平成 25 年度卒業論文

論文題目

振舞いの構造類似性を用いた Covert Channel 分析

神奈川大学 工学部 電子情報フロンティア学科
学籍番号　200902753
島田 琢士

指導担当者　木下宏揚　教授

目 次

図 目 次

第1章　序論

1.1　背景

昨今,CGM（Consumer Generated Media）と呼ばれる, ソーシャルネットワーキングサービス（SNS）, ブログ, COI（Community Of Interest）サイト等, インターネットなどを活用して消費者が内容を生成していくメディアや, ネットワークを通じ, ソフトウェアやデータなどにサービスという形でオンデマンドにアクセスできるクラウドコンピューティングなどが普及してきている. また, 個人のプライバシーを含む情報や企業機密を含んだデータをネットワークシステムに組み込むことで, インターネットの利便性は益々高まっている. これらにより, 誰でも情報へのアクセスが容易となり, 人々がインターネットを日常的に利用する社会へと遷移してきている. しかし, 公然の情報だけでなく, 機密情報にさえ誰でも自由にアクセスできる状況は好ましくない. そこで, データにアクセスできるユーザを制御するセキュリティ技術として, アクセス制御（Access Control）が発展してきた. これにより, 権限を持たないユーザによるデータの閲覧や改ざんの脅威から護られている.

それでも, アクセス制御上の脆弱性によって情報漏えいが起きる可能性がある. 情報漏えいを引き起こす脆弱性の一つに,Covert Channel と呼ばれる不正な情報通信路がある.

Covert Channel はシステム側の欠陥によって発生するものではなく, アクセス制御で許可されているユーザから連鎖的に情報が伝播することで, 本来権限のないユーザが情報にアクセス可能になるといった脆弱性である. 当然ながら, 機密データを持つネットワーク上に Covert Channel が存在すると情報漏えいというリスクが顕在化する.

このような背景から,Covert Channel の検出技術や制御技術の発展が望まれている.

1.2 問題点

急速に発展し, 巨大化・複雑化したインターネットが原因でアクセス権限も複雑に絡み合うようになり, その結果ネットワーク内では不正な情報経路が発生し, 情報流出の危険性が増大してしまっている.

従来の主な情報漏えい対策は, 個人情報や企業機密のような秘密情報という, データを取り巻く対策が主な目的であった. 例えば, 誰がどの情報にアクセス可能かどうかを規定するアクセス制御リスト（Access Control List）によって情報漏えいの危険を監視するためのモデルやシステム, アルゴリズムなどの研究である. しかし, このような把握したコミュニティの ACL のみを用いた Covert Channel の解析だけでは検出できないアクセス権の矛盾が存在する場合があるといった問題点がある.

また, 情報漏えいを防ぐように情報フィルターで制御してしまうことで, 自由な情報のやり取りを制限してしまう恐れもある.

1.3 関連研究

いままでの我が研究室の Covert Channel 分析では, 把握したコミュニティの ACL のみを用いた Covert Channel の解析や, アクセス行列の振舞いの類似度による Covert Channel の検出などを行ってきた. これら分析の手法として Tanimoto 係数を用いた研究が挙げられる. また, 外部の研究では後ろ向き推論システムなども行われている.

1.4 提案

本論文では,Covert Channel 発生の要因の一つである,READ,WRITE の連鎖であるアクセス行列を振舞いと定義し, その振舞いをグラフと

して抽出し, その構造類似性から Mason を用いて群れを作り情報漏えいのメカニズムの分析を行うことで, 不正な閲覧・改ざんを防ぐ手段を提案するものである. この際, 公開しても差し支えない情報の漏えいはある程度許容し, 秘すべき情報の漏えいを最低限に抑えることを目的とする. これにより, アクセスの連鎖という人の活動による情報漏えいの被害を最小限に止めることが期待できると考えている.

第2章　基礎知識

2.1　アクセス制御

アクセス制御とは, 情報資源の不正利用を防止するために, 決められた規則に従って資源へのアクセスを制限することである．正当なユーザにのみ情報のアクセスを許可することにより, システムやデータの機密性と完全性を保持するものである.

2.1.1　任意アクセス制御

任意アクセス制御（DAC）は, 主体の識別子やグループ名などに基づいてオブジェクトへのアクセスを制限する方式であり, オブジェクトに誰がアクセスできるかは, オブジェクトの所有者が任意に決定することのできるアクセス制御の方式である．任意アクセス制御の特徴の一つは, オブジェクトの所有者にアクセス制御ポリシーの決定権を移譲することで, 中央集権的ではなく柔軟にシステムが運用できることである.[?]

2.1.2　強制アクセス制御

強制アクセス制御（MAC）は，主体やオブジェクトに割り当てられたセキュリティレベルと, 形式的に定義された認可モデルとに基づいてオブジェクトへのアクセスを制限するアクセス制御の方式である．強制アクセス制御は情報フローの厳密な制御を実現するために考え出された仕組みであり, 管理者によって決められたセキュリティポリシーに従ってすべてのアクセスが決定される. そのため, ユーザ自身が生成したリソースであっても自由にアクセスできるとは限ら

ない. 当初から主に軍関連システムとして利用されてきたが, 近年ではよりセキュアなオフィス用オペレーティングシステムとしても使われるようになってきた.[?]

2.1.3 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り, そこには制御したいサービスや組織の構造が反映される.

一般に, アクセス制御のための基本的な要素は,Subject（主体）,Object（客体）である. 要素間の関係は, もっとも単純な型では, Permission（read,write, ¬ read，¬ write）であり, これら 3 つ（アクセストリプル）を如何に扱うかによって, アクセス行列モデル, あるいはケーパビリティモデルに分類される. アクセス制御する上で, 主体がいかなる Permission を客体に対して許可されるかを決定するために, アクセストリプルという基本構造の中に, 主体と主体の関係, 主体と客体の関係, 客体と客体の関係が考慮される必要が生じる. 更に, アクセス制御システムとして十分に考慮しなければならない問題として Covert Channel の問題がある.

Covert Channel とは, 情報を「記号作用部」（書かれたもののタイトル等）,「記号内容部」（記号意味部）に分類したとき, 主体による客体の read,write の連鎖によって記号意味部が伝搬し, 本来記号作用部としてはアクセスが許可されていない, 主体がその記号意味部を読み取れる, あるいは書き込める現象をいう. 主体と客体の関係, 及び Covert Channel の視点でセキュリティモデルとして以下がある.

- Role Based Access Control モデル
  主体の役割に着目したモデル. アクセス権限がユーザが組織内で現在果たしている役割（role）に基づいて決定される. つまり, 複数の subject の権限をまとめて管理することができるものである.

- Chinese Wall モデル
  組織の競合関係に着目し,Covert Channel 分析について扱うことができるモデル. 一般的な規則は「あるビジネス分野のある企業と最近係わった人間は, 同じ分野の別の企業の書類には近づいてはならない」というように, 利害の対立を防ぐ仕組みを提供しているものである.

- Bell and LaPadula モデル
  セキュリティレベルに基づく階層モデルであり, 下位レベルへの情報の流れを禁止している. Covert Channel は起こらないが, 単独では使い勝手が悪い. Biba Integrity モデルは,Bell and LaPadula モデルとは逆に上位レベルへの情報の流出を禁止している. ユーザの行動とは独立にセキュリティポリシーが遂行される強制アクセス制御（MAC）である.

- Ownership モデル
  所有という概念に着目したモデル. 主体が, 所有関係にある客体に対してアクセス権限を変更, 設定できるというものである. 誰が何をどのような理由で所有しているのかという, 主体の所有制が問題となる.

2.2 Covert Channel

2.2.1 間接情報フロー

アクセス行列において,Subject ,Object ,Permission をアクセストリプルと定義し, そのアクセストリプル間で起きる不正な情報の経路を Covert Channel と定義する.

【定義】アクセス行列 : Subject, Object, Permission からなるアクセストリプルの集合をアクセス行列と呼ぶ.

- Subject : 人の名前
- Object : 情報リソースの名前
- Permission : Subject が Object を Read ,Write する権限

とする. これは,CovertChannel をアクセス行列の中に限定し, 行動から生じる情報漏えいを分析・制御する事を目的とするためである. ここで R は Read 権限,W は Write 権限,RW は R ∧ W 権限, Φ は¬ R ∧¬ W 権限, 空欄は権限が不確定を意味する.

【定義】Covert Channel : アクセス行列において, アクセス禁止の Permission に矛盾する情報フローのことを Covert Channel と呼ぶ.

即ち,
SubjectSi（i = 1.2.…）,Sj（j = 1.2.…）, 但し i ≠ j
ObjectOn（n = 1.2.…）,Sm（m = 1.2.…）, 但し n ≠ m
PermissionPR,W,RW, ¬ RW
アクセストリプル < Si,On,P >について,
If < Si,On,¬ R >,
AND if < Sj,On,R > ∧ < Sj,Om,W > ∧ < Si,Om,R >,
Then Covert Channel（Si,Sj,Om（Om））.

と定義される.

この定義では,SubjectSi はObjectOn と言う名前とその内容をREAD禁止< Si,On,¬ R >であるにもかかわらず,SubjectSj がObjectOn の内容をREADし,ObjectOm にそれをWRITEし,SubjectSi がObjectOm の内容からObjectOn の内容をREADする事< Si,Om,R >によって, アクセストリプル< Si,On,¬ R >と矛盾する結果が生じることが示される.

図2.1 は, 矢印の流れでCovert Channel が起こることを表している.

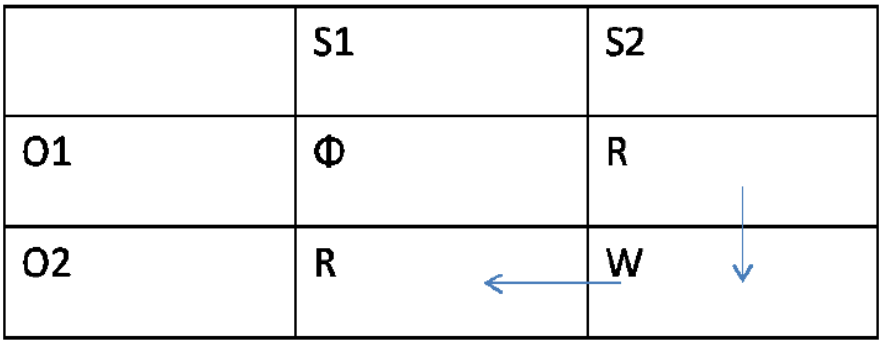


図 2.1: Covert Channel(間接情報フロー)

2.2.2 実際に生じる Covert Channel

不正な情報経路であるCovert Channel を全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Covert Channel（隠れた経路）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, Covert Channel 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.

例えば以下のような例が挙げられる.

- 会社の機密データを社外へ持ち出したり, 社外の人間（社外のPC）でも見られるようにする.
- SNS の個人データが掲示板やブログ等不特定多数へ流出スパイウェア等, 個人 PC から情報を持ち出すためにこれを用いて通信を行い, 検知を困難とする.

WWW 等, 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えいが起こりうる. 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 フローレベル

Covert Channel において, ある Subject から Subject へと情報が流出する回数をフローレベルと定義する. 関わる Subject の数によって,Covert Channel の程度は決まり,Subject 数を k としてフローレベル k と表記できる. フローレベル k はフローレベル 2 が基となっている如何なるフローレベル k においてもフローレベル 2 をベースとして生成される性質である.

この性質から, フローレベル 2 を抑えればそこから発生しうる多くのフローレベル k を防ぐことが出来, Covert Channel 分析と評価ができる.

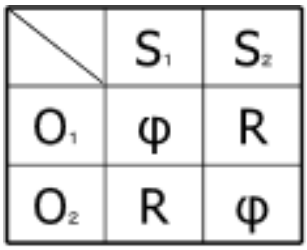
2.2.4 情報フィルタ

情報フィルタとは Covert Channel 検出時にその Covert Channel が無くなるように特定の権限を変更することである. 情報フィルタの具体的な処理を以下にまとめていく.

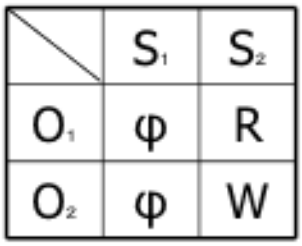
図 2.1 のように Covert Channel が発生して検出された場合, 以下, 図 2.1 の 1.2.3.4. のいずれかを適用すれば Covert Channel が解消される.

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 権限を削除

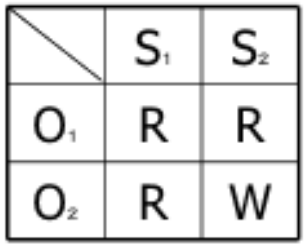
1.(S1,O1) の READ 権限を削除



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 権限を削除

図 2.2: 情報フィルタ

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.

2.3 Hadoop

Hadoop は, メタ言語（タグ) を含むインデックスを生成するシステムでクラウド内に散らばったリソースのファイル名, ファイルの内容の語を収集, 分析し, インデックスとして纏める機能を持つ. Hadoop は Map 関数と Reduce 関数という二つの関数の組み合わせを定義する

だけで, 大規模データに対する様々な計算問題を解決できる. HadoopはMap フェーズとReduce フェーズの二つから成り立っている. Mapとは, 情報の分解・抽出を行う関数でこのフェーズで大量の情報を分解し, 必要な情報を抜き出して出力する. この時,Map 関数によってkey/value のペアの集合が生成される. Reduceとは,Map フェーズで抽出された情報を集約し, それに対して計算を行い結果を出力する. 計算処理は,Worker と呼ばれるノードが行う. 図 2.4 ではWorker はW で示されている.

2.4 群知能

2.4.1 Boid

1987年, アメリカのリレイグ・レイノズルによって考案・製作された人工生命シミュレーションプログラムである. Boid のモデル名は鳥もどきという意味（Bird-android）.Boid の群れを実現させる振る舞いは３つの要素からなる. 「衝突の回避」,「群れの中心に向かう」,「向きを合わせる」といった３つのルールを規定するだけで鳥の群れをシミュレーションすることができる.[?]

2.4.2 「衝突の回避」Separation

引き離し. 近くの鳥や物体に近づきすぎたらぶつからないように離れるルールである. Boid 同士が近づきすぎたら, 前を飛んでいる Boid はスピードを速くして, 後ろを飛んでいる Boid はスピードを遅くするといったように, 衝突の回避のために Boid はそれぞれ自分にとっての最適距離を持っている. 障害物に対しては, それにぶつからないように方向転換して衝突を避けるようにする.[?]

2.4.3 「群れの中心に向かう」Cohision

結合. 鳥たちが多くいる方向へ向かって飛ぶルール. 鳥たちが多くいる方へ向かって飛ぶというのは, おおざっぱにいうと群れの中心（重心）方向にむかうということ. つまりこのルールは,Boid に群れの中心の方向へ飛んでいくとを指示している. この群れの中心は, 全 Boid の位置（座標）の平均として求める.[?]

2.4.4 「向きを合わせる」Alingment

整列. 近くの鳥たちと飛ぶスピードや方向を合わせようとするルール. 同じ方向にあまり距離を空けないように飛ぶようにする. このルールはある一定の距離より遠ざかり過ぎてしまったら, 前の Boid

はスピードを遅くし, 後ろを飛んでいる Boid はスピードを速くするといったようにすることで実現することができる.[?]

2.4.5 群れるふるまい

集まる力の源は”家族的類似”であり, 群知能のパラメータとして表現される.

群れる正の力

- 群れの中心に向かう力： Cohision
- 隣人と家族的類似行為をする： Alingment
- 行為の濃度： Pheromone

Pheromone： 行為の重要性を表現する.Pheromone は揮発性であり, 濃度が濃いほど重要な行為を表す.

群れる負の力

- 群れから排除する力, 離れる力： Separation

2.5 家族的類似

ルートヴィヒ・ウィトゲンシュタインの言語ゲームで振る舞う particle は, 家族的類似性によって群れを作ると定義した. 家族的類似性は言語と行為の類似性を表現するものである. 家族的類似性は同値関係でもないし, 等価関係でもない. 常に変動しつつ, 少しづつ似ているエンティティの集まりである. しかし, それは自己から見れば, 同値関係であってもよいし等価関係であってもよい. そのような個人個人の意味論を統合して世界を記述する意味論が無いということである. 家族的類似性は不確実な世界の中の同類の定義である. 家族的類似性は公的言語世界の観察視点の同類の定義である. 群知能にお

いて群れる振る舞いは家族的類似に基づいている. 集まる力の源は,「家族的類似」であり,群知能のパラメータとして表現される.

2.6 Cl-GBI 法

大量に蓄積された電子データから興味深い有用な知識を獲得するデータマイニングにおいて, 近年, 複雑な構造を有するデータを扱うためにグラフ構造データを対象としたグラフマイニングが活発に研究されている. その一手法である Graph Based Induction（GBI）法は, ノードペアを遂次拡張（チャンク）することにより, グラフ中に頻繁に現れる典型的なパターンを高速に発見することができる. また,GBI 法のチャンキング時の曖昧性及びチャンクすることによる探索空間の不完全性などの問題を軽減した Beam-wise GBI（B-GBI）法も提案されている.

しかしながら,GBI 法及び B-GBI 法は部分的に重複するパターンを同時に抽出できない. Chunkingless GBI（Cl-GBI）法では, ノードペアをチャンクせずに一つの塊として捉えること（疑似チャンキング）で重複パターンの抽出を可能とした.

2.6.1 データマイニング

データの中に潜んでいる価値ある情報や知識を掘り出すことを目的とした (大規模データに対応可能な) データ処理技術である. この研究では Cl-GBI 法という一つのデータマイニングを, 大量のアクセス行列の中から制限を付けて特定のデータを抜き出すのに用いる.[?]

2.6.2 グラフマイニング

コンピュータネットワーク, 鉄道路線, 道路交通網, 神経回路などが形作る, 幾つかの頂点（ノード）を結ぶ網の目状の構造を, グラフ構造と呼ぶ. 例示から分かるように, この構造を持ったデータは多岐に渡り, なおかつそのネットワークの経路の最適化や構造推定, 変化点の検出は, 我々の生活基盤を大きく改善する可能性がある. 信号の切り替えのタイミングが改善されれば, 渋滞は緩和されるであろうし, 送電網が最適化されれば送電ロスが軽減され, 電気料金は値下げされ

るであろう. グラフマイニングとは, 上記のような目標を達成するために, グラフ構造が持つ性質を調べる事を指す. そのため得たい情報に対応する様々な手法が存在する. 簡単な一例としては, ノードのグループ分けが考えられる.[?]

2.6.3 Cl-GBI 法のアルゴリズム

入力：グラフデータベース D, ビーム幅 b, 疑似チャングの繰り返しの最大数 N, 最低支持度 θ ,
出力：典型的なパターンの集合 S（初期値は空集合）

Step1 D 中のグラフから隣接する 2 つのノードから成る全てのペアを抽出する. レベル 2 以降については, 2 つのノードのうち少なくとも一方は新しく登録された疑似ノードから成るペアのすべてを抽出する.

Step2 抽出されたペアの頻度を数える. ここで, θ よりも低い頻度のペアは削除する.

Step3 ”Step1”で抽出されたペアの中から頻度の高い順に b 個のペアを選び, それぞれを抽出パターンとして S に加える. この時, ペアを構成するノードが疑似ノードであれば元のパターンに復元してから S に加える. 疑似チャンクすべきペアがない場合, もしくは, レベルが N の場合はここで終了する.

Step4 ”Step3”で選ばれたペアにそれぞれ新しいラベルを割り当てる. ただし, グラフは書き変えない. そして,”Step1”に戻る.

2.6.4 深さ優先探索

深さ優先探索アルゴリズムは木の最初のノードから, 目的のノードが見つかるか行き止まりまで深く下がって行く. スタートか根ノードから始まり, ずっと下ったところの葉ノードまで探索する. もしゴール

ノードが見つからなかったら引き返し,次のまだ通っていないツリーを葉に到達するまで探す.[?]

2.7 マルチエージェントシミュレータ

2.7.1 マルチエージェント

自らの価値基準に従って自分の行為を自由に選択できるような自律的なが多数共存する環境である, エージェントは, 構成要素のうち特に主体性を持ち, 自律的に行動するものである. エージェント同士の相互依存関係があり, エージェント同士の相互作用により, やがてシステム全体の流れのようなものが創発され, その流れが今度は逆にエージェントにフィードバックされ, また個々のエージェントのふるまいを決定していく.[?]

2.7.2 Mason

2 次元・3 次元の空間を使用した物理・社会モデル用のシミュレーションライブラリ. 適切なクラスを継承してモデルを実装することでそれに合わせた GUI が提供される. モデル開発というより,MASON ライブラリを使用したアプリケーション開発という色合いがどうしても強くなってしまう. 従来のシミュレータに比べ高速で, 客観的に第三者的な視点から観察することができる. 実装言語は Java.[?]

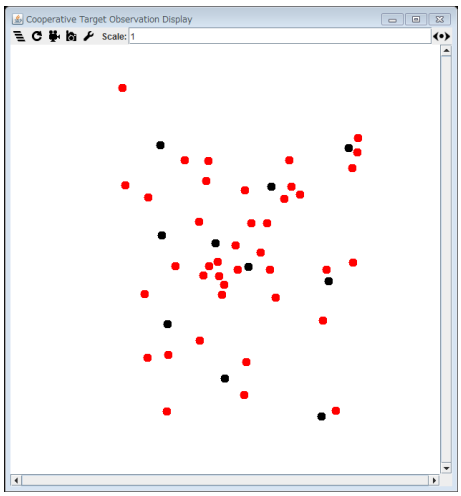


図 2.3: Mason のシミュレーション例 (Cooperative Target Observation)

第3章　提案

本研究での提案フローは以下の通りである.

1. Cl-GBI 法を用いてアクセス行列の振舞いの構造分布行列, グラフを抽出する
2. 構造分布行列, グラフから振舞い間の類似度を算出する
3. 算出した類似度を, 類似した振舞いが群れを生成するためのパラメータとして与える.
4. 振舞いが群れる事を Mason にて実証する
5. 群れに生じる Covert Channel の分析を行う
6. Covert Channel が検出されれば, 群れを護るよう制御を行う

以下, 本章では提案の詳細について述べる.

3.1　Cl-GBI法で構造分布行列の抽出および類似度を算出する

膨大なアクセス行列で制約として, 一つもしくは複数抜き出したいペアを設定する (設定したペアの数分, 大きな群れができる). そこから疑似ノードが生成され, 部分構造が抜き出される. その過程で構造分布行列, グラフが求まる.

3.1.1　”構造類似性”を用いての一致度, 不一致数の算出

二つのグラフの似たグラフ, 部分構造から一致度, 不一致数を求める.

1. 比較する 2 つのグラフから同じラベルを持つノードをそれぞれ一つずつ用意する.
2. 構造分布行列を利用し,2 つのノードの一致度 (C)，不一致数 (E) を計算する. ただし,
   - 各部分構造毎に二つのノードに含まれている個数の最小値にその部分構造に含まれるノード数を重みとしてかけ, その総和を一致度とする.
   - 2 つのノードにおいて, 各部分構造の数の差を計算し, その総和を不一致数とする.
3. 同じノードラベルを持つ他のすべてのノードのペアについて（2）の処理を行う.
4. 計算された各ノードペアの中で, 一致度の割合が最大のペアのノードを各グラフより取り除き各数値を保存する.
5. 同じラベルを持つノードのペアが存在しなくなるまで（1）～（4）の処理を繰り返す.
6. 保存されている一致度と不一致数のそれぞれの総数から一致度の割合を求め, これを類似度とする.

3.1.2 グラフの構造分布行列表現

Cl-GBI 法により抽出された部分グラフを用いることによって, 各グラフは, それを構成するノードと部分グラフとの関係に基づいて表現することができる. 対象とするグラフ集合から抽出された部分グラフ集合を P, 抽出された部分グラフ数を J, ノード nki を含む部分グラフ pj2P,j =1,2, ⋯ ,J, の数を mkij=mk(nki; pj) とおくと, 任意のグラフ Gk は以下に示す行列 Mk で表現することができる.

$$M_k = \begin{bmatrix} m^k_{11} & m^k_{12} & \dots & m^k_{1n} \\ m^k_{21} & m^k_{22} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ m^k_{I_k1} & \dots & \dots & m^k_{I_kJ} \end{bmatrix}$$

この行列 Mk を構造分布行列と呼び,Gk の構造上の特徴が表現されているとみなす. これは,Cl-GBI 法の実行過程の情報から容易に作成することができる.

3.1.3 グラフ間類似度の算出方法

本稿では, まず任意の 2 つのグラフ中に含まれる共通ラベルを持つノード集合ごとに, 各ノード間の類似性を定義し, 次に, そのノード間の類似性を用いてグラフ間の構造類似性を定義する. その際, 各ノード間の類似性は, 関連している部分グラフの共通数の多いノードペアから評価していくこととする. そのため, 対象とするグラフ間で同一ラベルを持つノード数が異なる場合には, 余分のノードは評価対象としない. たとえば, 比較対象となる一方のグラフにのみ, あるノードラベルを持つノードが多数存在しても, その多くのノードは構造類似性の尺度には, 直接は反映しない考え方である. しかし, 一方にしか含まれないノードは, 共通するノードと関連を持つ部分グラフの情報によって間接的に反映される. まず, 任意のグラフ対に対して, 構造類似性を表す尺度であるノード間類似度を定義する. 今, 比較対象となるグラフを G1,G2 とする. 簡単のために, ノードラベルを一種類, ノード数については,G1 の方が G2 より多いとする. また, 対象となるグラフ集合全体から抽出されている部分グラフの数を J, それぞれの構造分布行列を M1,M2, さらに, 部分グラフ pi を構成するノード数を size(pi) とする. この時, グラフ G1 のノード x とグラフ G2 のノード y のノード間類似度 r12xy およびノード間相異値 d12xy を以下のように定義する.

$$r_{xy}^{12} = \sum_{j=1}^{J} \alpha_j \min(m_{xj}^1, m_{yj}^2)$$

$$1 \leq \alpha_j \leq size(p_j)$$

$$d_{xy}^{12} = \sum_{j=1}^{J} \beta_j |m_{xj}^1 - m_{yj}^2|$$

$$1 \leq \beta_j \leq \alpha_j$$

$$m_{ij}^1 = m^1(n_i^1, p_j) \in M_1$$

$$i = 1, 2, \cdots \quad , I_1$$

$$m_{ij}^2 = m^2(ni^2, p_j) \in M_2$$

$$i = 1, 2, \cdots \quad , I_2 \; I_2 \leq I_1$$

これらを用いて, ノード間類似度 S_{xy}^{12} を以下のように定義する.

$$S_{xy}^{12} = \frac{r_{xy}^{12}}{r_{xy}^{12} + d_{xy}^{12}} \tag{3.1}$$

3.2 群知能による群れの構成

Cl-GBI 法により求めた類似度を群れの引力 $|fa|$ のパラメータとして各 Particle に設定し, 振舞いの群れを作る. この群れが作られることを Mason にて実証する.

類似度による群れの引力

$$\left|\left|f_a\right|\right|_{ij} = \left|c_{1(i)} \cdot (T_{ij} - T_{(i)t})\right| \tag{3.2}$$

群れる力 ： fa_{ij}
$c_{1(i)}$ ： 係数
T_{ij} ： 求めた類似度 S_{xy}^{12}
任意の j に対する T_{ij}のしきい値 ： $T_{(i)t}$

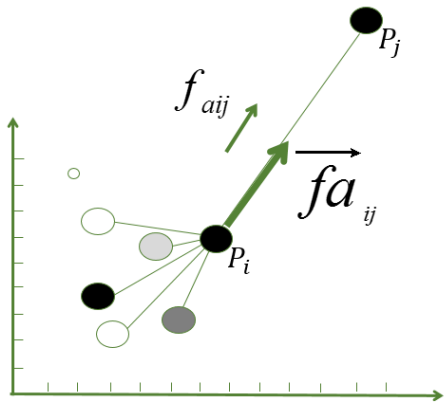


図 3.1: Particle(i) の Particle(j) に対する群れの引力

3.3 群れに生じる Covert Channel の分析

類似した振舞いで群れを作るようにしたが, この群れには Covert Channel を生じさせる振舞いも含まれている可能性がある. そのような振舞いを検出するために一度アクセス行列に展開し Covert Channel を分析する. しかし, アクセス行列上の振舞いの Covert Channel の計算量は膨大である. そこで Covert Channel の計算を分散処理をして行う.

reduce
A をまとめて Ak' をつくる.

(A → A0') → A1' ⋯→ Ak'

Covert Channel 分析
Ak' の中の covert channel Pk' を求める.
→ もし Pk' が無ければ,(O,S) については終了.
for Pk' 上の Oi,Sj について A'k-1 における covert channel を P'k-1 とする.

入力：アクセス行列 Ak=(S,O,E)
S：subject 集合
O：object 集合
E：有辺集合
s → o：s は o に write 可能
o → s：s は o を read 可能
出力：全ての x∈ O に対して，x を始点とする covert channel x → s が E に無いような x から s∈ S への有向道がもしあれば，そのうち最短の物を全て出力.
O' = x, S' = s ∈ S — x → s ∈ E
※幅優先探索用の初期化
V0 = O', Vs = S', Eh = x → s — s ∈ S'
while (Eh! = ϕ)
O' = o ∈ O Vo — s → o ∈ E for some s ∈ S'
S' = s ∈ S Vs — o → s ∈ E for some o ∈ O'
Eh = Eh ∪ o → s ∈ E — o ∈ O', s ∈ S' ∪ s → o ∈ E — s ∈ S', o ∈ O'
V0 = V0 ∪ O'
Vs = Vs ∪ S'

V0,Vs,Eh を出力　END(次の x へ)

第4章　結論

第5章　謝辞

卒業したい

2013年　11月
島田 琢士

参考文献

[1] 森住 哲也, 鈴木 一弘, 能登 正人, 木下 宏揚 :
”マルチエージェントに基づく遺伝的なアクセス行列制御” 電子情報通信学会 SITE 研究会

[2] 久保 直也
”群知能を適用したアクセス制御システム” 神奈川大学 木下研究室（2011）

[3] 鈴木久夫, 臼田啓介, 辻井重男, 森住哲也, 辻井重男 :
”後向き推論システムを用いた Covert Channel の検証”

[4] 安竹 有輝
”Cl-GBI 法によるふるまいのグラフの類似に基づく群れのモデルの提案” 神奈川大学 木下研究室（2012）

[5] 森住 哲也
”直観主義論理の意味論に基づく統合セキュリティモデル”
情報セキュリティ大学院大学

[6] 工藤 道治（日本 IBM）: 電子情報通信学会知識ベース 3 群 7 編 コンピュータネットワークセキュリティ 2 章 アクセス制御

http://www.ieice-hbkb.org/files/03/03gun_07hen_02.pdf

[7] ”データマイニング”

http://www.datamining.sakura.ne.jp/11haikei.html

[8] ”グラフマイニング”

http://www.eb.waseda.ac.jp/murata/research/graph

[9] ”探索アルゴリズム”
http://cis.k.hosei.ac.jp/ rhuang/Miccl/AI-0/2012-AI-0-L4-Jver.pdf

[10] ”MAS コミュニティ マルチエージェントとは”
http://mas.kke.co.jp/modules/tinyd4/index.php?id=3

[11] ”Mason”
http://cs.gmu.edu/˜eclab/projects/mason/

[12] 小松 充史
”Covert Channel 分析制御のためのに推論を導入した情報フィルタに関する研究” 2006 年度神奈川大学修士論文

[13] 戸田 瑛人, 森住 哲也, 鈴木 一弘, 木下 宏揚 :
”検索システムに組み込むセキュリティモデルに関して” 社団法人電子情報通信学会（2009）

[14] 内山 竜佑
”多様性を実現する群知能のふるまいのモデル” 神奈川大学 木下研究室（2011）

[15] 和田 貴久, 大野 博之, 稲積 宏誠 : ”部分構造情報を用いたグラフクラスタリング手法の検討” 人工知能学会全国大会（第 20 回）

[16] 和田 貴久, 大野 博之, 稲積 宏誠 : ”対象グラフ集合の特性を反映した構造類似性の提案 ” DBSJ Letters, Vol.6, No.1, pp.185-188

[17] 高林健登,Phu Chien Nguyen, 大原剛三, 元田浩, 鷲尾隆 :
”グラフ構造データからの特徴的なパターン抽出における制約に基づく探索制御” 2006 年度人工知能学会全国大会（第 20 回）論文集

[18] S. Luke, G. C. Balan, L. Panait, C. Cioffi-Revilla and S. Paus: “MASON: A Java multi-agent simulation library”, Proceedings

of Agent 2003 Conference on Challenges in Social Simulation (2003).

第6章　質疑応答

Q. この振舞い (を用いること) が, 現実のアクセス制御と対応する部分は何処か.

A.