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

木下研究室　鈴木　遼 (工学研究科 電気電子情報工学専攻 201070087)

あらまし

本稿では情報間の推論的依存関係をハイパーグラフを用いたモデル化し, それによる課題の提起する．そして依存関係と ACL のリストが与えられたグラフに対して安全な頂点着色が存在するかを目的としたアルゴリズムを提案する．このアルゴリズムはそもそも ACL 自体に問題があり, 情報を実際に read する, しないに関わらずどのようなリストを与えたとしても推論による情報漏えいが発生してしまうような ACL であるかを判定するアルゴリズムである．

1　はじめに

昨今, インターネット技術の発展に伴い, 人類が創出する情報量は爆発的に増加している．また, mixi や twitter などの SNS やクラウドサービスの台頭によりそれらの情報へのアクセスおよび解析が容易になった．このような情報爆発時代においては, 膨大な情報を高速に解析する攻撃に対応できる新しい情報漏えい対策が必要となる．

従来の主な情報漏えい対策の目的は, 個人情報や企業機密のような秘密情報がそのまま丸ごと漏えいしないようにすることであった．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, アルゴリズムなどが研究されている [3, 6, 7, 8].

しかし, 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もある．例えば, twitter は「今から○○へ行ってきます」「到着なう」のように, それ自体は秘密ではない情報を時刻情報と共に発信する場であるが, 過去にその人物が発信した全ての出発・到着情報と時刻を twitterAPI などを用いて自動取得しそれをもとに推論解析することで発信者の住所の詳細な部分的情報を抽出できる可能性がある．また, SNS では知人同士のつながりのようなコミュニティのリンクをたどることが可能なため, ある人物の本名や所属等の個人情報が特定されると, その情報を元に, その知人の本名や所属までもが推論によって暴かれてしまう可能性がある．

このような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, そのためには情報間の推論的依存関係を記述するモデルが必要である．そこで本論文では, ハイパーグラフを用いたモデル化を提案し, そのモデル上で考えるべき課題を提起する．

本論文の構成は次のとおりである．第 2 節では, covert channel と呼ばれる ”隠れた情報経路 ”による情報漏えいをアクセス制御リストを用いて監視するモデルを紹介し, そのモデルだけでは推論による情報漏えいを防げないことを示す．第 3 節では, グラフの頂点彩色について紹介したのち, 推論的依存関係の有向グラフによるモデル化を試みる．また, 推論による情報漏えい対策のためにアクセス制御リストの修正が必要な場合があることを示し, 今後の課題とする．第 4 節では, 有向グラフでは表現できない依存関係を有向ハイパーグラフを用いることでモデル化することを提案する．第 5 節では, 推論を考慮したグラフに対して安全なリスト着色をすることが出来ない ACL を判別するアルゴリズムについて提案する．

2　Covert Channel

2.1　ACL と Covert Channel

情報へのアクセスは主に情報リソースへの read と write によって実現される．そこで, 情報のアクセス許可を, read 可能か否か, write 可能か否かで表現した Access Control List(ACL) というものがある．ACL は Fig.1 の左図のように行列で表現できる．このような行列をアクセス行列と呼ぶ．この図では, O1, O2 が情報リソースを表し, S1 は O1 に対して read 可能で O2 に対して read と write 可能であり, S2 は O1 に対しては read も write も不可能で O2 に対して read 可能であることを意味している．一般には O をオブジェクト（客体）, S をサブジェクト（主体）と呼ぶ.

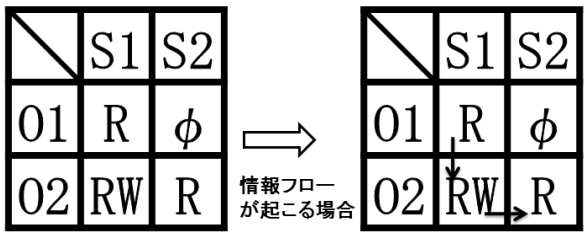

図 1: アクセス行列と covert channel

しかし, このアクセス行列をよく見ると, もし S1 が O1 を read し, その情報を O2 に write してしまうと, その O2 を S2 が read することによって O1 に記述されていた情報が S2 に漏えいしてしまう．したがってこのアクセス行列には O1→S1→O2→S2 という情報漏えい経路が隠されていたことになる．このような隠れた情報漏えい経路を ACL 上の covert channel（以下 covert channel）と呼ぶ [6] .

A hypergraph-based model and algorithm against information leakage by inference, Ryo SUZUKI(kinoshita laboratory,Graduate School of Electrical, Electronics and Information Engineering).

アクセス行列と covert channel は有向グラフによってモデル化できる．グラフとは頂点集合とそれらを結びつける辺の二つで構成され,「点とそれを結ぶ線」の「つながり方」に着目して問題を考えるためのデータ構造である．つながり方だけではなく, 辺でつながれた 2 頂点の順序を考慮してその有向性を矢印で表現したものを特に有向グラフと呼ぶ．その場合, 有向性のないグラフは無向グラフと呼ばれ, 有向グラフと区別される．

一般に, グラフは $G = (V, E)$ と記述される．ここで, V は頂点集合, E は V の 2 頂点対の集合である．$u, v \in V$, $(u, v) \in E$ のとき, G が無向グラフならば, $(v, u) \in E$ かつ $(u, v) = (v, u)$ であるが, G が有向グラフのときはそうとは限らない．ある頂点 $u \in V$ から, ある頂点 $v \in V$ へ辺をたどって到達可能な時, その到達経路を (u, v) 歩道といい, どの頂点も高々 1 回しかたどらない (u, v) 歩道を (u, v) 道と呼ぶ．

Fig.1 の左図のアクセス行列をグラフで表現したものを Fig.2 に示す．

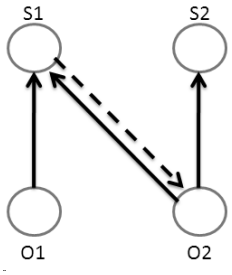

図 2: アクセス行列のグラフ表現

このグラフにおいて, 実線は read, 破線は write を意味する．グラフ G 上で, O_i から S_i への直接の矢印が無い, 即ち, $(O_i, S_i) \notin E$ にも関わらず, O_i から S_i への (O_i, S_i) 歩道が存在するときその歩道は covert channel を意味する．したがって, グラフによってモデル化することで ACL 上の covert channel 検知問題に対してグラフの歩道発見アルゴリズムを適用できるようになる [8]．また, グラフ化することで人間に対して視覚的に危険を訴えることができる．

2.2 推論による情報漏えい

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．推論攻撃への対策はデータベースのセキュリティ課題として研究されてきた．推論攻撃の一種として, 情報間の統計的な関連に注目した研究がある [3]．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．例えば, ある発言者が同じ地名, 若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間に何らかの関連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全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, そのためには情報間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無い．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．Fig.3 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 3: 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．このようなリストを作ることそれ自体も重要で難しい問題であるが, 本研究の目的は, このようなリストが与えられたときに, そのリストを解析し covert channel を検知するために必要なモデルを考察することである．

ACL は直接的にオブジェクトを読み書きできるかどうかだけを表したリストであるから, 仮に ACL 上では covert channel が無かったとしても, 推論によって ACL に反する情報アクセスが可能かもしれない．即ち, ACL と Fig.3 のような依存関係リストを合わせて初めて発覚する covert channel があり得るということである．次の節では, ACL と依存関係リストを同時にグラフ表現することを試みる．

3 有向グラフと頂点彩色によるモデル化

3.1 推論による頂点彩色のグラフ表現

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とし, 依存関係リスト上で $O_{i_1}, \cdots, O_{i_k} \in V$ から $O_j \in V$ が導出可能ならば有向辺 $(O_{i_1}, O_j), \cdots, (O_{i_k}, O_j)$ 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 可能ならば, $S_i \in L(O_j)$ とし O_j の色リストに S_i を加える．例として, Fig.3 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 Fig.4 に示す．

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 を塗るとしよう．例えば, Fig.4 において O_1, O_2, O_3 が全て S_1 で塗られたとする．このとき, 推論

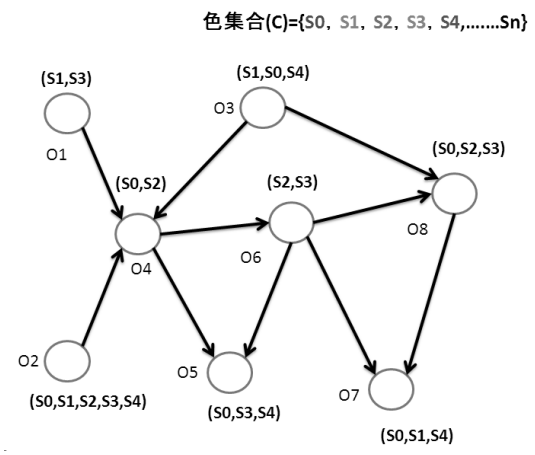

図 4: 推論的依存関係と ACL の有効グラフ表現

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 が全て 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 できないことになっているので, これは推論による情報漏えいを意味している．このとき, その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 を終点とする全ての有向辺 $(u_1, v), \ldots, (u_k, v)$ 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．

このモデル上で考えるべき課題は状況に応じて様々であると思われるが, 本論文では ACL そのものに含まれる問題を見直すための初歩的な課題を提起する．

3.2 ACL 修正問題

情報が実際に read されるタイミングには時差がある．例えば, 前節の例では, S_1 が O_1, O_2 を read したときに, その後 O_3 が読み込まれたときの危険性を検知し S_1 が O_3 を read する前に O_3 の ACL を修正し, $L(O_3)$ から S_1 を削除してしまえば良い．そのためには, オブジェクトの読み込み状況を常に監視し未来に起こり得る P 着色を随時計算しそれが推論に対して安全かどうかを判定するアルゴリズムが必要である．

また, そもそも最初から ACL に問題があるケースもある．Fig.5 は O_6 が S_2 に read されると, S_2 は O_3 を read できなくなる．なぜなら, 推論によって S_2 は O_8 を導出できてしまうからである．すると O_3 は S_1 しか読めない．S_1 が O_3 を read すると, 同様に, O_2 は S_0 しか読めなくなる．すると, O_4 も S_0 しか読めないので, もし S_0 が O_2 と O_4 を read すると, S_0 は推論によって O_5 を導出してしまう．よって, このグラフは推論に対して安全な P 着色が存在しないので, ACL を見直さなければならない．このようなグラフが与えられたときに, ACL を見直すべきかどうかを判定するためには, 安全な P 着色が存在するかどうかを判定するアルゴリズムが必要である．

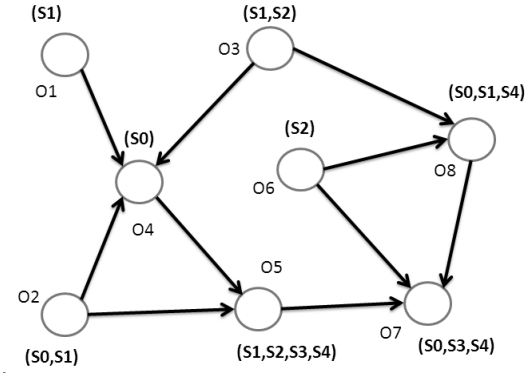

図 5: 色が塗れない object があるグラフ構造

4 有向ハイパーグラフによるモデル化

4.1 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．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 [1, 4]．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 の部分集合族である．

グラフとは異なり，ハイパーグラフは紙上に図示するのが困難である．そのため，グラフのような図解をされることは少なく，集合論の用語で表されることが多い．

4.2 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，空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．

グラフの頂点着色, 頂点彩色, P 着色, リスト着色, リスト彩色はいずれもハイパーグラフに拡張可能であるが, ハイパーグラフのリスト彩色についてはあまり研究がなされていない [2].

4.3 提案モデル

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とし, 依存関係リスト上で $O_{i_1}, \cdots, O_{i_k} \in V$ から $O_j \in V$ が導出可能ならば有向辺 $(\{O_{i_1}, \ldots, O_{i_k}\}, \{O_j\})$ 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 可能ならば, $S_i \in L(O_j)$ とし O_j の色リストに S_i を加える．ハイパーグラフを Fig.6 に示す.

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 を塗るとしよう．例えば, Fig.6 において O_1, O_2 が全て 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 が全て 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 できないことになっているので, これは推論による情報漏えいを意味している．このとき, その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 を終点とする全ての有向辺 S, T の始点集合 S が同一色で塗られているならば, T の頂点も同じ色で塗られなければならない.」とする．

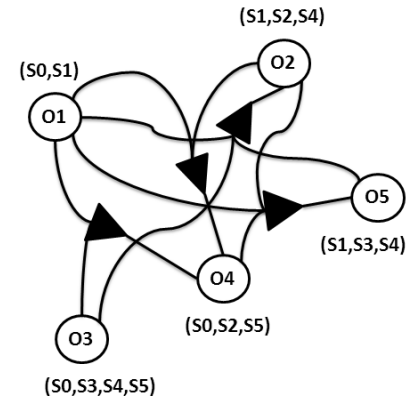

図 6: 有向ハイパーグラフを用いた提案データ構造

5 提案アルゴリズム

本稿で読み込み不可能のオブジェクトが存在しないリストの組み合わせを判別するのが目的のため, 効率の考慮はされていない. また, ここで述べる read の許可は実際に情報の read をすることを表しているわけではない。まず最初に, 与えられたグラフ全体のノードに含まれているリストを見て $L = 1$ のノードに含まれているサブジェクトの read を許可する．$L = 1$ のノードが複数ある場合, ノードを結ぶ線の数が少ないものから許可を与える．$L = 1$ のノードに read の許可を全て与えたときに推論によりサブジェクトを導出できるノードが存在するときそのノードには導出元のサブジェクトを与える．このとき既に推論によりノードの L に存在しないサブジェクトが与えられている場合．その ACL は不自由な ACL となる．その次に $L \geqq 2$ のノードは L の値が少ないものから順番に許可を与える．全てのノードに許可を与えたら推論により導出されるサブジェクトが存在するのであればそのノードは着色に沿ってサブジェクトを変更する．このようにしてサブジェクトの組み合わせを全て行い, 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 ACL は推論による情報漏えいを考慮したとき, 読み込めないオブジェクトが存在する不自由な ACL とする．

6 まとめ

本論文では, 推論による情報漏えいを未然に防ぐために必要な推論的依存関係をハイパーグラフによってモデル化することを提案し, 推論攻撃に対する安全性を定義した．アクセス制御リストモデルにおける covert channel 解析だけでは推論による情報漏えいを防げないことを示した．推論的依存関係をハイパーグラフで表現し, アクセス制御リストを色リストで表現することで, ハイパーグラフ上の頂点着色問題に帰結できる可能性を示した．推論による情報漏えい対策のためにアクセス制御リストの修正が必要な場合があることを示した．アクセス制御リストの修正を行う必要がある ACL を判別するためにグラフを用いて安全な頂点着色が存在するかを検出するアルゴリズムについて提案した．アクセス制御リストを効果的に修正するためには, 提案モデル上でどのような問題設定をすれば良いか, また, 本稿で提案した安全性の定義以外にも安全性は考えられるかなどの考察．またアルゴリズムを用いて安全な頂点着色の組み合わせの数などからどのようにして ACL の自由度を評価するかなどの考察が今後の課題である．

参考文献

[1] Giorgio Gallo, Giustino Longo, Sang Nguyen, Stefano Pallottino: ”DIRECTED HYPERGRAPHS AND APPLICATIONS”,(1992), http://citeseerx.ist.psu.edu/viewdoc /summary?doi=10.1.1.54.829.

[2] Penny Haxell, Jacques Verstracete: ”List coloring hypergraphs”, THE ELECTRONIC JOUNAL OF COMBIN The electronic journal of combinatorics 17, #R129, (2010).

[3] 木下宏揚: ”データベースのアクセス制御に関する考察”, Symposium on Cryptography and Information Security(SCIS1996),SCIS96-10D(1996).

[4] Andre Kundgen, Eric Mendelsohn, Vitaly Voloshin: Colouring planar mixed hypergraphs, The electronic journal of combinatorics 7, #R60 ,(2000).

[5] 河合博之, 柴田幸夫: ”関数に基づく集合分割と有向ハイパーグラフ”, IPSJ SIG Technical Report,p.41-46(2005).

[6] 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” , 博士論文, 博第 3 号, 情報セキュリティ大学院大学,(2008).

[7] 戸田瑛人, 市瀬浩, 鈴木一弘, 森住哲也, 木下宏揚: ”プッシュ型 Web システムに於ける情報フロー制御の提案”, 信学技報,vol.109,no.4,SITE2009-2,pp.51-56, (2009).

[8] 戸田瑛人, 森住哲也,(鈴木一弘), 木下宏揚: ”MapReduce を用いたクラウドの情報漏洩解析”, Symposium on Cryptography and Information Security(SCIS2010),3E4-2,(2010).