(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4414092号
(P4414092)
(45)発行日　平成22年2月10日(2010.2.10)　(24)登録日　平成21年11月27日(2009.11.27)

(51)Int.Cl.　F I
G06F 21/24 (2006.01)　G06F 12/14 520A

請求項の数 26 (全 26 頁)

(21)出願番号	特願2000-553884 (P2000-553884)
(86)(22)出願日	平成11年6月9日 (1999.6.9)
(65)公表番号	特表2002-517853 (P2002-517853A)
(43)公表日	平成14年6月18日 (2002.6.18)
(86)国際出願番号	PCT/US1999/012914
(87)国際公開番号	WO1999/064947
(87)国際公開日	平成11年12月16日 (1999.12.16)
審査請求日	平成18年6月9日 (2006.6.9)
(31)優先権主張番号	09/096,679
(32)優先日	平成10年6月12日 (1998.6.12)
(33)優先権主張国	米国 (US)

(73)特許権者　500046438
マイクロソフト　コーポレーション
アメリカ合衆国　ワシントン州　９８０５２－６３９９　レッドモンド　ワン　マイクロソフト　ウェイ
(74)代理人　100077481
弁理士　谷　義一
(74)代理人　100088915
弁理士　阿部　和夫
(72)発明者　マイケル　エム.　スイフト
アメリカ合衆国　９８１０５　ワシントン州　シアトル　１　アベニュー　ノースイースト　４２２０

審査官　大塚　良平

最終頁に続く

(54)【発明の名称】制限付きトークンを介した最小権限

(57)【特許請求の範囲】
【請求項１】
リソースの各々に関連づけられたセキュリティ情報に対して、アプリケーションの各々に関連づけられたアクセストークン内の情報に基づいて、リソースへのアプリケーションのアクセスを決定するセキュリティメカニズムを有するシステムにおいて、システムリソースへのアプリケーションのアクセスを制限するコンピュータ実施方法であって、
前記アプリケーションに関する制限情報をストアするステップであって、前記制限情報は、前記リソースへの前記アプリケーションのアクセスに関連するステップと、
前記アプリケーションを実行する要求を受信するステップと、
親トークンおよび前記制限情報に基づいて制限付きアクセストークンを作成するステップであって、前記制限付きアクセストークンは、前記親トークンに関連する削減されたアクセスを提供し、
アクセス情報を前記親トークンから前記制限付きアクセストークンにコピーするステップと、
少なくとも１つの権限を前記親トークンに関連する前記制限付きアクセストークンから削除するステップと
を備えるステップと、
前記制限付きアクセストークンを前記アプリケーションに関連づけるステップと
前記制限付きアクセストークンで前記リソースへの前記アプリケーションのアクセスを試みるステップと、

アクセスが否定された場合、前記親トークンで前記リソースへの前記アプリケーションのアクセスを試みるステップと
を備えたことを特徴とする方法。
【請求項２】
前記アプリケーションを実行するステップと、前記制限付きアクセストークンを前記アプリケーションの前記アクセストークンとして使用して、前記システムリソースへのアクセスを試みるステップとをさらに備えたことを特徴とする請求項１に記載の方法。
【請求項３】
前記アプリケーションに関する制限情報をストアする前記ステップは、前記アプリケーションがアクセスを有している少なくとも１つのファイルを識別するステップを含むことを特徴とする請求項１に記載の方法。
【請求項４】
前記アプリケーションに関する制限情報をストアする前記ステップは、前記アプリケーションを１つのファイルに制限するステップを含むことを特徴とする請求項３に記載の方法。
【請求項５】
前記アプリケーションに関する制限情報をストアする前記ステップは、前記アプリケーションが起動することができる少なくとも１つの他のアプリケーションを識別するステップを含むことを特徴とする請求項１に記載の方法。
【請求項６】
制限付きアクセストークンを作成する前記ステップは、通常のトークンからアクセス情報を前記制限付きアクセストークンにコピーするステップと、少なくとも１つの制限付きセキュリティ識別子を前記制限付きアクセストークンに追加するステップとを含むことを特徴とする請求項１に記載の方法。
【請求項７】
少なくとも１つの制限付きセキュリティ識別子を前記制限付きアクセストークンに追加する前記ステップは、前記アプリケーションに対応する制限付きセキュリティ識別子を追加するステップを含むことを特徴とする請求項６に記載の方法。
【請求項８】
親トークンから制限付きアクセストークンを作成する前記ステップは、前記親トークン内の対応するセキュリティ識別子の属性情報に関連して、前記制限付きアクセストークン内のセキュリティ識別子の属性情報を、該セキュリティ識別子を介して否定のみのアクセスに使用するように変更するステップを含むことを特徴とする請求項１に記載の方法。
【請求項９】
前記アプリケーションの機能の少なくとも一部を少なくとも２つのグループに分割するステップと、少なくとも１つが前記制限付きアクセストークンであるアクセストークンを各グループについて作成するステップと、前記制限付きアクセストークンを前記グループのうち少なくとも１つのグループに関連づけるステップとをさらに備えたことを特徴とする請求項１に記載の方法。
【請求項１０】
前記アプリケーションの機能の少なくとも一部を少なくとも２つのグループに分割する前記ステップは、機能を権限付き部分と権限なし部分に分割するステップを含み、前記制限付きアクセストークンを前記グループのうち少なくとも１つのグループに関連づける前記ステップは、前記制限付きアクセストークンを前記権限なし部分に関連づけるステップを含み、前記親トークンを前記権限付き部分に関連づけるステップをさらに備えたことを特徴とする請求項９に記載の方法。
【請求項１１】
前記親トークンで前記アプリケーションへのアクセスを試みる前記ステップは、前記システムのユーザからの応答を受け取るステップを含むことを特徴とする請求項１に記載の方法。

【請求項１２】
前記ユーザにプロンプトで前記応答を促すステップをさらに備えたことを特徴とする請求項１１に記載の方法。
【請求項１３】
前記親トークンは、関連する増大したアクセスを伴うより高い親トークンを有し、前記親トークンで前記アプリケーションへのアクセスを試みる前記ステップは、前記システムが前記親トークンに対してアクセスを拒否した場合に、前記より高い親トークンで前記アプリケーションへのアクセスを試みるステップをさらに含むことを特徴とする請求項１に記載の方法。
【請求項１４】
親トークンから制限付きアクセストークンを作成する前記ステップは、前記親トークン内の対応するセキュリティ識別子の属性情報に関連して、前記制限付きアクセストークン内のセキュリティ識別子の属性情報を、該セキュリティ識別子を介して否定のみのアクセスに使用するように変更するステップを含むことを特徴とする請求項１に記載の方法。
【請求項１５】
前記親トークンは通常のトークンであり、親トークンから制限付きアクセストークンを作成する前記ステップは、制限付きセキュリティ識別子を前記親トークンに関連する前記制限付きアクセストークンに追加するステップを含むことを特徴とする請求項１に記載の方法。
【請求項１６】
前記親トークンは少なくとも１つの制限付きセキュリティ識別子をその中に有する制限付きアクセストークンであり、親トークンから制限付きアクセストークンを作成する前記ステップは、少なくとも１つの制限付きセキュリティ識別子を前記親トークンに関連する前記制限付きアクセストークンから削除するステップを含むことを特徴とする請求項１に記載の方法。
【請求項１７】
前記制限付きアクセストークンで前記アプリケーションにアクセスを試みる前記ステップは、プロセスを前記制限付きアクセストークンに関連づけるステップを含むことを特徴とする請求項１に記載の方法。
【請求項１８】
前記親トークンで前記アプリケーションにアクセスを試みるステップ前記は、プロセスを前記親トークンに関連づけるステップを含むことを特徴とする請求項１に記載の方法。
【請求項１９】
システムリソースへのアプリケーションのアクセスを制限する、コンピューティング環境におけるシステムであって、前記システムは、リソースの各々に関連づけられたセキュリティ情報に対して、アプリケーションの各々に関連づけられたアクセストークン内の情報に基づいて、リソースへのアプリケーションのアクセスを決定するセキュリティメカニズムを有し、
前記アプリケーションに関する制限情報をストアする手段であって、前記制限情報は、前記リソースへの前記アプリケーションのアクセスに関連する手段と、
前記アプリケーションを実行する要求を受信する手段と、
親トークンおよび前記制限情報に基づいて制限付きアクセストークンを作成する手段であって、前記制限付きアクセストークンは、前記親トークンに関連する削減されたアクセスを提供し、
アクセス情報を前記親トークンから前記制限付きアクセストークンにコピーする手段と、
少なくとも１つの権限を前記親トークンに関連する前記制限付きアクセストークンから削除する手段と
を備える手段と、
前記制限付きアクセストークンを前記アプリケーションに関連づける手段と

前記制限付きアクセストークンで前記リソースへの前記アプリケーションのアクセスを試みる手段と、
アクセスが否定された場合、前記親トークンで前記リソースへの前記アプリケーションのアクセスを試みる手段と
を備えたことを特徴とするシステム。
【請求項２０】
前記アプリケーションを実行する手段と、前記制限付きアクセストークンを前記アプリケーションの前記アクセストークンとして使用して、前記システムリソースへのアクセスを試みる手段とをさらに備えたことを特徴とする請求項１９に記載のシステム。
【請求項２１】
前記アプリケーションに関する制限情報をストアする前記手段は、前記アプリケーションがアクセスを有している少なくとも１つのファイルを識別する手段を含むことを特徴とする請求項１９に記載のシステム。
【請求項２２】
前記アプリケーションに関する制限情報をストアする前記手段は、前記アプリケーションを１つのファイルに制限する手段を含むことを特徴とする請求項２１に記載のシステム。
【請求項２３】
前記アプリケーションに関する制限情報をストアする前記手段は、前記アプリケーションが起動することができる少なくとも１つの他のアプリケーションを識別する手段を含むことを特徴とする請求項１９に記載のシステム。
【請求項２４】
制限付きアクセストークンを作成する前記手段は、通常のトークンからアクセス情報を前記制限付きアクセストークンにコピーする手段と、少なくとも１つの制限付きセキュリティ識別子を前記制限付きアクセストークンに追加する手段とを含むことを特徴とする請求項１９に記載のシステム。
【請求項２５】
少なくとも１つの制限付きセキュリティ識別子を前記制限付きアクセストークンに追加する前記手段は、前記アプリケーションに対応する制限付きセキュリティ識別子を追加する手段を含むことを特徴とする請求項２４に記載のシステム。
【請求項２６】
親トークンから制限付きアクセストークンを作成する前記手段は、前記親トークン内の対応するセキュリティ識別子の属性情報に関連して、前記制限付きアクセストークン内のセキュリティ識別子の属性情報を、該セキュリティ識別子を介して否定のみのアクセスに使用するように変更する手段を含むことを特徴とする請求項１９に記載のシステム。
【発明の詳細な説明】
【０００１】
（発明の分野）
本発明は、一般にコンピュータシステムに関し、より詳細には、コンピュータシステムのセキュリティの改善に関する。
【０００２】
（発明の背景）
コンピューティングでは、あるタスクを実行するために必要な権限より多い権限を有するユーザによってタスクが実行されると、ユーザが不注意でコンピュータリソースに害を与えるリスクが増大する。例として、一組のファイルが管理者権限を持つユーザによってのみ削除できる場合、管理者が、管理者によって実施される必要のない別のタスクを実行している時に、不注意でこれらのファイルを削除する可能性がある。管理者がより少ない権限を有するユーザだった場合、目的とするタスクは依然として実行されるが、不注意な削除は起こり得ないであろう。
【０００３】

したがって、コンピュータセキュリティにおける認識された目的は最小権限というコンセプトであり、この中でタスクを実行するユーザはそのタスクを実行するために必要な絶対最小限の権限（またはグループのメンバーシップなどの識別）で実行すべきである。しかし、ユーザのアクセス権および権限を追加したり削除したりする便利な方法はない。たとえば、ウィンドウズＮＴオペレーティングシステムでは、ユーザがログオンすると、そのユーザの証明に基づいてそのユーザ用のアクセストークンが構築される。アクセストークンは、ユーザがそのセッションに関して有することになるアクセス権および権限を決定する。この結果、ユーザはこのセッションの間および任意の将来のセッションについて、試みられた各タスクに関してこれらの権限を有することになる。理想的には管理者が多数の識別をセットアップし、各タスクに関して異なる権利を有する異なるユーザとしてログオンできるが、これは煩わしく複雑過ぎる。さらに、自動的な実行がないので、安全を意識している管理者でも、ただ目的としないアクションを実行する可能性を避けるためにのみ、異なるタスクが実行されるたびに新しい識別でログオフし、ログオンしなおす見込みはない。

【０００４】

簡単に言えば、権限レベルまたはアクセス権を変更する便利な方法、および、ドメイン管理者によって作成された権限より細かい細分性でさらに権限を制限する方法がないということである。他のオペレーティングシステムは、最小権限ではほとんど実行できないという同様な問題を有する。

【０００５】

（発明の概要）

簡単に言えば、本発明は制限付きのアクセストークンを介して最小権限、または何らかの方法で削減されたアクセスを実行するためのメカニズムを提供する。制限付きのアクセストークンはセキュリティメカニズムが、プロセスが既存のアクセストークンの変更された制限付きのバージョンに基づいたリソースへのアクセスを有するかどうかを決定することを可能にする。制限付きトークンは既存のトークンに基づき、既存のトークンより少ないアクセスを有する。制限付きトークンは、親トークン内のアクセスを許可する１つまたは複数のセキュリティ識別子の属性を、制限付きトークン内でアクセスを否定する設定に変更する、および／または親トークン内に存在する１つまたは複数の権限を制限付きトークンから削除することによって、既存の（親）トークンから作成できる。さらに、制限付きのセキュリティ識別子は制限付きトークン内に置くことができる。

【０００６】

使用に際しては、プロセスが制限付きトークンに関連づけられ、制限付きのプロセスがリソースについてアクションを実行しようと試みると、カーネルモードのセキュリティメカニズムはまず、ユーザに基づいたセキュリティ識別子および目的のタイプのアクションを、そのリソースに関連づけられた識別子およびアクションのリストに対して比較する。制限付きトークン内に制限付きのセキュリティ識別子がない場合、アクセスはこの第１の比較の結果によって決定される。制限付きトークン内に制限付きのセキュリティ識別子がある場合、このアクションに関する第２のアクセスチェックは、制限付きのセキュリティ識別子を、リソースに関連づけられた識別子およびアクションのリストに対して比較する。制限付きセキュリティ識別子を有するトークンを使用して、このプロセスは第１のアクセスチェックおよび第２のアクセスチェックの両方がパスした場合のみ、リソースへのアクセスを許可される。

【０００７】

アプリケーションプログラムは、関連づけられた制限情報をストアしている場合がある。アプリケーションが起動されると、制限情報に基づいてそのアプリケーションに関して制限付きトークンが作成される。この方法で、削減されたアクセスが自動的にそのアプリケーションに関して実行される。アプリケーションは権限付き部分および権限なし部分など、異なるアクセスレベルに分割でき、これによって、そのアプリケーションを介してユーザが実行できるアクションを自動的に制限する。また、システムはユーザのプロセスを制

限付きトークンで実行することを強制し、次いで、一時的にユーザの通常のトークンで実行することによって制限されているアクションを特にオーバーライドする明確なアクションをユーザから要求することによって、削減されたアクセスで実行することができる。

【０００８】

他の利点は図面と共に以下の詳細な説明から明らかになろう。

【０００９】

（詳細な説明）

例としてのオペレーション環境

図１および以下の議論は、本発明が実現できる適切なコンピューティング環境の簡単で一般的な説明を提供することを意図している。必要ではないが、本発明は、パーソナルコンピュータによって実行されるプログラムモジュールなどのコンピュータ実行可能な命令の一般的なコンテキスト内で説明される。一般に、プログラムモジュールはルーチン、プログラム、オブジェクト、構成要素、データ構造などを含み、特定のタスクを実行するか、特定の抽象的なデータタイプを実装する。さらに、当業者であれば本発明は、手持ちデバイス、マルチプロセッサシステム、マイクロプロセッサに基づいたまたはプログラミング可能な消費者電気製品、ネットワークＰＣ、ミニコンピュータ、メインフレームコンピュータなどを含む他のコンピュータシステム構成でも実行できることが理解されるであろう。本発明はまた、通信ネットワークを介してリンクされた遠隔処理デバイスによってタスクが実行される分散コンピューティング環境内でも実行できる。分散コンピューティング環境では、プログラムモジュールはローカルメモリ記憶デバイスおよび遠隔メモリ記憶デバイスの両方に位置する場合がある。

【００１０】

図１を参照すると、本発明を実現するための一例としてのシステムは、従来のパーソナルコンピュータ２０などの形の汎用コンピューティングデバイスを含み、汎用コンピューティングデバイスは処理ユニット２１、システムメモリ２２、およびシステムメモリから処理ユニット２１までを含む種々のシステム構成要素を結合するシステムバス２３を含む。システムバス２３はいくつかのタイプのバス構成のうち任意の構成であって、その中にはメモリバスまたはメモリコントローラ、周辺バス、種々のバスアーキテクチャのうち任意のアーキテクチャを使用するローカルバスを含んでもよい。システムメモリは読取り専用メモリ（ＲＯＭ）２４およびランダムアクセスメモリ（ＲＡＭ）２５を含む。起動時などにパーソナルコンピュータ２０内の要素の間で情報を転送する役に立つ基本的なルーチンを含む基本入出力システム２６（ＢＩＯＳ）は、ＲＯＭ２４内に格納されている。パーソナルコンピュータ２０はさらに、図示されてはいないがハードディスクから読み出したり書き込んだりするためのハードディスクドライブ２７、取外し可能磁気ディスク２９から読み出したり書き込んだりするための磁気ディスクドライブ２８、および、ＣＤ－ＲＯＭまたは他の光媒体など、取外し可能光ディスク３１から読み出したり書き込んだりするための光ディスクドライブ３０を含む場合がある。ハードディスクドライブ２７、磁気ディスクドライブ２８、および光ディスクドライブ３０はそれぞれ、ハードディスクドライブインタフェース３２、磁気ディスクドライブインタフェース３３、および光ドライブインタフェース３４によってそれぞれシステムバス２３に接続されている。ドライブとその関連付けられたコンピュータ可読媒体は、パーソナルコンピュータ２０のためにコンピュータ可読命令、データ構造、プログラムモジュールおよびほかのデータの、不揮発性の格納を提供する。ここに説明された例としての環境は、ハードディスク、取外し可能磁気ディスク２９および取外し可能光ディスク３１を使用しているが、当業者であれば磁気カセット、フラッシュメモリカード、ディジタルビデオディスク、ベルヌーイカートリッジ、ランダムアクセスメモリ（ＲＡＭ）、読取り専用メモリ（ＲＯＭ）などの、コンピュータによってアクセス可能なデータを格納できる他のタイプのコンピュータ可読媒体も例としての動作環境内で使用できることが理解されるであろう。

【００１１】

オペレーティングシステム３５（好ましくはＷｉｎｄｏｗｓ　ＮＴ）、１つまたは複数の

アプリケーションプログラム３６、ほかのプログラムモジュール３７およびプログラムデータ３８を含むいくつかのプログラムモジュールがハードディスク、磁気ディスク２９、光ディスク３１、ＲＯＭ２４またはＲＡＭ２５に格納できる。ユーザは、キーボード４０およびポインティングデバイス４２などの入力デバイスを介してパーソナルコンピュータ２０にコマンドおよび情報を入力できる。他の入力デバイス（図示せず）は、マイクロフォン、ジョイスティック、ゲームパッド、サテライトディッシュ、スキャナなどを含む。これらおよび他の入力デバイスはしばしば、システムバスに結合されたシリアルポートインタフェース４６を介して処理ユニット２１に接続されているが、パラレルポート、ゲームポートまたは汎用シリアルバス（ＵＳＢ）などの他のインタフェースによって接続されている場合もある。モニタ４７または他のタイプの表示デバイスもまた、ビデオアダプタ４８などのインタフェースを介してシステムバス２３に接続されている。モニタ４７の外に、パーソナルコンピュータは典型的にはスピーカおよびプリンタなどの他の周辺出力デバイス（図示せず）を含む。

【００１２】

パーソナルコンピュータ２０は、遠隔コンピュータ４９など１つまたは複数の遠隔コンピュータへの論理接続を使用してネットワーク化された環境内で動作する場合がある。遠隔コンピュータ４９は、他のパーソナルコンピュータ、サーバ、ルータ、ネットワークＰＣ、ピアデバイスまたは他の共通ネットワークノードであり、典型的にはパーソナルコンピュータに関連して上記に説明された要素の多くまたはすべてを含むが、メモリ記憶デバイス５０のみが図１に示されている。図１に描かれた論理接続は、ローカルエリアネットワーク（ＬＡＮ）５１および広域ネットワーク（ＷＡＮ）５２を含む。このようなネットワーク化環境はオフィス、全社的コンピュータネットワーク、イントラネットおよびインターネットに普通に見られる。

【００１３】

パーソナルコンピュータ２０はＬＡＮネットワーク化環境内で使用されると、ネットワークインタフェースまたはアダプタ５３を介してローカルネットワーク５１に接続される。パーソナルコンピュータ２０はＷＡＮネットワーク化環境内で使用されると、典型的にはモデム５４または他の手段を含み、インターネットなど広域ネットワーク５２上で通信を確立する。モデム５４は内部的である場合も外部的である場合もあるが、シリアルポートインタフェース４６を介してシステムバス２３に接続されている。ネットワーク化された環境では、パーソナルコンピュータ２０またはその一部に関して描かれたプログラムモジュールは、遠隔のメモリ記憶デバイス内に格納される場合がある。示されたネットワーク接続は例としてのものであって、コンピュータの間で通信リンクを確立する他の手段も使用できることが明らかであろう。

【００１４】

一般的なセキュリティモデル

本発明の好ましいセキュリティモデルは、Ｗｉｎｄｏｗｓ　ＮＴセキュリティモデルに関して説明される。しかし、本発明をＷｉｎｄｏｗｓ　ＮＴオペレーティングシステムに限定する意図はなく、逆に、本発明はオペレーティングシステムのレベルでセキュリティチェックを実行する任意のメカニズムで動作し、利益を提供することを目的としている。

【００１５】

一般に、Ｗｉｎｄｏｗｓ　ＮＴオペレーティングシステムでは、ユーザはプロセス（およびそのスレッド）を介してシステムのリソースにアクセスすることによってタスクを実行する。簡単に説明するために、プロセスおよびそのスレッドは概念上等価と見なされ、以後は簡単にプロセスと呼ぶ。また、Ｗｉｎｄｏｗｓ　ＮＴ内ではオブジェクトによって表される、ファイル、共有メモリおよび物理デバイスを含むシステムのリソースは、本明細書では通常リソースまたはオブジェクトと呼ばれる。

【００１６】

ユーザがＷｉｎｄｏｗｓ　ＮＴオペレーティングシステムにログオンし認証されると、セキュリティコンテキストがそのユーザのためにセットアップされ、この中にはアクセスト

ークン６０の構築も含まれる。図２の左側に示すように、従来のユーザベースのアクセストークン６０は、ＵｓｅｒＡｎｄＧｒｏｕｐｓフィールド６２を含む。ＵｓｅｒＡｎｄＧｒｏｕｐｓフィールド６２は、セキュリティ識別子、すなわち、ユーザの証明およびそのユーザが属するグループ（たとえば編成内のグループ）を識別する１つまたは複数のグループＩＤ６６に基づいたセキュリティ識別子（セキュリティＩＤまたはＳＩＤ）を含む。トークン６０はまた、そのユーザに割り当てられた任意の権限を一覧する権限フィールド６８を含む。たとえば、このような権限の１つは、管理レベルのユーザに、特定のアプリケーションプログラミングインタフェース（ＡＰＩ）を介してシステムクロックを設定する能力を与える場合がある。権限は、アクセス制御チェック、これは次に説明されるが、権限がない場合にオブジェクトへのアクセスを許可する前に実行されるアクセス制御チェックに優先することに注意されたい。

【００１７】

次に詳細に説明され、図３に一般に示されるように、オブジェクト７２へのアクセスを所望するプロセス７０は、所望するアクセスのタイプを指定し（たとえばファイルオブジェクトへの読取り／書き込みアクセスを得るなど）、および、カーネルレベルでは関連付けられたトークン６０をオブジェクトマネジャー７４に提供する。オブジェクト７２はそれに関連付けられたカーネルレベルのセキュリティ記述子７６を有し、オブジェクトマネジャー７４はセキュリティ記述子７６およびトークン６０をセキュリティメカニズム７８に提供する。セキュリティ記述子７６の内容は、典型的にはオブジェクトの所有者（たとえば制作者）によって決定され、一般に（任意に）アクセス制御エントリーのアクセス制御リスト（ＡＣＬ）８０を含み、各エントリーについて、そのエントリーに対応する１つまたは複数のアクセス権（許可されたアクションまたは拒否されたアクション）を含む。各エントリーはタイプ（拒否または許可）インジケータ、フラグ、セキュリティ識別子（ＳＩＤ）およびアクセス権をビットマスクの形で含み、各ビットは許可に対応する（たとえば、１つのビットは読取りアクセス、１つのビットは書込みアクセス、など）。セキュリティメカニズム７８はトークン６０内のセキュリティＩＤおよびプロセス７０によって要求されたアクション（複数可）のタイプをＡＣＬ８０内のエントリーと比較する。許可されたユーザまたはグループの一致が発見され、所望のアクセスのタイプがそのユーザまたはグループに対して許可可能な場合、オブジェクト７２へのハンドルはプロセス７０に戻されるが、その他の場合はアクセスは拒否される。

【００１８】

例として、ユーザを「会計」グループのメンバーとして識別するトークンを持つユーザが読取りおよび書込みアクセスで特定のファイルオブジェクトにアクセスしたいとする。ファイルオブジェクトが、ＡＣＬ８０のエントリー内で許可されたタイプの「会計」グループ識別子を有し、そのグループが読取りおよび書き込みアクセスを使用可能にする権利を有する場合、読取りおよび書込みアクセスを許可するハンドルは戻されるが、その他の場合はアクセスは拒否される。効率上の理由から、セキュリティチェックはプロセス７０がまずオブジェクト７２（作成または開く）にアクセスしようと試みた時のみ実行され、したがってそのオブジェクトに対するハンドルはそれを介して実行できるアクションを制限するように、アクセス情報のタイプを格納することに注意されたい。

【００１９】

セキュリティ識別子７６はまた、システムＡＣＬまたはＳＡＣＬ８１を含み、これは監査されるべきクライアントアクションに対応するタイプ監査のエントリーを含む。各エントリー内のフラグは監査が成功したオペレーションまたは失敗したオペレーションのどちらを監視するかを示し、エントリー内のビットマスクは監査されるべきオペレーションのタイプを示す。エントリー内のセキュリティＩＤは、監査されるユーザまたはグループを示す。たとえば、ファイルオブジェクトへの書込みアクセスを有しないグループのメンバーがそのファイルに書き込もうと試みた場合いつでも決定できるように特定のグループが監査される状況を考えてみる。そのファイルオブジェクトに関するＳＡＣＬ８１は、その中にグループセキュリティ識別子と、適切に設定された失敗フラグおよび書込みアクセスビ

ットを有する監査エントリーを含む。その特定のグループに属するクライアントがそのファイルオブジェクトに書き込みしようとして失敗するといつでも、そのオペレーションは記録される。

【００２０】

ＡＣＬ８０は、（すべての権利または選択された権利に関して）グループユーザにアクセスを許可する識別子ではなくて、グループユーザのアクセスを拒否するためにマークされる。１つまたは複数の識別子を含む場合があることに注意されたい。たとえば、ＡＣＬ８０内にリストされた１つのエントリーは、他の場合に、「グループ$_3$」のメンバーにオブジェクト７２へのアクセスを許可するが、ＡＣＬ８０内の他のエントリーは特に、「グループ$_{24}$」のすべてのアクセスを拒否する場合がある。トークン６０が「グループ$_{24}$」セキュリティＩＤを含んでいる場合、アクセスは「グループ$_3$」のセキュリティＩＤの存在にかかわらず拒否されることになる。もちろん、セキュリティチェックが正しく機能するために、セキュリティチェックは、”グループ$_3$”エントリーを介したアクセスを許可しないようにアレンジされ、その後に、すべてのＤＥＮＹ（拒否）エントリーをＡＣＬ８０の前面に置くなどによって、グループ$_{24}$エントリーの「ＤＥＮＹ　ＡＬＬ（すべて拒否）」状態をチェックする。この構成（ａｒｒａｎｇｅｍｅｎｔ）は、グループの残りのメンバーの各々を個別にリストしてそのアクセスを許可する必要なく、グループの、１人または複数の分離したメンバーを個別にＡＣＬ８０内で排除できるので、向上した効率を提供することが明らかであろう。

【００２１】

アクセスのタイプを指定する代わりに、コーラー（ｃａｌｌｅｒ）はＭＡＸＩＭＵＭ＿ＡＬＬＯＷＥＤアクセスを要求することもでき、これによって、１つのアルゴリズムが通常のＵｓｅｒＡｎｄＧｒｏｕｐｓリスト対ＡＣＬ８０内の各々のエントリーに基づいて、許可される最大のアクセスタイプを決定することに注意されたい。より詳しくは、アルゴリズムは所与のユーザのための権利を蓄積している識別子のリストをウォークダウンする（すなわち、種々のビットマスクをＯＲする）。権利が一度蓄積されると、ユーザに蓄積された権利が与えられる。しかし、ウォークスルーの間にユーザ識別子またはグループ識別子および要求された権利に一致する拒否エントリーが発見されると、アクセスは拒否される。

【００２２】

制限付きのトークン

本発明の一態様によれば、制限付きのトークンは次のように、既存のアクセストークン（制限付きまたは制限なし）から作成される。また次に説明するように、制限付きトークンが任意の制限付きセキュリティＩＤを含む場合、そのトークンは追加のアクセスチェックを受け、その中で制限付きセキュリティＩＤはオブジェクトのＡＣＬ内のエントリーに対して比較される。制限つきトークンはまた、本発明と同じ譲受人に付与されている“制限付きのトークンを使用したセキュリティモデル”というタイトルの共に係属中の米国特許出願内で説明されており、その特許出願は本発明と共に同時に出願されており、制限つきトークンはその中で完全に引例によって含まれている。

【００２３】

制限付きトークンの主な用途は、プロセスに、それ自体のトークンの制限付きのバージョンを持つ新しいプロセスを作成させることである。制限付きのプロセスはついで、リソースに関して実行できるアクションにおいて制限される。たとえば、ファイルオブジェクトリソースは、関連付けられた制限付きトークンの中に同じマイクロソフトワード制限付きＳＩＤを有する制限付きプロセスのみがそのファイルオブジェクトにアクセスできるように、単一の制限付きＳＩＤ、すなわち、マイクロソフトワードのアプリケーションプログラムを識別する単一の制限付きＳＩＤをそのＡＣＬの中に有する場合がある。オリジナルユーザはさらにオブジェクトにアクセスする必要があるので、ＡＣＬもまた、ユーザアクセスおよびマイクロソフトワードプログラムを許可するアクセス制御エントリーを含む必要があることに注意されたい。そのため、たとえば、ブラウザを介してダウンロードされ

たコードなど、信用できないコードは、制限付きプロセス内で実行される。制限付きプロセスは、制限付きトークン内にマイクロソフトワード制限付きセキュリティＩＤを有しない（不正である可能性のある）コードがファイルオブジェクトへアクセスすることを防ぐ。

【００２４】

セキュリティ上の理由から、異なるトークンを伴うプロセスの作成は通常は、ＳｅＡｓｓｉｇｎＰｒｉｍａｒｙＴｏｋｅｎ権限として知られる権限を必要とする。しかし、プロセスが制限付きトークンと関連付けられるようにするために、制限付きトークンが主トークンから導出されている場合に、プロセス管理は別のプロセスへの十分なアクセスを持つ１つのプロセスが、その主トークンを制限付きトークンに変更できるようにする。新しいプロセスのトークンのＰａｒｅｎｔＴｏｋｅｎＩｄを既存のプロセスのトークンのＴｏｋｅｎＩｄと比較することによって、オペレーティングシステム３５はプロセスがそれ自体の制限付きバージョンを作成しているだけであることを確認する。

【００２５】

制限付きトークン８４は親トークンより少ないアクセスしか有せず（すなわち、親トークンの権利と権限のサブセットを有するため）、ユーザ情報またはグループ情報のみに基づいてアクセスを許可または拒否するのみでなく、そのオブジェクトにアクセスしようと試みるプロセスのタイプ（またユーザまたはグループ）に基づいてオブジェクトへのアクセスを防止することなどが可能である。制限付きのトークンはまた、親トークンがこれらのＳＩＤを介してアクセスを許可している場合でも、「ＵＳＥ＿ＦＯＲ＿ＤＥＮＹ＿ＯＮＬＹ」と特にマークされた１人または複数のユーザまたはグループのセキュリティＩＤを介したアクセスを許可しない場合もあり、および／または、親トークンに存在する権限を削除してしまう場合もある。

【００２６】

したがって、アクセスを削減する１つの方法は、制限付きトークン内の１つまたは複数のユーザセキュリティ識別子および／またはグループのセキュリティ識別子の属性を変更して、アクセスを許可するのではなくアクセスを許可できないようにすることである。ＵＳＥ＿ＦＯＲ＿ＤＥＮＹ＿ＯＮＬＹとマークされたセキュリティＩＤは、アクセスを許可する目的のためには効果的に無視されるが、そのセキュリティＩＤに関して「ＤＥＮＹ（拒否）」エントリーを有するＡＣＬは依然としてアクセスが拒否される。例として、制限付きトークン８４（図３）内のグループ$_2$のセキュリティＩＤが、ＵＳＥ＿ＦＯＲ＿ＤＥＮＹ＿ＯＮＬＹとマークされている場合、ユーザのプロセスがグループ$_2$を許可されたものとしてリストしているＡＣＬ８０を有するオブジェクト７２にアクセスしようと試みた時、このエントリーは効果的に無視され、プロセスは他のなんらかのセキュリティＩＤによってアクセスを得なければならないことになる。しかし、ＡＣＬ８０が要求されたタイプのアクションに関して、グループ$_2$をＤＥＮＹとリストしたエントリーを含んでいる場合、１度テストされると、他のセキュリティＩＤにかかわらずアクセスは許可されない。

【００２７】

セキュリティＩＤはいくつかのオブジェクトのＡＣＬの中で「ＤＥＮＹ」としてマークされている可能性があり、それによってその識別子を削除するとこれらのオブジェクトへのアクセスを拒否するのではなく許可することになってしまうため、ユーザのトークンからセキュリティＩＤをただ除去するだけではオブジェクトへのアクセスは安全に削減できないことに注意されたい。したがって、本発明は制限付きのトークン内でＳＩＤの属性がＵＳＥ＿ＦＯＲ＿ＤＥＮＹ＿ＯＮＬＹに変更できるようにする。さらに、このＵＳＥ＿ＦＯＲ＿ＤＥＮＹ＿ＯＮＬＹセキュリティチェックをオフにするためのメカニズムは提供されていない。

【００２８】

制限付きトークン内でアクセスを削除する別の方法は、親トークンに関して１つまたは複数の権限を削除することである。たとえば、管理権限を伴う通常のトークンを有するユーザは、そのユーザが特にシステムに別の情報を与えなければ、そのユーザのプロセスが権

限のない制限付きトークンで実行するようにシステムを設定する場合がある。これはユーザが管理者の範囲内で意図的に行動していない時に発生する可能性のある、偶然のエラーを防ぐことを理解されたい。同様に、プログラムがユーザの権限に応じて異なるモードで実行するように開発され、これによって管理レベルのユーザはあるオペレーションを実行するために管理権限でプログラムを実行しなければならないが、より基本的なオペレーションを実行するためには低減された権限でオペレーションを実行しなければならない場合がある。ここで再び、これはこのようなユーザが通常のオペレーションを実行しようとしているだけなのに、高められた権限で実行している時に生じる可能性のある深刻なエラーを防ぐのに役立つ。

【００２９】

トークンのアクセスを低減するさらに別の方法は、そこに制限付きのセキュリティＩＤを追加することである。制限付きのセキュリティＩＤは、プロセス、リソースオペレーションなどを表す数であり、ＧＵＩＤへ接頭辞を追加するかまたは暗号ハッシュなどを介して生成された数字を追加するなどによってユニークになっており、これらのセキュリティＩＤを他のセキュリティＩＤから区別する情報を含む場合もある。本発明に必ず必要ではないが便利のために、ＧＵＩＤをセキュリティＩＤに変換したり、セキュリティＩＤを人間が可読な形態で表示したりなど、アプリケーションおよびユーザをセキュリティＩＤにインタフェースさせるための種々のアプリケーションプログラミングインタフェース（ＡＰＩ）が提供される。

【００３０】

アプリケーション（プロセス）要求アクセスに基づいてリソースへのアクセスを制限することに加えて、同様にリソースの制限付き使用に基づいて特定のセキュリティＩＤを開発する場合もある。例として、「ＵＳＥ＿ＷＩＮＤＯＷＳ」などのセキュリティＩＤは、ウィンドウステーションおよびデスクトップのデフォルトのＡＣＬに置かれ、制限付きトークンの中に対応するＳＩＤを有するプロセスによってのみアクセスを許可する場合がある。同様に、プリンタオブジェクトのデフォルトＡＣＬは、そのデフォルトのＡＣＬの中に「ＵＳＥ＿ＰＲＩＮＴＩＮＧ」ＳＩＤを含み、プロセスが制限付きトークンの中でリストされたこのセキュリティＩＤのみを伴う制限付きプロセスを作成し、これによって制限付きプロセスはプリンタにはアクセスできるが他のリソースにはアクセスできないようにする場合もある。他のリソースにアクセスするために、多くの他のセキュリティＩＤも実現できることが理解されるであろう。

【００３１】

図３に示すように、制限付きセキュリティＩＤは、制限付きトークン８４の特別なフィールド８２内に置かれ、アクションを要求しているプロセスを識別することなどを行う。以下にさらに詳細に説明するように、少なくとも１つのユーザ（またはグループ）セキュリティＩＤおよび少なくとも１つの制限付きセキュリティＩＤの両方についてオブジェクトへのアクセスを許可するように要求することによって、オブジェクトは要求側プロセス（またユーザまたはグループ）に基づいて選択的にアクセスを許可できる。たとえば、ファイルオブジェクトなどのオブジェクトは、マイクロソフトワード、マイクロソフトエクセルまたはＷｉｎｄｏｗｓ　エクスプローラのプロセスがそれにアクセスすることを許可するが、他のプロセスへのアクセスを拒否する場合がある。さらに、許可されたプロセスの各々は、異なるアクセス権が与えられる場合もある。

【００３２】

この設計は、ユーザのコンテキスト内で異なるプロセスが実行することを制御するための、大きな融通性と細分性を提供する。これらの機能に関する１つの期待される使用モデルは、信頼できるアプリケーションと信頼できないアプリケーションとの区別を含む。「アプリケーション」は一般的な意味で、所与のセキュリティコンテキストの下で「ユーザモード」で実行できる任意のコードの一部分を記述する。たとえば、マイクロソフトワードなどのアプリケーションはＡｃｔｉｖｅＸ制御からのプロセスとして実行され、ＡｃｔｉｖｅＸは既存のプロセス内にロードされ実行される場合がある。マイクロソフトのインタ

ーネットエクスプローラなどの他のアプリケーションを起動するアプリケーションは、このインフラストラクチャを使用した「信頼モデル」を導入する可能性がある。

【００３３】

例として、インターネットエクスプローラなどのアプリケーションは制限付きトークンを使用して異なるプロセスの下で信頼できない実行可能なコードを実行し、これらのプロセスがユーザの全体的なアクセス権および権限の中で行うことを制御できる。この目的のために、インターネットエクスプローラアプリケーションは、それ自体のトークンから制限付きトークンを作成し、どの制限付きセキュリティＩＤを制限付きトークン内に置くかを決定する。ついで、信頼できない実行可能コードは制限付きのコンテキストがアクセスできるオブジェクトにのみアクセスするように制限される。

【００３４】

さらに、制限付きＳＩＤおよび他の制限に対応するエントリーは、監査目的のためにＳＡＣＬ８１のフィールドに置くことができる。たとえば、リソースのＳＡＣＬは、インターネットエクスプローラプログラムがそのリソースの読取りまたは書込みアクセスを試みるたびに、および／またはＵＳＥ＿ＦＯＲ＿ＤＥＮＹ＿ＯＮＬＹとマークされたＳＩＤの使用が監査される場合にはいつでも監査するように設定することができる。単純化のために、監査はこのあと詳細に説明しないが、制限付きＳＩＤを介したアクセス制御に関して説明されたコンセプトは監査動作にも適用可能であることは容易に理解されるであろう。

【００３５】

既存のトークンから制限付きトークンを作成するために、ＮｔＦｉｌｔｅｒＴｏｋｅｎと名付けられたアプリケーションプログラミングインタフェース（ＡＰＩ）が提供され、その内容は次のとおりである。

【００３６】

【表１】

```
NTSTATUS
NtFilterToken (
    IN HANDLE ExistingTokenHandle,
    IN ULONG Flags,
    IN PTOKEN_GROUPS SidsToDisable OPTIONAL,
    IN PTOKEN_PRIVILEGES PrivilegesToDelete OPTIONAL,
    IN PTOKEN_GROUPS RestrictingSids OPTIONAL,
    OUT PHANDLE NewTokenHandle
    );
```

【００３７】

ＮｔＦｉｌｔｅｒＴｏｋｅｎ　ＡＰＩは、ＣｒｅａｔｅＲｅｓｔｒｉｃｔｅｄＴｏｋｅｎと名付けられたＷｉｎ３２　ＡＰＩの下でラップされ、ＣｒｅａｔｅＲｅｓｔｒｉｃｔｅｄＴｏｋｅｎの内容は次のとおりである。

【００３８】

【表２】

```
WINADVAPI
BOOL
APIENTRY
CreateRestrictedToken(
    IN HANDLE ExistingTokenHandle,
    IN DWORD Flags,
    IN DWORD DisableSidCount,
    IN PSID_AND_ATTRIBUTES SidsToDisable OPTIONAL,
    IN DWORD DeletePrivilegeCount,
    IN PLUID_AND_ATTRIBUTES PrivilegesToDelete OPTIONAL,
    IN DWORD RestrictedSidCount,
    IN PSID_AND_ATTRIBUTES SidsToRestrict OPTIONAL,
    OUT PHANDLE NewTokenHandle
    );
```

【００３９】

図２および図４～５に表示されたように、これらのＡＰＩ８６は共同して機能し、制限付きでも制限なしでも既存のトークン６０をとり、変更された（制限付き）トークン８４をそこから作成する。ログオンしたユーザのインスタンスに関する識別情報を含む制限付きトークンの構造は、ＰａｒｅｎｔＴｏｋｅｎＩｄ，ＲｅｓｔｒｉｃｔｅｄＳｉｄＣｏｕｎｔ、およびＲｅｓｔｒｉｃｔｅｄＳｉｄｓの３つの新しいフィールドを含む（以下に太字で示されている）。

【００４０】

【表３】

```
Typedef struct _TOKEN {
  TOKEN_SOURCE TokenSource;                    // Ro: 16 バイト
  LUID TokenId;                                // Ro: 8 バイト
  LUID AuthenticationId;                       // Ro: 8 バイト
  LUID ParentTokenId;                          // Ro: 8 バイト
  LARGE_INTEGER ExpirationTime;                // Ro: 8 バイト
  LUID ModifiedId;                             // Wr: 8 バイト

  ULONG UserAndGroupCount;                     // Ro: 4 バイト
  ULONG RestrictedSidCount;                    // Ro: 4 バイト
  ULONG PrivilegeCount;                        // Ro: 4 バイト
  ULONG VariableLength;                        // Ro: 4 バイト
  ULONG DynamicCharged;                        // Ro: 4 バイト

  ULONG DynamicAvailable;                      // Wr: 4 バイト (Mod)
  ULONG DefaultOwnerIndex;                     // Wr: 4 バイト (Mod)
  PSID_AND_ATTRIBUTES UserAndGroups;           // Wr: 4 バイト (Mod)
  PSID_AND_ATTRIBUTES RestrictedSids;          // Ro: 4 バイト
  PSID PrimaryGroup;                           // Wr: 4 バイト (Mod)
  PLUID_AND_ATTRIBUTES Privileges;             // Wr: 4 バイト (Mod)
  PULONG DynamicPart;                          // Wr: 4 バイト (Mod)
  PACL DefaultDacl;                            // Wr: 4 バイト (Mod)

  TOKEN_TYPE TokenType;                        // Ro: 1 バイト
  SECURITY_IMPERSONATION_LEVEL
                    ImpersonationLevel;        // Ro: 1 バイト

  UCHAR TokenFlags;                            // Ro: 4 バイト
  BOOLEAN TokenInUse;                          // Wr: 1 バイト

  PSECURITY_TOKEN_PROXY_DATA ProxyData;        // Ro: 4 バイト
  PSECURITY_TOKEN_AUDIT_DATA AuditData;        // Ro: 4 バイト
  ULONG VariablePart;                          // Wr: 4 バイト (Mod)
} TOKEN, * PTOKEN;
```

【００４１】
通常の（制限なしの）トークンがＣｒｅａｔｅＴｏｋｅｎ　ＡＰＩを介して作成される時、ＲｅｓｔｒｉｃｔｅｄＳｉｄｓフィールドもＰａｒｅｎｔＴｏｋｅｎＩｄフィールドも空であることに注意されたい。
【００４２】

制限付きトークン８４を作成するために、このプロセスは適切なフラグ設定および／または入力フィールド中の情報を伴うＣｒｅａｔｅＲｅｓｔｒｉｃｔｅｄＴｏｋｅｎ ＡＰＩを呼び出し、これはＮｔＦｉｌｔｅｒＴｏｋｅｎ ＡＰＩを順番に起動する。図４のステップ４００の初めに示すように、ＮｔＦｉｌｔｅｒＴｏｋｅｎ ＡＰＩは、ＤＩＳＡＢＬＥ＿ＭＡＸ＿ＳＩＤＳと名付けられたフラグが設定されているかどうかをチェックする。このフラグは新しい、制限付きトークン８４の中にあるグループに関してすべてのセキュリティＩＤがＵＳＥ＿ＦＯＲ＿ＤＥＮＹ＿ＯＮＬＹとマークされていなければならないことを示す。このフラグは、各々のグループを個別に識別する必要なく、トークン内のグループ（多くのグループである可能性がある）を制限するための便利な方法を提供する。フラグが設定されている場合、ステップ４００はステップ４０２に分岐し、ステップ４０２では新しいトークン８４内のグループセキュリティＩＤの各々について、ＵＳＥ＿ＦＯＲ＿ＤＥＮＹ＿ＯＮＬＹを示すビットを設定する。

【００４３】

ＤＩＳＡＢＬＥ＿ＭＡＸ＿ＳＩＤＳフラグが設定されていない場合、ステップ４００はステップ４０４に分岐し、ＮｔＦｉｌｔｅｒＴｏｋｅｎ ＡＰＩのＳｉｄｓＴｏＤｉｓａｂｌｅフィールド内にセキュリティＩＤが個別にリストされているかどうかをテストする。図４のステップ４０４で示されたように、オプションのＳｉｄｓＴｏＤｉｓａｂｌｅ入力フィールドが存在する時、ステップ４０６では、そこにリストされ、また、親トークン６０のＵｓｅｒＡｎｄＧｒｏｕｐｓフィールド６２内にも存在する任意のセキュリティＩＤは、新しい制限付きトークン８４のＵｓｅｒＡｎｄＧｒｏｕｐｓフィールド８８内でＵＳＥ＿ＦＯＲ＿ＤＥＮＹ＿ＯＮＬＹとして個別にマークされる。上記のようにこのようなセキュリティＩＤは、アクセスを拒否するためにのみ使用でき、アクセスを許可するためには使用できず、さらに、あとから削除または使用可能にはできない。したがって、図２に示された例では、グループ$_2$のセキュリティＩＤは、ＮｔＦｉｌｔｅｒＴｏｋｅｎ ＡＰＩ８６のＳｉｄｓＴｏＤｉｓａｂｌｅ入力フィールド内にグループ$_2$セキュリティＩＤを指定することにより、制限付きトークン８４内でＵＳＥ＿ＦＯＲ＿ＤＥＮＹ＿ＯＮＬＹとしてマークされる。

【００４４】

フィルタプロセスはついで図４のステップ４１０に続き、ここではＤＩＳＡＢＬＥ＿ＭＡＸ＿ＰＲＩＶＩＬＥＤＧＥＳと名付けられたフラグがテストされる。このフラグは同様に、新しい、制限付きトークン８４内のすべての権限を削除すべきであることを示す、便利なショートカットとして設定できる。このように設定された場合、ステップ４１０はステップ４１２に分岐し、ステップ４１２では新しいトークン８４からすべての権限が削除される。

【００４５】

フラグが設定されていない場合、ステップ４１０はステップ４１４に分岐し、ここではオプションのＰｒｉｖｉｌｅｇｅｓＴｏＤｅｌｅｔｅフィールドが確認される。ＮｔＦｉｌｔｅｒＴｏｋｅｎ ＡＰＩ８６が呼ばれた時に存在する場合は、ステップ４１６で、この入力フィールドにリストされまた既存のトークン６０の権限フィールド６８にも存在する任意の権限は、新しいトークン８４の権限フィールド９０から個別に削除される。図２に示された例では、「権限$_2$」から「権限$_m$」として示された権限は、ＮｔＦｉｌｔｅｒＴｏｋｅｎ ＡＰＩ８６のＰｒｉｖｉｌｅｇｅｓＴｏＤｅｌｅｔｅ入力フィールド内にこれらの権限を指定することによって、新しいトークン８４の権限フィールド９０から削除されている。上記のように本発明の１つの態様によれば、これは、トークン内で使用可能な権限を削減する機能を提供する。このプロセスは図５のステップ４２０に続く。

【００４６】

制限付きトークン８４を作成する時に、ステップ４２０でＲｅｓｔｒｉｃｔｉｎｇＳｉｄｓ入力フィールド内にＳＩＤが存在していた場合、親トークンが通常のトークンか、または親トークン自体が制限付きＳＩＤを有する制限付きトークンであるかどうかに関して決定が行われる。ＡＰＩ、ＩｓＴｏｋｅｎＲｅｓｔｒｉｃｔｅｄがステップ４２２で呼び出

され、親トークンのＲｅｓｔｒｉｃｔｉｎｇＳｉｄｓフィールドを（ＮｔＱｕｅｒｙＩｎｆｏｒｍａｔｉｏｎＴｏｋｅｎ　ＡＰＩを介して）照会してこれがＮＵＬＬでないかどうかを確認することによってこの問題を解決し、ここでＮＵＬＬでなかった場合、親トークンは制限付きトークンであり、ＡＰＩはＴＲＵＥ（真）を戻す。テストが満足できなかった場合、親トークンは通常のトークンでありＡＰＩはＦＡＬＳＥ（偽）を戻す。続くステップ４２６または４２８のために、トークン自体は制限付きであるが制限付きＳＩＤを有しない親トークン（すなわち、権限が削除されているかおよび／またはＵＳＥ＿ＦＯＲ＿ＤＥＮＹ＿ＯＮＬＹ　ＳＩＤＳである場合）は、制限付きでないとして処理される可能性があることに注意されたい。

【００４７】

ステップ４２４では、親トークンが制限付きである場合、ステップ４２４はステップ４２６に分岐し、ステップ４２６では、親トークンの制限付きセキュリティＩＤフィールドと、ＡＰＩの制限付きセキュリティＩＤ入力リストの両方にある任意のセキュリティＩＤは、新しいトークン８４の制限付きセキュリティＩＤフィールド９２に置かれる。制限付きセキュリティＩＤが両方のリストになければならないため、制限付き実行コンテキストが、制限付きセキュリティＩＤフィールド９２にさらなるセキュリティＩＤを追加することを阻止し、この場合、アクセスを減らすのではなく効果的にアクセスを増やすことになる。同様に、ステップ４２６で共通なセキュリティＩＤがなかった場合、少なくとも１つの制限付きＳＩＤを新しいトークン内の元のトークンから取り去るなどによって、作成された任意のトークンはそのアクセスを増やすことなく制限されなければならない。その他の場合は、新しいトークン内の空の制限付きＳＩＤフィールドはそのトークンが制限されていないことを示し、この場合、アクセスを減らすのではなく効果的にアクセスを増やすことになる。

【００４８】

あるいは、ステップ４２４で親トークンが通常のトークンであると判断された場合、ステップ４２８で新しいトークン８４のＲｅｓｔｒｉｃｔｉｎｇＳｉｄｓフィールド９２は入力フィールド内にリストされたものに設定される。これはセキュリティＩＤを追加するが、次に詳細に説明されるように制限付きＳＩＤを有するトークンが第２のアクセステストを受けるため、アクセスは実際には減らされることに注意されたい。

【００４９】

最後に、ステップ４３０も実行され、ここで新しいトークン８４内のＰａｒｅｎｔＴｏｋｅｎＩｄ９３は既存の（親）トークンのＴｏｋｅｎＩｄに設定される。これはオペレーティングシステムに、通常は親トークン以外には許可されていない場所で、そのトークンの制限付きのバージョンを使用するプロセスをのちに許可するオプションを提供する。

【００５０】

特に図６～８を参照して本発明の動作の説明に戻ると、図６に表示されたように、制限付きプロセス９４が作成され、読取り／書込みアクセスでファイルオブジェクト７０を開こうと試みている。オブジェクト７２のセキュリティ記述子内では、ＡＣＬ８０はそこにリストされた多くのセキュリティＩＤおよび、各ＩＤに関して許可されたタイプのアクセスを有し、ここで「ＲＯ」は読取りのみのアクセスが許可されていることを示し、「ＷＲ」は読取り／書込みアクセスを示し、「ＳＹＮＣ」は同期化アクセスが許可されていることを示す。他の場合は「ＸＪｏｎｅｓ」が許可されたグループ内のメンバーシップを介してアクセスを許可されている場合でも、「ＸＪｏｎｅｓ」は特にオブジェクト７２へのアクセスを拒否されていることに注意されたい。さらに、関連付けられたこのトークン８４を有するプロセス９４は、このエントリーは「ＤＥＮＹ」（すなわち、ＵＳＥ＿ＦＯＲ＿ＤＥＮＹ＿ＯＮＬＹ）とマークされているため、トークン８４内の「バスケットボール」セキュリティＩＤを介して任意のオブジェクトへのアクセスを許可されないことになる。

【００５１】

セキュリティの目的のために、制限付きセキュリティコンテキストは第一に、Ｗｉｎｄｏｗｓ　ＮＴカーネル内で実現される。オブジェクト７２にアクセスしようと試みるために

、プロセス９４はオブジェクトマネジャー７４に、アクセスが望ましいオブジェクトを識別する情報および、所望されるアクセスのタイプを提供する（図８、ステップ８００）。オブジェクトマネジャー７４はこれに応答して、ステップ８０２に示されたように、セキュリティメカニズム７８と共同して機能し、トークン８４内にリストされた（プロセス９４と関連付けられている）ユーザセキュリティＩＤおよびグループセキュリティＩＤをＡＣＬ８０内のエントリーに対して比較し、所望のアクセスが許可されるべきか拒否されるべきかを決定する。

【００５２】

ステップ８０４で一般に示されているように、アクセスがリストされたユーザまたはグループに関して許可されていない場合、セキュリティチェックはステップ８１４でアクセスを拒否する。しかし、ステップ８０４でアクセスチェックのユーザ部分およびグループ部分の結果が許可可能なアクセスを示した場合には、セキュリティプロセスはステップ８０６に分岐し、制限付きトークン８４が任意の制限付きセキュリティＩＤを有しているかどうかを決定する。有していない場合、追加の制限はなく、ここでアクセスチェックは完了し、ステップ８１２において、ユーザアクセスおよびグループアクセスのみに基づいてアクセスは許可される（そのオブジェクトへのハンドルが戻される）。このようにして、通常のトークンは本質的に以前と同じようにチェックされる。しかし、ステップ８０６によって決定されたように、トークンが制限付きセキュリティＩＤを含んでいる場合、ついでステップ８０８によって、制限付きセキュリティＩＤをＡＣＬ８０内のエントリーと比較することによって、第２のアクセスチェックが実行される。ステップ８１０でこの第２のアクセステストがアクセスを許可した場合、そのオブジェクトへのアクセスはステップ８１２で許可される。そうでない場合、アクセスはステップ８１４で拒否される。

【００５３】

図７で論理的に示すように、トークン８４の中に制限付きセキュリティＩＤが存在する時はいつでも、２つの部分からなるテストがこのように実行される。トークン８４内のセキュリティＩＤおよび所望のアクセスビット９６をオブジェクト７２のセキュリティ記述子に対して考慮することによって、通常のアクセステスト（ビットごとのＡＮＤ）および制限付きセキュリティＩＤのアクセステストの両方は、プロセスがそのオブジェクトへのアクセスを許可されるようにアクセスを許可しなければならない。本発明に必要ではないが上記のように、通常のアクセステストが最初に行われ、アクセスが拒否された場合には、さらなるテストは必要ではない。トークン内にＡＣＬの識別子に一致するセキュリティＩＤがないため、またはＡＣＬエントリーが特に、その中にあるセキュリティ識別子に基づいてトークンへのアクセスを拒否したためのどちらの理由でも、アクセスは拒否されることに注意されたい。別法としては、トークンを制限付きＳＩＤの多数の組を有するように構成し、たとえば、組Ａ　ＯＲ（組Ｂ　ＡＮＤ組Ｃ）がアクセスを許可するなど、これらのＳＩＤの評価をカバーするさらに複雑なブール式を伴う場合もある。

【００５４】

このように図６に示された例では、トークン８４（フィールド９２）内の制限付きＳＩＤのみが「インターネットエクスプローラ」を識別する一方、オブジェクトのＡＣＬ８０内には対応する制限付きＳＩＤがないため、図６に示された例では、オブジェクト７２へのアクセスはプロセス９４へは許可されない。ユーザは通常のトークンで実行するプロセスを介してオブジェクトへアクセスする権利を有していたが、プロセス９４はＡＣＬ内に「インターネットエクスプローラ」ＳＩＤ（非ＤＥＮＹ）を有するオブジェクトのみにアクセスできるようにするように、制限された。

【００５５】

アクセスのタイプを指定する代わりに、コーラーが、指定されたＭＡＸＩＭＵＭ＿ＡＬＬＯＷＥＤアクセスを有する場合、これによって上記のように、アルゴリズムは最大のアクセスを決定するＡＣＬ８０をウォークスルーする。制限付きトークンで、１つでも任意のタイプのユーザアクセスまたはグループアクセスが許可された場合、ユーザおよびグループの実行に続いて許可可能なアクセス権のタイプ（複数可）は、第２の実行に関して所望

のアクセスとして指定され、第２の実行はＲｅｓｔｒｉｃｔｅｄＳｉｄｓリストをチェックする。このようにして、制限付きトークンは通常のアクセスより少ないかまたは等しいアクセスを許可されることが確認される。

【００５６】

最後に、本発明のセキュリティモデルは、他のセキュリティモデルと共に使用できることに注意されたい。たとえば、オペレーティングシステムの上に常駐する、機能に基づいたセキュリティモデルは、本発明のオペレーティングシステムレベルのセキュリティモデルの上で使用することができる。

【００５７】

制限付きトークンを介した最小権限

一般に、本発明は削減されたアクセス権および／または権限で実行しているユーザをシステムが自動的に実行することに関する。簡単に説明するために、以下用語「アクセス」または「権限」は、プロセスがリソースを使用する能力のコンテキスト内で使用される時には、権限またはセキュリティ識別子のいずれか、または、両方の何らかの組合せをさす。このように、制限付きトークンは、１つまたは複数の権限を削除するか、および／またはＳＩＤをＵＳＥ＿ＦＯＲ＿ＤＥＮＹ＿ＯＮＬＹに設定するかのいずれかによって、親トークンのアクセスに関して削減された「アクセス」を有する。制限付きトークンはまた、親トークンが通常のユーザに基づいたトークンであり制限付きトークンがその中にＳＩＤを有する場合、または親トークン自体が制限付きＳＩＤを有し、（子の）制限付きトークンがその中により少ない制限付きＳＩＤを有する場合も、削減されたアクセスを有する可能性がある。同様に、以下使用されるように、増加されたアクセスが実際には、トークン内にリストされた実際の権限を介してではなく、トークン内のセキュリティＩＤから生じた可能性があっても、増加されたまたは高められた「権限」での実行は、増加された「アクセス」での実行と同じになる。

【００５８】

いずれの場合でも、最小の（すなわち、何らかの方法で削減された）権限が実行できる第１の方法は、制限をアプリケーションに論理的に接続することである。より具体的には、制限付き実行のコンテキストは、オペレーティングシステムが各アプリケーションに関して、および各リソースに関して別の制限付きセキュリティＩＤを作成することを可能にする。オペレーティングシステムは次いで、アプリケーションがどんなリソースにアクセスする必要があるかを知っている安全なアプリケーションランチャを含み、制限付きトークンを介して、アプリケーション（すなわちそのプロセス）がこれらのリソースのみにアクセスするように制限することが可能になる。

【００５９】

したがって、本発明の別の態様によれば、および図９に示されているように、アプリケーションプログラム１１０（図９）は関連づけられた制限情報１１２を有する場合がある。たとえば、アプリケーションはデフォルトの制限と共に出荷されるか、および／または、管理者などがアプリケーションをインストールするときに制限を設定する場合がある。情報１１２は、アプリケーションがどのファイルまたはディレクトリにアクセスするか、アプリケーションを実行するためには管理者が必要かどうか、またはアプリケーションが任意の他のプログラムを起動する必要があるかどうかなど、制限を含む可能性がある。システムはこの制限情報１１２をデータベース、または他の非揮発性メモリなどに格納する。たとえば、ウィンドウズエクスプローラなどのプログラムランチャ１１４は、この情報をエクスプローラリンクファイル内に格納できる。

【００６０】

プログラムランチャ１１４は実行されると制限情報１１２を読み出し、格納された情報に基づいて通常の、ユーザに基づいたトークン１１６から制限付きトークン１２２を作成する。この結果、アプリケーションプログラム１１０は制限付きトークン１２２がアクセスを許可したリソース１２４のみにアクセスするように制限される。たとえば、制限付きトークン１２２を介して、ゲームプログラム１１０をそれ自体のデータファイルにのみアク

セスするように制限できる。

【００６１】

この目的のために、図１０に示されたように、制限付きトークン１２２はその制限付きＳＩＤフィールド内に、たとえば、図１０に「ＧＡＭＥ３３」と示されたようなアプリケーションを識別する制限付きＳＩＤを含む。また図１０に示されたように、ゲームの各データファイルに関連づけられたＡＣＬ（たとえば、リソースオブジェクト１２４）は、制限付きトークン１２２内に置かれた制限付きＳＩＤ（複数可）に対応する１つまたは複数の識別子（これも「ＧＡＭＥ３３」と示されている）を含む。上記のようにアクセス評価が実行されると、ユーザＳＩＤおよび制限付きＳＩＤの両方がセキュリティ記述子内のエントリーに一致しているため、アプリケーション１１０はこのファイルオブジェクト１２４へのアクセスを許可される。しかし、オペレーティングシステム３５を介して管理者によって決定されたように、システム内の他のファイルのセキュリティ記述子にはこのような「ＧＡＭＥ３３」ＳＩＤがないので、これによって、ゲームプログラム１１０がこれらの他のファイルにアクセスすることを防ぐ。

【００６２】

他の使用法は、アプリケーションのアクセスを、ファイルの範囲ではなく編集されている１つのファイルのみに許可することによってウイルスを制御することである。この方法で、文書を他の文書へアクセスさせないことによって、マクロウイルスは効果的に止められる。

【００６３】

アプリケーションへのアクセスを制限するさらに別の方法は、アプリケーション自体を制限付き部分および制限なし部分に分割することによるものである。もちろん、アプリケーションは追加の細分性を有し、制限レベルに基づいて２つ以上の部分に分割される場合もある。例として図１１に示されたように、アプリケーションプログラム１３０は管理タイプのアクティビティおよび非管理タイプのアクティビティの間で分割された機能を有する場合がある。この方法で、高められた権限で実行する管理者は、新しい機能をアプリケーションに追加したり、またはデフォルトの挙動を設定したりなどのタスクを実行できるようになる。これは制限付きトークンを介して、非管理者に一定の機能を含むダイナミックリンクライブラリ（ＤＬＬ）、および／またはデフォルトの情報を格納するデータファイルへのアクセスを否定するなど、多くの方法で実施される可能性があることに注意されたい。他の方法は、たとえば、権限なしと指定された機能のための制限付きトークンおよび、権限付き機能のための通常のトークンなど、実行しようと試みている各機能の各プロセスにトークンを関連づけることである。いずれの場合でも、より少ないアクセスを有する通常のユーザはデータ入力およびデータ保存など通常の機能を実行でき、より高いレベルのユーザは管理的な機能を実行することができる。もちろん、いくつかの機能は、任意のタイプの有効なユーザによるアクセスを許可することによって、論理的に両方のグループに入れることもできる。

【００６４】

分割された部分は、アクセスに関しては相互に排他的である場合があることに注意されたい。たとえば、管理者用の一定の機能へのアクセスを許可または否定するなど、制限付きトークンを使用すると、管理者が管理者モードで実行している時に通常の機能を実行できないか、または逆に管理者が通常モードで実行している時に管理機能を実行できないように、アプリケーションを書くことができる。これは、非管理的なタスク（たとえばデータ入力）を通常のモードで実行している管理者が、権限付きの機能を介して不注意に何らかの損害（たとえばファイルの削除）を行うことを防ぐ。さらに、オペレーションオペレーションのモードを強調するために、アプリケーションはそれが実行されているモードに応じて異なる表示（たとえば異なる色スキーム）を有することもできる。

【００６５】

さらに、本発明はどのプログラムも通常は管理者権限で実行できないことを指定する能力も提供する。これは、ユーザに、安全なデスクトップ（すなわち、オペレーティングシス

テムによって管理されているデスクトップ）から管理者権限でプログラムを起動するように要求することによって実行できる。これはユーザに、プログラムを信用するだけではなく、管理者権限を明示的に使用することを強要する。

【００６６】

最小の（または何らかの方法で削減された）権限でユーザにオペレーションさせるさらに別の方法は、各ユーザが必要な量のアクセスのみを許可する制限付きトークンで実行するようなシステムデフォルトを有することである。一般に、資格を持つユーザが増大されたアクセスを必要とするタスクを行う必要がある場合、ユーザまたはオペレーティングシステムは明示的なオペレーションを実行し、そのアクセスを得る必要がある。この目的のために、制限付きトークンに関連づけられたプロセスを有するユーザは一時的に、彼または彼女のプロセスを増大されたアクセスを有するトークン（制限付きトークンまたは通常のトークン）に関連づけることができる。タスクが実行されると、強化されたアクセスは次いで、制限付きトークンをユーザのプロセスに復元することによって削除される。

【００６７】

図１２は、システムが最小のまたは削減されたアクセスでオペレーションを実行する方法を示している。ステップ１２００で開始するオペレーションにおいては、ユーザの通常のトークンより少ないアクセスを有する制限付きトークンがユーザのために作成される。上記のように、これはユーザＳＩＤおよびグループＳＩＤの属性をＵＳＥ＿ＦＯＲ＿ＤＥＮＹ＿ＯＮＬＹに変更し、通常の親トークンに関連する１つまたは複数の権限を削除する、および／または制限付きＳＩＤを制限付きトークンに追加することによって達成される。次いで、ステップ１２０２において、制限付きトークンはユーザの制限付きプロセスに関連づけられる。またステップ１２０２に示されたように、プロセスがリソースへアクセスしようと試みた時、上記のように、そのリソースのセキュリティ記述子に対して制限付きトークンを使用してアクセス評価が実行される。２つ以上のレベルの制限が可能であることに注意されたい（たとえば、通常のトークン、通常のトークンから作成された第１の制限付きトークン、および第１の制限付きトークンから作成された第２の制限付きトークン）。２つ以上のレベルが望ましい場合、ステップ１２０２でプロセスと関連づけられた制限付きトークンは、典型的には、もっとも少ないアクセスを伴うトークンである。ユーザは次いで、各タスクについて削減された（たとえば、もっとも可能性が少ない）アクセスレベルを使用するようなデフォルトになる。

【００６８】

ステップ１２０４では、オペレーティングシステム（すなわち、その中のセキュリティメカニズム）は、（望ましいタイプの）アクセスが許可されているかどうかを決定し、許可されている場合ステップ１２２０に分岐し、ステップ１２２０では、適切なタイプのアクセスが許可され、タスクが実行される。制限付きＳＩＤが存在し、アクセスが権限を介していない時は、アクセス評価は上記のように、２つの部分のアクセスチェックを含むことに注意されたい。

【００６９】

本発明によれば、ステップ１２０４でアクセスが許可されなかった場合、オペレーティングシステムはアクセスを否定するのではなく、ユーザに増大されたアクセスを伴うトークンを使用してリソースへアクセスする追加の機会を与える場合がある。この目的のために、ステップ１２０６では、ユーザのトークンが制限付きトークンであるかどうかを決定するテストが行われる。この決定は、制限付きトークンがＰａｒｅｎｔＴｏｋｅｎＩＤフィールド内で識別された非ＮＵＬＬの親トークンを有するため、トークンのそのフィールドを介して行うことができる。トークンが親（ステップ１２０６）を有しない場合、そのユーザは制限があるかないかにかかわらず彼または彼女の通常のトークンでアクセス権を有しないので、アクセスはステップ１２２２ですぐに否定される。

【００７０】

別法としては、トークンがステップ１２０６で親を有する場合、システムはステップ１２１０でユーザに、ユーザが増大されたアクセスレベル、すなわち制限付きトークンの親ト

ークンでそのリソースへのアクセスを再び試みたいかどうかを決定するようにプロンプトで促される。この方法で、ユーザは危険の可能性がある状況、すなわち、なにか異常なことが未決定になっていることに気づかされる。ユーザが増大されたアクセスでタスクを実行することを試みないように決定した場合、ステップ１２１２はステップ１２２２に分岐し、ここでアクセスが否定される。しかし、ユーザがアクションが本当に望ましいと決定した場合（たとえば、実際にディスクドライブ上のすべてのファイルを削除しようとする場合）、ユーザはそのプロンプトに肯定的に答え、これによってステップ１２１２はステップ１２１４に分岐し、ステップ１２１４ではアクセスは親トークンをプロセスに関連づけることによって増大され、評価が再び実行される。アクセスが許可された場合（ステップ１２１６）、要求されたタスクはステップ１２１８で実行される。ステップ１２１６でアクセスが許可されない場合、ステップ１２０６が再び実行され、トークンが親トークンを有するかどうかを決定する。この方法で、２つ以上のレベルの制限がサポートされる。

【００７１】

上記のメカニズムは資格のあるユーザに対してアクセスを否定する目的ではなく、資格のあるユーザに、より高いアクセスレベルを必要とするイベントが要求されたことを警告するのが目的であることに注意されたい。ユーザはイベントを実行する前に、第２の、明確なアクションをとらなければならない。しかし、どの状況でも、ユーザはユーザの通常のトークンによって許可されている以上の追加の権限は有しない。

【００７２】

このように、たとえば、管理者は彼または彼女のタスクを実行するために制限付きトークンをセットアップする。制限付きトークンは管理者を、グループの非管理者に許可された権限およびアクセス権で実行するように制限する。上記のようにこの方法で１度セットアップすると、管理者は要求されたアクションが通常のユーザより高いレベルであることをプロンプトで表示されずに、システムを損壊するようなことを不注意にしてしまうことはなくなる（たとえばディスクドライブ上のすべてのファイルを削除するなど）。プロンプトおよび応答メカニズムは、特定のオーバーライドのみが管理レベルのアクションを許可するようにする。

【００７３】

前記の詳細な説明からわかるように、制限付きトークンを介して最小の（または何らかの方法で削減された）権限でオペレーションを実行する、改善されたセキュリティモデルが提供される。実行は自動的であり、たとえば、アプリケーションに基づくか、アプリケーション内に書き込まれるか、および／またはプロンプトおよび応答メカニズムを介してシステムによって提供できる。

【００７４】

本発明は種々の修正および代替の構成を可能にするが、そのうち所定の例示された実施形態が図に示され、上記に詳細に説明された。しかし、本発明を開示された具体的な形に制限する意図はなく、逆に、本発明の趣旨と範囲に含まれるすべての変形例、代替の構成、等価物をカバーすることが目的であることを理解されたい。

【図面の簡単な説明】

【図１】　本発明を組み込むことのできるコンピュータシステムを表す構成図である。

【図２】　既存のトークンから制限付きトークンを作成することを一般に表す構成図である。

【図３】　プロセスがリソースにアクセスできるかどうかを決定するための種々の構成要素を一般に表す構成図である。

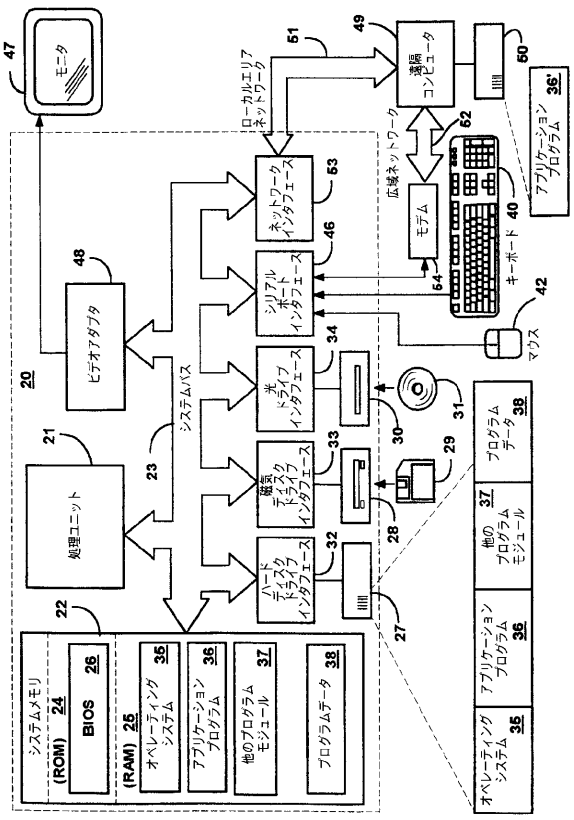
【図４】　既存のトークンから制限付きトークンを作成するためにとられる一般のステップを表す流れ図を含む。

【図５】　既存のトークンから制限付きトークンを作成するためにとられる一般のステップを表す流れ図を含む。

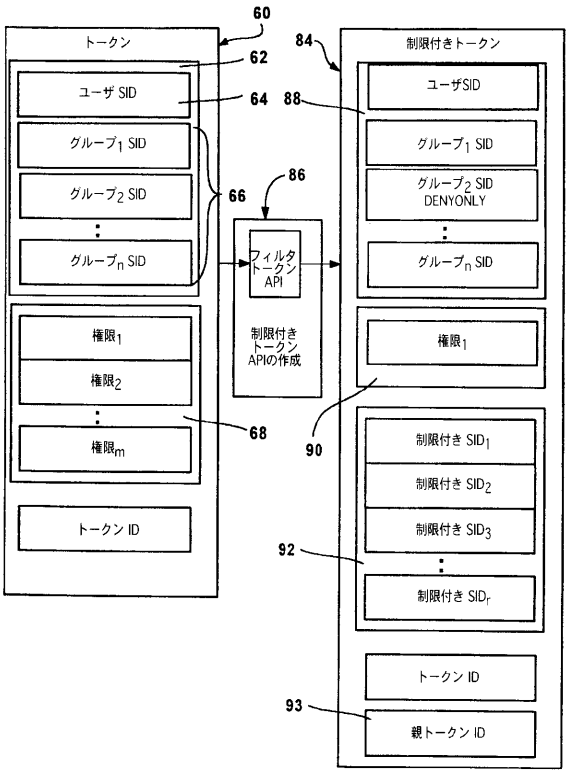
【図６】　リソースにアクセスを試みる、関連づけられた制限付きトークンを有するプロセスを一般に表す構成図である。

【図７】　関連づけられた制限付きトークンを有するプロセスのオブジェクトへアクセスすることを決定するための論理を一般に表す構成図である。
【図８】　リソースへのプロセスアクセスを許可するかどうかを決定する時にとられる一般のステップを表す流れ図である。
【図９】　本発明の一態様による、削減された権限を持つアプリケーションプログラムを自動的に実行するための種々の構成要素の構成図である。
【図１０】　本発明の一態様による、自動的にそこに関連づけられた制限付きトークンを有し、リソースにアクセスしようと試みるプロセスを一般に示す構成図である。
【図１１】　本発明の一態様による、権限付き部分と権限なし部分に分割されたアプリケーションプログラムを示す図である。
【図１２】　本発明の一態様による、削減されたアクセスで実行するユーザを実行するために取られる一般的なステップを示す流れ図である。

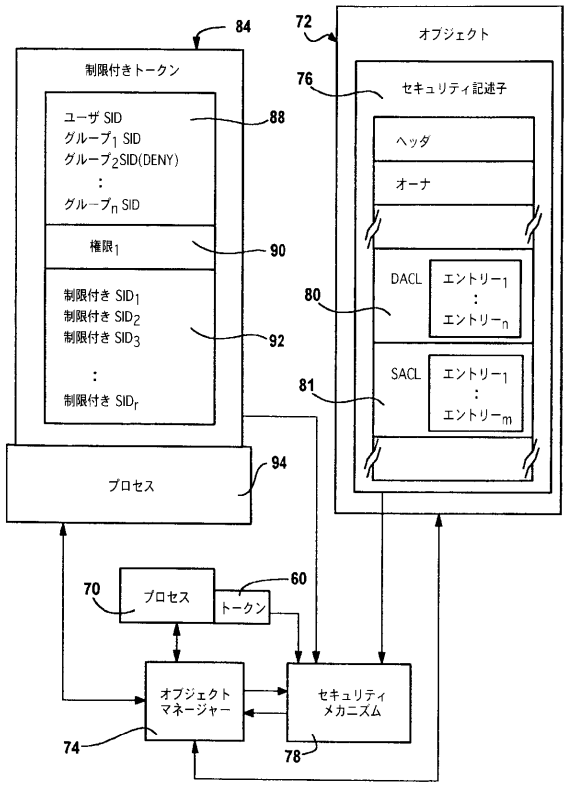
【図１】



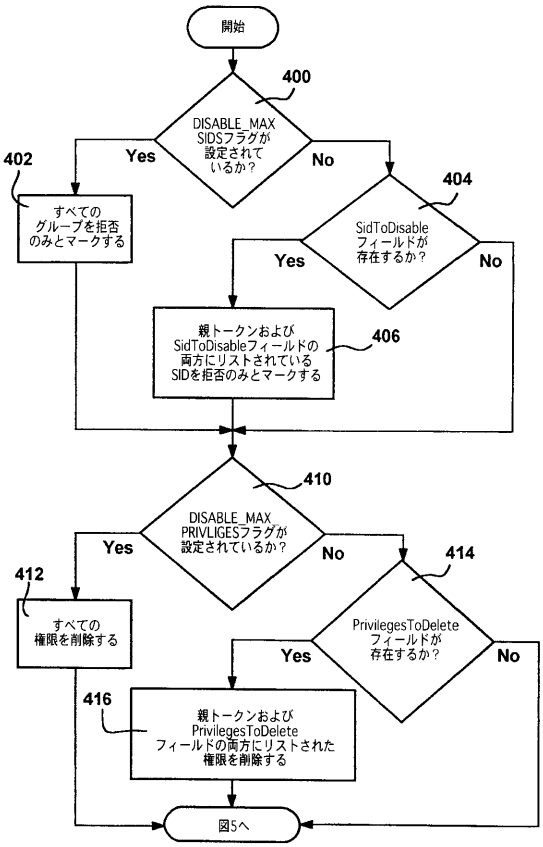
【図２】



【図３】


制限付きトークン
84
ユーザ SID
グループ1 SID
グループ2SID(DENY)
:
グループn SID
88
権限1
90
制限付き SID1
制限付き SID2
制限付き SID3
:
制限付き SIDr
92
プロセス
94
オブジェクト
72
セキュリティ記述子
76
ヘッダ
オーナ
DACL
エントリー1
:
エントリーn
80
SACL
エントリー1
:
エントリーm
81
プロセス
70
トークン
60
オブジェクト
マネージャー
セキュリティ
メカニズム
74
78

【図４】


開始
400
DISABLE_MAX
SIDSフラグが
設定されて
いるか？
Yes
No
402
すべての
グループを拒否
のみとマークする
404
SidToDisable
フィールドが
存在するか？
Yes
No
406
親トークンおよび
SidToDisableフィールドの
両方にリストされている
SIDを拒否のみとマークする
410
DISABLE_MAX
PRIVLIGESフラグが
設定されているか？
Yes
No
412
すべての
権限を削除する
414
PrivilegesToDelete
フィールドが
存在するか？
Yes
No
416
親トークンおよび
PrivilegesToDelete
フィールドの両方にリストされた
権限を削除する
図5へ

【図５】

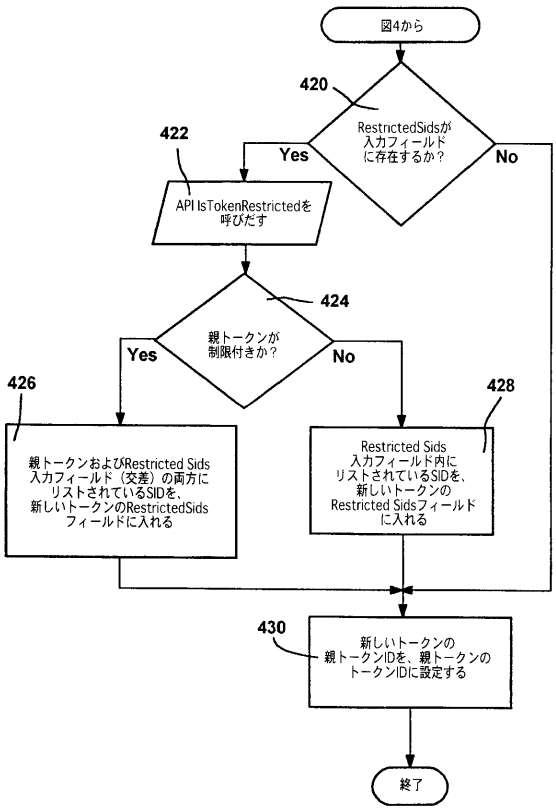
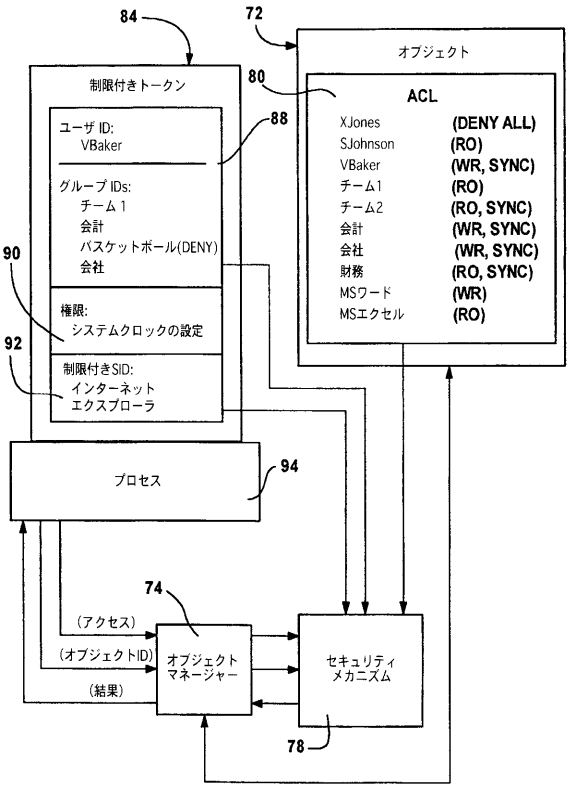
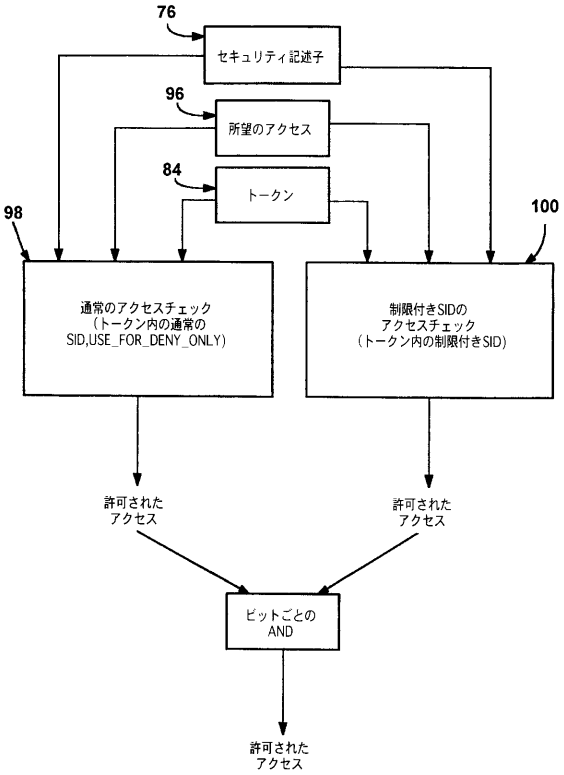


図4から
420
RestrictedSidsが
入力フィールド
に存在するか？
Yes
No
422
API IsTokenRestrictedを
呼びだす
424
親トークンが
制限付きか？
Yes
No
426
親トークンおよびRestricted Sids
入力フィールド（交差）の両方に
リストされているSIDを、
新しいトークンのRestrictedSids
フィールドに入れる
428
Restricted Sids
入力フィールド内に
リストされているSIDを、
新しいトークンの
Restricted Sidsフィールド
に入れる
430
新しいトークンの
親トークンIDを、親トークンの
トークンIDに設定する
終了

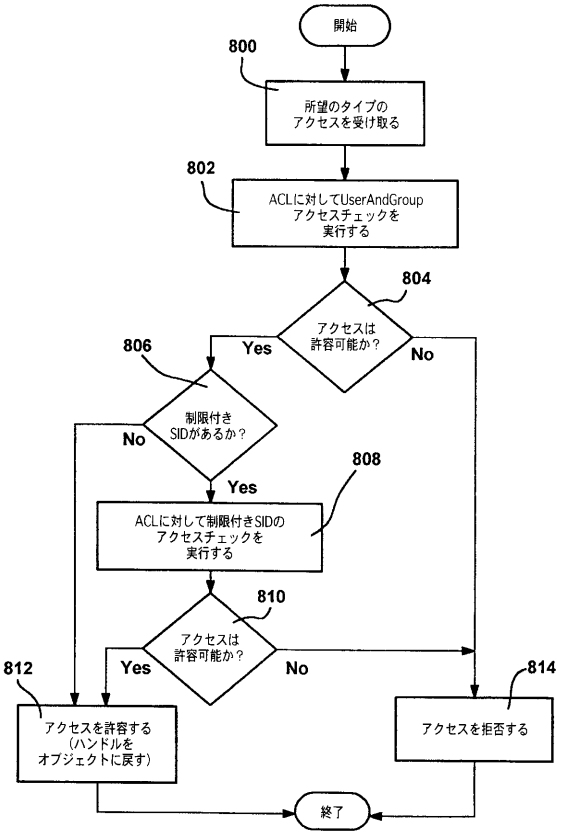
【図６】


84
制限付きトークン
ユーザ ID:
VBaker
88
グループ IDs:
チーム 1
会計
バスケットボール(DENY)
会社
90
権限:
システムクロックの設定
92
制限付きSID:
インターネット
エクスプローラ
プロセス
94
72
オブジェクト
80
ACL
XJones (DENY ALL)
SJohnson (RO)
VBaker (WR, SYNC)
チーム1 (RO)
チーム2 (RO, SYNC)
会計 (WR, SYNC)
会社 (WR, SYNC)
財務 (RO, SYNC)
MSワード (WR)
MSエクセル (RO)
74
(アクセス)
(オブジェクトID)
(結果)
オブジェクト
マネージャー
セキュリティ
メカニズム
78

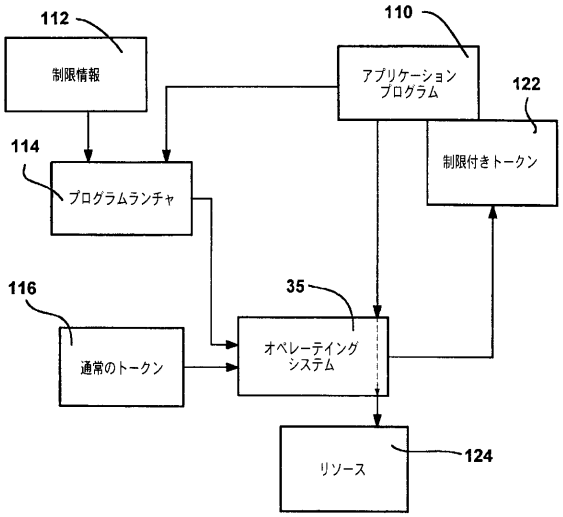
【図７】


76
セキュリティ記述子
96
所望のアクセス
84
トークン
98
100
通常のアクセスチェック
（トークン内の通常の
SID,USE_FOR_DENY_ONLY）
制限付きSIDの
アクセスチェック
（トークン内の制限付きSID）
許可された
アクセス
許可された
アクセス
ビットごとの
AND
許可された
アクセス

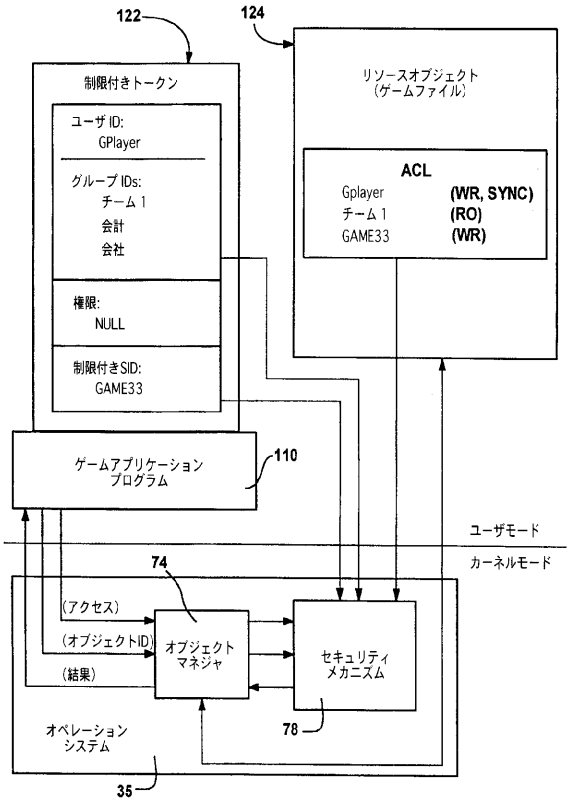
【図８】


開始
800
所望のタイプの
アクセスを受け取る
802
ACLに対してUserAndGroup
アクセスチェックを
実行する
804
アクセスは
許容可能か？
Yes
No
806
制限付き
SIDがあるか？
No
Yes
808
ACLに対して制限付きSIDの
アクセスチェックを
実行する
810
アクセスは
許容可能か？
Yes
No
812
アクセスを許容する
（ハンドルを
オブジェクトに戻す）
814
アクセスを拒否する
終了

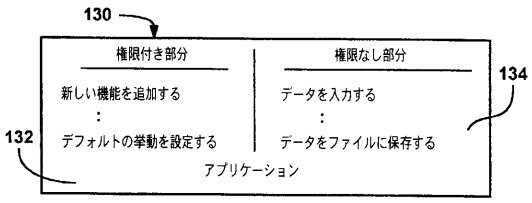
【図９】


112
制限情報
110
アプリケーション
プログラム
122
制限付きトークン
114
プログラムランチャ
116
通常のトークン
35
オペレーティング
システム
リソース
124

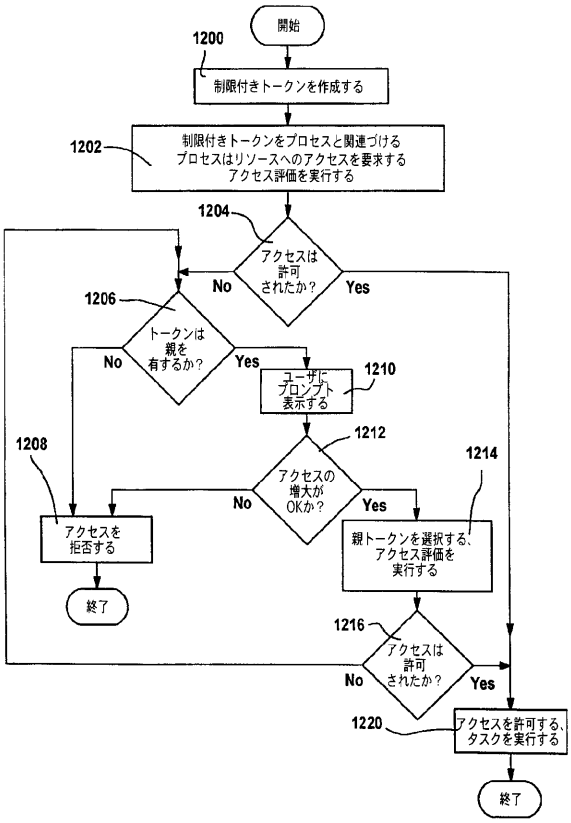
【図１０】


122
124
制限付きトークン
ユーザ ID:
GPlayer
グループ IDs:
チーム 1
会計
会社
権限:
NULL
制限付きSID:
GAME33
リソースオブジェクト
（ゲームファイル）
ACL
Gplayer (WR, SYNC)
チーム 1 (RO)
GAME33 (WR)
ゲームアプリケーション
プログラム
110
ユーザモード
カーネルモード
74
（アクセス）
（オブジェクトID）
（結果）
オブジェクト
マネジャ
セキュリティ
メカニズム
78
オペレーション
システム
35

【図１１】


130
権限付き部分
新しい機能を追加する
：
デフォルトの挙動を設定する
権限なし部分
データを入力する
：
データをファイルに保存する
134
132
アプリケーション

【図１２】


開始
1200
制限付きトークンを作成する
1202
制限付きトークンをプロセスと関連づける
プロセスはリソースへのアクセスを要求する
アクセス評価を実行する
1204
アクセスは許可されたか？
No
Yes
1206
トークンは親を有するか？
No
Yes
1210
ユーザにプロンプト表示する
1212
アクセスの増大がOKか？
No
Yes
1214
1208
アクセスを拒否する
終了
親トークンを選択する、アクセス評価を実行する
1216
アクセスは許可されたか？
No
Yes
1220
アクセスを許可する、タスクを実行する
終了

フロントページの続き

(56)参考文献 特開平３－６６４０（ＪＰ，Ａ）
国際公開第９６／１３１１３（WO，Ａ１）
米国特許第５７６１６６９（ＵＳ，Ａ）
福永勇二，Windows NT基礎の基礎▲２▼ ユーザー管理とセキュリティ，ｎｅｔＰＣ，日本，株式会社アスキー，１９９７年１２月，第２巻，第12号，pp.118－121
ＯＳ／２ Ｊ２．０ テクニカル・ライブラリー プログラミングの手引き 上巻，日本，日本アイ・ビー・エム株式会社，１９９２年１０月，第１刷，pp.7-4－7-7
鈴木信夫，Windows NTの実像 第３部 NT Executive，日経バイト，日本，日経ＢＰ社，１９９３年１０月 １日，第117号，pp.102－113

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 9/46 - 9/54
G06F 21/24