

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4685782号
(P4685782)

(45) 発行日 平成23年5月18日 (2011.5.18)

(24) 登録日 平成23年2月18日 (2011.2.18)

(51) Int. Cl.		F I	
G06F 21/24	(2006.01)	G06F 12/14	540A
G06F 12/14	(2006.01)	G06F 12/14	510A
G06F 12/00	(2006.01)	G06F 12/00	537H
G09C 1/00	(2006.01)	G09C 1/00	660D

請求項の数 18 (全 16 頁)

(21) 出願番号	特願2006-532376 (P2006-532376)	(73) 特許権者	502303739
(86) (22) 出願日	平成16年4月1日 (2004.4.1)		オラクル・インターナショナル・コーポレーション
(65) 公表番号	特表2007-500912 (P2007-500912A)		アメリカ合衆国、94065 カリフォルニア州、レッドウッド・ショアーズ、オラクル・パークウェイ、500
(43) 公表日	平成19年1月18日 (2007.1.18)	(74) 代理人	100064746
(86) 国際出願番号	PCT/US2004/010360		弁理士 深見 久郎
(87) 国際公開番号	W02005/003940	(74) 代理人	100085132
(87) 国際公開日	平成17年1月13日 (2005.1.13)		弁理士 森田 俊雄
審査請求日	平成19年3月30日 (2007.3.30)	(74) 代理人	100083703
(31) 優先権主張番号	10/459,811		弁理士 仲村 義平
(32) 優先日	平成15年6月11日 (2003.6.11)	(74) 代理人	100096781
(33) 優先権主張国	米国 (US)		弁理士 堀井 豊
前置審査			

最終頁に続く

(54) 【発明の名称】 データベースのカラムを暗号化するための方法および装置

(57) 【特許請求の範囲】

【請求項1】

プロセッサおよび記憶装置を含むコンピュータによって実行される、前記記憶装置に保持されているデータベースのカラム内部にあるデータの暗号化を容易にするための方法であって、

前記プロセッサが、データベースオペレーションを実行するコマンドを受取ることを含み、前記コマンドは、前記データベースへの参照および更新オペレーションを含む複数のオペレーションからなり、

前記プロセッサが、前記コマンドをパースして、解析木を作成することと、

前記プロセッサが、前記解析木を調べて、前記解析木の中で参照されるカラムが暗号化されたカラムであるか否かを判断することと、

前記解析木の中で参照されるカラムが暗号化されたカラムであれば、前記プロセッサが、前記データベースオペレーションの実行中において、前記暗号化されたカラムへアクセスすることを容易にするために、1つ以上の暗号化コマンドを含むように自動的に前記コマンドを変換することと、

前記プロセッサが、前記データベースオペレーションに前記暗号化されたカラムからの参照オペレーションを含むか否かを判断することと、

前記データベースオペレーションに前記暗号化されたカラムからの参照オペレーションを含めば、前記プロセッサが、前記参照オペレーション中に、前記暗号化されたカラムから検索されるデータを復号化するために、復号化オペレーションを含むように前記データ

ベースオペレーションを変換することと、

前記プロセッサが、前記コマンドが前記暗号化されたカラムへの更新オペレーションを含むか否かを判断することと、

前記コマンドが前記暗号化されたカラムへの更新オペレーションを含めば、前記プロセッサが、前記更新オペレーション中に、前記暗号化されたカラムの中で更新されるデータを暗号化するために、暗号化オペレーションを含むように前記更新オペレーションを変換することとを含む、方法。

【請求項 2】

カラムが暗号化される場合、前記方法は、さらに、前記プロセッサが、前記カラムのための暗号化鍵を識別することを含む、請求項 1 に記載の方法。

10

【請求項 3】

前記解析木を調べることは、さらに、

前記コマンドが前記データベースの中の前記カラムを暗号化する明示的なコマンドを含むかどうかを判断することと、

もし、そうである場合、前記カラムを暗号化することを含む、請求項 1 に記載の方法

。

【請求項 4】

前記解析木を調べることは、さらに、

前記コマンドが前記カラムのための暗号化鍵を変更するオペレーションを含むかどうかを判断することと、

もし、そうである場合、現在の暗号化鍵を用いて前記カラムを復号化することと、新しい暗号化鍵を用いて前記カラムを暗号化することを含む、請求項 1 に記載の方法。

20

【請求項 5】

前記解析木を調べることは、さらに、

前記コマンドが前記データベースの中の前記カラムを復号化する明示的なコマンドを含むかどうかを判断することと、

もし、そうである場合、前記カラムを復号化することを含む、請求項 1 に記載の方法

。

【請求項 6】

前記解析木を調べることは、さらに、

前記コマンドが前記カラムのための暗号化アルゴリズムを変更する明示的なコマンドを含むかどうかを判断することと、

もし、そうである場合、先の暗号化アルゴリズムを使用して前記カラムを復号化することと、新しい暗号化アルゴリズムを使用して前記カラムを暗号化することを含む、請求項 1 に記載の方法。

30

【請求項 7】

コンピュータによって実行されると、データベースのカラム内部にあるデータの暗号化を容易にするための方法をコンピュータに実行させる命令を記憶する、コンピュータが読込可能な記憶媒体であって、前記方法は、

データベースオペレーションを実行するコマンドを受取ることを含み、前記コマンドは、前記データベースへの参照および更新オペレーションを含む複数のオペレーションからなり、

40

前記コマンドをパースして、解析木を作成することと、

前記解析木を調べて、前記解析木の中で参照されるカラムが暗号化されたカラムであるか否かを判断することと、

前記解析木の中で参照されるカラムが暗号化されたカラムであれば、前記データベースオペレーションの実行中において、前記暗号化されたカラムへアクセスすることを容易にするために、1 つ以上の暗号化コマンドを含むように自動的に前記コマンドを変換することと、

前記データベースオペレーションに前記暗号化されたカラムからの参照オペレーション

50

を含むか否かを判断することと、

前記データベースオペレーションに前記暗号化されたカラムからの参照オペレーションを含めば、前記参照オペレーション中に、前記暗号化されたカラムから検索されるデータを復号化するために、復号化オペレーションを含むように前記データベースオペレーションを変換することと、

前記コマンドが前記暗号化されたカラムへの更新オペレーションを含むか否かを判断することと、

前記コマンドが前記暗号化されたカラムへの更新オペレーションを含めば、前記更新オペレーション中に、前記暗号化されたカラムの中で更新されるデータを暗号化するために、暗号化オペレーションを含むように前記更新オペレーションを変換することとを含む、
コンピュータが読込可能な記憶媒体。

10

【請求項 8】

カラムが暗号化される場合、前記方法は、さらに、前記カラムのための暗号化鍵を識別することを含む、請求項 7 に記載のコンピュータが読込可能な記憶媒体。

【請求項 9】

前記解析木を調べることは、さらに、

前記コマンドが前記データベースの中の前記カラムを暗号化するオペレーションを含むかどうかを判断することと、

もし、そうである場合、前記カラムを暗号化することとを含む、請求項 7 に記載のコンピュータが読込可能な記憶媒体。

20

【請求項 10】

前記解析木を調べることは、さらに、

前記コマンドが前記カラムのための暗号化鍵を変更する明示的なコマンドを含むかどうかを判断することと、

もし、そうである場合、現在の暗号化鍵を用いて前記カラムを復号化することと、新しい暗号化鍵を用いて前記カラムを暗号化することとを含む、請求項 7 に記載のコンピュータが読込可能な記憶媒体。

【請求項 11】

前記解析木を調べることは、さらに、

前記コマンドが前記データベースの中の前記カラムを復号化する明示的なコマンドを含むかどうかを判断することと、

もし、そうである場合、前記カラムを復号化することとを含む、請求項 7 に記載のコンピュータが読込可能な記憶媒体。

30

【請求項 12】

前記解析木を調べることは、さらに、

前記コマンドが前記カラムのための暗号化アルゴリズムを変更する明示的なコマンドを含むかどうかを判断することと、

もし、そうである場合、先の暗号化アルゴリズムを使用して前記カラムを復号化することと、新しい暗号化アルゴリズムを使用して前記カラムを暗号化することとを含む、請求項 7 に記載のコンピュータが読込可能な記憶媒体。

40

【請求項 13】

データベースのカラム内部にあるデータの暗号化を容易にするための装置であって、

データベースオペレーションを実行するコマンドを受取るように構成される受取機構を含み、前記コマンドは、前記データベースへの参照および更新オペレーションを含む複数のオペレーションからなり、

前記コマンドをパースして、解析木を作成するように構成されるパーズ機構と、

前記解析木を調べて、前記解析木の中で参照されるカラムが暗号化されたカラムであるか否かを判断するように構成される調査機構と、

前記解析木の中で参照されるカラムが暗号化されたカラムであれば、前記データベースオペレーションの実行中において、前記暗号化されたカラムへアクセスすることを容易に

50

するために、1つ以上の暗号化コマンドを含むように自動的に前記コマンドを変換するように構成される変換機構と、

前記データベースオペレーションに前記暗号化されたカラムからの参照オペレーションを含むか否かを判断する機構と、

前記データベースオペレーションに前記暗号化されたカラムからの参照オペレーションを含めば、前記参照オペレーション中に、前記暗号化されたカラムから検索されるデータを復号化するために、復号化オペレーションを含むように前記データベースオペレーションを変換する機構と、

前記コマンドが前記暗号化されたカラムへの更新オペレーションを含むか否かを判断する機構と、

前記コマンドが前記暗号化されたカラムへの更新オペレーションを含めば、前記更新オペレーション中に、前記暗号化されたカラムの中で更新されるデータを暗号化するために、暗号化オペレーションを含むように前記更新オペレーションを変換する機構とを含む、
装置。

【請求項 14】

カラムが暗号化される場合、前記カラムのための暗号化鍵を識別するように構成される識別機構をさらに含む、請求項 13 に記載の装置。

【請求項 15】

前記コマンドが前記データベースの中の前記カラムを暗号化する明示的なコマンドを含むかどうかを判断するように構成される判断機構と、

前記コマンドが前記データベースの中の前記カラムを暗号化する前記明示的なコマンドを含む場合、前記カラムを暗号化するように構成される暗号化機構とをさらに含む、請求項 13 に記載の装置。

【請求項 16】

前記コマンドが前記カラムのための暗号化鍵を変更するオペレーションを含むかどうかを判断するように構成される判断機構と、

現在の暗号化鍵を用いて前記カラムを復号化するように構成される復号化機構と、

新しい暗号化鍵を用いて前記カラムを暗号化するように構成される暗号化機構とをさらに含む、請求項 13 に記載の装置。

【請求項 17】

前記コマンドが前記データベースの中の前記カラムを復号化する明示的なコマンドを含むかどうかを判断するように構成される判断機構と、

前記コマンドが前記データベースの中の前記カラムを復号化する前記明示的なコマンドを含む場合、前記カラムを復号化するように構成される復号化機構とをさらに含む、請求項 13 に記載の装置。

【請求項 18】

前記コマンドが前記カラムのための暗号化アルゴリズムを変更するオペレーションを含むかどうかを判断するように構成される判断機構と、

現在の暗号化アルゴリズムを用いて前記カラムを復号化するように構成される復号化機構と、

新しい暗号化アルゴリズムを用いて前記カラムを暗号化するように構成される暗号化機構とをさらに含む、請求項 13 に記載の装置。

【発明の詳細な説明】

【技術分野】

【0001】

背景

発明の分野

この発明はデータベースのセキュリティに関する。より具体的には、この発明は、データベース内部のデータをカラム単位で透過的に暗号化および復号化するための方法および装置に関する。

10

20

30

40

50

【背景技術】

【0002】

関連技術

データベースのセキュリティは、多くのデータベースシステムにおいて重要な機能である。データベースシステムでは、セキュリティはデータベースシステム内部のデータを暗号化することによって達成されることが多い。現在、データベースシステムの中に記憶されるデータを暗号化するための2つの主要なアプローチがある。第1のアプローチは「一括暗号化」として特徴付けられることができ、データベースファイル全体で暗号オペレーションを実行する。第2のアプローチは、暗号オペレーションをデータベース内部の特定の機密事項を扱うカラムに選択的に適用する。

10

【0003】

機密事項を扱うデータは特定のテーブルの中に記憶されるだけではないので、一括暗号化は典型的にデータベース全体の暗号化を必要とする。機密事項を扱うデータは他のデータベースオブジェクトの中にも現われるであろう。たとえば、機密事項を扱うデータはインデックス、アンドゥーおよびリドゥーログの修正用レコード、ならびに一時的な分類領域の中に現われるであろう。これらのデータベースオブジェクトはデータベースシステム全体によって共用されるように設計されるので、暗号化されるデータもあれば、暗号化されないデータもあるというように、これらのデータベースオブジェクト内部のデータを分けることは実用的ではない。

【発明の開示】

20

【発明が解決しようとする課題】

【0004】

一括暗号化は実施が比較的容易であり、データベースにアクセスするアプリケーションに透過的であるが、その一方で重大な欠点がある。これらの欠点の中で主なものはシステムの性能劣化である。データベースファイル全体を暗号化または復号化するには長い時間を要する。このようなシステムでは、再入力オペレーションがデータベースファイル全体を復号化し、その後再暗号化することを含み得る。これらのオペレーションは多くの時間を要する可能性があり、そのためにこの解決法は大規模なオンライントランザクション処理の配置に適さない。また、データレコードがファイルから復号化された後に、データベースレコードがプレーンテキストとして共用記憶域の中で露出されるので、システムのセキュリティは損なわれ得る。

30

【0005】

第2のアプローチはデータベース内部の機密事項を扱うそれらのカラムのみに暗号化を限定するものであり、理論的には暗号オペレーションを実行することに関わるオーバーヘッドを低減し得る。しかしながら、このアプローチを使用する、現在使用可能なシステムには、いくつかの大きな欠点がある。暗号化および復号化オペレーションは、暗号化されたカラムのいずれの参照にも明示的に適用されなければならない。たとえば、社会保障番号が「123456789」である顧客のクレジットカード番号を検索するコマンドの発行を要望するアプリケーションは以下のコマンドを発行するであろう。

【0006】

40

【数1】

```
select credit_card_number from tab where ssn = '123456789'
```

【0007】

しかしながら、これらのカラムの両方が暗号化されると、クエリーはたとえば以下のような復号化コマンドを含むように修正されなければならない。

【0008】

【数 2】

```
select decrypt(credit_card_number) from tab where decrypt(ssn) = '123456789'
```

【0009】

暗号化および復号化機能が暗号アルゴリズムを選択するためのインターフェイスも与えなければならず、アプリケーションが鍵管理を与えなければならないということに注目されたい。

【0010】

したがって、データベースシステムメーカーが要求するのは透過的な暗号化および復号化オペレーションであるにもかかわらず、この第2のアプローチでは、暗号化および復号化オペレーションはアプリケーション開発者に透過的ではない。機密事項を扱うカラムがアクセスされると、暗号化または復号化機能はカラムデータに明示的に適用されなければならない。このようなランタイム機能の実行をユーザに透過的で、かつ安全にするために、アプリケーションスキーマオブジェクトは大きく変更されなければならない。たとえば、機密事項を扱うカラムを有するテーブルは、暗号機能を隠すために、ビューにされなければならない。これは、ビューおよびテーブルが同じ名前空間の中に存在し、名前を共有できないために、ベースオブジェクトの名前が変更されなければならないことも意味する。ビューの挿入または更新によってベーステーブルの中のデータが暗黙的に暗号化されるように、トリガが作成される必要がある。さらに、辞書式順序を失っている暗号化されたデータを用いることによってのみ、サーバがインデックスを構築し得るので、インデックスサポートは制限される。これは、暗号化および復号化オペレーションがインデックス処理層と統合されることができないためである。

【0011】

したがって、必要とされるのは、データベースシステム内部のデータをカラム単位で透過的に暗号化および復号化するための方法および装置である。

【課題を解決するための手段】

【0012】

概要

この発明の1つの実施例は、データベースのカラム内部にあるデータの暗号化を容易にするシステムを与える。システムは最初に、データベースオペレーションを実行するコマンドを受取ることによって作動する。次に、システムはコマンドをパースして、解析木を作成する。システムはその後、解析木を調べて、解析木の中で参照されるカラムが暗号化されたカラムであるかどうかを判断する。解析木の中で参照されるカラムが暗号化されたカラムであれば、システムは、データベースオペレーションを実行する間に、暗号化されたカラムにアクセスすることを容易にするために1つ以上の暗号オペレーションを含むように暗黙のうちに解析木を変換する。

【0013】

この実施例の変形例では、データベースオペレーションが暗号化されたカラムからの参照オペレーションを含む場合、平文を与えるために、参照オペレーションの間にシステムは解析木を変換して、暗号化されたカラムから検索されるデータを復号化する。

【0014】

さらに他の変形例では、コマンドが暗号化されたカラムへの更新オペレーションを含む場合、暗号化されたデータをデータベースの中に与えるために、更新オペレーションの間にシステムは解析木を変換して、暗号化されたカラムの中で更新されるデータを暗号化する。

【0015】

さらに他の変形例では、解析木の中で参照されるカラムが暗号化される場合、システムはカラムのための暗号鍵を識別する。鍵は、各々のコマンドに対して、カラムへのすべて

10

20

30

40

50

のアクセスに対して一度だけ回復される。

【0016】

さらに他の変形例では、解析木を調べることは、ユーザコマンドがデータベースの中で現在暗号化されていないカラムを暗号化する明示的な要求であるかどうかを判断することを含む。ユーザコマンドがデータベースの中で現在暗号化されていないカラムを暗号化する明示的な要求であれば、システムはカラムを暗号化する。

【0017】

さらに他の変形例では、解析木を調べることは、ユーザコマンドがカラムのための暗号化鍵を変更する明示的な要求であるかどうかを判断することを含む。ユーザコマンドがカラムのための暗号化鍵を変更する明示的な要求であれば、システムは現在の暗号化鍵を用いてカラムを復号化し、新しい暗号化鍵を用いてカラムを暗号化する。

10

【0018】

さらに他の変形例では、解析木を調べることは、ユーザコマンドがデータベースの中の暗号化されたカラムを復号化する明示的な要求であるかどうかを判断することを含む。ユーザコマンドがデータベースの中の暗号化されたカラムを復号化する明示的な要求であれば、システムはカラムを復号化する。

【0019】

さらに他の変形例では、解析木を調べることは、ユーザコマンドがカラムのための暗号化アルゴリズムを変更する明示的な要求であるかどうかを判断することを含む。ユーザコマンドがカラムのための暗号化アルゴリズムを変更する明示的な要求であれば、システムは現在の暗号化アルゴリズムを用いてカラムを復号化し、新しい暗号化アルゴリズムを用いてカラムを暗号化する。

20

【発明を実施するための最良の形態】

【0020】

詳細な説明

以下の記載は任意の当業者がこの発明をなし、使用することができるように提示され、特定の適用例およびその要件との関連で与えられる。開示される実施例に対するさまざまな修正が当業者に容易に明らかであり、ここに規定される一般的な原理は、この発明の精神および範囲を逸脱することなく、他の実施例および適用例に適用されるであろう。したがって、この発明は示される実施例に限定されるように意図されるのではなく、ここに開示される原理および特性と一致する、最も広い範囲を与えられるように意図される。

30

【0021】

この詳細な説明に記載されるデータ構造およびデータコードは典型的には、コンピュータシステムによって使用されるコードおよび/またはデータを記憶し得る任意の装置または媒体であろう、コンピュータが読込可能な記憶媒体上に記憶される。これは、ディスクドライブ、磁気テープ、CD（コンパクトディスク）およびDVD（デジタル多用途ディスクまたはデジタルビデオディスク）などの磁気および光学記憶装置、ならびに伝送媒体に含まれるコンピュータの命令信号（それらの信号がその上で変調される搬送波を伴うか、または伴わない）を含むが、これらに限定されるものではない。たとえば、伝送媒体はインターネットなどの通信ネットワークを含んでもよい。

40

【0022】

データベースシステム

図1は、この発明の実施例に従うデータベースシステムを図示する。データベースシステムはクライアント102、サーバ104、およびデータベース106を含む。クライアント102は概して、ネットワーク上に計算機能を含む任意のノードを含むことができ、ネットワーク全域で通信を行なうための機構を含んでもよい。

【0023】

サーバ104は概して、クライアントからの計算および/またはデータ記憶リソースの要求を処理するための機構を含む任意の計算ノードを含み得る。サーバ104は1つ以上のクライアントと通信し、各々のクライアントにサービスを与える。この通信は典型的に

50

は、インターネットまたは企業のイントラネットなどのネットワーク（図示せず）全域での通信である。サーバ104は、計算およびデータベースサービスを提供するように一斉に働くサーバのクラスタとして実現されてもよい。

【0024】

データベース106は、不揮発性記憶装置の中にデータを記憶するための任意のタイプのシステムを含み得る。これは、磁気記憶装置、光学記憶装置、および磁気光学記憶装置に基づくシステム、ならびにフラッシュメモリおよび/またはバッテリーバックアップメモリに基づく記憶装置を含むが、これらに限定されるものではない。データベース106はサーバ104に直接に結合されるか、または企業のイントラネットもしくはインターネットなどのネットワーク全域でアクセスされ得る。

10

【0025】

動作中に、クライアント102はデータベースコマンドをサーバ104に送る。これらのコマンドは典型的には構造化照会言語（SQL）などのデータベース言語の状態であり、データベース106上での参照および更新オペレーションを含み得る。これらの参照または更新オペレーションのいずれかが暗号化されたカラム上でのオペレーションを含む場合、オペレーションは図2 - 図5とともに以下に記載されるように処理される。

【0026】

サーバ

図2は、この発明の実施例に従うサーバ104を図示する。サーバ104はクライアントインターフェイス202、コマンドパーサ204、コマンド変換器206、暗号ユニット208、およびデータベースインターフェイス210を含む。クライアントインターフェイス202はクライアント102と通信して、クライアント102からのコマンドを受取り、クライアント102からのコマンドに応答する。これらのコマンドは、データベース106上で機能する、サーバ104のためのSQLコマンドを含み得る。

20

【0027】

コマンドパーサ204は、コマンドを、そのコマンドを含む個々の要素（オペランド、演算子など）にパースする。コマンドパーシングは当該技術に周知であり、この記載の中でさらに論じられることはない。

【0028】

コマンド変換器206はコマンドのパースされる要素を調べて、データベース106内部の暗号化されたカラムに関連する任意の参照または更新オペレーションを突きとめる。暗号化されたカラムに関連する参照または更新オペレーションを突きとめるのと同時に、コマンド変換器206は暗号化されたカラムにアクセスするために必要な暗号オペレーションを含むようにオペレーションを変換する。これらの変換するオペレーションは、図4Aおよび図4Bとともに以下に詳細に記載される。

30

【0029】

暗号ユニット208は鍵管理、暗号化、および復号化などの暗号オペレーションを実行する。多くの標準的な鍵管理システムのうちのいずれかがこのシステムとともに使用され得る。暗号化および復号化は、データ暗号化規格（DES）、トリプルDES、または先進暗号化規格（AES）などの任意の受入可能なアルゴリズムを使用して実行され得る。さらに、これらの暗号化アルゴリズムは、セキュアハッシュアルゴリズム1（SHA-1）またはメッセージダイジェスト5（MD5）などの統合技術と組合せられ得る。

40

【0030】

データベースインターフェイス210は、データベース106にアクセスするための機構を含む。これらのアクセスするオペレーションは、データベース106からデータを検索し、データベース106内部のデータを記憶または更新することを含み得る。コマンドの変換およびコマンドの実行は同じ事象シーケンスの中で起こらないかもしれないということに注目されたい。コマンドの実行はコマンドの変換よりも後に起こってもよい。

【0031】

データベースオペレーションを変換すること

50

図3は、この発明の実施例に従って、暗号オペレーションを含むようにデータベースオペレーションを変換するプロセスを図示するフローチャートを提示する。システムは、データベースオペレーションを実行するコマンドが受取られると、始動する(ステップ302)。次に、システムはコマンドをパースして、解析木を作成する(ステップ304)。システムはその後、この解析木を調べて、参照されるカラムまたは暗号化されたデータに関連付けられる式を突きとめる(ステップ306)。

【0032】

参照されるカラムを突きとめた後、システムはカラムが暗号化されているかどうかを判断する(ステップ308)。カラムが暗号化されていれば、システムは暗号オペレーションを含むように、この暗号化されたカラム上でのオペレーションを変換する(ステップ310)。この変換するプロセスはユーザに透過的であることに注目されたい。

10

【0033】

ステップ308でカラムが暗号化されていない場合、またはステップ310で暗号オペレーションを含むようにコマンドを変換した後、システムはコマンドの中で指定されるオペレーションを実行し、それによってコマンドを完了する(ステップ312)。コマンドの変換およびコマンドの実行は同じ事象シーケンスの中で起こらないかもしれないということに注目されたい。コマンドの実行はコマンドの変換よりも後に起こってもよい。

【0034】

解析木

図4Aは、この発明の実施例に従う、変換されない解析木を提示する。システムは、解析木を作成する、入力されるコマンドをパースする。たとえば、図4Aに提示される解析木は、以下のコマンドがどのようにパースされるかを図示する。

20

【0035】

【数3】

```
UPDATE employee SET sal = 1.01 * sal;
```

【0036】

演算子「*」は左のサブツリー(1.01)と右のサブツリー(sal)とを乗算し、その結果を「sal」カラムの中のデータベースに戻す。この解析木は、「sal」カラムが暗号化されていないと仮定する。

30

【0037】

図4Bは、この発明の実施例に従う、変換される解析木を提示する。この解析木は、カラム「sal」が暗号化されていると仮定する。再び、演算子「*」はその左のサブツリーとその右のサブツリーとを乗算する。しかしながら、右のサブツリーは復号化演算子「DO」を含むように変換されている。DO演算子は与えられるパラメータを使用して、左のサブツリー(この例では、sal)のカラムを復号化する。DOに与えられるパラメータは、アルゴリズム識別子「alg_id」および「GK」演算子の結果である。「GK」演算子は、与えられるパラメータを使用してカラム暗号化鍵を検索するゲット鍵(get key)演算子である。「GK」演算子への入力パラメータは、鍵管理タイプ、マスタ鍵識別子、およびカラム鍵識別子である。

40

【0038】

DOが「sal」カラムを復号化し、結果が「*」演算子によって1.01を乗算された後、暗号化演算子「EO」は結果を暗号化し、「sal」カラムでの記憶のためにデータベースに戻すように結果を渡す。EO演算子への入力は、「*」演算子の結果、アルゴリズム識別子「alg_id」、および「GK」演算子の結果である。鍵は変化していないので、復号化のための「GK」演算子の結果は暗号化のためのEOと共用されることに注目されたい。事実、「GK」演算子は、「sal」カラム全体を更新するために、一度だけ呼出される。

50

【 0 0 3 9 】

参照されるカラムのための暗号手法を含むコマンド

図 5 は、この発明の実施例に従って、カラムのための暗号手法を含むコマンドを実行するプロセスを図示するフローチャートを提示する。システムは、データベースオペレーションを実行するコマンドが受取られると、始動する（ステップ 5 0 2）。次に、システムはコマンドをパースして、解析木を作成する（ステップ 5 0 4）。システムはその後、この解析木を調べて、参照されるカラムのための暗号手法をコマンドが含むかどうかを判断する（ステップ 5 0 6）。参照されるカラムのための暗号手法をコマンドが含む場合、システムは参照されるカラム上で暗号オペレーションを実行する（ステップ 5 0 8）。

【 0 0 4 0 】

例として、コマンドが以下のような暗号化鍵を変更するコマンドであれば、システムは最初にカラムのためのメタデータを更新する。

【 0 0 4 1 】

【 数 4 】

```
ALTER TABLE employee MODIFY (ssn REKEY);
```

【 0 0 4 2 】

システムは次に、実行のために以下の UPDATE ステートメントに変換される更新ステートメントを暗黙的に発行する。

【 0 0 4 3 】

【 数 5 】

```
UPDATE (employee SET ssn = ENCRYPT( DECRYPT(ssn,
    k_algorithm_id, GET_KEY(key_mgn_type, master_key_id,
    col_key_id)), k_algorithm_id, GET_KEY(key_mgn_type,
    master_key_id, new_col_key_id));
```

【 0 0 4 4 】

概観

この発明は、カラムまたはカラムの属性の細かさで、データの暗号化を与える（オブジェクトデータベースの場合）。この暗号化は、データベース内部の暗号化されたカラムにアクセスするアプリケーションに透過的である。

【 0 0 4 5 】

カラムの守秘性は、内蔵の、またはユーザ定義の暗号化および復号化機能に依存することなく、カラムの特性の一部としてサポートされる。制約またはデータタイプなどの任意の他のカラムの特性と同様に、カラムの暗号特性はデータ記述言語（DDL）コマンドを使用して、いつでも定義および変更され得る。以下は、暗号化されたカラムの特性を定義および変更する典型的な管理タスクの例である。

【 0 0 4 6 】

機密事項を扱うデータは、以下のようなステートメントを使用して、異なる暗号化アルゴリズムで再暗号化され得る。

【 0 0 4 7 】

【 数 6 】

```
ALTER TABLE employee MODIFY (ssn ENCRYPT USING 'AES128');
```

【 0 0 4 8 】

システムは最初にカラムのためのメタデータを更新する。システムは次に、実行のための

10

20

30

40

50

以下のUPDATEステートメントに変換される更新ステートメントを暗黙的に発行する。

【 0 0 4 9 】
【 数 7 】

```
UPDATE (employee SET ssn = ENCRYPT( DECRYPT(ssn,  
    k_algorithm_id, GET_KEY(key_mgn_type, master_key_id,  
    col_key_id)), AES128, GET_KEY(key_mgn_type,  
    master_key_id, col_key_id));
```

10

【 0 0 5 0 】

セキュリティ要件は、暗号化鍵が定期的に変更されることを必要とするかもしれない。暗号化鍵を変更することは、パラグラフ [0 0 4 0] に記載されるように達成され得る。

【 0 0 5 1 】

その代わりに、暗号化されたデータをプレーンテキストの中で使用可能にするように判断がなされると、以下のコマンドが使用され得る。

【 0 0 5 2 】
【 数 8 】

```
ALTER TABLE employee MODIFY (ssn DECRYPT);
```

20

【 0 0 5 3 】

システムは最初にカラムのためのメタデータを更新する。システムは次に、実行のための以下のUPDATEステートメントに変換される更新ステートメントを暗黙的に発行する。

【 0 0 5 4 】
【 数 9 】

```
UPDATE (employee SET ssn = ( DECRYPT(ssn,  
    k_algorithm_id, GET_KEY(key_mgn_type, master_key_id,  
    col_key_id))));
```

30

【 0 0 5 5 】

テーブルが作成されるときに、カラムは暗号化されるカラムとして宣言されることも可能である。以下のDDLコマンドは、従業員テーブルの作成中に従業員テーブルのSSNおよび給料フィールドを暗号化する例を与える。

【 0 0 5 6 】

【数 1 0】

```

CREATE TABLE(
  name          VARCHAR2(30),
  employee_id   NUMBER(10),
  SSN           NUMBER(9) ENCRYPT USING 'DES3' AND HASH
                USING 'MD5',
  address       VARCHAR2(256),
  city          VARCHAR2(80),
  state         VARCHAR2(80),
  zip-code      VARCHAR2(10),
  salary        NUMBER(10) ENCRYPT,
  date_of_birth DATE,
  title         VARCHAR2(30)
);

```

10

【 0 0 5 7】

カラムが暗号化されるカラムとして指定されると、そのカラムの中のデータはすべてカラム暗号化鍵を用いて暗号化される。この鍵は、サーバのメタデータテーブルの中に記憶される前に、1つ以上のマスタ鍵によってラップされる。マスタ鍵の検索は、選択される鍵管理スキームおよびマスタ鍵の記憶位置に依存する。これは、任意のカラムの暗号オペレーションのために、サーバが最初にマスタ鍵を見つけ、暗号化されたカラム暗号化鍵を復号化するためにマスタ鍵を使用しなければならないということを意味する。

20

【 0 0 5 8】

透過的な暗号オペレーションのランタイムサポートは、サーバ上に3つの内部演算子を導入することに基づく。3つの内部演算子とは、(1)カラム暗号化鍵検索、(2)データの暗号化、および(3)データの復号化である。

【 0 0 5 9】

鍵検索演算子は、カラム暗号化鍵の識別、マスタ鍵の識別、および鍵管理タイプを受入れるために、引数を有する。この演算子はプレーンテキストの状態でカラム暗号化鍵を戻す。

30

【 0 0 6 0】

暗号化および復号化演算子は、暗号化されたカラムデータ、カラム暗号化鍵、および暗号化アルゴリズムの識別を識別するための引数を有する。鍵検索演算子がステートメントの実行ごとに一度だけ評価されることが望ましいので、鍵検索演算子は暗号化/復号化演算子から分けられる。

【 0 0 6 1】

ステートメントのパーズの際に、復号化演算子は、暗号化されたデータをサーバから受取ることになるカラムの属性のまわりに暗黙的に追加され得る。変換の後、式の中のカラムのみを典型的に参照することは、あたかも復号化機能が明示的に適用されるかのよう以下と等価である。

40

【 0 0 6 2】

【数 1 1】

```

DECRYPT (column, k_algorithm_id, GET_KEY(key_mgn_type,
                                         master_key_id, col_key_id))

```

【 0 0 6 3】

挿入値または更新値としてデータベースの中に与えられることになる式の値のために、暗号化演算子は以下のように追加される。

【 0 0 6 4】

50

【数 1 2】

```
ENCRYPT (expression, k_algorithm_id, GET_KEY(key_mgn_type,
      master_key_id, col_key_id))
```

【0065】

DECRYPTおよびENCRYPTコマンドへの引数の値が、暗号化されたカラムに影響を与えるDDLコマンド時に維持されるメタデータの中に含まれるので、これらのコマンドへの引数は、暗号化されたカラムデータを除き、パース時に既知であることに注目されたい。これらの引数は共用記憶域におけるステートメントコンテキストの一部である。しかしながら、これらの引数は機密事項を扱う情報を明かすことはない。暗号化鍵自体は実行時に一度だけ検索され、ユーザセッションの実行ごとの記憶域の中にのみ現われることになる。鍵の暗号化のためのアルゴリズムは情報管理用の設定可能なパラメータであり得るか、またはオプションの引数がGET_KEYコマンドに追加され得る。

10

【0066】

実行時に、上記に記載されるようにステートメントコンテキスト上で暗黙的に変換が行われるために、暗号化されたデータは式の評価の前に復号化される。挿入値および更新値のための式の評価が永続的な記憶に入った後、プレーンテキストは暗号化される。これはまた、暗号化および復号化の間、カラムのネイティブなデータタイプ形式が保存されることを保証する。したがって、式の評価のための既存の実現化例は影響を受けない。

20

【0067】

たとえば、従業員テーブルの中の給料カラム「sal」が暗号化されると仮定されたい。昇給のための以下の更新ステートメント

【0068】

【数 1 3】

```
UPDATE employee SET sal = 1.01 * sal WHERE empno = 999999;
```

【0069】

は、実際には以下のように実行されることになる。

30

【0070】

【数 1 4】

```
UPDATE (employee SET sal = ENCRYPT(1.01 * DECRYPT(sal,
      k_algorithm_id, GET_KEY(key_mgn_type, master_key_id,
      col_key_id)), k_algorithm_id, GET_KEY(key_mgn_type,
      master_key_id, col_key_id))
WHERE empno = 999999);
```

【0071】

鍵検索演算子は、どのマスタ鍵またはカラム暗号化鍵も普遍的に固有の鍵自体の識別を有する、複数の鍵管理スキームをサポートすることが可能である。特定のカラムを保護するカラム暗号化鍵が複数のコピーを有することができ、各々のコピーが異なるマスタ鍵によってラップされることに注目されたい。

40

【0072】

以下は、システムの柔軟性を示す例である。システムは変数「SERVER_HELD」によって識別される鍵管理タイプを有すると仮定されたい。このスキームでは、カラム暗号化鍵のすべてはサーバのワレットの中に保存される単一のマスタ鍵によってラップされる。管理者は、任意の数のマスタ鍵を生成するためにワレットマネージャを使用してもよい。しかしながら、以下のようなSQLコマンドが発行されると、サーバはデータベー

50

スのために1つのマスタ鍵のみを選択することになる。

【0073】

【数15】

ALTER DATABASE MASTER KEY my_db_ms_key;

【0074】

したがって、my_db_ms_keyは鍵の外部名である。サーバはまた、鍵に関連付けられる普遍的に固有の識別を作成する。サーバは現在のマスタ鍵の識別のみを記憶し、一方で鍵自体はワレットの中に残る。ワレットマネージャはデータベースの一部ではないため、データベース作成時にマスタ鍵の採用も行なわれるであろう。上記のコマンドもまた、データベースにおけるすべてのカラム暗号化鍵の再入力を必要とすることに注目されたい。しかしながら、暗号化されたカラムデータは影響を受けない。

10

【0075】

サーバが新しいカラム暗号化鍵を生成するか、または暗号化されたカラムを管理するDDLのうちの1つの結果として古いカラム暗号化鍵を置換えると、カラム暗号化鍵はサーバのマスタ鍵によってラップされる。カラム暗号化鍵ID、マスタ鍵ID、および暗号化されたカラムの鍵情報は、上記に記載されるように、鍵検索演算子GET_KEYのためのパラメータとして使用される。「SERVER_HELD」鍵管理タイプに基づいて、演算子はワレットアプリケーションプログラムインターフェイス(API)を介してワレットの中にサーバのマスタ鍵を見つけることができ、それによって暗号化および復号化オペレーションの両方のために使用されるプレーンテキストのカラム暗号化鍵を回復することができる。

20

【0076】

明らかに、鍵検索演算子のロジックは鍵管理タイプによって駆動される。鍵検索は常に、カラム暗号化鍵の識別およびマスタ鍵の識別を見る。新しい鍵管理スキームは、平文と暗号文との間の透過的なデータ変換の実施に影響を与えることなく、容易にシステムに接続され得る。これらの普遍的に固有の識別によって、演算子がランタイム時に鍵を見つけ得る限り、鍵はどこにでも記憶され得る。異なる鍵管理タイプをサポートするために、DDLコマンドを強化するか、または新しいDDLコマンドを加算する必要があるかもしれない。

30

【0077】

この発明の実施例の前述の記載は、例示および説明の目的でのみ提示されてきた。前述の記載は網羅的であるように、または開示される形態にこの発明を限定するように意図されるものではない。したがって、多くの修正および変更が当業者に明らかとなる。さらに、上記の開示はこの発明を限定するように意図されるものではない。この発明の範囲は特許請求の範囲によって規定される。

【図面の簡単な説明】

【0078】

【図1】この発明の実施例に従うデータベースシステムを図示する。

40

【図2】この発明の実施例に従うサーバを図示する。

【図3】この発明の実施例に従う、暗号オペレーションを含むようにデータベースクエリを変換するプロセスを図示するフローチャートを提示する。

【図4A】この発明の実施例に従う変換されない解析木を提示する。

【図4B】この発明の実施例に従う変換される解析木を提示する。

【図5】この発明の実施例に従う、カラムのための暗号手法を含むコマンドを実行するプロセスを図示するフローチャートを提示する。

【 図 1 】

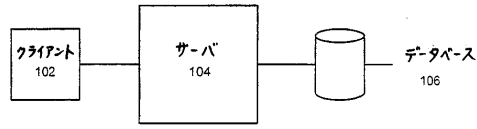


FIG. 1

【 図 2 】

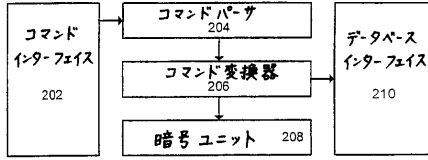


FIG. 2

【 図 3 】

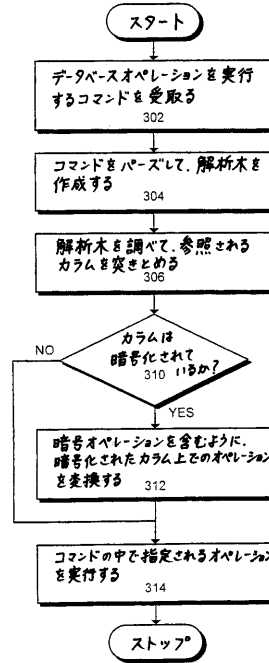


FIG. 3

【 図 4 A 】

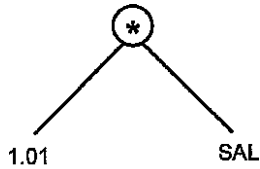


FIG. 4A

【 図 4 B 】

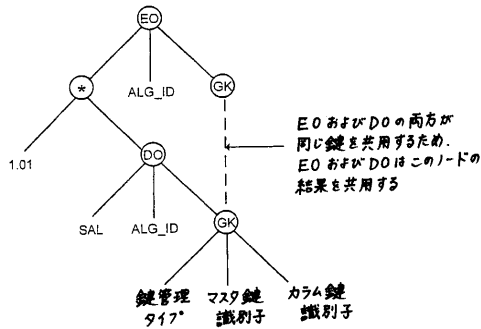


FIG. 4B

【 図 5 】

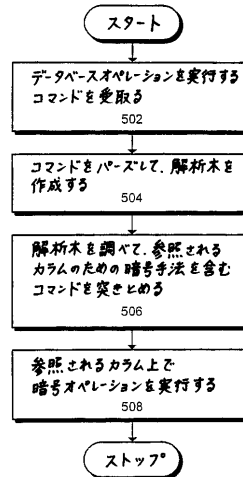


FIG. 5

フロントページの続き

(74)代理人 100098316

弁理士 野田 久登

(74)代理人 100109162

弁理士 酒井 将行

(72)発明者 レイ, チョン・ハイ

アメリカ合衆国、94502 カリフォルニア州、アラメダ、スイート・ロード、352

(72)発明者 キーフ, トーマス

アメリカ合衆国、94080 カリフォルニア州、サウス・サンフランシスコ、カペイ・サークル、37

(72)発明者 ウォン, ダニエル・エム

アメリカ合衆国、94080 カリフォルニア州、サウス・サンフランシスコ、アマリリス・コート、27

審査官 岸野 徹

(56)参考文献 国際公開第2002/029577(WO, A1)

特開平11-187013(JP, A)

特開2001-358705(JP, A)

特表2004-528615(JP, A)

特開2001-101055(JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 21/24

G06F 12/00

G06F 12/14

G09C 1/00