ロールベースアクセス制御情報の多バージョン並行処理制御を利用した監査ログトラッキング手法

近藤誠一[†] 白木宏明[†] 大沼聡久[†]
小宮崇[†] 五月女健治[†,††] 虎渡昌史[†,†††]

近年，企業の機密情報や個人情報が外部へ流出する事件が多発しており，社会問題となっている．情報漏洩に対するセキュリティ対策として，行為の実施前にユーザ認証，アクセス制御を行うとともに，実施した行為をログとして収集・蓄積し，監視・監査・分析を行うという二面の方式がとられる．アクセス制御に関しては，ロールベースアクセス制御（RBAC: Role-Based Access Control）モデルに基づく手法が一般的である．しかし，RBAC を企業に適用する場合，人事情報と連動させた運用の効率化が課題となる．また，実施した行為の蓄積ログが長期間にわたる場合，その間に人事異動，セキュリティポリシの改訂が発生するため，RBAC 情報の変更に連動させた各行為の監視・監査・分析が課題となる．本論文では，行為の実施前のユーザ認証・アクセス制御と，実施後のログの監視・監査・分析の両者の整合性をとった多バージョンアイデンティティアクセス制御情報管理方式について示す．

Audit Log Tracking Method Using Multiversion Concurrency Control of Role-Based Access Control (RBAC) Information

Seiichi Kondo,[†] Hiroaki Shiraki,[†] Akihisa Oonuma,[†]
Takashi Komiya,[†] Kenji Saotome[†,††] and Masashi Torato[†,†††]

Recently confidential information leakage is becoming a serious issue. In order to prevent the leakage from insiders, two kinds of measures are taken: (1) user authentication and authorization before accessing confidential information. (2) audit trail after accessing confidential information. Role-based access control (RBAC) models are known as a powerful and generalized approach to security management. However, the enterprise system needs to increase in efficiency of personnel information provisioning to the security management system and to monitor audit log which are accumulated over the long period of time after changing staff reassignment and a security policy. In this paper, we propose a new identity and access management method which can be used both before and after accessing confidential information using multiversion concurrency control.

1. はじめに

近年，企業の機密情報や個人情報が外部へ流出する事件が多発しており，社会問題となっている．情報漏洩に対するセキュリティ対策として，紙文書・媒体・機器・建造物に対する「物理セキュリティ」，計算機上の情報の漏洩・改ざん・偽造の脅威に対する「情報セキュリティ」，企業内のネットワークへの不正侵入・攻撃の脅威に対する「ネットワークセキュリティ」がとられる．従来，物理セキュリティ，情報セキュリティ，ネットワークセキュリティの個々の観点から，対策システムを個別に導入してきたが，さまざまな脅威に対してワンストップで対応していくためには，体系的な導入が有効であると考えられる．そこで，我々は，ユーザ認証，アクセス制御，ファイル暗号化等の情報セキュリティと，入退室管理システム等の物理セキュリティを統合したトータルソリューションとして情報漏洩防止ソリューションを開発した[1)~4)]．

個々の脅威に対応した対策システムを個別に導入すると，以下に示す課題が生じる．

- ユーザ情報，セキュリティポリシを統一させるための運用管理の効率化

† 三菱電機株式会社情報技術総合研究所
Information Technology R&D Center, Mitsubishi Electric Corporation
†† 法政大学大学院イノベーション・マネージメント研究科
Hosei Business School of Innovation Management
††† 三菱電機インフォメーションシステムズ株式会社
Mitsubishi Electric Information Systems Corporation

- 複数のセキュリティコンポーネントのユーザ認証手段によるセキュリティ強度の統一と利用者の利便性向上
- 入退室管理装置等の非 PC を含むさまざまな情報機器，個別のセキュリティツールからの広域分散環境でのログ収集・管理

これらの問題の解決のため，本ソリューションは，以下に示す方式を採用した．

- ユーザ認証情報，アクセス制御情報を一元管理し，人事情報と連動させた変更管理機能の提供
- 個別ツールに対する ID の統一と，共通の認証手段，アクセス制御ポリシによる認証・認可機能の提供
- 個々の機器，ツール対応のコンポーネント層と，それを制御するコントロール層に分離したコンポーネント指向ログ収集機能の提供

このように，情報漏洩対策として，多岐にわたるセキュリティ対策を可能としてきた．さらに，このような対策は，導入後も，継続して運用を行い，改善していくことが求められている[5]．その際，ユーザ認証・アクセス制御で用いたアイデンティティ情報と，行為の実施後を示すログとの関連付けが必要となる．

本論文では，行為の実施前のユーザ認証・アクセス制御と，実施後のログの監視・監査・分析の両者の整合性をとったアイデンティティアクセス制御情報管理方式について示す．3 章では，人事システムと連動した運用管理効率化を実現するため，RBAC モデル[6)~8)]をもとにしたユーザ情報，アクセス制御情報の LDAP（Lightweight Directory Access Protocol）上での実装・運用方式について示す．4 章では，収集されたログの継続的な監査を可能とするアクセス制御情報の多バージョン並行処理制御を利用した監査ログトラッキング手法について示す．

2. 基 本 事 項

2.1 情報漏洩防止ソリューション

本論文で述べる情報漏洩防止ソリューションの体系を**図 1** に示す．

2.1.1 情報セキュリティコンポーネント

- ファイル暗号化システム
  特定フォルダ等の一括暗号化・自動暗号化を行う．また，共有サーバ上の機密情報を暗号化して保管し，人事情報に連動したアクセス制御を実現する．
- デバイス制御ソフトウェア
  USB メモリ，DVD 等のリムーバブルメディアへの書き込み禁止制御を行う．

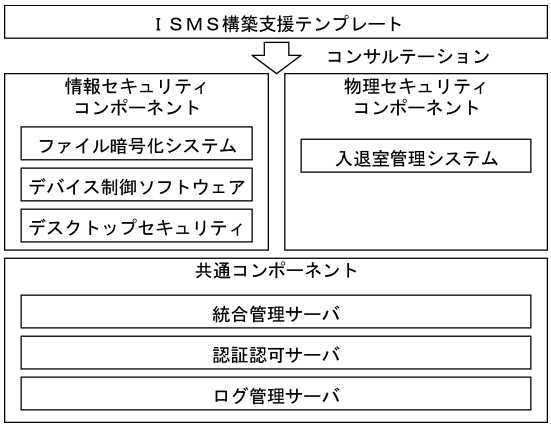


図 1 情報漏洩防止ソリューションの体系
Fig. 1 Architecture of information leak prevention solution.

- デスクトップセキュリティ
  IC カード，指紋照合，パスワード，PKI 認証等の多様なユーザ認証手段により PC へのログイン制御を行う．

2.1.2 物理セキュリティコンポーネント

- 入退室管理システム
  非 PC の ID コントローラがユーザのアクセス制御情報を持ち，IC カード，指紋照合等のユーザ認証により入退室の制御を行う．

2.1.3 共通コンポーネント

- 統合管理サーバ
  情報セキュリティコンポーネントおよび物理セキュリティコンポーネントのユーザ認証・認可で用いられるユーザ情報，アクセス制御情報を一元管理し，運用管理者向けに統合ツールを提供する．
- 認証・認可サーバ
  統合管理サーバで管理されるユーザ情報およびアクセス制御情報をもとに，ファイル暗号化システム，デバイス制御ソフトウェア，デスクトップセキュリティに対して，ユーザ認証および認可決定を行うとともに，Web 業務アプリケーションのシングルサインオンを行う．また，入退室管理システムに対しては，ユーザ情報，アクセス制御情報を配布する．
- ログ管理サーバ
  情報漏洩防止ソリューションを構成する各コンポーネントが出力する各種セキュリティログの収集，統合管理を行う．

2.2 ロールベースアクセス制御（RBAC）

2.2.1 ロールベースアクセス制御モデル

ロールベースアクセス制御（RBAC: Role-Based

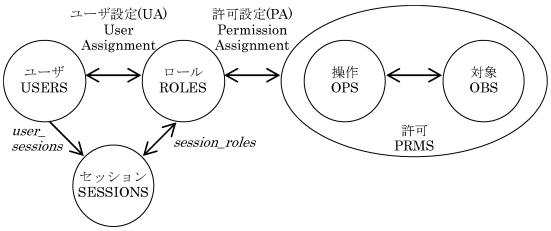


図 2 コア RBAC
Fig. 2 Core RBAC.

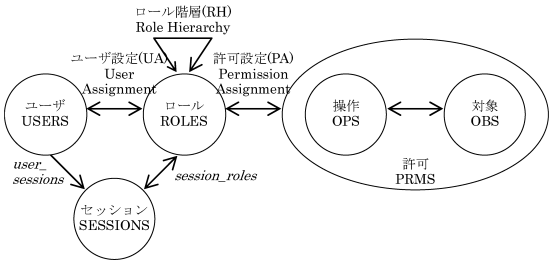


図 3 階層 RBAC
Fig. 3 Hierarchical RBAC.

Access Control）モデルは，セキュリティ管理の一般的な手法として知られている[7)~9)]．また，業務アプリケーションだけでなく，データベース管理システム，OS でも採用されている[10),11)]．RBAC では，ユーザ情報，アクセス制御情報をロールに設定する．その効果として，組織，役職といったユーザ属性情報と，セキュリティの対象となるファイル，アプリケーションに対するアクセス制御情報の変更管理を独立して行うことが可能となる．また，ロールの階層化，グループ化といった柔軟な運用が可能となる．

2.2.2 NIST RBAC モデル

NIST（National Institute of Standards and Technology）で示されているコア RBAC モデルを**図 2** に示す[7)]．コア RBAC は以下の要素から成り立つ．

- ユーザ（USERS）
  人そのものが定義される．エージェント等擬似的なものへの拡張が可能である．
- ロール（ROLES）
  組織，役職等，意味的な役割を示すグループが定義される．
- 対象（OBS）
  セキュリティで守るべきファイル，フォルダ等のデータの格納場所，プリンタ等のシステムリソースが定義される．
- 操作（OPS）
  参照，編集，印刷等，対象に対して行われる操作が定義される．
- 許可（PRMS）
  RBAC で保護される対象への操作の許可が定義される．対象と操作は，多対多の関係にある．
- セッション（SESSIONS）
  ユーザとユーザに割り当てられたロールの部分集合とのマッピングが定義される．

RBAC では，図 2 に示すように，ユーザとロールとの関係であるユーザ設定：UA（User Assignment），許可とロールとの関係である許可設定：PA（Permission Assignment）からなる．図中，両矢印は多対多の関係を示す．ユーザは，各セッションにおいて，(1) 設定されているロールのいくつかを要求，(2) 要求されたロールが，要求時点で許されている場合は，そのロールを獲得，(3) ロールに紐づけられている許可に従ったアクセス制御を行う．

コア RBAC の拡張として，**図 3** に示すように，組織構造を考慮した階層 RBAC が提唱されている[7)]．階層 RBAC では，ロール階層を導入し，ロールの継承関係を定義することが可能である．

3 章において，RBAC モデルをベースとして，日本の組織，役職の階層構造に即したユーザ設定（UA）手法，情報漏洩対策として多岐にわたる機器，セキュリティツールに対応した許可設定（PA）手法，および，それらの LDAP 上での実装・運用方式について示す．

2.3 データベースにおける並行処理制御

2.3.1 並行処理制御

RBAC モデルで定義された要素を追加，変更，削除，参照する複数のトランザクションは，通常，並行に処理される．トランザクションが並行処理されるデータベース管理システムでは，トランザクションが直列可能（serializable）な場合，一貫性が保証されていると見なして，読み出し処理，書き込み処理を並行処理制御が行われる[12),13)]．$R(x)$ をデータ x の読み出し処理，$W(x)$ をデータ x の書き込み処理，A_i をトランザクション，s_i をスケジュールとする．たとえば，

$$s_1 = A_1 : R(x)R(y) \qquad W(x)$$
$$A_2 : \qquad R(y)W(y)$$

は，等価な直列スケジュールとして，

$$s_2 = A_1 : R(x)R(y)W(x)$$
$$A_2 : \qquad R(y)W(y)$$

が存在するため，直列可能である．一方，

$$s_3 = A_1 : R(x)W(x) \qquad R(y)W(z)$$
$$A_2 : \qquad R(x)W(y)$$

は，等価な直列スケジュールが存在しないため，直列可能ではない．直列可能性については，厳密には，さまざまなクラスが定義されている[13)]．直列可能を保証

する並行処理制御方式として，二相ロック方式と時刻印方式が知られている[12),13)]．ここでは，4 章で提案する方式で用いる時刻印方式について示す．

時刻印方式は，トランザクションの到着順に時刻印を付け，この時刻印の順番の直列スケジュールと等価となるように制御を行う．時刻印順とならない場合，処理単位の後退復帰で扱うことになる．

(i) 各トランザクション T_i に到着順に時刻印 t_i を割り当てる．

(ii) 各データに対して読み出し時刻印 t_r と書き込み時刻印 t_w を割り当てる．

(iii) 次の場合は，トランザクション T_i の 1 つの操作を実行できる．
a) 読み出し：データの t_w が t_i より小さい．
b) 書き込み：データの t_r が t_i より小さい．
c) b) の場合，データの t_w が小さければ実際にデータに書き込むがそうでなければ書き込みは行わない．

(iv) 読み出し，書き込みの場合に a)，b) の条件が満足されないと，T_i を後退復帰する．この場合，後退復帰の連鎖についても扱う必要がある．$R_i(x)$ を時刻印 i を持つトランザクションによるデータ x の読み出し処理，$W_i(x)$ を時刻印 i を持つトランザクションによるデータ x への書き込み処理とする．a)，b)，c) の各処理により，
a) $W_j(x)$ の後の $R_i(x)$ $(t_j < t_i)$
b) $R_j(x)$ の後の $W_i(x)$ $(t_j < t_i)$
c) $W_j(x)$ の後の $W_i(x)$ $(t_j < t_i)$，または，
$W_j(x)$ の前の $W_i(x)$ $(t_j > t_i)$
の関係が保証される．

2.3.2 多バージョン並行処理制御

書き込み処理の前の値をつねに保持し，データの t_w がトランザクションの時刻印より大きい読み込み処理の際に，トランザクションを後退復帰させないで，保持している書き込み処理の前の値を読むことにより，直列可能性を保証することが可能となる．

たとえば，前述のスケジュール s_3 について，考察する．

$$s_3 = A_1 : R(x)W(x) \qquad R(y)W(z)$$
$$A_2 : \qquad R(x)W(y)$$

トランザクション A_2 の $R(x)$ 処理を，トランザクション A_1 の $W(x)$ 処理の前の値を読み込むようにすることにより，$A_2 \rightarrow A_1$ の順の直列スケジュールと等価となる．また，トランザクション A_1 の $R(y)$ 処理を，トランザクション A_2 の $W(y)$ 処理の前の値を読み込むようにすることにより，$A_1 \rightarrow A_2$ の順の直列スケジュールと等価となる．このように，書き込み前の値を保持することによって直列可能な場合，多バージョン直列可能（multiversion serializable）と呼ばれている[13)]．では，以下の 3 種類の処理の競合のうち，a)，b) は交換可能，c) は交換不可能としている．

a) $W(x)$ の前の $W(x)$
b) $R(x)$ の前の $W(x)$
c) $W(x)$ の前の $R(x)$

このような競合の交換可能性によって判定される多バージョン直列可能なクラスは，競合多バージョン直列可能（conflict multiversion serializable）と呼ばれ，効率良く判定が可能であることが示されている[13)]．

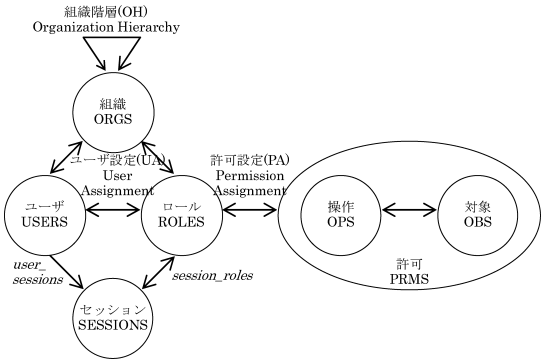


図 4 階層型組織 RBAC
Fig. 4 Hierarchical organization RBAC.

3. 情報漏洩防止ソリューションにおける RBAC の実装

3.1 抽象ロールを用いたユーザ設定（UA）

情報漏洩防止ソリューションでは，日本の一般的な組織構造である階層構造に特化し，初期導入コスト，および，人事異動，組織改変時の運用コストを抑えることを目標とした．2.2 節で示した NIST の RBAC では，階層構造を持つ組織をロールとし，ロール自身を階層構造で定義する．また，ロールはアクセス性能を考慮して，ユーザの集合として定義する方式が一般的である．この場合，人事異動や組織改変時に，ユーザ情報の変更に合わせてロールのグループ内容の再設定が必要となり，その作業負荷が課題となる．

本章では，上記課題に対して，**図 4** に示すように，人事情報に対応するユーザ，組織をロールから独立させる構造をとった．すなわち，階層構造を持つ組織を新たに設けるとともに，組織，および，ユーザの持つ属性情報を指定する抽象化されたロールを導入した．本方式におけるオブジェクト間の設定を以下に示す．

- ユーザ–組織間
  ユーザはフラットな構造とし，所属する組織との関係を持つ．組織は複数のユーザを所属員として持つ．また，ユーザは，複数組織の兼務を認める．したがって，ユーザと組織の間は，多対多の関係を持つ．
- ロール–組織・ユーザ間
  ロールに所属するユーザを指定するため，役職等のユーザの持つ属性情報を要素とする論理式を用いて間接的に指定する抽象ロールを提供する．論理式の要素として組織を指定する場合は，その下位の組織に属するユーザも，ロールに所属するものとする．
- ロールの自動生成
  所属組織，役職に関しては，利便性を考慮して，各組織，役職のみを要素とするロールを自動生成する機能を提供する．

これらの構造を利用して，初期導入時，人事異動時，組織改変時に，以下の操作を行う．

- 初期導入時
  人事システム等で管理されている組織情報，人事情報を，ユーザ，組織にマッピングし，ユーザ–組織間の設定を行う．また，組織，役職といった人事情報に則した抽象ロールを自動生成する．その結果，企業等で一般的に用いられている組織，役職の構造に修正を加えず，アクセス制御情報定義から独立した構造で初期導入時のユーザ設定が可能となる．
- 人事異動時
  組織間の異動，役職の変更については，ユーザの属性変更のみで対応する．組織，ロールの変更，すなわち，ユーザ設定の変更は不要である．
- 組織変更時
  組織，役職の追加，削除，変更時は，人事情報の変更に合わせて，ユーザ，組織の変更を行う．また，抽象ロールを自動変更する．

3.2 多様な対象に対応した許可設定

情報漏洩防止ソリューションにおける許可設定では，図 1 に示したファイル暗号化システム，デバイス制御ソフトウェア，デスクトップセキュリティ，入退室管理システム，Web 業務アプリケーションシングルサインオンシステムといった用途の異なるセキュリティコンポーネントごとの運用コストが課題となる．

本ソリューションでは，以下の設定を可能とした．

- 対象の階層構造
  入退室管理装置では，ビル/フロア/サーバルーム等の特殊部屋，ファイルでは，フォルダ構造，Web 業務アプリケーションでは，メインアプリケーション/サブアプリケーションに基づくメニュー構成といったように，対象自身の階層構造に基づき，許可設定を可能とした．
- 操作の設定
  RBAC モデルに基づき，操作と対象を関連づけて指定する．入退室管理装置における操作は開錠，ファイルの操作は参照/更新/印刷/コピー，Web 業務アプリケーションは起動といったように，対象ごとに異なる．
- 論理式によるロールの設定
  ユーザ設定では，ユーザ属性値，組織の論理式によって指定することにより，さまざまなセキュリティツールに共通なロールの作成を可能にした．一方で，セキュリティツールごとに必要とされる対象の粒度は，たとえば，入退室管理装置とファイルでは数，操作が大きく異なる．そこで，ユーザ設定に加えて，許可設定においてもロールを要素とする論理式による設定を可能とした．

3.3 LDAP での実装，運用方式

日本の一般的な組織構造，セキュリティの対象であるフォルダ/ファイルの構造，業務アプリケーションの構造である階層構造を効率良く処理するために，本ソリューションでは，LDAP を採用した．**図 5** に LDAP 上の構造を示す．以下の 4 種類の木からなる．

- ユーザ（USRS）
  階層を持たないフラットな構造を持つ．ユーザ認証情報（パスワード，証明書，指紋等の生体情報，IC カード ID 等），アクセス制御に用いる属性情報（所属組織，役職等）を持つ．所属，役職が兼務の場合は，同一人物が複数のノードを持つことを示す．
- 組織（ORGS）
  日本の一般的な組織構造に対応した階層構造を持つ．
- ロール（ROLES）
  ユーザ設定として，ユーザ属性値を要素とする論理式で表現する．組織を指定した場合は，下位の組織を包含する．
- 許可（PRMS）
  対象と，操作の対をノードとする．許可設定として，ロールの論理和で表現する．セキュリティツールごとに，木を持ち，階層構造を許す．操作の種類は，セキュリティツールに依存する．

以下の手順で，認可決定を行う．

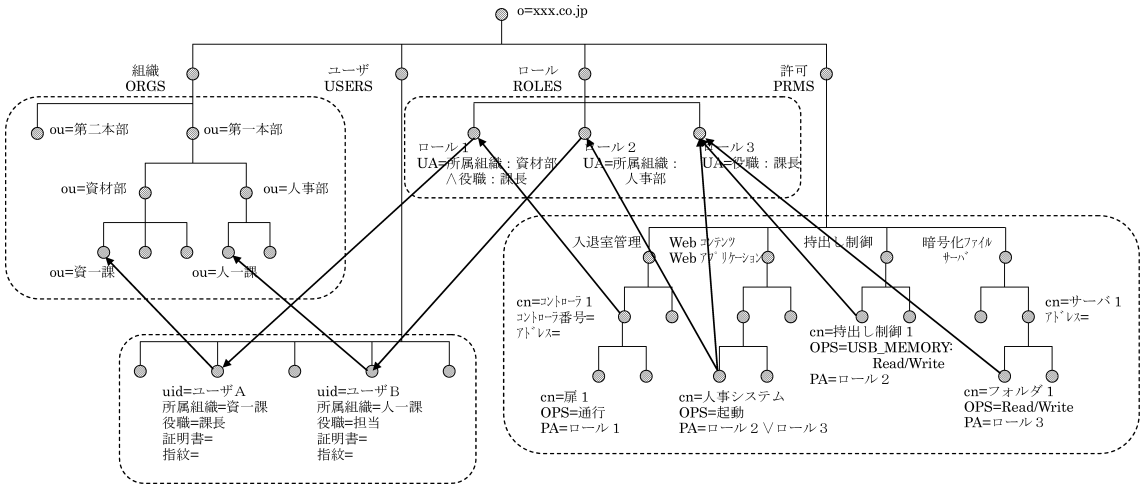


図 5 LDAP での階層型組織 RBAC の実装
Fig. 5 Implementation of hierarchical organization RBAC on LDAP.

① 対象となる行為に対応する許可を取り出す.
② 許可ごとのロールを取り出す.
③ ロールとユーザ属性を比較して認可判定を行う.
④ 指定された属性が，所属組織の場合は，上位の組織について認可判定を行う.
⑤ 論理和の関係にある許可，ロールすべてについて，①〜④ を繰り返す.

たとえば，図 5 におけるユーザ B が人事システムを起動する場合の認可判定を行う場合について，以下に示す.
① 人事システムの許可を参照する．この場合は，該当する許可のノードは 1 つである.
② 論理和の関係にあるロール 2，ロール 3 を取り出す.
③ ロール 2 の条件は，所属組織が人事部であり，ユーザ B の所属組織である人一課と異なるため，認可されない.
④ 組織を参照し，人一課の上位である人事部が一致するため，人事システムの起動が認可される.

本方式では，ロールは実行時にユーザ情報・組織情報を参照するため，人事異動時に参照とは独立して変更された情報に，即座に追随可能となる．また，日本企業では，組織主体のユーザ管理が行われるため，組織に結び付けたアクセス制御が行われる．そこで，組織情報のロールを自動的に生成する機能を提供することにより，許可の指定を容易に可能とした.

3.4 実装例と性能評価

本方式は，組織構造，配属，役職といった人事上の事実と，セキュリティ上の設定であるアクセス制御設定を分離し，その間を論理式により間接的に関連づけ

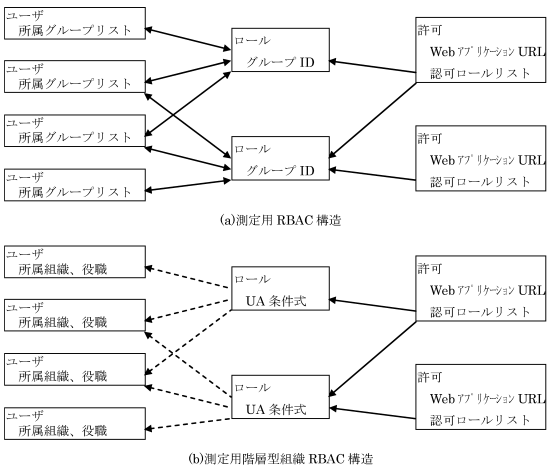


図 6 ユーザ設定の実装
Fig. 6 Implementation of UA.

たユーザ設定の実装方式が RBAC 方式と比較して大きな特徴となっている．本節では，従業員 5,000 人規模の企業をモデルとして RBAC と本方式の性能評価結果を示す．1 ユーザあたり，1 Kbyte と仮定すると，5,000 人規模のユーザ情報は，5 Mbyte 程度である．大企業といわれる 10 万人規模でも，100 Mbyte 程度とメモリ上で操作可能な範囲内であり，今回の目的から十分であると考える[14].

アクセス制御の対象として，Web アプリケーションとし，そのシングルサインオンを実現するシステムを想定する.

RBAC のユーザ設定の実装として，**図 6** に示すように，IBM Tivoli Access Manager [15]，Plumtree Corporate Portal [16] 等で一般的に用いられているユーザ

表 1 測定用データ構造
Table 1 Data structure for measurement.

(a) 組織構造とユーザ数

	組織数	ユーザ数	役職
本部	4	4	本部長
事業所	6	8	事業所長
部	150	450	部長
課	450	4,500	課長，担当
計	610	5,000	

(b) 認可設定，ユーザ数設定

オブジェクト数	10
ロール数	1,874
オブジェクトに設定するロール数	
1 オブジェクトあたり	2
総数	16
UA 条件式	2
所属グループ数	10

をグループとして直接リンクによって関連づける方式と比較する．すなわち，性能を重視した実装方式と比較して，運用を重視した本方式が，性能面で容認可能な範囲内であるかの検証を行った．

3.4.1 測 定 方 法

（1）データ構造

社員数 5,000 名規模を想定し，5 階層とする．**表 1** にデータ構造を示す．ロールは，すべての組織，役職ごとに自動生成し，そのうち，16 個を認可可能としてオブジェクトに設定するものとする．UA 条件式は，これまでの実装例から，1〜2 項目程度の AND または OR のことが多いので，2 項目とした．

（2）認可方式

① 実行時認可判定

指定された Web アプリケーションの URL と認証ユーザを入力として実行時に認可判定を行う．

・RBAC 方式

ユーザ認証段階で，所属グループリストをディレクトリサーバから参照して保持しているものと仮定する．

(1-1) 指定された Web アプリケーションをキーとしてディレクトリを検索して，許可されているロールを求める．

(1-2) (1-1) で求めたロールに登録されているグループをディレクトリサーバから検索して，ユーザが所属するグループと (1-1) で求めたロールを比較し，認可判定を行う．

・階層型組織 RBAC 方式

ユーザ認証段階で，上位を含めた所属組織，役職をディレクトリサーバから参照して保持しているものと仮定する．

(2-1) 指定された Web アプリケーションをキーとして，ディレクトリを検索して許可されているロールを求める．

(2-2) ロールの属性として持つ UA 条件式をディレクトリを検索して求める．

(2-3) ユーザの所属，役職と UA 条件式の比較により，認可判定を行う．

② ユーザ認証時認可判定

ユーザ認証時にアクセス可能な業務アプリケーションのリストを生成する．

・RBAC 方式

(3-1) ディレクトリを利用してユーザ認証を行い，ディレクトリを検索して所属するグループ（複数）を求める．

(3-2) すべての業務アプリケーションについてディレクトリを検索してアクセスが許可されているロールを求める．

(3-3) グループとロールの関連から，認証されたユーザに認可されている Web アプリケーションのリストを求める．

・階層型組織 RBAC 方式

(4-1) ディレクトリを利用してユーザ認証を行い，ディレクトリを検索して所属，役職を求める．

(4-2) すべての業務アプリケーションについて，ディレクトリを検索してアクセスが許可されているロールを求める．

(4-3) (4-2) で求められたロールの属性として持つ UA 条件式をディレクトリ検索により求める．

(4-4) ユーザの所属，役職と論理式の比較により，認証されたユーザに認可されている Web アプリケーションのリストを求める．

（3）測定環境

以下の 2 台のサーバを 100BaseT で接続し，認証・認可サーバからディレクトリサーバへ LDAP インタフェースでアクセスするものとする．

(a) 認証認可サーバ
- H/W CPU：Pentium4 2.66 GHz
  メモリ：1 GB，HDD：40 GB
- S/W 認証・認可プログラム
  Java 1.4.2_07
  Windows 2003 Server

(b) ディレクトリサーバ
- H/W CPU：Pentium4 3.0 GHz
  メモリ：1 GB，HDD：40 GB
- S/W SunONE Directory Server 5.2
  Windows2003 Server

表 2 実行時認可判定 ①
Table 2 On-demand authorization.

	応答時間 (msec)				スループット (回/sec)
RBAC	(1-1)	(1-2)		計	
シングル	0.519	0.990		1.509	662.856
10 多重	3.391	6.300		9.691	1,031.914
階層型組織	(2-1)	(2-2)	(2-3)	計	
シングル	0.562	0.990	0.012	1.575	634.856
10 多重	3.441	6.442	0.029	9.942	1,005.874

表 3 ユーザ認証時認可判定 ②
Table 3 Authorization on authentication.

	応答時間 (msec)					スループット (回/sec)
RBAC	(3-1)	(3-2)	(3-3)		計	
シングル	0.662	1.319	9.382		11.363	90,277
10 多重	3.861	5.257	65.754		74.872	133.561
階層型組織	(4-1)	(4-2)	(4-3)	(4-4)	計	
シングル	1.317	1.175	9.421	0.153	12.210	81.486
10 多重	7.750	5.009	66.399	0.201	79.561	125.690

3.4.2 測 定 結 果

① 実行時認可判定の Web アプリケーション起動時に認可判定を行う場合の測定結果を**表 2** に示す．20,000 回の認可判定を行った際の段階ごとの平均応答時間とスループットをシングルアクセス，10 多重アクセスの 2 種類について示す．

② ユーザ認証時認可判定のユーザ認証時に，アクセス可能なアプリケーションの認可判定をまとめて行う場合の測定結果を**表 3** に示す．20,000 回のユーザ認証を行った際の段階ごとの平均応答時間とスループットをシングルアクセス，10 多重アクセスの 2 種類について示す．

3.4.3 考 察

本論文で提案する階層型組織 RBAC 方式の応答時間，スループットは，RBAC 方式と比較して，最大で 10%程度であり，許容範囲内であることを確認した．

① 実行時認可判定の場合，LDAP の参照回数は等しく，差異は，グループの直接比較と，UA 条件式の判定にある．LDAP 検索を含む全体の応答時間に占める割合は，2%未満で，実用上，問題ないと考える．

② ユーザ認証時認可判定の場合，差異は，階層型組織 RBAC における階層を含む組織参照，グループの直接比較と UA 条件式の判定に起因する．本手続きは，ユーザ認証時に行われ，その後，シングルサインオンでアプリケーションが起動される場合には，いずれの方式でも，認可可能なアプリケーションのリストを参照するため，差異は生じない．スループットの差が 5%以内で，測定に用いた設備環境においても，10 多重の場合，100 回/sec 以上あり，実用上，問題がないと考える．

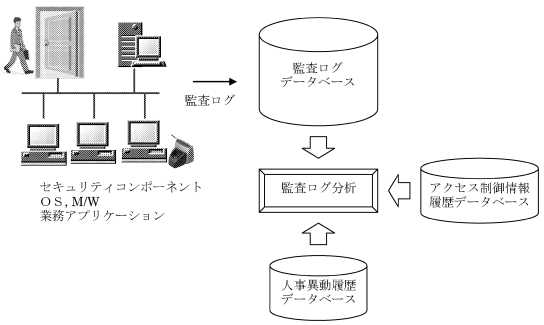


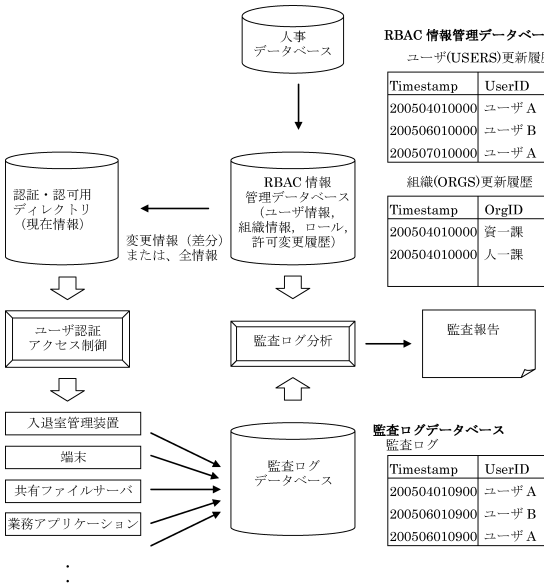
図 7 人事異動履歴を利用した監査ログ分析
Fig. 7 Audit log analysis using staff reassignment history.

また，図 6 に示したユーザ–グループを直接リンクで結ぶ方式をとる場合，UA 条件式にあたるものをセキュリティポリシの一部として保持し，それに合致するように，ユーザ–グループの関係を，維持する必要がある．すなわち，定期的にバッチで，ユーザとグループ間のリンクをセキュリティポリシに従って更新する必要がある．一方，階層型組織 RBAC では，人事異動に合わせた維持のみで，現状の把握も直感的で確認が容易である．

4. RBAC 情報の多バージョン管理によるセキュリティログ監査

4.1 人事情報に連動したセキュリティログ監査概要

個人情報，機密情報漏洩の原因の多くは内部の犯行と考えられており，企業では，セキュリティ対策の説明責任が問われている．そこで，個人の行為をログとして一定期間，たとえば，数年単位で蓄積し，その正当性を監査する方策がとられる．一方で，長期間蓄積されたログの正当性を監査するためには，日々更新される人事情報，セキュリティ対象の変更に追随する必要がある．しかし，監査のためのセキュリティログは，警告ログだけでなく，アクセスログすべてを記録する必要があるため，一般に，膨大となる．したがって，個々のログに，行為を行ったユーザの属性情報を含めることは現実的ではない．そこで，**図 7** に示すように，人事異動履歴を保持して，アクセス制御情報履歴と関連づけた監査を行うシステム構成となる．本章では，3 章で示したユーザ設定の人事に関わる部分を独立させた RBAC 方式を利用したセキュリティログ監査方式を提案する．



ユーザ(USERS)更新履歴

Timestamp	UserID	Action	Attributes
200504010000	ユーザ A	modify	資一課,課長
200506010000	ユーザ B	add	人一課,担当
200507010000	ユーザ A	delete	

ロール(ROLES)更新履歴

Timestamp	RoleID	Action	UA
200504010000	ロール 1	add	所属組織:資材部∧役職:課長
200504010000	ロール 2	add	所属組織：人事部
200506010000	ロール 3	add	役職：課長

組織(ORGS)更新履歴

Timestamp	OrgID	Action	Attributes
200504010000	資一課	add	資材部
200504010000	人一課	add	人事部

許可(PRMS)更新履歴

Timestamp	ObjID	Action	OPS	PA
200504010000	扉 1	add	通行	ロール 1
200504010000	人事システム	add	起動	ロール 2
200506010000	人事システム	modify	起動	ロール 2∨ロール 3

監査ログ

Timestamp	UserID	OPS	ObjID
200504010900	ユーザ A	通行	扉 1
200506010900	ユーザ B	起動	人事システム
200506010900	ユーザ A	起動	人事システム

図 8 監査ログシステムの構成
Fig. 8 Architecture of audit log system.

4.2 多バージョン RBAC 情報の静的管理によるセキュリティログトラッキング

図 8 に RBAC の多バージョン管理データベースを利用した監査ログシステムの全体構成を示す．RBAC 情報であるユーザ更新履歴，組織更新履歴，ロール更新履歴，許可更新履歴の各テーブルによって，それぞれの更新履歴を，監査ログによって，各セキュリティコンポーネントで実施された行為の履歴を管理することとする．Timestamp は，認証・認可用ディレクトリに対する更新トランザクションの時刻印を示す．各更新履歴テーブルの項目 Action は，認証・認可用ディレクトリに対する更新操作である，add/delete/modify の 3 種類の値のいずれかを持つ．また，Attributes，UA，PA は各項目の属性情報を示す．監査ログに対応する時点での RBAC 情報を関連づけるため，図 8 に示したテーブルを加えて，静的管理するものとする．更新履歴テーブルは更新という操作を管理するのに対して，これらのテーブルは，値の有効期間を示す．**図 9** に示すテーブルは，2.3.2 項に示した競合多バージョン直列可能を満たすように，ディレクトリの更新時に，手続 4.1 の手続によって，レコードを追加するものとする．なお，null 値として，すべての時刻印 t_i について，$null_{min} < t_i$，および，$t_i < null_{max}$ となる 2 種類を用いる．

手続 4.1：

(1) add

BeginTS を更新トランザクションの時刻印に，EndTS を $null_{max}$ として，新しい行を挿入する．

(2) delete

ユーザ，組織，ロール，許可に対応する ID で，EndTS が $null_{max}$ 値の行を選択し，EndTS を，更新トランザクションの時刻印に更新する．

ユーザ(USERS)

BeginTS	EndTS	UserID	Attributes
200504010000	200507010000	ユーザ A	資一課，課長
200506010000	$null_{max}$	ユーザ B	人一課，担当

組織(ORGS)

BeginTS	EndTS	OrgID	Attributes
200504010000	$null_{max}$	資一課	資材部
200504010000	$null_{max}$	人一課	人事部

ロール(ROLES)

BeginTS	EndTS	RoleID	UA
200504010000	$null_{max}$	ロール 1	所属組織:資材部∧役職:課長
200504010000	$null_{max}$	ロール 2	所属:人事部
200506010000	$null_{max}$	ロール 3	役職:課長

許可(PRMS)

BeginTS	EndTS	ObjID	OPS	PA
200504010000	$null_{max}$	扉 1	開錠	ロール 1
200504010000	200506010000	人事システム	起動	ロール 2
200506010000	$null_{max}$	人事システム	起動	ロール 2∨ロール 3

図 9 RBAC 情報の静的管理テーブル
Fig. 9 Static management tables for RBAC information.

(3) modify
上記，delete，add 両者の更新，挿入を行う． □
これらのテーブルを用いて，2.3.2 項に示した競合多バージョン直列可能となるように，手続 4.2 に，指定されたユーザ，およびオブジェクトに関して，ユーザ属性情報，組織，ロール，許可を監査ログに付加したトラッキングを行う手続きを示す．

手続 4.2：
・特定ユーザのトラッキング
(1) トラッキングを行うユーザ，および，期間を指定する．
(2) 指定されたユーザ，期間の条件に合うログを，監査ログテーブルから選択する．
(3) (2) で選択された各行に対して以下の操作を行う．
(3-1) UserID が一致し，$BeginTS \leqq Timestamp < EndTS$ を満足する行をユーザテーブルから選択する．
(3-2) (3-1) で選択された Attributes 内の組織に対応する OrgID が一致し，$BeginTS \leqq Timestamp < EndTS$ を満足する行を組織テーブルから選択する．
(3-3) 同様の手順で，該当するロール，許可を選択する．
(3-4) 選択された許可の operation が，監査ログの operation に矛盾しないことを確認するとともに，ユーザ属性情報，組織，ロール，許可を付加して出力する．
・特定オブジェクトのトラッキング
(1) トラッキングを行うオブジェクト，操作，および，期間を指定する．
(2) 指定されたオブジェクト，操作，期間の条件に合うログを，監査ログテーブルから選択する．
(3) (2) で選択された各行に対して，特定ユーザのトラッキングと同様の手続きを行い，監査ログの operation に矛盾しないことを確認するとともに，ユーザ属性情報，組織，ロール，許可を付加して出力する．□

4.3 多バージョン RBAC 情報の動的生成によるセキュリティログトラッキング

4.2 節では，RBAC 情報の更新トランザクションにより，情報の有効期間を持つ新たなテーブルを用意する方式を示した．しかし，この方式では，更新トランザクションの実行時の負荷，有効期間を持つテーブルの維持管理が課題となる．通常，ユーザ，オブジェクトをキーとした詳細なトラッキングは，セキュリティ事故発生時，警告発生時等に限られる処理であり，利用頻度は高くない場合が多い．一方で，図 8 に示した RBAC 情報の変更履歴は，セキュリティの観点から，維持管理が必須であると考えられる．そこで，本節では，RBAC 情報の変更履歴から，監査対象となるログに関連する行のみを選択して，2.3.1 項で示した時刻印方式を多バージョンに適用することにより，トラッキングの対象となる部分のみの有効期間を持つテーブルを動的に生成して，監査を行う方式を示す．

トラッキングトランザクション実行時に，動的に，履歴から抽出されたトランザクションの個々の命令を時刻印方式に従ってシミュレートし，後退復帰する部分を過去のデータとして記録して利用する．動的に生成するテーブルは，図 9 に示した RBAC 情報の静的管理テーブルと同一のスキーマをとることとする．すべての書き込みトランザクションを実行後，読み込みトランザクションを実行するため，各データに対しては，書き込み時刻印のみを管理するのみで十分である．

手続 4.3 に，2.3.2 項に示した競合多バージョン直列可能を満たすように，指定されたユーザ，およびオブジェクトに関して，動的に管理テーブルを生成して，ユーザ属性情報，組織，ロール，認可を監査ログに付加したトラッキングを行うための手続きを示す．

手続 4.3：
・特定ユーザのトラッキング
(1) トラッキングを行うユーザ uid，および，期間（開始時刻 bts，終了時刻 ets）を指定する．
(2) 指定されたユーザ，期間の条件に合う監査ログを監査ログテーブルから選択する．
(3) UserID が uid に一致し，かつ，Timestamp t が，$bts \leqq t \leqq ets$ を満たす行，または，$t < bts$ を満たす行のうち，最大の時刻印を持つ行を，ユーザ更新履歴テーブルから選択する．
(4) 選択された行の時刻印 t_i に従って，Action ごとに以下の操作を行う．
(4-1) add
更新トランザクションの時刻印 t_i を $BeginTS$ に，$EndTS$ を $null_{max}$ として，新しい行を挿入する．uid に一致する行がすでに存在した場合には，以下の操作を行う．
・$BegintTS < t_i \wedge ti < EndTS$ を満足する行 r が存在する場合は，r の EndTS を t_i に，挿入した行の $EndTS$ を r の $EndTS$ に置き換える．
・$t_i < BeginTS$ の行のみが存在する場合は，その最小値 t'_i を求め，挿入した行の $EndTS$ を，t'_i に置き換える．
・$BeginTS$ が $null_{min}$ である行 r が存在する場合は，$EndTS$ をその行の $EndTS$ に置き換え，行 r を削除する．
(4-2) delete
uid に一致する行が存在する場合は，以下の操作を行う．

・$BeginTS < t_i \wedge t_i < EndTS$ を満足する行 r が存在する場合は，$EndTS$ を t_i で置き換える．$EndTS$ が $null_{max}$ でない場合は，$BeginTS$ が $null_{min}$，$EndTS$ が行 r の $EndTS$ となる新しい行を挿入する．
・$BeginTS < t_i$ を満足する行が存在しない場合は，$BeginTS$ が $null_{min}$，$EndTS$ が t_i となる新しい行を挿入する．
uid に一致する行が存在しない場合は，$BeginTS$ が $null_{min}$，$EndTS$ が t_i となる新しい行を挿入する．
(4-3) modify
(4-2) delete，(4-1) add に対応する削除，挿入を行う．
(5) ユーザ情報から参照される組織，ロール，許可について，id をそれぞれに置き換えて，(3)，(4) の手続きを実行する．
(6) (2) で選択された各行に対して，時刻印 t_j に従って，以下の操作を行う．
(6-1) UserID が一致し，$BeginTS \leqq t_j < EndTS$ を満足する行をユーザテーブルから選択する．
(6-2) (6-1) で選択された Attributes 内の組織に対応する OrgID が一致し，$BeginTS \leqq t_j < EndTS$ を満足する行を組織テーブルから選択する．
(6-3) 同様の手順で，該当するロール，許可を選択する．
(6-4) 選択された許可の operation が，監査ログの operation に矛盾しないことを確認するとともに，ユーザ属性情報，組織，ロール，許可を付加して出力する．

・特定オブジェクトのトラッキング

(1) トラッキングを行うオブジェクト，操作，および，期間を指定する．
(2) 指定されたオブジェクト，操作，期間の条件に合うログを，監査ログテーブルから選択する．
(3) (2) で選択された各行に対して，特定ユーザのトラッキング手続 (3)，(4)，(5)，(6) を行い，監査ログの operaion に矛盾しないことを確認するとともに，ユーザ属性情報，組織，ロール，許可を付加して出力する． □

4.4 考　　察

4.4.1 手続 4.1，4.2 と手続 4.3 の比較

セキュリティトラッキングの手法として，手続 4.1，4.2 において，あらかじめ RBAC 情報の静的管理テーブルを生成して，トラッキングを行う方式を，手続 4.3 において，注目する人，オブジェクトに限定して動的に RBAC 情報管理テーブルを生成してトラッキングを行う方式を示した．多バージョン管理するユーザ，組織，ロール，許可のうち，構成要素，更新数が多く，性能に関与するユーザについて，考察する．

ユーザ数 n 人，1 人あたりの異動による平均更新数を m 回/年，1 ユーザあたりに必要とする容量を u Kbyte とすると，t 年の RBAC 情報静的管理テーブルが保持する容量は，$n \times m \times u \times t$ (Kbyte) となる．

たとえば，従業員 5,000 人で，人事異動が，1 回/人年，1 ユーザあたりの容量を，ディレクトリで管理する標準的な人のオブジェクトである inetOrgPerson を用いた場合の 1 Kbyte，10 年間の RBAC 情報静的管理テーブルを保持する場合，

$$5{,}000 \times 1 \times 1\,(\text{Kbyte}) \times 10 = 50{,}000\,(\text{Kbyte})$$

である．1 ユーザあたり，10 件で，ユーザ ID でインインデクス検索が可能なため，通常の RDBMS を利用した場合，トラッキングの際の参照性能は，特に大きな問題とならないと考える．

一方，手続 4.1 と，手続 4.3 を比較すると，RBAC 情報管理情報の生成手続に関しては，静的/動的で大きな差異はないため，トラッキング解析を行う人の数に依存する．以下に目的に応じた選択指針を示す．
(1) 特定ユーザのトラッキング
インシデント発生時に注目する人，オブジェクトに着目してトラッキングを行う．一般に，インシデントの発生が，ユーザの数を上回ることは考えにくいため，手続 4.3 の方式が有効であると考える．
(2) 監査，不審者解析等を目的とした全件解析
ロール等の設定ミス発見，法・ガイドライン遵守，不審者解析等を目的に，ログの全件解析を行う．たとえば，金融庁検査マニュアル[17] では，実行時のアクセス制御判定は，リアルタイム性に限界があるため，事後を加えた二重チェックを推奨している．また，不正行為を発見するための不審行動分析では，さまざまな属性情報をもとに，多次元的に，全件の統計解析を行う手法がとられる[18]．

このように，全ユーザを対象にした統計解析が目的の場合は，手続 4.1，4.2 の方式が有効であると考える．

4.4.2 RBAC 方式と階層型組織 RBAC 方式の比較

ログの監査・解析を行う際，関連づける情報として，本論文で提案した階層型組織 RBAC 方式では，人事情報（所属組織，役職）といった事実と，セキュリティを目的として設定したアクセス制御情報の 2 種類に分離して管理する．一般の RBAC 方式と階層型組織 RBAC の多バージョン管理について考察する．
(1) RBAC 方式

- 人事情報
  ユーザ–グループの関係は，一般に人事情報をもとに，組織，役職を組み合わせて作成される．しかし，可逆は保証されないため，ログ情報を補足

するためには，別途，ユーザ・組織情報の多バージョン管理が必要となる．すなわち，4.1.1 項で示したテーブルと同程度のものが必要となる．

- アクセス制御情報
  アクセス制御で直接用いられる情報として，ユーザ–グループ間の関係の多バージョン管理が必要となる．ユーザ–グループ間の関係の数は，セキュリティポリシに依存し，1 ユーザあたりの所属するグループの数は，1〜20 程度と考えられる．1 ユーザあたりの所属するグループ数を 10 と仮定すると，ユーザ数 × 10 のデータの多バージョン管理が必要となる．すなわち，4.4.1 項で示した静的管理テーブルの 10 倍の件数を必要とする．

(2) 階層型組織 RBAC 方式

- 人事情報
  内包しているため，新たなテーブル管理は不要である．
- アクセス制御情報
  ロールごとに，認可条件式として保持しており，ロールの数に一致し，ユーザ数に依存しない．ロールの総数は，セキュリティポリシに依存するが，ユーザ数に比して，全体に影響を与えない数であることが一般的である．

階層型組織 RBAC 方式の人事情報とセキュリティ情報の分離，および，その間を条件式により間接的に関連づけるというユーザ設定方式が，ログ解析で必要となるマスタ情報の多バージョン管理に効果があることを示した．実装時の選択の総合判断は，3.4 節で示した実行時性能とのトレードオフとなる．

5. おわりに

本論文では，情報漏洩防止対策として，システムで施されるさまざまな行為の実施前のユーザ認証・アクセス制御と，実施後に行われるログの監視・監査・分析に注目し，両者の整合性をとったアイデンティティ・アクセス管理方式について提案した．実施前のユーザ認証・アクセス制御方式として，人事システムと連動した運用管理効率化を実現するため，組織情報をロールから独立させた方式を示した．本方式では，RBAC を構成するロール（ROLES），許可（PRMS）と，人事情報にかかわるユーザ（USERS），組織（ORGS）を分離することにより，一般的な企業構造であるセキュリティ管理部門と人事部門の操作範囲を一致させ，人事異動時の運用コスト削減をはかった．

また，RBAC を構成する各情報の変更履歴を利用した多バージョン並行制御による監査ログトラッキング手法について提案した．多バージョン管理の方式として，ディレクトリ変更時に静的に作成する方式と，変更履歴をもとに参照トランザクションが必要とする部分のみ，動的に生成する方式を示した．ユーザ情報とセキュリティ情報を分離して論理式で関連づける本論文で提案した階層型組織 RBAC と組み合わせると，頻繁に人事異動が行われる企業，官公庁等の組織において，継続的なログの監査を行う場合，有用である．

今後は，システム運用管理の目的で利用される対象（OBJS）の構成管理情報[19] をもとに，監査ログと対象の所在との関連づけを検討する．

参 考 文 献

1) 二井，中嶋，近藤，伊藤：三菱情報漏洩防止ソリューション，三菱電機技報，Vol.78，No.4 (2004).
2) 青木，佐伯，長浜，近藤：個人情報保護法，e 文書法にも対応可能なトータルセキュリティソリューション，三菱電機技報，Vol.79, No.4 (2005).
3) 情報漏洩防止ソリューション．http://www.mitsubishielectric.co.jp/security/info/
4) 情報漏洩防止ソリューション．http://www.mdis.co.jp/security/solution02/index.html
5) ITIL Security Management, *ITIL Book, OGC*, (1999).
6) Ferraiolo, D. and Kuhn, R.: Role-Based Access Control, *Communications of the 15th NIST-NSA National computer security Conference* (1992).
7) Ferraiolo, D., Sandhu, R., Gavrila, S. and Kuhn, R.: Proposed NIST Standard for Role-Based Access Control, *ACM Trans. Information and System Security*, Vol.4, No.3 (2001).
8) Yao, Y., Moody, K. and Bacon, J.: A Model of OASIS Role-Based Access Control and its Support for Active Security, *CSACMAT'01* (2001).
9) OASIS: Core and Hierarchical Role Based Access Control (RBAC) profile of XACML, Version 2.0, Committee Draft 01 (2004).
10) Ramaswamy, C. and Sandhu, R.: Role-Based Access Control Features in Commercial Database Management Systems, *Proc. 21st national Information Systems security Conference* (1998).
11) Smalley, S. and Fraser, T.: A Security Polisy Configuration for the Security-Enhanced Linux (2001). http://www.nsa.gov/selinus/policy.pdf
12) 上林彌彦：データベース，昭晃堂 (1986).
13) Papadimitriou, C.: *The Theory of Database Concurrency Control*, Computer Science Press (1986).
14) Wang, X., Schulzrine, H., Kandlur, D. and

Verma, D.: Measurement and Analysis of LDAP Performance, *International Conference on Measurement and Modeling of Computer Systems, ACM SIGMETRICS* (2000).
15) 日本 IBM: IBM Tivoli Access Manager Base 管理者ガイドバージョン 5.1 (2004).
16) Plumtree, The Enterprise Web Technology Leader, Plumtree Corporate Portal 5.0: Administration Student Guide 日本語版，プラムツリーソフトウェアジャパン株式会社 (2004).
17) 金融庁：金融検査マニュアル（預金等受入金融機関に係る検査マニュアル）(2004).
18) セキュリティリコメンデーションシステム. http://www2.mdit.co.jp/service/logauditor/
19) ITIL Service Support, *ITIL Book, OGC* (2000).

(平成 17 年 6 月 20 日受付)
(平成 17 年 10 月 13 日採録)

（担当編集委員　高倉 弘喜）

近藤　誠一（正会員）

1984 年京都大学大学院工学研究科情報工学専攻修士課程修了．同年三菱電機（株）入社．1989〜1992 年（財）新世代コンピュータ技術開発機構（ICOT）出向．統合データベース，システム間連携，情報セキュリティシステムに関する研究開発に従事．

白木　宏明

1996 年東京工業大学大学院情報理工学研究科数理・計算科学専攻修士課程修了．同年三菱電機（株）入社．イントラネット情報共有システム，運用管理システムの研究開発を経て，現在，情報セキュリティシステムに関する研究開発に従事．

大沼　聡久

1986 年早稲田大学理工学部機械工学科卒業．同年三菱電機（株）入社．言語プロセッサの開発を経て，現在，情報セキュリティシステムに関する研究開発に従事．

小宮　崇

2001 年佐賀大学大学院工学系研究科電子工学専攻修了．同年三菱電機（株）入社．Web シングルサインオンシステムに関する研究開発に従事．

五月女健治（正会員）

1979 年大阪大学基礎工学部情報工学科卒業．同年三菱電機（株）入社．現在，同社より法政大学大学院イノベーション・マネジメント研究科に出向中．Web シングルサインオンシステム，LDAP ディレクトリサービスの応用に関する研究に従事．

虎渡　昌史（正会員）

1983 年慶應義塾大学大学院工学研究科電気工学専攻修士課程修了．同年三菱電機（株）入社．2003 年より三菱電機インフォメーションシステムズ（株）出向．UNIX 開発，ジョブ制御ミドルウェア開発を経て，現在，運用管理技術，情報セキュリティシステムに関する研究開発に従事．