地名, もしくは駅名などの単語を頻繁に発言している場合, この地名や駅名と発言者との間に何らかの関係があることが統計的に分析できてしまう. 推論は統計的手法以外にも様々なものがあり, それらすべての推論解析攻撃に対抗するためには膨大な情報群とその間にある推論関係を常に監視し何らかの問題を未然に検知して警告するようなシステムが必須である. しかしそのためには情報館の推論的依存関係を記述するモデルが必要である. どのような推論手法であっても, いくつかの情報からある情報を導くことに変わりはない. その依存関係をモデル化できれば, 推論手法によらない対策を考えることができるかもしれない. 次の図はあるオブジェクト集合におけるオブジェクト間の依存関係を洗い出してリスト化したものである.

推論元オブジェクト	推論	導出オブジェクト
O1,O2,O3	⇒	O4
O4,O6	⇒	O5
O3,O6	⇒	O8
O6,O8	⇒	O7
O4	⇒	O6

図 2.4: object 間の依存関係リスト

リストの 2 行目は, オブジェクト O1,O2,O3 と, ある推論によってオブジェクト O4 が導出されてしまうことを意味している.

2.4 グラフ理論

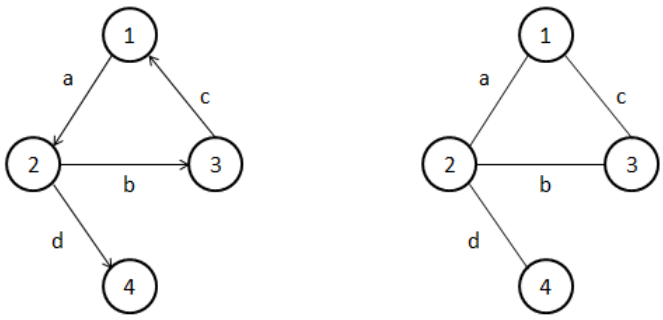
2.4.1 グラフ理論とは

グラフ理論が対象とするグラフは点 (頂点, ノードとも呼ばれる) の集合と, 辺,(枝, リンクとも呼ばれる) の集合で構成される. 統計で扱う折れ線グラフや棒グラフといった数値の変化量を示すためのグラフとは異なる. 辺は 2 点間を接続する線分として定義され, 点に何らかの意味を与え場合に, その対応する 2 点間の関係を表す [10]. たとえば, グラフは集積回路の配線パターンの表現に用いられ, 点はピンや抵抗, 電源などの回路要素, 辺はそれらを接続する配線に対応する, といったようにグラフ理論では通信ネットワークや集積回路といった非常に複雑なシステムを, グラフのシンプルな表現方法を用いて表すことで, 余分な情報を取り去り, 本質のみを表現することを可能とする.

2.4.2 グラフの諸定義と性質

グラフの基礎

グラフは点集合 V と辺集合 E の組として $G = (V, E)$ と表される. 辺は V の 2 点を結ぶ線分として表され, 向き (通常は矢印で表すことが多い) がある場合とない場合がある. 前者を有向グラフ, 後者を無向グラフと呼ぶ.



(1)有向単純グラフ　　(1)無向単純グラフ

図 2.5: グラフの例

ウォーク, トレイル, パス, 連結性

図.2.5(1) において点 1, 辺 a, 点 2, 辺 d, 点 4, というように点とそれに接続する有向辺の交代列を有向ウォークと呼ぶ. また点 4 は点 1 から到達可能であるという. 同図 (2) において同様に点 1, 辺 a, 点 2, 辺 d, 点 4, という点と無向辺の交代列を無向ウォークと呼ぶ. いずれの場合も含まれる辺の総数をそのウォークの長さという. 点は重複して現れるかもしれないが辺は重複しないウォークをトレイルと呼ぶ. さらに　点が重複して現れることのないトレイルをパスと呼ぶ. 点 u から点 v への有向パスと点 v から点 u への有向パスがともに存在するとき u と v は強連結であるという. 任意の 2 点間が強連結である有向グラフを強連結グラフという. 強連結な部分で極大なものを強連結成分という.

辺や点の除去, ブロック

辺の開放除去はその辺を取り去ることである.(図.2.6) 辺の縮約とはその辺をまず開放除去し, さらに両端点を一点に縮約する操作である.(図.2.7) 点の除去とはその点を取り去るとともにそこに接続するすべての辺を開放除去することである.(図.2.8)

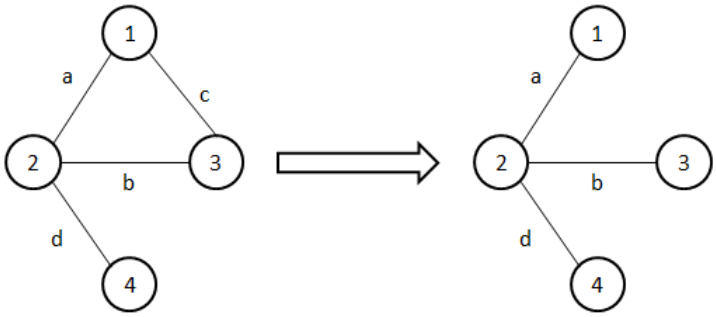


図 2.6: 辺 c の開放除去

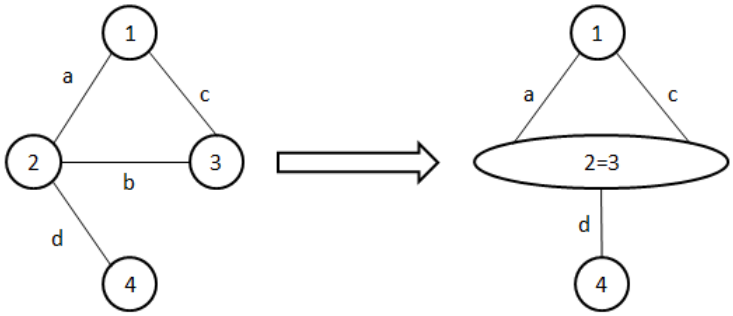


図 2.7: 辺 b の縮約

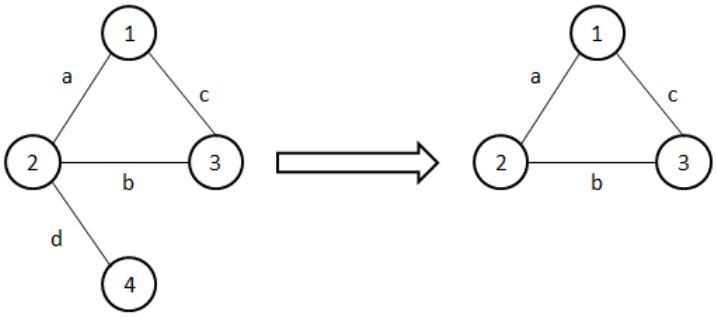


図 2.8: 点 4 の除去

辺の開放除去や点の除去によってグラフが非連結になる場合があるが, 辺の縮約はグラフの連結性は保持する. それを除去すること, あるいはそれを開放除去することによって連結成分数が 1 以上増加あるいは辺をそれぞれ切断点, 切断辺と呼ぶ. 複数の点あるいは辺の除去も同様に考えればよく, それぞれ切断点集合あるいは切断辺集合が定義できる. 辺集合 E' の縮約も, すべての辺 $e \in E'$ の縮約を同時に実行することで定められる. グラフ G に対して, 辺集合 E' の開放除去あるいは縮約により構成されるグラフをそれぞれ $G - E'$ あるいは $G\langle E' \rangle$ と表す.

2.4.3 ダイクストラ法

各辺 e に重み $w(e)$ がつけられた辺重み付きグラフ $G = (V, E)$ と G の点 v_0 が与えられたとき, 点 v_0 から G の各店への最短パスを求める問題を単一出発点の最短路問題と呼ぶ. ただし, この問題における点 v から w へのパスのうち重みの総和が最小のもののことである. ここでは点 v_0 を出発点と呼び, 点 v と点 w への最短パス上の重みの総和を v と w の間の最短距離とよぶ. 次の図は辺重み付きグラフの例である. 辺の近くに書いてある自然数がその辺の重みである. 例えばこのグラフで点 a を出発点にした場合の a から各点への最短パスをダイクストラ法を用いて求めることができる.

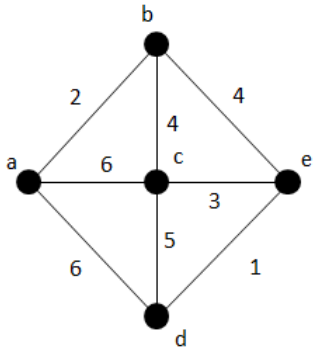


図 2.9: 辺重み付きグラフ

ダイクストラ法はオランダの計算機科学者エドガー・ダイクストラにより 1956 年に公表された単一出発点の最短路問題を解くアルゴリズムである. ダイクストラ法は出発点から各点までの最短距離を求めるようにアルゴリズムを設計することもできるが, 最短距離を同時に最短パスを構成するための補助情報も求めるように設計する. ダイクストラ法は応用範囲の広いアルゴリズムであり, インターネット上のルーティングや交通機関を乗り継いで目的地に行くための経路の探索などに

応用されている. 辺重み付きグラフ $G=(V,E)$ およびその出発点 v_0 が与えられたとき, ダイクストラ法では最初に G の各点 $v \in V$ に出発点 v_0 から点 v までの最短距離の暫定値を表すラベル $\delta\quad(v)\quad \geqq 0$ を割り当て, 手続きの進行とともにその値を減少させて真の最短距離に近づけていく. ラベル $\delta\,(v)$ の初期値は $\delta\,(v_0)$ かつ $v \neq v_0$ に対しては $\delta\,(v)=+\infty$ とする. 実際にはすべての辺の重みの総和よりも大きい任意の値を設定してもよい. グラフ G の各点に割り当てるラベルには永久ラベルと仮ラベルがある. 永久ラベルはその点までの最短距離が確定しているときに割り当て, 仮ラベルは最短距離が確定していない点に割り当てる. なお出発点 v_0 には永久ラベル $\delta\,(v_0)=0$ が割り当てられている. またダイクストラ法を適用している場合, 各点 v に v のラベルが永久ラベル化仮ラベルのどちらであるかをラベルとは別に設定する必要がある. 最短パスを求めるために各点 v に v_0 から v への最短パス上で終点 v に隣接している点の候補を補助情報として記録する. この補助情報を $\mathrm{Pre}(v)$ で表し, 点 v に補助情報が存在しないことを $\mathrm{Pre}(v)=\perp$ で表し, 常にこの式が成り立っている. 次にダイクストラ法の手法を示す. ここで扱うグラフは図.2.9 とする.

手順

(1) 点 a に隣接している点 b,d については辺 (a,b), 辺 (a,d) の重みが最短であるので, 永久ラベルを割り当てる. また補助情報として最も各点に近い点 (ここでは点 a) の補助情報を Pre(b)=a,Pre(d)=a とする.

(2) 点 c に関しては経路が 3 通り存在するので, 重みの総和が最も小さいものを永久ラベルとする (赤枠の値). 更に補助情報として Pre(c)=a を割り当てる.

(3) 最後に点 e にも隣接している各点からの重みの総和が最も小さいものを永久ラベルとして割り当て, 最短距離を決定する. また補助情報は Pre(e)=b とする.

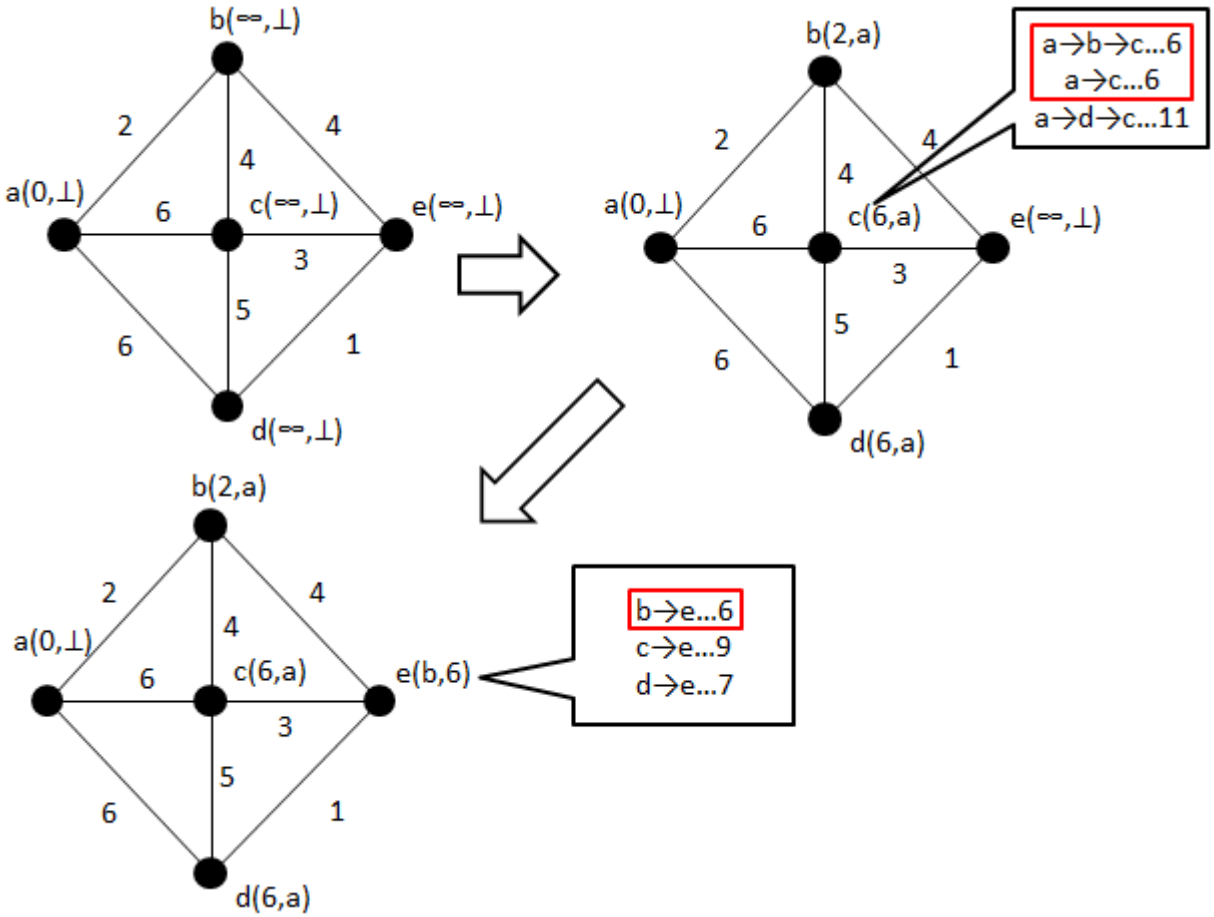


図 2.10: ダイクストラ法による最短経路の決定

2.5 有向グラフと頂点着色によるモデル化

2.5.1 グラフの頂点着色

グラフ G=(V,E) の頂点着色とはグラフの頂点に色を塗ることである. すなわち色集合と写像 c:V → C を与えることである. 特に (v, w) ∈E であるようなどの 2 頂点 v, w ∈V についても $c(v) \neq c(w)$ となるとき c を頂点彩色と呼ぶ. 色数 $|C|$ が制限されているときに頂点彩色可能かどうかを判定する問題や, 可能ならばその彩色を求める問題はグラフ理論において重要な問題として研究されており, 様々なアルゴリズムが考案されている. 一般に, ある条件 P を満たす頂点着色を P 着色と呼ぶ. 頂点着色は次のように一般化できる. グラフの頂点にあらかじめ色のリストが与えられているとする. 即ち, 写像 $L : V \to 2^c$ が与えられているとする. このこのとき G の頂点着色 c で, すべての頂点 v ∈Vに対してc(v)∈L(v)を満たすものをGのリスト

着色と呼ぶ.任意の頂点v∈Vに対してL(ｖ)=Cならば通常の頂点着色である.cが頂点着色の時は特にリスト彩色と呼ぶ.

2.5.2 推論による頂点着色のグラフ表現

まず, オブジェクト間の依存関係リストを次のようにグラフ化する. 頂点集合 V をオブジェクト集合都市, 依存関係リスト上で $O_{i1}...O_{i1} \in V$ から $O_j \in V$ が導出ならば有向辺 $(O_{i1}, O_j), ..., (O_{ik}, O_j)$ を描く. さらにそのグラフに対する色リストを用いて ACL を次のように表現する. ただしここでは議論を簡単にするために ACL において read 可能か否かのみに着目する. まず, 色集合 C をサブジェクト集合とする. そして,ACL 上で, あるサブジェクト $S_i \in C$ があるオブジェクト $O_j \in V$ を read 可能なら,$S_i \in L(O_j)$ とし O_j の色リストに S_i を加える. 例として図.2.4 をグラフで表現し, ある ACL に従って色リストを与えたものを次に示す.

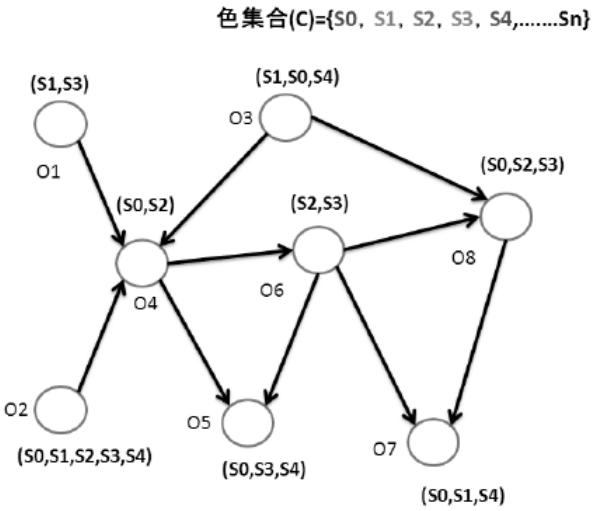


図 2.11: 推論的依存関係と ACL の有向グラフ表現

サブジェクト S_i がオブジェクト O_j を read したとき頂点 O_j に色 S_i を塗るとする. 例えば図.2.5 において O_1,O_2,O_3 がすべて S_1 で塗られたとする. このとき, 推論によって O_4 が導出可能なことを考えれば O_1,O_2,O_3 がすべて S_1 で塗られた時点で O_4 も S_1 で塗るべきである. しかしその一方で,$S_1 \notin L(O_4)$ 即ち,ACL 上では S_1 は O_4 を read できないこととなっているので, これは推論による情報漏えいを意味している. このとき, 頂点着色はリスト着色の定義にも反していることに注目すると推論による情報漏えいに関する安全性を次のように定義できる. 定義:ある P 着色がリスト着色ならばその着色は推論に対して安全であるという. ここで P=「任意の頂点 v に対して v を終点とするすべての有向辺 $(u_i, v), ..., (u_k, v)$ の始点 u_i が同一色で塗られているならば $c(v)$=$c(u_i)$ でなければならない.」とする.

2.5.3 有向グラフ表現の問題点

有向グラフでは表現不可能な依存関係リストが存在する. 例えば次の図において,O_1,O_2 から O_4 が導出でき,O_1,O_3 からも O_4 が導出できる. これを行こうグラフで表現しようとすると O_1,O_2,O_3 から O_4 へ有向辺を描くことになり, しかしそれでは O_1,O_2,O_3 から O_4 が導出できるという意味になってしまう. この問題を回避するために有効ハイパーグラフを用いる.

推論元オブジェクト	推論	導出オブジェクト
O1,O2	⇒	O4
O1,O3	⇒	O4
O1,O2,O4	⇒	O5
O1,O3,O5	⇒	O2

図 2.12: 有向グラフでは表現不可能なリスト

2.6 ハイパーグラフ

2.6.1 ハイパーグラフの定義

グラフにおいて辺とは 2 頂点対のことであった [5]. これは辺は 2 個の頂点からなることを意味する. この個数制限を自由にすることで一般化したものがハイパーグラフである. ハイパーグラフは H=(V,E) と記述される. ここで V は頂点集合,E$\subseteq$ 2^V は V の部分集合である.

2.6.2 有向ハイパーグラフの定義

有向ハイパーグラフ H=(V,E) は頂点集合 V と有向辺の集合 E から構成される. ここで有向辺とはからではない互いに素な V の 2 つの部分集合 S,T の順序対 (S,T) である.

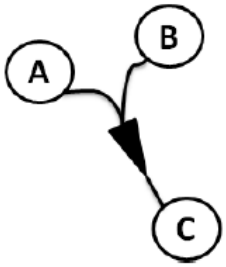


図 2.13: 有向ハイパーグラフ

2.7 マイナンバー制度

2.7.1 マイナンバー制度について

マイナンバー制度とは国民一人一人に個人番号と呼ばれる固有の番号を割り当て, 諸手続きの簡略化を目的とした制度である. アメリカなどではすでにこの制度

が適用されており, 医療, 介護, 年金などの社会保障などの分野で利用されている。しかし, なりすましなどによって番号が売買されており, 主にネット犯罪が横行している. 政府はこのなりすまし犯罪に対策を練っているが, 解決には至ってない.

2.7.2 日本におけるマイナンバー制度

日本でのマイナンバー制度では、国民一人一人に個人番号カードが交付される。この個人番号カードには IC チップが搭載されており、この IC チップには氏名, 住所, 生年月日, 性別, 個人番号, 本人写真が記録される。このようにプライバシー性の高い情報 (地方税関係情報, 年金給付関係情報などの特定個人情報) は記録されない. 他には総務省が定めた公的個人認証に係る電子証明書など, 市町村が条例で定めた事項などが記載される. この個人番号カードから流出する可能性があるのは基本的な 4 情報などである.

2.7.3 マイナンバーにおける推論

マイナンバー制度では様々な機関をまたがり様々な情報がやり取りされる. 更にその情報にはマイナンバーを通じて多くの情報が紐づけられている. 第 2 章で推論による情報漏えいについて示したように, 多いとは言えない情報から推論によって情報漏えいが起きてしまう可能性がある. 具体的な例を次に示す.

- 企業が自治体に提出する給与所得支払調書には次のような情報が格納されているとする.

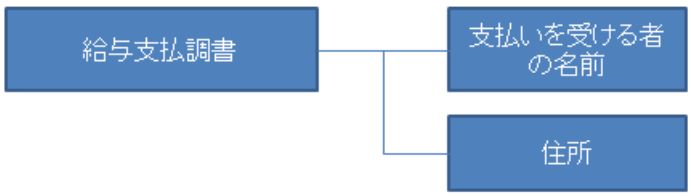


図 2.14: 給与支払調書に格納されている情報

- 市役所が扱う印鑑登録証明書には次のような情報が格納さてているとする.

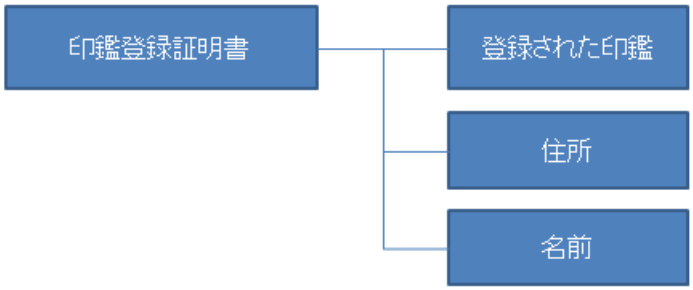


図 2.15: 印鑑登録証明書に格納されている情報

これらの情報について推論を行うと, 本来なら登録された印鑑は見れないはずだが, 住所と名前からの推論によって登録された印鑑が見えてしまう可能性がある.

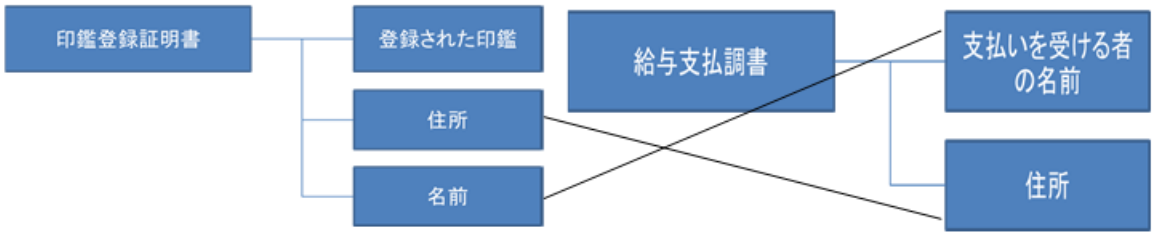


図 2.16: 推論されてしまう情報

このような推論による情報漏えいを防ぐためにハイパーグラフを用いる.

第3章　提案

3.1　目的

マイナンバー制度においては, 個人番号によって保有するディジタル化された個人情報を名寄せし, 管理する [5]. 個人番号に対して個人情報を紐づけることによって個人番号が漏えいした時にその者の様々な個人情報の漏えいにつながる危険性は否定できない. よってハイパーグラフによって個人番号と個人番号に紐づけられたデータを複数の推論から導出可能な組み合わせをすべて算出し, 推論による頂点着色を満たすリストの組み合わせが一つも存在しなければそのアクセス制御リストは推論による情報漏洩を考慮した時読み込めないオブジェクトが存在する不自由な ACL をを発見する問題設定を行うことを目的とする。

3.2　提案モデル

提案モデルにはハイパーグラフを用いる.

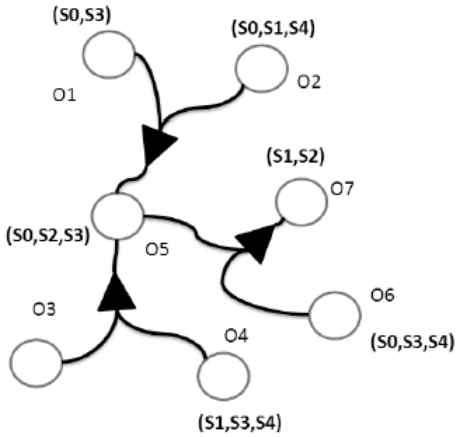


図 3.1: 複数の推論ができるグラフ

推論によるリスト着色が一つでも存在しなければその ACL は read することができないオブジェクトが存在することになる。そのような ACL はアルゴリズムの目的は与えられた依存関係とリストで構成されたグラフがリストの組み合わせが一

つでも存在していれば, その ACL は推論に対して安全なリストの組み合わせが存在していることを示す. 上記の図において (O1,O2) → (O5),(O3,O4) → (O5) だったとき O1 と O2 が S0 で塗られ,O3 と O4 が S3 で塗られていたら O5 を S0 と S3 のどちらで塗るのかという問題がある. (O5,O6),(O7) で,O6 に S3 が塗られていて,O7 の L に S3 が無いときもし O5 を S0 で塗ったとき O5 と O6 が S3 →推論で O7 も S3 になってしまうケースを見逃す可能性がある. このような場合は O5 に S0 を与える場合と S3 を与える場合の両方を記述することにする. このようにアクセス制御リストの修正が必要な場合もあり, 効果的に修正するためにはどのような問題設定をすればよいかを考察する.

参考文献

[1] 鈴木遼:”推論による情報漏えい防止のためのオブジェクト関係の視覚化”pp.5-6(2009)

[2] ”社会保障・税番号制度”,http://www.cas.go.jp/jp/seisaku/bangoseido/

[3] 石井夏生利:”マイナンバー制度とプライバシー・情報セキュリティ”, マイナンバーシンポジウム,pp2-8(2012)

[4] Avita Katalt, Pranjal Gupta, Mohammad Wazid, R.H. Goudar, Abhishek Mitta l , Sakshi Panwar and Sanjay Joshi:”Authentication and Authorization: Domain Specific Role Based Access Control Using Ontology”,Proc.Intelligent Systems and Control (ISCO),pp439-444,Jan. 2013

[5] 鈴木遼:”推論による情報漏えい防止のためのハイパーグラフによる依存関係のモデル化とアルゴリズム”(2011)

[6] Rick Kuhn:”Role Based Access Control”,ProcInromation Technology council(ITI),pp6-13,(2013)

[7] 溝口理一郎:”オントロジー工学”, オーム社 (2005)

[8] 小林夏生:”只見町　インターネット・エコミュージアムの「キーワード」検索の改善”pp.25(2013)

[9] "「マイナンバー」制度に潜む危険",
http://157.14.215.152/page/library/kaihou/2406_03_mynumber.html

[10] 船曳信生, 渡邉敏正, 内田智之, 神保秀司, 中西透:”グラフ理論の基礎と応用”, 井立出版 (2012)