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

永瀬 宏[†] 井上清一[†] 四七秀貴[††]

全順序関係を持つセキュリティレベル設定アルゴリズムの研究については，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 SLA アルゴリズム（Security Level Assignment）が発表されている．これは，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それぞれ独立したグラフで表し，それらを重ね合わせたグラフを解析して設計可能判定性とレベル設定を行うアルゴリズムである．SLA で設定されるレベルは，設定が可能なレベルの許容値の中で最小値となることから，レベルを上げても要求実現には矛盾が起こらない場合がある．そこで本論文ではレベルの許容値の中で最大値のレベルを決定する逆 SLA アルゴリズムを新たに提案し，これをもとにレベルの変更可能な範囲を導出する．

Access Right Assignment Method which Based on Margin of Hierarchical Security Level

Hiroshi Nagase,[†] Kiyokazu Inoue[†] and Hideki Shina[††]

For the research of assigning totally ordered security levels, the SLA (Security Level Assignment) algorithm was proposed from graph theoretic approach. In the SLA algorithm, processing requirement to flow data and confidentiality requirement to inhibit data flow, among entities, are firstly expressed in two independent graphs. Then, two graphs are mixed to an integrated requirement graph. By analyzing the integrated requirement graph, design capability is evaluated, and security levels are assigned to entities. Since security levels, assigned by the SLA algorithm, take minimal value within the allowable range of levels, security levels of appropriate entities may be raised without contradicting to original requirements. Hence, this paper newly proposes inverse SLA algorithm to find the maximal allowable security levels, and to determine freedom of security levels assigned to entities.

1. はじめに

情報化社会の進展にともない，情報処理システムのセキュリティ技術の重要性も増大している．セキュリティとはシステムの安全性を高め，故意や偶然の不正からシステムを保護することと一般に定義されている，情報処理システムのセキュリティ機能設計は，技術面のみならず，運用面，管理面等の多様な要素が組み合わされる．情報処理システムのセキュリティ機能設計とは，いずれの要素にせよ，セキュリティ要求（設計要求）からセキュリティ機能（設計機能）を設計することである．このうち本論文では，主に技術面に着目し，システムに対するセキュリティ上の要求（機密漏洩防止要求，改ざん防止要求等）を満足するように，脅威の防止・検出といったセキュリティ機能を設計することについて検討する．

セキュリティを考慮したシステムを構築する場合，設計対象となるシステムが小規模ならば，そのセキュリティ設計も容易であり，人間が手動で行うことも可能である．しかし，大規模な情報処理システムの場合，人手による設計は非常に困難となり，設計の誤り等の危険性も考えられる．そこで，セキュリティ設計を容易かつ正確に行うための “セキュリティ設計支援” が望まれ研究がなされている．

セキュリティ設計に関連した研究として，アクセス制御等，技術的側面に着目した研究としては，情報漏洩を防止するためのアクセス制御方式[1)~4)]や，セキュリティレベル設定アルゴリズムの研究[5)~7)]が存在する．アクセス制御技術とは，主体（動作を起こすもの，ユーザ，プログラム等），客体（主体により作用を受けるもの，ファイル，ディレクトリ等）をエンティティとし，各エンティティ間に定められたアクセス規定に従ってエンティティ間のアクセスを許可・禁止する技

† 金沢工業大学情報工学科
Department of Information and Computer Engneering, Kanazawa Institute of Technology
†† NTT ネットワークサービスシステム研究所
NTT Network Service Systems Laboratories

術である．アクセス制御方式に関する研究については，主（客）体すべてに階層的なセキュリティレベルを割り当て，このレベルに応じたアクセスが制御される “強制アクセス制御” や，アクセスの主（客）体がアクセスを許可するためのアクセス権限を保持し，その権限の存在によりアクセスが制御される “任意アクセス制御” に分類できる．

歴史的には 1960 年代の Multics が階層的セキュリティレベルを使った強制アクセス制御方式であったのに対して，商用 OS はケーパビリティに基づく任意アクセス制御が主体である．これは運用の柔軟性を優先させたからであるが，結果としてセキュリティ上の対策が十分であるとはいいきれない．1999 年 6 月に国際規格として承認され，同年 12 月に発行された ISO/IEC15408 に含まれる CC（Common Criteria）によると，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，任意アクセス制御よりも高いランク付けがされている[8)~10)]（なお，セキュリティレベルという用語について以後，了解性を損なわない範囲でレベルと略称する場合がある．また同様に，階層的セキュリティレベルを使った強制アクセス制御を，階層的アクセス制御と略記する）．

階層的アクセス制御は，Multics が発表されてから，1973 年，Bell と LaPadula によって初めてマルチレベルセキュリティ方針の数学的モデルとして，厳密に形式化された[3)]．このモデルは開発者の名前を取って Bell and LaPadula Model（BLP Model）と呼ばれている．

CC のもととなった TCSEC（Trusted Computer System Evaluation Criteria）の基準では，Multics は B2 のセキュリティ評価を受けているが，商用システムとしてはあまり普及していない．その理由として階層が多過ぎる（0〜63 の 64 階層を持つ）ため，システム管理者がレベルを設定することが非常に困難であるという問題が考えられる．この問題については階層数の低減とともに，アクセス要求や機密要求に合致するレベル設定を，システム管理者に代わって支援する機能が必要となる．

機械的なセキュリティレベル設定アルゴリズムの研究については，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 SLA アルゴリズム（Security Level Assignment）が発表されている[5)]．これは，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それぞれ独立したグラフで表し，それらを重ね合わせたグラフを解析して設計可能判定性とレベル設定を行うアルゴリズムである．

しかし，SLA は設定が可能なレベルの許容値の中で最小値のレベルを設定する．これは，SLA アルゴリズムはノードの探索がグラフの入口より始まり，幅優先探索で見つかったノード順にレベルを設定する仕組みを持つためである．そこで本論文では，最小のレベルを選択しなくても要求実現には矛盾が起こらない場合があることに着目する．レベルの変更可能な範囲を導き出すことにより，より設定範囲に幅を持たせることが可能となる．この結果，エンティティには複数のレベルを割り付けることができ，利用可能なレベルを状況に応じて選択することが可能となる．これは，現実の場面において，当初設定されていたレベル区分を，再度複数に区分しなおす状況に対応できるレベルの割付け方法となる．具体的には，当初学生というレベルに位置した人物が，後に定義された学生兼研究員というレベルに移される場合等，グラフを再計算することなく自由度の範囲でレベルを変更し，新しい区分に対応させることが可能である．

本論文では 2 章で階層制御の概念となる BLP モデルについて述べ，3 章で SLA ならびに逆 SLA アルゴリズムについて説明し，最後に 3 章で求められたレベルをもとに，設定可能なレベルの全組合せを導出する拡張 SLA アルゴリズムについて説明する．

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” という．以下では，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

階層的アクセス制御では，セキュリティレベルを定義し，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，ファイル等の主体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）に対比させたものである．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．順序関係には全順序と半順序があるが，本論文では第 1 ステップとして階層への適用が容易である全順序を取り上げる．なお，半順序関係は階層レベルの表現法も含めて抜本的な検討が必要であり，これらは今後の課題としたい．

アクセス規則については次の 2 通りの方法が現在利用されている．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 におけるスーパーユーザと一般ユーザの関係がこれに該当する．UNIX ではレベルはこの 2 つの区分しか存在しないが，スーパーユーザは下位ユーザである一般ユーザの権限をすべて持ち，さらにスーパーユーザのみが行使できる権限を，それに積み重ねて持つ．

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（情報のフロー等）が保証される構造を持つ．上位から下位への read，下位から上位への write を認め，その逆は許可しない．これは Bell and LaPadula モデル（以下 BLP モデルと略称）として知られており，情報は下位から上位に 1 方向にしかフローしないという特徴を有する．

2.2 Bell and LaPadula（BLP）モデル

Bell and LaPadula モデル（BLP モデル）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り，2.1 節に述べたように情報フローが 1 方向的である[4]．このため，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利点がある．以下，本モデルを詳細に述べる．

まず，コンピュータのアクセスに関わるユーザやファイル・プログラム等をエンティティと総称する．また，プログラムやユーザ等のアクセスを行うエンティティを主体（Subject），ファイル等のアクセスされるエンティティを客体（Object）とする．

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．同レベル間においては，書き込み，読み出しともに許可される．

たとえば，主体 s が客体 o を読み出すときは，主体 s のレベルは客体 o のレベルよりも上位でなければならない．つまり $L(s) \geq L(o)$ である必要がある（$L(x)$ はエンティティ x の階層レベルを表す）．主体 s が客体 o に書き込むときは，主体 s のレベルは客体 o のレベルよりも下位でなければならない．つまり $L(s) \leq L(o)$ である必要がある．

したがって，客体の読み出しは上位セキュリティレベルの主体に限定され，これは BLP の情報フローに準じたフローであるため，情報リークは発生しない．また，下位セキュリティレベルの主体は上位セキュリティレベルの客体を読み出せないため，下位セキュリティレベルの主体への情報リークは発生しない．この性質により，上位レベルのデータの機密性が確保される．また，この性質を “＊性質（スタープロパティ）” と呼ぶ．＊性質を図式化したものを**図 1** に示す．

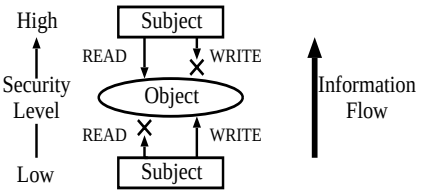

図 1 Bell and LaPadula モデル
Fig. 1 Bell and LaPadula model.

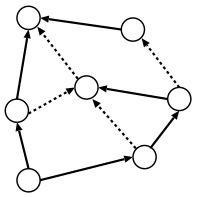

図 2 統合要求グラフ
Fig. 2 Integrated requirement graph.

3. レベル設定の自由度

機密要求と処理要求をともに満たすセキュリティレベル設定法として，SLA アルゴリズムが知られている．SLA はグラフ解析上では深さ優先探索となっている．本章では SLA と同一の機能を幅優先探索で行う，新 SLA アルゴリズムを提案する．一般的に深さ優先探索と幅優先探索のオーダーは，すべての根を網羅するという点においては同じ値である．しかし，すべての根を網羅しながらグラフの要求を満たすレベルを各エンティティ割り付けるという処理の場合，1 つのエンティティに対して複数回の探索が行われる場合がある．そこで，新 SLA アルゴリズムは処理を幅優先探索を用いたラベル付けとレベル設定の 2 つに分けることで，SLA のレベル設定を逆順に実行する逆 SLA アルゴリズムが容易に実現されることを示す．最後にこれらのレベル設定法をもとに，設定可能なレベルの最大許容範囲（これを自由度と呼ぶ）を求める．

3.1 新 SLA アルゴリズム

SLA ではまず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含む統合要求グラフを作成する．例を**図 2** に示す．図中，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．アルゴリズムは

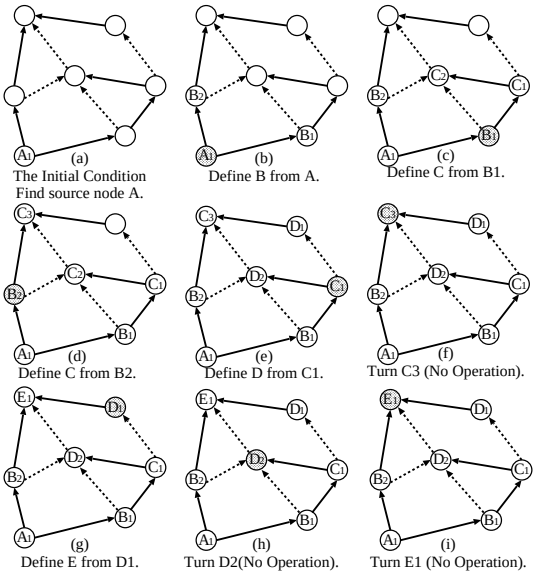

図 3 SLA レベル設定における準備
Fig. 3 Preparation for level assignment by SLA.

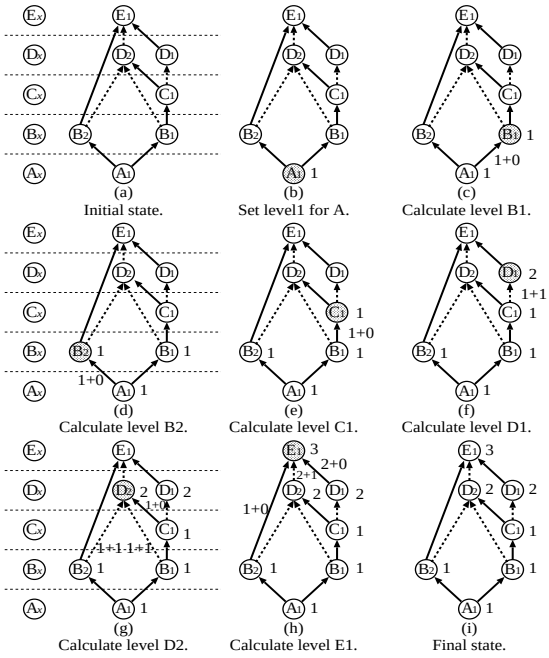

図 4 SLA によるレベル設定
Fig. 4 Level assignment by SLA.

エンティティへのラベル付けとレベル設定の 2 段階からなる．

● エンティティへのラベル付け

（S1）処理要求，機密要求の入力辺がない入次数 0 のエンティティを見つけ，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（**図 3** (a)）．

（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．例では，最初の時点では（S1）によってラベル付けされた A_1 のみが存在し，その A_1 に対してはまだ（S3),（S4）が実行されていないため，A_1 が現エンティティとして選択される．

（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処理を停止する．

（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（図 3 (b)）．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，終了する（図 3 (c)〜(i)）．

● レベル設定

（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（**図 4** (a)）．

（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（図 4 (b)）．

（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，例では最初に 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．例では最初，B 階層に 2 つのエンティティが存在し，どちらもまだ未処理であるため，1 という数値（B_1 エンティティ）が選択される．この選択されたエンティティを，現エンティティとする．

（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処理を停止する．（S15）の処理は，対で記憶されているエンティティのすべてに対して行う．
（S16）ループのチェック（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．

新 SLA アルゴリズムの実行によって決定されたラベルの例を図 3 に，レベル設定の例を図 4 に示す．この図では辺に付随する不等号は省略している．円の中の記号はエンティティに割り振った記号，円の横の数字はレベルを示す[4]．

従来の SLA アルゴリズムの実現においては，深さ優先探索が行われている．その手法を，本論文でこれから述べる逆 SLA および拡張 SLA にも適用した場合，深さ優先探索で生じる試行錯誤の部分（エンティティに探索済みのチェックを付けたり外したりしながら全ノードをたどる）は，各々のアルゴリズムでそのつど行われる．一方，3.1 節に述べた新 SLA アルゴリズムでは，エンティティの探索順序の決定は，最初にラベル付け処理という形で切り離し，レベル付けはラベルの順序に従って一括して行うことができる．この処理により，新 SLA アルゴリズムを用いた場合は，逆 SLA ならびに拡張 SLA アルゴリズムでエンティティの探索順序を改めて求める必要はなく，ラベル付け処理で決定された順に従って，容易に逆 SLA アルゴリズムを実行することができる．以上の理由から新 SLA アルゴリズムにおいては，（S1）から（S5）のラベル付け処理を新たに追加している．

深さ優先探索において試行錯誤の生じる理由は，次のように説明できる．深さ優先探索では，まずグラフの終点までをあるルートで探索してレベルを割り付ける．しかし，引き続き別のルートの探索を継続したとき，すでにレベルが設定済みのエンティティに再度，高いレベルを上書きする場合がある．これは，後にたどったルートが先にたどったルートよりも多くの機密要求辺を含み，かつ，後にたどったルートが最終的に先にたどったルートに入る場合に発生する．例として図 2 に示すグラフでは，深さ優先探索を用いてレベル付けを行った場合，C_2 および D_3 の 2 つのエンティティにおいて，一度割り付けたレベルが後に別のレベルで上書きされる．これは，終点までを A_1，B_1，C_1，D_2，E_1 の順序でたどるルートよりも，後に A_1，B_1，D_2，E_1 の順序でたどるルートに含まれる機密要求辺の数が多いためである．

3.2 Reversed SLA（逆 SLA）

処理要求，機密要求が与えられたとき各エンティティにおいて，3.3 節に自由度として述べる複数のレベル候補が存在した場合に，SLA アルゴリズムはその候補の中から最小値のレベルを割り当てる．そこで，グラフ探索の方向やレベル割当ての計算をまったく正反対に行うと，後述する原理により，すべてのエンティティに対してグラフの要求を満たす範囲で最大のレベルを割当てることができる．

逆 SLA アルゴリズムの具体的な動作を以下に示す．
（R1）逆 SLA アルゴリズムにおいて未処理であり，記号および数値の最も大きいエンティティを選択し，以下の処理を行う．これは，SLA アルゴリズムにおいてたどったエンティティ順と逆の順序にたどるということであり，例では E_1 エンティティが始点となり，レベルは 3 である．
（R2）現エンティティから出ている各要求を列挙する．
（R3）各々の要求に対して，処理要求であれば矢印の先のエンティティと同じ値を，機密要求であれば矢印の先のエンティティよりも 1 小さい値を求める．
（R4）求まった値の中で一番小さい値のものを，現エンティティのレベルとして設定する．
（R5）すべてのエンティティについて（R1）から（R4）の処理を行う．

以上より，最大レベルを導き出すことが可能となる．このアルゴリズムを “逆 SLA（Reversed SLA）” と呼ぶ．図 4 (i) で得られた有向グラフにおいて逆 SLA を実行し，最大レベルを設定する過程を**図 5** に示す．

SLA ならびに逆 SLA アルゴリズムでは，それぞれのエンティティに次のようにレベルが割り当てられた（$L(A_1, B_1) = (1, 2)$ という表記は，エンティティ A_1 のレベルが 1，エンティティ B_1 のレベルが 2 である

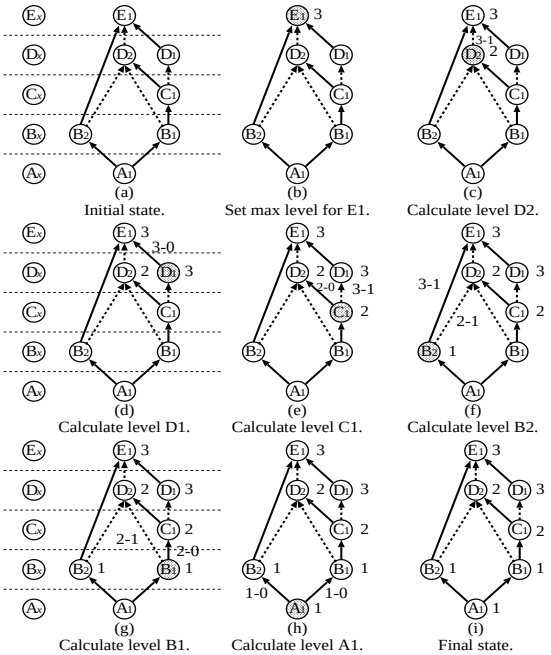

図 5 逆 SLA によるレベル設定
Fig. 5 Level assignment by reversed SLA.

ことを示す).

$$SLA \quad L(A_1, B_1, B_2, C_1, D_1, D_2, E_1) = (1, 1, 1, 1, 2, 2, 3)$$

$$\text{逆 } SLA \quad L(A_1, B_1, B_2, C_1, D_1, D_2, E_1) = (1, 1, 1, 2, 3, 2, 3)$$

したがって，2 つのアルゴリズムではエンティティ C_1, D_1 のレベルに差が生じることが分かる．

ここで，まず SLA アルゴリズムについて，すべてのエンティティにグラフの要求を満たす範囲でつねに最小値が割り付けられる理由を示す．

図 4 のアルゴリズムで述べたように，新 SLA ではレベルの設定が下位層のエンティティから順に段階的に行われる．このレベル設定中のグラフを模式的に**図 6** に示す．いま図 6 において，すでにレベル決定を終えたエンティティ $E_{ik}, E_{ik+1}, \cdots, E_{il}$ の集合を S_i とする．また，集合 S_i と有向枝で直接に接続される，レベル未設定のエンティティ $E_{i+1\,1}, \cdots, E_{i+1\,j}, \cdots, E_{i+1\,m}$ の集合を S_{i+1} とする．このとき新 SLA では集合 S_{i+1} に含まれる任意のエンティティ，たとえば $E_{i+1\,j}$ のセキュリティレベル $L(E_{i+1\,j})$ を次の式で求める．

$$L(E_{i+1\,j}) = max\{L(E_{it}) + \delta(\overrightarrow{E_{it}E_{i+1\,j}})\ ;\ t \in S_{ij}\}$$

ここでエンティティ $E_{i+1\,j}$ に向かう有向枝で $\overrightarrow{E_{it}E_{i+1\,j}}$ で接続されるエンティティ E_{it} の集合を

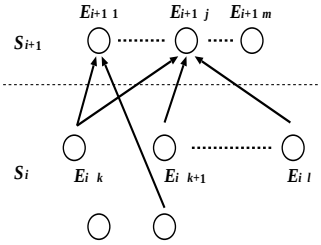

図 6 最小レベル割付けの方法
Fig. 6 Minimum level assignment method.

S_{ij}（集合 S_i の部分集合）とする．また，

$$\delta(\overrightarrow{E_{it}E_{i+1\,j}}) = 1 \ (\text{辺}\overrightarrow{E_{it}E_{i+1\,j}}\text{に機密要求あり})$$
$$= 0 \ (\text{辺}\overrightarrow{E_{it}E_{i+1\,j}}\text{に機密要求なし})$$

である．以上の計算方法によりエンティティ $E_{i+1\,j}$ には割付け可能な最小のレベルが割り付けられる．同様に集合 S_{i+1} の各エンティティには各々，最小のセキュリティレベルが割り付けられる．すでに得られているエンティティの集合 S_i に新たに求まった集合 S_{i+1} を追加して，全体を集合 S_{i+1} と再定義し，以上の計算を繰り返すことにより，グラフ全体として最小のセキュリティレベルが各エンティティに割り付けられる．

一方，逆 SLA アルゴリズムは上位層のエンティティから下位層に向かって，新 SLA と同様に段階的なレベル設定を行うので，グラフ全体としては最大のセキュリティレベルが各エンティティに割り付けられる．

3.3 自 由 度

SLA アルゴリズムと逆 SLA のレベル設定結果より，設定される最小レベルと最大レベルに，差ができるエンティティが存在する場合があることが分かる．したがって，そのエンティティは設定可能なレベルの許容範囲を持っているといえる．

双方のアルゴリズムを適用した結果，レベルに許容範囲が生じる理由は以下のように説明できる．まず，グラフに割り付けられた最大レベルより 1 を引いた数の機密要求辺を通るパスに着目する．これをクリティカルパスと呼ぶことにする．クリティカルパスにおいては SLA ならびに逆 SLA の双方のアルゴリズムでレベルに差が生じることはない．一方，クリティカルパスとは別に分岐して存在しているパスについては，クリティカルパスに含まれる機密要求辺より等しいかあるいは少ない数の機密要求辺数となる．したがって，機密要求辺が少ないパスについては，レベルとして設定可能な値には範囲が生じる．

ここで，任意のエンティティ e について，設定可能なレベルの許容範囲の最小値を L_{min}，最大値を L_{max} と

する．このとき，“エンティティ e は L_{min} から L_{max} までのレベルの自由度を持つ” と呼び，エンティティ e の自由度 LM を

$LM(e) = [L_{min}, L_{max}]$ ただし $(L_{min} \leq L_{max})$

と表記する．

また，統合要求グラフ G 上に存在する n 個のエンティティ $e_1, e_2, \cdots, e_n$ について，各々のレベルの自由度 LM が，

$$\begin{aligned} LM(e_1) &= [L_{min1}, L_{max1}] \\ LM(e_2) &= [L_{min2}, L_{max2}] \\ &\vdots \\ LM(e_n) &= [L_{minn}, L_{maxn}] \end{aligned}$$

であるとする．ここで，エンティティ e が持つ自由度のパターン数を $P(e)$ と表記すると，各エンティティのパターン数は，

$$\begin{aligned} P(e_1) &= (L_{max1} - L_{min1}) + 1 \\ P(e_2) &= (L_{max2} - L_{min2}) + 1 \\ &\vdots \\ P(e_n) &= (L_{maxn} - L_{minn}) + 1 \end{aligned}$$

と表すことができ，

$$LPT = \prod_{l=1}^{n} P(e_l)$$

となる LPT は，“要求の矛盾なく設定できる可能性のある” セキュリティレベルのパターンの数を示す．実際にはエンティティのレベルの自由度には，他のエンティティのレベルに依存する場合が存在するため，必ずしも “LPT = 設定可能レベルのパターン数” とはならない．

4. レベル設定値の全組合せを導出する方法

逆 SLA により，各エンティティが許容できる最大レベルを得ることが可能となった．しかし，実際にはあるエンティティの自由度は，他のエンティティのレベルに左右されるといった依存性が存在する．したがって，逆 SLA のみではレベルの自由度を完全に導出したとはいえない．よって，依存性を考慮した各エンティティの自由度を求める拡張 SLA アルゴリズムを以下に示す．

4.1 拡張 SLA アルゴリズム

図 2 に示す統合要求グラフに対する自由度の解析アルゴリズム（拡張 SLA アルゴリズム）を以下に示す．

（E1）SLA，および逆 SLA のアルゴリズムを用いたレベル設定において，双方のアルゴリズムで同一のレベルが設定されたエンティティは，レベルの変更が不可能であるため，これらのエンティティをまず，拡張 SLA アルゴリズムに使うエンティティの候補から除外する（図 7 の FixedEntity の範囲）．

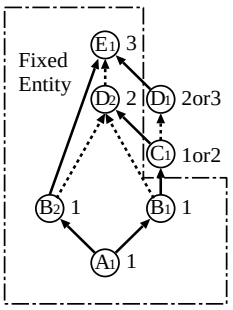

図 7 拡張 SLA アルゴリズムの適用範囲抽出
Fig. 7 The applicable range extraction of the extend SLA.

（E2）残ったエンティティ群において，レベルの変更が不可能であるとされたエンティティ群から入ってくる矢印を検出し，その矢印の先となるエンティティを，拡張 SLA アルゴリズムを実行する始点とする．

（E3）残ったエンティティ群において，拡張 SLA アルゴリズムの未処理の階層を，記号の小さな順に選択する．これを現階層とする．ただしレベル変更の不可能な領域に存在する階層は選択しない．

（E4）現階層のエンティティから，拡張 SLA アルゴリズムにおいて未処理のもので一番ラベルの値の低いものを選択する．これを現エンティティとする．

（E5）現エンティティに入っている矢印を列挙し，矢印の根であるエンティティの全レベルと，現エンティティの全レベルの組合せをすべて列挙する．ただし，ループが検出されている組合せのエンティティの場合は，矢印の根のエンティティと現エンティティが同じレベルとなる組合せのみを列挙する．

（E6）列挙された組合せには，エンティティ間の要求に合わないもの（機密要求であるのに，両端のエンティティが同レベルの組合せである等．**図 7** においては C_1，D_1 エンティティともにレベル 2 が割り付けられた場合）が存在する．それらの組合せを要求の種類から判別し，適合する組合せのみを残す．

（E7）（E3）から（E6）までの処理を繰り返す．全階層について処理を行い，アルゴリズムが終了する．最終的に，適合するレベルの組合せのみが残る．

このとき，設定可能なレベルが 3 種類存在することが分かる．**表 1** のように，拡張 SLA アルゴリズムは複数のレベル設定パターンを出力する．したがって，レベルの自由度が多いほどレベルの設定に汎用性

表 1 拡張 SLA 実行結果
Table 1 Result of extend SLA.

	A_1	B_1	B_2	C_1	D_1	D_2	E_1
Pattern1	1	1	1	1	2	2	3
Pattern2	1	1	1	1	3	2	3
Pattern3	1	1	1	2	3	2	3

を持たせることができる．たとえば，このとき，“エンティティ C_1 と D_2 は同じレベルである” という制約が処理・機密要求とは別に存在している場合，表の Pattern3 を選択すればよい．

5. むすび

処理要求と接続要求が与えられて，階層的アクセス制御のレベルを割り付ける方法は，従来から SLA アルゴリズムとして知られている．SLA ではエンティティをノード，要求を有向辺とするグラフを深さ優先探索でたどることにより，各エンティティに許容可能な最下位レベルを割り付ける．

これに対して本論文では，レベル設定の問題をグラフのラベル付けとレベル割り付けの 2 段階に分解する新 SLA アルゴリズムをまず，提案している．新 SLA はグラフのラベル付けを幅優先探索で行い，その結果グラフが多段化されるため，レベル割付けは機械的に行うことができる．得られるセキュリティレベルは SLA と同じく，許容可能な最下位のレベルである．

本論文の後半では，ラベル付けされたグラフを逆方向にたどってレベル付けする逆 SLA アルゴリズムを提案している．逆 SLA は各エンティティに許容可能な最上位レベルを割り付ける．この部分においては，新 SLA におけるラベル付け処理により試行錯誤の探索をなくし，容易な逆 SLA の実行を可能とした．

以上の新 SLA ならびに逆 SLA の実施により，得られる各エンティティのセキュリティレベル値は，ある許容範囲を持つことになる．この点に着目して，本論文の最後にはレベル設定可能な解をすべて列挙する方法を提案した．このようなレベル設定の自由度を確保することにより，レベル設定の運用に柔軟性が生まれる利点が得られることを示した．

参考文献

1) Jajodia, S. and Kogan, B.: Integrating an Object-Oriented Data Model with Multilevel Security, *1990 IEEE Symp. on Security and Privacy*, pp.76–85 (1990).
2) Fernandez, E.B., Gudes, E. and Song, H.: A Security Model for Object-Oriented Databases, *1989 IEEE Symp. on Security and Privacy*, pp.110–115 (1989).
3) Bell, D.E. and LaPadula, L.J.: Secure Computer System: Unified Exposition and Multics Interpretation, Technical Report, MTR-2997 (available as NTIS AD-A023588), MITRECorp. (1976).
4) Araki, T., Morizumi, T., Nagase, H., Takenaka, T. and Yamashita, K.: Security Level Assignment By Graph Analysis, *Issues on Cryptography and Information Security, IEICE Trans.*, Vol.74, No.8, pp.2166–2175 (1991).
5) 四七秀貴，奥田裕子，永瀬　宏：アクセス行列から階層的レベルを導出する方法についての考察，第 17 回情報理論とその応用シンポジウム（*SITA'94*），Vol.23, No.3, pp.157–160 (1994).
6) Shina, H., Okuda, Y. and Nagase, H.: Total Ordered Security Level Assignment Which Inhibits Entity Inference, *International Symp. on Information Theory & Its Application* (*ISITA'94*), pp.273–276 (1994).
7) Nagase, H., Shina, H. and Edmundo, G.: Security Planning in Data Processing System, *Korea-Japan Joint Workshop on Information Security and Cryptolog* (*JW-ISC'93*), pp.226–234 (1993).
8) 情報処理振興事業協会（IPA）：http://www.ipa.go.jp/SECURITY/ccj/archive/s200003/15408intro.pdf, ISO/IEC15408 セミナー，(2000).
9) 佐々木良一，宝木和夫，櫻庭健年，寺田真敏，浜田成泰：インターネットセキュリティ基礎と対策技術，オーム社 (1996).
10) 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).

(平成 11 年 11 月 30 日受付)
(平成 12 年 6 月 1 日採録)

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を経て平成 5 年金沢工業大学工学部情報工学科教授就任，以来情報セキュリティ，計算機アーキテクチャの研究に従事．

井上 清一（学生会員）

平成 10 年金沢工業大学工学部情報工学科卒業．平成 12 年同大学大学院工学研究科修士課程（情報工学）修了．同大学大学院博士課程在学中．現在，アクセスコントロールの研究に従事．

四七 秀貴

平成 6 年金沢工業大学工学部情報工学科卒業．平成 8 年同大学大学院工学研究科修士課程（情報工学）修了．同年，日本電信電話株式会社に入社，現在 NTT ネットワークサービスシステム研究所に勤務，以来交換システムソフトウェア，分散処理技術の研究に従事．