グループを用いた階層的アクセス制御方式

中盛友紀[†] 永瀬宏[†]

ファイルシステムにおけるアクセスコントロールを強制的に実施する有力な手法として，BLP（Bell and LaPadula）モデルを原則とした階層的運用がある．この運用では，情報フローを下位から上位に 1 方向化するため，機密性に優れたシステムの構築が可能である．また，多数のファイルを扱う状況のために，機械的に階層レベル情報を付加する SLA（Security Level Assignment）アルゴリズムが用意されている．しかし，BLP モデルを原則とした運用は，アクセス権限が階層的で，かつ単一の実世界を対象とする運用方式である．したがって，複数のグループが連携・競合し，各グループのアクセス権限が階層的な実世界の運用，すなわち複数の BLP モデルが存在する場合の運用が考慮されていない．そこで本稿では，グループが存在する場合の階層的アクセス制御手法について提案する．また，ネットワークの分野などで増加してきている動的なグループ形成に対応するために，SLA アルゴリズムを拡張したグループ化アルゴリズムの提案を行う．

Group Oriented Hierarchical Access Control System

Yuki Nakamori[†] and Hiroshi Nagase[†]

The hierarchical access control using the BLP (Bell and LaPadula) model is known as effective technique of the mandatory access control in a file system. Since this access control restricts information flow to one-way from low level to high level, the system of high confidentiality can be built. Moreover, SLA (Security Level Assihnment) algorithm is prepared in order to determine the hierarchical level information for many files. The access control design based on the BLP model operates only in a single computer with hierarchical access right. Therefore, use in the real world is not considered where the plural groups with hierarchical access rights cooperate or conflict, that is equivalent to plural BLP models. This paper proposes the hierarchical access control technique when the group exisit. An algorithm which extends the existing SLA algorithm is proposed to correspond to increasing demands of dynamic group formation issued in network community.

1. はじめに

階層的概念を導入したアクセス制御方式に BLP（Bell and LaPadula）モデルがある．このモデルでは，情報のフローを下位から上位に 1 方向化するため，UNIX などで主に用いられているアクセス制御リスト（ACL）よりも機密性に優れたシステムの構築が可能である．また，階層的なレベル情報を持つファイルシステムを設計するうえで，そのレベル情報を決定する手法に SLA（Security Level Assignment）アルゴリズムがある．システム管理者は，ユーザが望むアクセス要求（以下，要求と述べる）と，特定の情報フローを禁止する機密性条件を実現するために，このアルゴリズムを用いて機械的に階層レベルを割り当てる．

一方，ファイルのアクセス許可などを設定する場合，ユーザ単位で行うのは不便であり，ユーザをグループに分類して管理するのが一般的である．従来，グループは固定的であり，あらかじめ定められているものであった．しかし，現在，ネットワークの分野などで，グループが流動的に形成される事例が増えてきている．ナレッジマネージメントなどのグループごとの知識共有，グループウェアにおけるコンテンツの共有，P2P などである．これらの分野では，グループの再編，追加，削除などが頻繁に起こる．要求が先に存在し，それを満たすように適切なグループを形成する方法が求められている．

また，グループ内は通常フラットなものであり，グループに帰属する要素はすべて平等でなくてはならないという考え方も従来より存在する．しかし，グループ内に階層を導入することにより，グループ内要素の差別化を行い，きめ細かなセキュリティ設定が可能となる．

しかし，BLP モデルを原則とした階層的アクセス

† 金沢工業大学情報工学科
Department of Information and Computer Engineering, Kanazawa Institute of Technology

制御では，グループという概念が考慮されていない．また，SLA アルゴリズムは単一の計算機に対する権限設定の方法であり，要求に基づいてレベルの設定のみを行うもので，グループを形成することはできない．そこで本稿では，SLA アルゴリズムを用いたグループ化アルゴリズムを提案する．また，実社会において必ず存在しうるグループという概念を用いた際の階層的アクセス制御手法について提案する．

2. セキュリティモデル

機密情報の管理，保護，配布に関する規則や行動規範を形式的に定義したものをセキュリティモデルと呼ぶ．本章では，階層的アクセス制御に準拠したセキュリティモデルを述べる．

2.1 階層的アクセス制御

アクセスに関わるユーザやファイル，プログラムなどをエンティティと総称する．階層的アクセス制御では，レベルを定義し，すべてのエンティティをいずれかのレベルに割り付ける．そして，レベル間の順序関係に基づくアクセス規則を設け，それにより機密性を実現する．ここでの順序関係とは，ファイルなどのエンティティが持つ機密性の高低を，数学の概念である順序関係（たとえば数値の大小関係等）に対応させたものである．順序関係に基づくアクセス制御は，アクセスする任意の 2 者を考えたとき，この対比された順序関係だけでアクセス許可，不許可を決定する制御である．

アクセス規則については次の 2 通りの方法がある．第 1 はレベルの上下関係をアクセス権限の範囲に対応させる方法であり，権限の集合を考えた場合に，上位は下位の行使できる権限を積み上げた権限を与えられる構造を持つ．例としては，UNIX 系 OS におけるスーパユーザと一般ユーザの関係がこれに該当する．

第 2 はレベルの上下関係を情報フローの方向に対応させるものであり，権限の集合を考えた場合に，行使できる権限を選別し，権限を行使したときにある性質（情報のフローなど）が保証される構造を持つ．この構造を持つセキュリティモデルとして Bell and LaPadula モデル（以下 BLP モデルと略称）がある．

2.2 Bell and LaPadula（BLP）モデル

1973 年に David Bell と Leonard LaPadula が提案した Bell and LaPadula モデル（BLP モデル）[4] は，階層的アクセス制御を導入したセキュリティモデルである．メインフレームのタイムシェアリングシステムに懸念をいだいた米国空軍の要請で考案された．

BLP モデルでは，階層的なレベルと非階層的なカテゴリの 2 つの要素からなるラベル（サブジェクトのラベルは clearance，オブジェクトのラベルは classification と呼ばれる）が定義され，それらの間における半順序関係（あるいは束）に基づいて情報フロー制御が実施される．

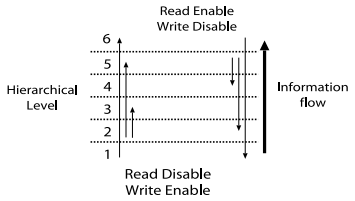


図 1 階層的アクセス制御の原則
Fig. 1 The principle of hierarchical access control.

階層的なレベル，すなわち全順序に限定した場合の BLP モデルは，いくつかのアクセス制御の規則に基づいて設計されている．具体的には，階層的なレベルが存在する中で，上位レベルに対しては書き込み，下位レベルに対しては読み出しのみ許可される．この規則により，情報フローが下位から上位へ 1 方向化され，階層レベルの上下関係で情報フローの有無が明確に判定できるという利点を持つ．この BLP モデルの情報フローにおける原則を**図 1** に示す．ここで図中の矢印は，情報フローの向きを示す．

この原則によって，UNIX オペレーティングシステムに見られる，意図しない情報のフローが，BLP モデルでは発生しにくい．UNIX が用いているアクセス制御手法はアクセス制御リスト（ACL）方式と呼ばれ，個々のファイルに対して細かなアクセス制御が可能である反面，与える権限を十分に検討しないと，予期しないファイルから情報がフローすることがある．

これに対して，BLP モデルは情報のフローの 1 方向性が存在するため，下位階層レベルに上位階層レベルの情報がフローすることはない．機密性設計の評価に関して，階層的セキュリティレベルを使った強制アクセス制御は，ACL などの任意アクセス制御よりも高いランク付けがされている[5]．

2.3 半順序 BLP モデルとの違い

全順序に限定した場合の BLP モデルでは，すべてのエンティティが比較可能となる．これに対し，半順序関係に基づく BLP モデルにおいては，関係を持たないエンティティどうしを比較不可能とすることができる．これはカテゴリの設定により実現される．1 つのエンティティは複数のカテゴリに帰属可能である．しかし，対象のエンティティが自分の属するカテゴリにすべて帰属していないとアクセスすることができない．そのため，階層レベルがアクセス条件を満たして

いても，カテゴリに関する条件が満たされていなければアクセスは不可能である．

半順序に基づいた BLP モデルでは，カテゴリの表現方法として集合を適用している．そのため，対象のエンティティがカテゴリに帰属しているかどうかは集合の包含関係により判断する．この際に集合どうしの演算が必要となる．アクセスは性急な処理を必要とするものであり，集合演算による遅延が生じるような運用では，快適なアクセス環境を得ることができない．

本稿が提案するグループを用いた階層的アクセス制御は，BLP モデルのサブクラスという位置づけにある．BLP モデルにおいて，半順序関係に基づくカテゴリに該当するものとして，グループを設定する．アクセスは同じグループに帰属しているエンティティに対してのみ可能とする．グループの表現にはローマ数字を用いる．これは一意な情報であり，集合よりも簡単に表現できる．また，1 つのエンティティは複数のグループに帰属することはできないという制限が加えられる．この制限により，エンティティのグループを表す識別子を見るだけで，帰属しているグループを瞬時に判断することができ，半順序 BLP モデルの集合演算よりもアクセスの可否を早く判断することができる利点を生じる．

各グループ内は全順序関係に基づく BLP モデルを用いて制御を行う．すなわち，全順序 BLP モデルが複数存在することになる．全順序に限定した BLP モデルは，2.2 節で述べたようにアクセス可能かどうかは階層レベルの上下により判断する．本稿が提案する制御方法は，グループの識別子と各グループごとの階層レベルを用いてアクセス制御を行う．これにより，全順序に限定した BLP モデルでは実現できなかった，関係を持たないエンティティどうしを比較不可能とすることが可能となる．関係を持つ，すなわち同じグループ内のエンティティどうしは各グループごとの階層レベルによって比較を行うことが可能である．

3. モデルの定義

SLA アルゴリズムでは，グラフ解析を用いて BLP モデルの原則に基づいた階層レベルを設定する．ここでは，本稿で扱うセキュリティモデルの各要素を定義する．

3.1 エンティティ

本稿におけるセキュリティモデルでは，コンピュータのアクセスに関わるユーザやファイル・プログラムなどをエンティティと総称する．また，各々のエンティティは，他のエンティティに対するアクセス要求を持つ．

3.2 アクセス要求

ユーザが指定できる各エンティティ間の要求は下記の 3 種類とする．

- データを流したい．
- データを流したくない．
- どちらでもよい．

原則として各エンティティ間はすべて，この 3 つの要求のうち，どれか 1 つに設定されていなければならない．しかしその場合，エンティティ数 n に対して，要求数は $n^2 - n$ と膨大になり，視覚的にも煩雑になる．この要求数は，n 個のエンティティから 2 つのエンティティを選ぶ組合せ ${}_nC_2$ より求められる．各エンティティ間の要求は双方向に指定できるため，${}_nC_2 \times 2 = n^2 - n$ となる．

そこで，本稿ではあらかじめデフォルトの要求を定めておく方法をとる．これは，要求が記述されていないエンティティ間の要求は，すべてこの定めた要求であると決めておくことである．たとえば，デフォルトの要求を「データを流したい」という要求とした場合は，要求が記述されていないエンティティ間はすべて，「データを流したい」という要求が存在するものと考える．

今回は，SLA アルゴリズムの定義に則り，エンティティ間に要求が記述されていない場合は，「どちらでもよい」という要求であるとする．「どちらでもよい」という要求は，データフローが許可されても禁止されてもよいという場合である．この要求が指定された場合，両エンティティ間はデータフローの許可・禁止のどちらに設定される可能性も持つ．これは他の 2 つの要求に比べ，ユーザが意識的に指定する可能性は少ないと考えられる．また，「データを流したい」「データを流したくない」という要求は，必ずデータフローの許可または禁止に設定されなければならない．SLA アルゴリズムはレベルの設定により，「どちらでもよい」とされるエンティティ間のデータフローを許可・禁止し，このユーザからの 2 つの要求に基づいたデータフローの許可または禁止を実現する．

SLA アルゴリズムでは，このユーザが指定する「データを流したい」という要求を処理要求，「データを流したくない」という要求を機密要求と表現している．

ユーザが各々希望する要求を指定した後，管理者がその要求をすべて満たすように SLA アルゴリズムでレベルを設定する．ここでいう管理者とはシステム設計時のセキュリティ設定における責任者を指す．管理者がレベルを設定した際に，ユーザが指定した要求の

帰結として,「データを流してはいけない」という要求が生じる．これは，ユーザの意図に基づく指定ではないため，本節のユーザが指定できる要求からは除外してある．

3.3 SLA アルゴリズム

SLA アルゴリズム[1),2)] は，グラフ解析に基づいたセキュリティレベル設定法である．データを流したいという処理要求とデータを流したくないという機密要求をグラフで表し，このグラフを解析して設計可能判定とレベル設定を行う．

前述のエンティティと要求を SLA アルゴリズムではグラフで表現する．エンティティはグラフのノード，アクセス要求はグラフの有効辺で表現する．ノードは丸印で表され，辺の接続によって他のエンティティとの情報フロー関係を表現する．ノードと辺の一例を，**図 2** に示す．

また，処理要求はグラフの実線の辺で表し，始点から終点への情報フローを示す．また，機密要求は破線の辺で表し，終点から始点への情報フローを禁止する．

ユーザから寄せられたすべての要求をつなぎ，グラフとしたものを統合要求グラフと呼ぶ．統合要求グラフの例を**図 3** に示す．

SLA アルゴリズムは，統合要求グラフの根の部分から処理を行い，各ノードを一意に識別するためのラベルとユーザの要求に矛盾しない階層レベルの最小値を割り付ける．レベル割付けはラベルの昇順で行われる．

3.4 グループ

本稿では，互いに利害関係にあり，要求でつながれているエンティティ群を 1 つのグループとする．また，異なったグループとの間では，原則としてエンティティどうしの情報フローは行われない．ただし，例外として，明示的に別グループに対する要求が存在するものについては，これを認めることとする．すなわち，

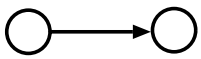


図 2 エンティティとアクセス要求の基本モデル
Fig. 2 Basic model of entity and access demand.

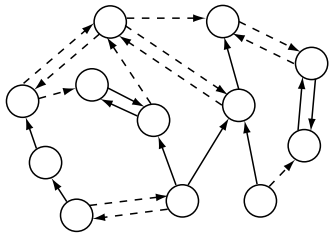


図 3 統合要求グラフ
Fig. 3 Integrated demand graph.

SLA アルゴリズムの権限設定が適用されるのはグループ内のみであり，異なるグループのエンティティ間に要求が設定されていない場合でも，そのエンティティ間は「どちらでもよい」と見なすことはできない．したがって，各グループは独立性の高いエンティティ群で構成される．

また，本アルゴリズムでは最小のグループ数でグループを形成する．グループ数は決まっておらず，増やすことも可能である．

3.5 グループ制御方法

本節では，グループ化アルゴリズムにより，グループが定められた後，階層的アクセス制御によってこれを制御する方法を示す．

グループが決定した後は，各グループごとに階層レベルを設定する．グループ内はこの階層レベルに基づき，アクセスが制御される．

異なったグループ間では原則として情報のフローが行われないが，例外として明示的に要求が示されている場合は，ケーパビリティを用いてこれを個別に制御する．ケーパビリティとは，そのエンティティがアクセス可能なエンティティを定めたものである．SLA アルゴリズムで定義する権限が最大権限であるのに対し，ケーパビリティで定義する権限は最少権限となる．ケーパビリティが設定されたエンティティ間のみデータフローを認め，それ以外のデータフローは禁止する．

異なるグループに属している 2 つのエンティティ間に処理要求があるときは，この 2 つのエンティティ間のみの情報フローを認める．すべてのグループ間のケーパビリティの設定が終了したら，グループ間のアクセスを設定したエンティティすべてに対して，機密要求を充足しているかの確認を行う．この際，同グループ間から 2 つ以上のエンティティにアクセスが設定されている場合は，レベルに応じてそのエンティティ間に要求を入れる必要がある．

別のグループのエンティティどうしは原則として情報フローが生じることはなく，また例外として情報フローを認める場合も通信を行いたいエンティティ間のみの権限を認めるケーパビリティを用いるので，その他のエンティティ間で情報フローが生じることはない．

4. グループ化アルゴリズム

本章では，与えられたエンティティと要求をもとに，グループを形成するためのアルゴリズムを述べる．

本アルゴリズムを実行する前に，管理者は情報をフローしたいという処理要求と，情報をフローしたくないという機密要求を収集し，図 3 のような統合要求グ

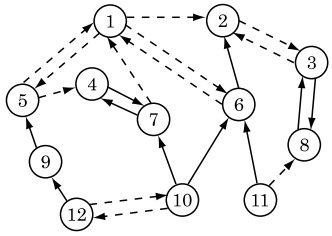
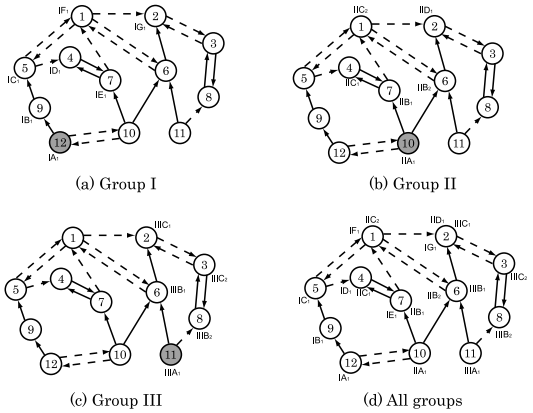


図 4 エンティティ番号付加後の統合要求グラフ
Fig. 4 Integrated demand graph with entity number.

表 1 相互機密要求表
Table 1 Mutual confidentiality demand table.

①	⑤
①	⑥
②	③
⑩	⑫



(a) Group I (b) Group II (c) Group III (d) All groups

図 5 ラベル設定後の統合要求グラフ
Fig. 5 Integrated requirement graph after label assignment.

ラフを作成する．図中，丸はエンティティを表す．実線の矢印は処理要求を表し，始点から終点に向かう情報フローを示す．また破線の矢印は機密要求を表し，終点から始点への情報フローを禁止する．

以下，図 3 を例として，本アルゴリズムの手順を述べる．

4.1 ラベル付け手順

グループ化アルゴリズムは，まず初めにエンティティへのラベル設定を行う．以下，その手順を示す．

(G1) エンティティを識別するためのエンティティ番号を各エンティティに任意に付加する．
図 3 の統合要求グラフにエンティティ番号を付加したものを**図 4** に示す．

(G2) エンティティどうしが相互に機密要求でつながれている組合せを検出し，相互機密要求表を作成して，エンティティ番号の組合せを記入する．
この相互機密要求表の例を**表 1** に示す．

(G3) (G2) で検出されたエンティティの中で，相互の機密要求以外，入力がないものを検出する．
例では，エンティティ⑩と⑫が検出される．

(G4) (G2) で検出されていないエンティティの中で，入力がないエンティティを検出する．
例では，エンティティ⑪が検出される．

(G5) (G3) と (G4) で検出されたエンティティを出発点とし，SLA アルゴリズムのラベルの設定を実行する．この手順は付録 A.1 のアルゴリズムに従う．ただし，これ以降の処理において，相互機密要求表に存在する組合せのエンティティに関しては，矢印が存在しないものとして扱う．また，複数の出発点がある場合は，それぞれラベルの前に I，II，III，$\cdots$ という記号を付加して，ラベルの設定を個別に行う．ここで記号 I，II，III，$\cdots$ はグループを表す．

(G6) 各々設定したラベルを 1 つにまとめる．

図 5 の (a)〜(c) はそれぞれエンティティ⑫，⑩，⑪を出発点としてラベルを設定した統合要求グラフである．また，(d) は別々に設定したラベルを 1 つの統合要求グラフにまとめたものである．

(G5) において，ラベルの設定を行う際，相互機密要求表に存在するエンティティ間の矢印を存在しないものとして扱っている．これは，双方向の機密要求を満たすためには，接続されたエンティティどうしを別のグループとして設定しなければならないからである．そのため双方向の機密要求をたどるパスはつながっていないものとしてラベル設定の処理を行う必要がある．

4.2 グループ化手順

この節では，複数のラベルを付けられたエンティティについて，ラベルを 1 つに決める手順を示す．この処理により，エンティティをグループに分けることができる．

本アルゴリズムにおいて，ラベルを 1 つ選ぶ操作が何度か出てくる．このときに，選んだラベルで固定する場合と選んだラベルとは別に予備として 1 つ以上のラベルを記憶しておく場合がある．前者のときのラベルを選ぶ操作を「ラベルを確定する」，またその状態を「ラベルが確定されている」と表現する．後者のときのラベルを選ぶ操作を「ラベルを選択する」，またその状態を「ラベルが選択されている」と表現する．なお，本文中でも分かりやすいようにこの操作および状態を表す際には，括弧（例：【確定】）で囲むこととする．

(G7) 複数のラベルが設定されているエンティティを列挙し，複数ラベル表を作成して，エンティティ番号，

ラベルとともにこの表に記入する．
複数ラベル表の例を**表 2** に示す．
(G8) 設定したラベルを相互機密要求表の対応したエンティティ番号の欄に記入する．
ラベルを記入した相互機密要求表の例を**表 3** に示す．
(G9) 統合要求グラフの全エンティティにおいて，ラベルが 1 つしか付けられていないエンティティは，ラベルを【確定】する．
(G10) 相互機密要求表において，対になっているエンティティ両方のラベルが【確定】または【選択】されている場合は，その組合せを表から削除する．ただし，その組合せのラベルが互いに同じグループのものである場合は，このエンティティの組合せの相互機密要求を実現できないため，失敗した旨を管理者へ通知し，アルゴリズムを停止する．ただし，このような場合は，6.5 節で述べる方法で対処できる．アルゴリズムが冗長になるため，この部分は別に記述している．
例では，表 3 におけるエンティティ⑩と⑫の組合せが，それぞれラベル IIA_1 と IA_1 に確定しており，ラベルのグループも異なるため，この組合せを相互機密要求表から削除する．
(G11) 相互機密要求表において，対になっているエンティティがお互いに同じグループのラベルを持っていない場合は，その組合せを表から削除する．
(G12) 相互機密要求表の中で，どちらか一方のエンティティのみ，ラベルが【確定】または【選択】されているものを選ぶ．該当する組合せが複数ある場合は，もう片方のエンティティに設定されているラベルの数が少ない方を選択する．
例では，まずエンティティ①と⑤の組合せが選ばれる．
(G13) ラベルが【確定】または【選択】されていないエンティティに，対のエンティティの選ばれているラベルと同じグループのラベルがある場合は，そのラベルを削除する．
例では，エンティティ⑤のラベルが IC_1 に確定しているため，エンティティ①のラベル IF_1 が削除される．
(G14) ラベルを削除した場合は，下記に示す処理を行う．

・ラベル削除の処理

(D1) ラベルを削除したエンティティを現エンティティとする．
例では，エンティティ①が現エンティティとなる．
(D2) 複数ラベル表から現エンティティの同ラベルを削除する．
(D3) 相互機密要求表に記述されている現エンティティすべてについて，同ラベルを消去する．
例では，エンティティ①と⑥の組合せにおけるエンティティ①のラベル IF_1 が削除される．
(D4) 現エンティティから出ている要求があれば，その出力先のエンティティをチェックし，ラベルを複数持っている場合は，現エンティティにおいて削除したラベルと同グループのラベルを削除する．ただし，出力先のエンティティが同グループのラベルを持つ別のエンティティからの入力を持っている場合は，この処理は行わない．
例では，現エンティティの出力先はエンティティ②であり，ラベルを複数持っている．エンティティ②への入力エンティティである⑥は，現在削除対象であるグループ I のラベルを持っていないので，エンティティ②のラベル IG_1 が削除される．
(D5) (D4) の処理によって，ラベルを削除したエンティティがある場合は，そのエンティティを現エンティティとし，(D2) からの処理を繰り返す．
例では，エンティティ②のラベルを削除したので，このエンティティを現エンティティとし，(D2) から処理を行う．

(G15) ラベルの削除の結果，そのエンティティのラベルが 1 つに定まった場合は，そのラベルを【確定】する．
例では，エンティティ①のラベル IF_1 が削除され，IIC_4 のみになるので，ラベルは IIC_4 に確定する．
(G16) ラベルが【確定】した場合は，下記に示す処理を行う．

・ラベルが確定した際の処理

(D6) ラベルが【確定】したエンティティを現エンティティとする．
例では，エンティティ①が現エンティティとなる．
(D7) 現エンティティへ入っている要求があれば，そ

表 2 複数ラベル表
Table 2 Plural label table.

Entity	Label		
①	IF_1	IIC_4	
②	IG_1	IID_1	$IIIC_1$
④	ID_1	IIC_1	
⑥	IIB_2	$IIIB_1$	
⑦	IE_1	IIB_1	

表 3 ラベル記入後の相互機密要求表
Table 3 Mutual confidentiality demand table with label.

Entity	Label			Entity	Label	
①	IF_1	IIC_4		⑤	IC_1	
①	IF_1	IIC_4		⑥	IIB_2	$IIIB_1$
②	IG_1	IID_1	$IIIC_1$	③	$IIIC_2$	
⑩	IIA_1			⑫	IA_1	

の要求を列挙する．

(D8) 各要求の根のエンティティをチェックし，現エンティティと同じグループのラベルを持つエンティティを選択する．同じグループのラベルを持つエンティティが複数ある場合は，ラベル数が少ないエンティティを優先する．選択するエンティティがない場合や，選択したエンティティのラベルがすでに【確定】または【選択】されている場合は，ここでこのラベルが確定した際の処理を停止する．

例では，エンティティ⑦が選択される．

(D9) 現エンティティと同じグループのラベルを，選択したエンティティのラベルとして【選択】する．ただし，選択したラベル以外のグループのラベルも削除せずに記憶しておく．記憶してあるラベルは括弧で囲む．ここで，相互機密要求表に選択したエンティティが存在する場合は，対になっているエンティティを参照し，ラベルが【確定】または【選択】されていないかを確認する．対になっているエンティティのラベルが【確定】または【選択】されている場合は，選択したラベルと対になっているエンティティのラベルとを比較し，同じグループであったならば，ラベルの【選択】処理を取り消したうえで (D8) の処理に戻り，今回選択したエンティティを除き，再びエンティティの選択を行う．

例では，エンティティ⑦のラベルが IIB_1 に選択される．ラベル IE_1 も記憶しておく．エンティティ⑦は，相互機密要求表に存在しないため，次の処理へ進む．

(D10) (D9) の処理によって，ラベルを【選択】したエンティティがある場合は，そのエンティティを現エンティティとし，(D7) からの処理を繰り返す．

例では，エンティティ⑦を現エンティティとし，(D7) からの処理を行う．

(G17) (G10) からの手順を繰り返す．この結果，処理を行うエンティティが存在しなくなったら (G18) の処理を行う．

(G18) 相互機密要求表において，複数のラベルが付けられているエンティティを選び，下記のラベル選択ルールに従い，ラベルを【選択】する．ラベル選択ルールは上から順に適用して行く．この際，選択されなかったラベルは削除せずに記憶しておく．

・ラベル選択ルール

(1) 着目しているエンティティへの入力先のエンティティのラベルが【確定】または【選択】されていない場合は，そちらのエンティティのラベル選択を先に行う．

(2) そのラベルと同じグループのラベルに【確定】または【選択】されているエンティティからの入力がないラベルは，予備のラベルとし，記憶する．

(3) 着目しているエンティティに複数の入力があり，それが処理要求と機密要求の場合は，処理要求の根のエンティティと同グループのラベルを選択する．

(4) ラベルにおける階層を表すアルファベット記号が小さいものを選択する．

(5) 階層を表すアルファベット記号が最も小さいものが複数あった場合はどれを選択してもよい．

(G19) (G18) で選択したエンティティと対で記憶されているエンティティのラベルがまだ【確定】または【選択】されていなければ，選択したラベルとは異なるグループのラベルかつラベル選択のルールに基づいたラベルを【選択】する．

(G20) (G18) から (G19) の処理を繰り返す．その結果，相互機密要求表ですべてのエンティティのラベルが 1 つに【確定】または【選択】された時点で，処理の繰返しを終了する．

(G20) までの処理が終わった時点での相互機密要求表を**表 4** に示す．なお，説明のため，表から削除された組合せについても記述している．

(G21) 複数ラベル表において，ラベルが【確定】または【選択】されていないエンティティについて，ラベル選択ルールに従い，ラベルを【選択】する．

例では，エンティティ④のラベルがラベル選択ルールの (3) に基づき，ラベル IIC_1 に定められる．

(G22) (G21) の処理を繰り返す．その結果，複数ラベル表ですべてのエンティティのラベルが【確定】または【選択】された時点で，処理の繰返しを終了する．このときの複数ラベル表を**表 5** に示す．

ラベル削除の処理を行うのは，そのラベルを経るパスが途中でなくなるためである．ラベル設定を行う際は，矢印で接続されているエンティティを幅優先探索の探索順序で設定していく．そのため，あるエンティティが存在しなくなると，その先に接続されているエ

表 4 グループ決定後の相互機密要求表
Table 4 Mutual confidentiality demand table after deciding the group.

Entity	Label	Entity	Label
①	IIC_2	⑤	IC_1
①	IIC_2	⑥	$IIIB_1$
②	IID_1	③	$IIIC_2$
⑩	IIA_1	⑫	IA_1

表 5 グループ決定後の複数ラベル表
Table 5 Plural label table after deciding the group.

Entity	Label	
①	IIC_2	
②	IID_1	
④	(ID_1)	IIC_1
⑥	$IIIB_1$	
⑦	(IE_1)	IIB_1

ンティティにはラベルは設定されない．エンティティのラベルが削除されるということは，ラベル設定においてそのエンティティが存在しなくなることと同じであり，そのエンティティのみを経ている上位階層のエンティティにラベルが設定されていてはならない．よって，ラベルを削除したエンティティにのみ接続されている上位階層のエンティティに対してもラベルの削除を行う必要がある．

ラベルが確定した際の処理は，エンティティが孤立しないために行う．ラベルが確定した場合，同グループのラベルを持つエンティティからの入力がなくては，そのエンティティは同グループから孤立してしまう．これを防ぐためにこの処理を行う．

5. グループにおける階層的アクセス制御

本章では，グループ分けされた統合要求グラフにおいて，階層的アクセス制御を実施するための手順を述べる

5.1 レベルの設定

(G23) すべてのエンティティのラベルが【確定】または【選択】されたら，グループごとに SLA アルゴリズムのレベル設定を実行し，レベルの設定を行う．この手順は付録 A.2 のアルゴリズムに従う．このとき，相互機密要求および別グループのエンティティに対する要求に関しては，矢印がないものとして扱う．

(G24) 正しくレベルが設定されれば，ここでアルゴリズムは停止する．

アルゴリズムが終了した際の統合要求グラフを**図 6** に示す．

レベル設定の際に，別グループのエンティティに対する矢印を存在しないものとして扱っている．これは，別グループのエンティティにそのグループのレベル情報を付加しないためである．レベル設定は幅優先探索を用いて行うため，矢印でつながれたエンティティすべてにレベルを付与してしまう．レベルが設定されると別グループのエンティティまで比較可能となってしまう．そのため，別グループのエンティティについては，矢印がないものとしてレベルを設定する．

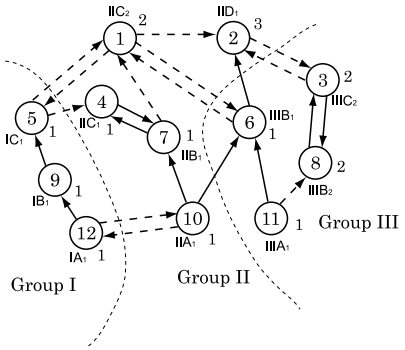


図 6 レベル設定後の統合要求グラフ
Fig. 6 Integrated demand graph after level assignment.

レベル設定でアルゴリズムが失敗し，アルゴリズムが停止する場合がある．これは，機密要求と処理要求が混在するループが存在する場合である．その場合は，失敗したエンティティを含むループを検出し，どれか 1 つのエンティティを別グループに変更することでグループ化することは可能である．この際に，記憶しておいたラベルを用いる．ただし，グループ間通信を設定する際に失敗するため，グループ間の要求も実現したい場合は，要求を見直す必要がある．しかし，要求を見直す際には，このループを構成する要求の中から，どれか 1 つの要求を断念する必要がある．すなわち，このループを構成する要求すべてを同時に満たすことは不可能である．

5.2 グループ間通信の設定

グループをまたがって要求が存在する場合は，その要求に従い，各々のエンティティ間をケーパビリティにより制御し，情報フローを可能にする．

(G25) グループをまたがる要求とその要求に関わるエンティティのみを抽出し，統合要求グラフを作成する．この際，同グループのエンティティを抽出する場合には，下位レベルのエンティティから上位レベルのエンティティに対して，処理要求と機密要求を付加する．同グループ内の同じレベルのエンティティに関しては，1 つにまとめて抽出する．

(G26) 抽出されたエンティティの処理要求に対して，ケーパビリティを設定する．

図 6 のように設定された統合要求グラフに対して，グループ間通信を設定したものを**図 7** に示す．エンティティには識別のためにグループ番号とレベルを負荷したラベルを与えている．

(G27) この統合要求グラフ全体の到達行列を求め，設定に矛盾がないことを確認する．到達行列[6] とは，処理要求行列のサイズを n とすると，ブール演算によっ

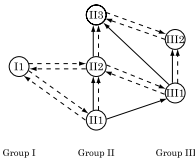
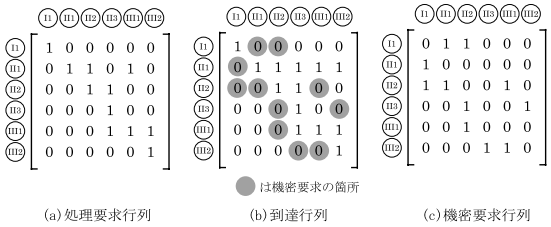


図 7 グループ間通信の設定
Fig. 7 Setting of the access between groups.



(a) 処理要求行列 (b) 到達行列 (c) 機密要求行列

図 8 到達行列による設定の確認
Fig. 8 Confirmation of setting by arrival matrix.

てこれを $n-1$ 乗した行列のことであり，これは処理要求によって生じる直接的なフローと間接的なフローをすべて表現したものである．したがって，到達行列と機密要求行列の両方で同時に “1” となる成分が存在しなければ，設定に矛盾がないといえる．

図 8 にそれぞれの行列の例を示す．

(G25) において，同グループ内の同じレベルのエンティティを 1 つにまとめて抽出している．これは，同じレベルのエンティティ間ではすべて情報フローが可能であるため，到達行列を求める際には，1 つのエンティティであると考えてよい．この作業により，計算時間も短縮できる．

また，同グループ内の下位レベルのエンティティから上位レベルのエンティティに対して，機密要求だけでなく処理要求も記入している．これは同じグループのエンティティ間の機密要求は，同じ方向の処理要求も持つためである．各エンティティにレベルが設定されているため，同グループ間ではすべてのエンティティが比較可能となる．そのため，下位レベルのエンティティから上位レベルのエンティティへは情報フローの可能性が生じる．

6. 評　　価

6.1 従来の方法との違い

従来の SLA アルゴリズムでは，双方向の機密要求があるような場合，これを矛盾として検出し，設定が不可能であることをシステム管理者に通知して，アルゴリズムは停止する．システム管理者はこの通知を受け，矛盾を含む要求を提示したユーザに対して要求を再検討するよう要請する必要があった．

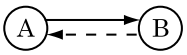


図 9 矛盾の基本モデル
Fig. 9 Basic model of inconsistent demand.

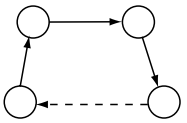


図 10 一般的な矛盾の例
Fig. 10 An example of demand including inconsistency.

矛盾は，機密要求を含む閉路のことである．本モデルにおいて，矛盾は 2 種類に分類される．1 つは要求自体の誤りによって生成される矛盾である．もう 1 つは，要求自体に誤りはないが，従来の SLA アルゴリズムでは実現不可能なものである．

前者の矛盾の代表的な例を**図 9** に示す．この例では，機密要求によりエンティティA からエンティティB に対する情報フローを禁じているにもかかわらず，処理要求によりエンティティA からエンティティB への情報フローを確保しようとしている．これは，要求自体に誤りがある．

ただし，統合要求グラフにおいてこの例そのものが存在する可能性は低い．ユーザが要求をシステム管理者へ通知する時点で，ユーザ自身が誤りに気が付く可能性が高いためである．要求自体の誤りによる矛盾は，複数のエンティティを経て構成されると考えられる．このような例を**図 10** に示す．本アルゴリズムの手順 (G24) のレベル設定アルゴリズムにおいて失敗するのは，このような矛盾が含まれている場合である．

双方向の機密要求は後者の矛盾にあたり，2 つのエンティティ間における情報フローを禁止したい場合に設定される．これは，要求としては想定されるものであり，要求自体に誤りはない．

矛盾を含む要求が与えられた場合において，各要求を恒常的に確保しておく必要がないケースに着目し，矛盾を含む要求を実現する時間分割を用いた階層的アクセス権設定法がある[3]．これは，矛盾部分を時間分割し，実行順序を考慮することで，機密要求に違反せずに矛盾部分の処理要求を満たすことを可能とする．

この時間分割を用いた階層的アクセス権設定法は，要求自体に誤りがある矛盾を実現するための方法である．すなわち，本アルゴリズムでは実現できず，手順 (G24) で失敗するような例を実現可能とする．しかし，矛盾部分における処理要求を時間分割し，実行順序の変更を行うものであり，矛盾部分に処理要求が含まれ

ていない双方向の機密要求には適用できない．

双方向の機密要求は，実際に想定される要求であるにもかかわらず，従来の SLA アルゴリズムおよび時間分割を用いた階層的アクセス権設定法では設定不可能であった．本アルゴリズムではこの点を改善し，双方向の機密要求が存在した場合は，その 2 つのエンティティを別グループとすることで，情報フローを禁止し，このような要求の実現を可能とした．

また，本アルゴリズムレベル設定において失敗するような統合要求グラフについても，本稿が提案するアルゴリズムと時間分割を用いた階層的アクセス権設定法を併用することにより，2 種類の矛盾を同時に実現することが可能となり，本アルゴリズムでは実現できない統合要求グラフにも適用できる可能性を持つ．

6.2 アルゴリズムの正当性

本アルゴリズムの正当性について，次の 3 点を証明する．

- 要求の充足性
  - 見つかった解は，処理要求，機密要求を満たしている．
- 探索の網羅性
  - 解を見落としていることはない．
- アルゴリズムの停止性
  - アルゴリズムは必ず停止する．

①要求の充足性

本アルゴリズムが処理要求，機密要求を満たさない解を導き出すと仮定する．このような解が導出されるのは，SLA アルゴリズムのレベル設定，またはグループ間通信の設定のときである．

SLA アルゴリズムのレベル設定により導かれる処理要求，機密要求を満たさない解とは，矢印の根のエンティティの方が矢印の先のエンティティよりもレベルが高く設定された解である．SLA アルゴリズムのレベル設定では，着目するエンティティに対して，入力された各矢印の根のエンティティと同じか 1 高いレベルを求め，その中から最も大きな値をレベルとして設定する．そのため，矢印の根のエンティティが矢印の先のエンティティよりもレベルが高く設定されることはない．

グループ間通信の設定により導かれる処理要求，機密要求を満たさない解とは，設定したケーパビリティが，グループ間の機密要求を満たしていない場合である．本アルゴリズムでは，ケーパビリティを設定した後，グループ間にまたがる要求とその要求に関わるエンティティについての到達行列を求め，機密要求を満たしているかの確認を行う．機密要求を満たしていない場合は，グループ間の要求を見直すようユーザに通告して，アルゴリズムは停止する．よって，処理要求，機密要求を満たさない解を正しい解だとして導き出すことはない．

以上のことから，本アルゴリズムが導き出した解は，処理要求，機密要求を満たしているといえる．

②探索の網羅性

本アルゴリズムのラベル設定は，幅優先探索法を用いて行われる．幅優先探索法は連結グラフであれば，すべてのエンティティを巡回することができるので，ラベルの付け忘れは発生しない．

レベル設定は幅優先探索法で設定されたラベルの順序にしたがって行われる．すべてのエンティティにラベルが付けられているため，レベル設定の漏れは生じない．

また，本アルゴリズムでグループ化を行う際は，まずエンティティがとりうるすべてのラベルを列挙する．ここから相互機密要求に矛盾したラベルを削除していくという方法を用いているため，解の見落としはないといえる．

③アルゴリズムの停止性

アルゴリズムの停止性について，ラベル設定，レベル設定，グループ化に分けて示す．

まず，ラベル設定においてアルゴリズムが停止しないと仮定する．ラベル設定では，幅優先探索法を用いてエンティティが巡回され，エンティティ数も有限であるため，ループを含まない統合要求グラフにおいては，アルゴリズムの停止に支障はない．そこで，アルゴリズムが停止しないためには，ループを含む統合要求グラフにおいて，ループの検出に失敗する必要がある．SLA アルゴリズムではラベル付けを行う際に，ラベルが未設定のエンティティからの入力をループを構成する可能性があるとして検出する．ラベルが未設定のエンティティはその時点では，未探索ということである．幅優先探索では，同一階層にあるエンティティの探索がすべて終わった時点において，未探索のエンティティは探索済みのエンティティよりも上位の階層に位置する．そのため，ラベルが未設定であるエンティティから入力される要求のみがループを構成する可能性を持つ．よって，ループの可能性がわずかでもある要求は必ず列挙されるため，ループの検出漏れは発生しない．このことから，アルゴリズムが停止しないために必要な条件を満たすことができず，仮定が矛盾すると説明できる．

レベル設定では，ラベルの順序に従い処理を行う．ラベルは有限であり，階層を表すアルファベットと同

一階層内のエンティティ識別子から構成され，どちらも昇順である．そのため，アルゴリズムの停止に支障がないことは自明である．

グループ化において，アルゴリズムが停止しないと仮定する．グループ化の際にアルゴリズムが停止しないのは，各エンティティのラベルが 1 つに定まっていない場合である．本アルゴリズムでは，エンティティが複数のラベルを持っている際には必ずラベルの選択を行う．このことより，仮定が成立しないことが分かる．

6.3 アルゴリズムの計算量

本アルゴリズムの計算量を決定する要因を以下に示す．

- 双方向の機密要求の検出
- エンティティのラベル設定
- グループ化
- 各グループ内でのレベル設定
- グループ間通信設定

ノードの数を n，辺の数を e として，それぞれの計算量を求める．

各エンティティにおける要求は表によって表される．双方向の機密要求の検出は，表に記載されている各々のエンティティの要求を見て検出するため，$O(n)$ となる．

エンティティのラベル設定は，SLA アルゴリズムを使用して行う．SLA アルゴリズムは幅優先探索を基にしているため，探索の計算量（辺をたどる回数）は通常の幅優先探索と同様に $O(n+e)$ となる．

グループ化における計算量は，ラベルの削除後や決定後に辺をたどる回数に依存する．この処理は，辺をたどった先のエンティティが複数のラベルを持たなくなるまで行われる．たどる順序は幅優先探索を用いるので，この処理 1 回の計算量は $O(n_1+e_1)$ となる．グループ化全体の計算量は，$O((n_1+n_2+\cdots+n_k)+(e_1+e_2+\cdots+e_k))$ となる．$n_1+n_2+\cdots+n_k$ は，最大でも グループ数 × エンティティ数 であり，$e_1+e_2+\cdots+e_k$ もそれに付随する辺の数なので，これは $O(n+e)$ であるといえる．

各グループ内でのレベル設定もラベル設定と同様，SLA アルゴリズムを用いているため，計算量は $O(n+e)$ となる．

グループ間通信の設定における計算量は，到達行列を作成する時間に依存する．到達行列に要する計算量は $O(m^4)$ となる．ここで，m はグループ間通信に関わるエンティティのみであり，同グループ内の同じレベルのエンティティをまとめて処理を行うので，最大でも各グループ内のレベル数を全グループ分加算したものであり，エンティティ n に対して非常に少ない数である．そのため，計算量としては除外して考えることができる．なお，シミュレーションプログラムを用いて算出したエンティティ数 n に対する m の値，および到達行列の作成に要する時間を 6.4 節の**表 6** に示した．

表 6 シミュレーションによる計算時間測定
Table 6 Simulation result of required calculating time.

エンティティ数	所要時間（秒）	到達行列の要素数	到達行列の作成に要する時間（秒）
100	0.018	7	0.000017
500	0.073	16	0.000051
1,000	0.257	25	0.000113

以上のことから，本アルゴリズムの計算量は $O(n+e)$ であるといえる．

6.4 計 算 時 間

エンティティ数に対する本アルゴリズムの計算時間を調べるため，グループ化アルゴリズムを実施するシミュレーションプログラムを作成し，実行時間を測定する．本シミュレーションプログラムは IBM のワークステーション Intellistation（CPU: Intel Pentium 4 3.40 GHz）に Red Hat Enterprise Linux 3 をインストールし，その環境上で C 言語を用いて作成した．統合要求グラフのエンティティ数を 100，500，1,000 と変化させ，グループ化に要する時間の変化を測定する．

n 個のエンティティに対して，グループ化アルゴリズムを実行し，その所用時間を測定した．また，グループ間通信を設定するために抽出するエンティティの数 m も求めた．各個数のエンティティ数で構成される異なったデータを用いて 3 回ずつ計測し，その平均を算出して表 6 にまとめた．ただし，使用したデータにおける要求はランダムに与えるものとする．

この m の値は，同時に到達行列の要素数の値でもある．ここで算出した m 個のエンティティに対して，到達行列を作成し，その際に要した時間も測定した．

表 6 から，実際の所用時間がほぼ 6.3 節で算出した計算量どおりに推移していることが分かる．今回のように，グループ数があらかじめ決められていない場合に，総当たりでグループ化を行うと，通常 $O(n!)$ の計算量を必要とする．$O(n!)$ は，NP 問題であり，n が大きくなると計算は不可能となる．本アルゴリズムでは，n と e の多項式時間でグループ化を行うことを可能とする．

また，グループ間通信を設定する際に抽出するエンティティの数 m が，エンティティ数に比較すると小さい値であることが分かる．エンティティ数が大きく

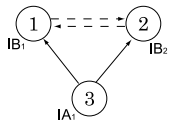


図 11 特殊な事例
Fig. 11 Special case.

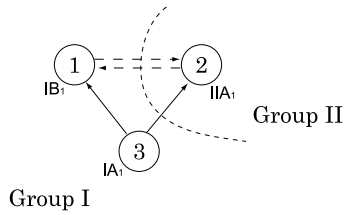


図 12 グループ決定後の状態
Fig. 12 State after deciding the group.

なるにともない，m の割合は小さくなっていることも分かる．

到達行列の作成に要する計算時間については，アルゴリズムの実行時間と比較して，非常に小さな値となった．到達行列を求めるための計算量は理論上は $O(m^4)$ であるが，ブール演算であることより，工夫次第で計算量を大幅に短縮することが可能である．

6.5 特殊な例

要求によっては，双方向に機密要求がある 2 つのエンティティに同グループのラベルしか付加されず，グループを分けることができない場合が生じる．この場合は，グループを増やすことで対応する．本アルゴリズムの (G10) の処理で失敗するような場合は，この方法を用いてグループ化を行う．このような事例の簡単な例を**図 11** に示す．

図 11 の統合要求グラフでは，出発点がエンティティ③のみであり，すべてのエンティティに同グループのラベルが付加される．しかし，エンティティ①とエンティティ②の間には双方向に機密要求が設定されており，この 2 つのエンティティを同グループにすることはできない．

選択可能なラベルがなく，双方向に機密要求が存在するエンティティどうしを別々のグループに分けることができない場合は，新たにグループを作成し，片方のエンティティにそのグループを割り当てる．

例では，グループ I とは別に，新しくグループ II を作成し，エンティティ②をこのグループの出発点とする．もし，エンティティ②に出力があれば，その接続先のエンティティにも同グループのラベルを設定する．そして，エンティティ②以降のグループ I のラベルを消去する．エンティティ③からエンティティ②への要求はグループ間のフローをケーパビリティにより設定することで実現する．グループ設定後の例を**図 12** に示す．このように，グループを増やすことにより，双方向の機密要求を満たすことが可能となる．

7. むすび

処理要求と機密要求が存在する状況において，それらの要求を矛盾なく実現する階層レベルを割り付ける方法として，SLA アルゴリズムが知られている．また，SLA アルゴリズムをベースとする拡張 SLA アルゴリズムは自由度と呼ばれる階層レベルの変更可能範囲を求めることにより，階層レベルの割付けに柔軟性を持たせ，より現実の業務に適用しやすいレベル設定を行うことができる．

これに対して本稿では，SLA アルゴリズムを拡張したエンティティのグループ化方法について提案した．従来，SLA アルゴリズムでは実現不可能であった双方向の機密要求を利用し，互いにフローを行いたくないエンティティどうしを別グループとすることにより，要求を満たすグループを形成できる可能性を生じる．グループ化アルゴリズムでは，双方向の機密要求を検出し，これを基にエンティティをグループに分け，また影響の及ぶ他のエンティティのラベルを適切に設定する．そして，グループをまたがった要求に対しては，ケーパビリティを用いて制御を行うことでこれを実現することを可能にした．

以上のグループ化アルゴリズムの実施により，要求が与えられた際にそれらをすべて満たすことのできるグループを形成し，それを基にした階層的運用が可能であることを示した．

謝辞 本研究の一部は文部科学省ハイテク・リサーチ・センター整備事業（平成 17 年度〜平成 21 年度）による私学助成を得て行われた．

参考文献

1) 永瀬 宏，井上清一，四七秀貴：階層的セキュリティレベルの自由度を用いたアクセス権設定法，情報処理学会論文誌，Vol.41, No.8, pp.2255–2263 (2000).
2) Araki, T., Morizumi, T., Nagase, H., Takenaka, T. and Yamashita, K.: Security Level Assignment By Graph Analysis, *IEICE Trans. Issues on Cryptography and Information Security*, Vol.74, No.8, pp.2166–2175 (1991).
3) 井上清一，中盛友紀，下川徳之，永瀬 宏：時間分割を用いた階層的アクセス権設定法，情報処理学会論文誌，Vol.44, No.8, pp.2031–2041 (2003).
4) Bell, D.E. and LaPadula, L.J.: Secure Com-

puter System: Unified Exposition and Multics Interpretation, Technical Report, MTR-2997 (available as NTIS AD-A023588), MITRE Corp. (1976).
5) Denning D.E.R.: *Cryptography and Data Security*, Addison-Wesley, Reading, Massachusetts (1982). 上園忠弘, 小嶋　格, 奥島晶子 (訳): 暗号とデータセキュリティ, 培風館 (1988).
6) Morizumi, T., Nagase, H., Takenaka, T. and Yamashita, K.: An Evaluation of Security Requirements Based on the Capability Model, *IEICE Trans.*, Vol.E-74, No.8, pp.2160–2165 (1991).

付　　録

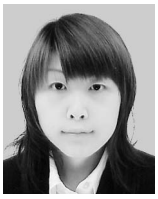
A.1 ラベル設定手順

(S1) 出発点となるエンティティに A_1 というラベルをふる．ここで記号 A はグラフ上の階層を意味し，数値 1 は同一階層内のエンティティ識別子である．
(S2) ラベルの付け終わっているエンティティの中で，以下に述べる (S3), (S4) がまだ実施されておらず，かつ一番ラベルの値の低いエンティティを選択する．これを現エンティティとする．
(S3) 現エンティティに入っている矢印を列挙し，ラベルがまだ付けられていないエンティティからの入力を検出する．検出された場合はループチェック用の記憶領域へ，現エンティティと現エンティティへつながるエンティティを，対で記憶する．ただし，この両エンティティをつなぐ矢印が機密要求であった場合は，このループは実現不可能であるため，アルゴリズムが (S3) で失敗した旨をユーザに通告してこのラベル設定アルゴリズムを停止する．
(S4) 現エンティティから出る要求の矢印を列挙し，順に現エンティティの記号を 1 つ進めた記号を用いて（たとえば B_1, B_2, $\cdots$, のように）矢印の先の各エンティティにラベル付けする．すでにラベル付けがされていたエンティティの場合はループチェック用の記憶領域を参照し，現在処理を行っているエンティティの組合せと同一のものが発見された場合は，ループと判断して記号の上書きは行わない．そうでなければ，新たな記号により上書きする．
(S5) (S2) からの手順を繰り返す．この結果，未処理のエンティティがなくなった時点でラベル設定アルゴリズムは終了する．

A.2 レベル設定手順

(S6) エンティティに割り振ったラベルを階層を示す記号順に列挙する．また，各エンティティを，要求に対応する矢印でつなぐ．
(S7) A_1 エンティティに対して，1 というレベルを設定する．
(S8) 未処理の階層を記号の小さな順で選択する．A 階層は (S7) で必ず 1 というレベルが設定され処理済みであるため，最初に B 階層が未処理かつ最小の記号となり，まず B 階層が選択される．これを現階層とする．
(S9) 現階層から，未処理のエンティティを数値の小さな順で選択する．この選択されたエンティティを，現エンティティとする．
たとえば，図 6 の GroupIII のような場合，B 階層に 2 つのエンティティが存在し，どちらも未処理であるため，1 という数値（B_1 エンティティ）が選択される．
(S10) 現エンティティへ入っている各要求を列挙する．
(S11) 各々の要求に対して，処理要求であれば矢印の根のエンティティと同じ値を，機密要求であれば矢印の根のエンティティよりも 1 高い値を求める．ただし，ループチェック用の記憶領域に存在する組合せのエンティティに関しては，矢印が存在しないものとして扱う．
(S12) 求まった値の中で，一番高い値のものを，現エンティティのレベルとして設定する．
(S13) 現階層に，未処理のエンティティがある場合は，引き続いて (S9) から (S12) 間の処理を行う．
(S14) 現階層のすべてのエンティティに対する処理が終わったとき，未処理の階層がある場合は，引き続いて (S8) から (S13) 間の処理を行う．
(S15) ループチェック用の記憶領域に存在するエンティティのレベルをチェックする．対で記憶されているエンティティのレベルがこの時点で同一でないならば，このループは実現不可能であるため，アルゴリズムが (S15) で失敗した旨をユーザに通告してこのレベル設定アルゴリズムを停止する．(S15) の処理は，対で記憶されているエンティティすべてに対して行う．
(S16) ループチェック (S3) および (S15) をすべてパスした場合は，レベル設定アルゴリズムは終了する．

(平成 17 年 6 月 20 日受付)
(平成 18 年 3 月 2 日採録)



中盛 友紀（学生会員）

平成 16 年金沢工業大学工学部情報工学科卒業．同大学大学院修士課程に在学中．現在，セキュリティファイルシステムの研究に従事．

永瀬 宏（正会員）

昭和 49 年慶應義塾大学理工学部計測工学科卒業．昭和 54 年同大学大学院工学研究科博士課程（電気工学）修了．同年日本電信電話公社に入社，ATR 通信システム研究所主任研究員．平成 5 年金沢工業大学工学部情報工学科教授就任．知的創造システム専攻教授を経て，現在，情報工学科教授．情報セキュリティ，計算機アーキテクチャの研究に従事．