多様なセキュリティ操作を考慮したオブジェクト指向データベースの構造

安東　学　　松浦　幹太　　馬場　章

東京大学大学院 情報学環・学際情報学府
〒 113-0033 東京都文京区本郷 7-3-1

E-mail: qq16206@iii.u-tokyo.ac.jp, kanta@iis.u-tokyo.ac.jp, baba@hi.u-tokyo.ac.jp

近年、データベースのコンテンツは多様化し、多階層化している。このようなコンテンツには、著作権等の問題が生じることもあり、不正使用によるトラブルも多様化する傾向にある。本稿では、秘匿性の保証だけでなく、多様なセキュリティ操作に適したデータベース構造を考察する。

Object-oriented Databases for Wide Range of Security Management

Manabu ANDO　　Kanta MATSUURA　　Akira BABA

Interfaculty Initiative in Information Studies,
Graduate School of Interdisciplinary Information Studies,
The University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan

E-mail: qq16206@iii.u-tokyo.ac.jp, kanta@iis.u-tokyo.ac.jp, baba@hi.u-tokyo.ac.jp

When the databases have various and hierarchical contents, it also varies the problems by frauds. In some application, intricate problems, such as infringement of copyright, can occur. In order to solve these problems, we describe the structure of database for wide range of security management.

1　はじめに

近年、データベースに格納されるコンテンツの形態は多様化している。テキストだけではなく、画像や音声、動画を含むデータベースが増加する傾向にある。また、単一の対象を格納するデータベースであっても、それを様々な手法でデジタル化し格納するデータベースが増加している。つまり、コンテンツが階層構造をもつようになっている。

アプリケーションによっては、著作権や特許権などの問題が生じることもあり、不正使用によるトラブルも多様化する傾向にある。そこで、要求されるセキュリティ管理は秘匿性の保証だけでなく、著作権管理など多様化している。この問題を解決するためには、コンテンツの形態に応じたセキュリティ操作が必要となる。従来のデータベースセキュリティの研究では、アクセス制御を中心としたものが多数を占めている ([1][2][3][4][5])。つまり、データの秘匿性を重視した立場の研究が大部分である。しかし、多様な形態のコンテンツを保護するには、単純なアクセス制御だけでは不十分である。

また、セキュリティポリシー設定のための研究も行われているが、これらも大部分は秘匿性を確保するための研究である。また、これらは一つのセキュリティポリシーに各データを適用するものである ([2][3])。しかし、[6][7] ではデータ形態に応じたセキュリティ操作のためのポリシー設定を考察している。この研究からも、多様な形態をもつコンテンツには、各形態に応じたセキュリティが必要であるということがわかる。

以上のことから、本稿では、各種セキュリティ操作に適したデータベースの構造について考察する。

2 対象とするデータベースと要求されるセキュリティ操作

2.1 対象とするデータベース

対象とするデータベースは、多様な形態を格納するものであり、また単一の対象を様々な形態で格納しているデータベースである。このような条件を満たすデータベースとして、オブジェクト指向データベースを取り上げる。以下にオブジェクト指向データベースの概要を述べる ([8])。

2.1.1 構成要素

- 主体
  オブジェクトに対し何らかの操作を行う要素を主体と呼ぶ。主体の例は、ユーザやプロセスなどである。

- オブジェクト
  オブジェクト指向データベースにおける客体のことであり、主体に操作される要素である。オブジェクトは複雑な構造をもつデータや参照構造をもつデータなど、様々な形態のデータを扱うことができる。また、オブジェクトは属性とメソッドをもつ。主体はメソッドによってオブジェクトの属性値にアクセスする。

- 関連
  オブジェクト間の何らかの結びつきを関連という。オブジェクト間のメッセージのやり取りによって処理が進むオブジェクト指向システムでは、関連が重要な役割を担う。

2.1.2 具体例

具体例として、製品情報データベースを取り上げる (図 1)。これは単一の製品に関する情報を格納したデータベースの例である。単一の製品情報ではあるが、形態としては「画像」型、「テキスト」型を含んでいる。

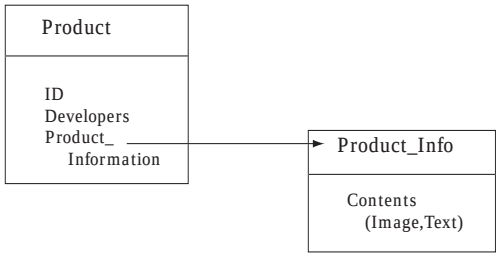

図 1: 製品情報データベース

2.2 要求されるセキュリティ操作例

多様な形態をもつオブジェクトに要求されるセキュリティ操作の具体例を以下に列挙する。

1. 暗号化
   オブジェクトがテキスト形態のとき、暗号化することが必要になる。また、暗号鍵と復号鍵の管理についても考慮するべきである。さらに、鍵管理を徹底することができれば、「内容が見えない」という意味でのアクセス制御も実現できる。
2. 認証子
   オブジェクトがテキスト形態のとき、そのテキストが改竄されていないことを証明するための認証子が必要である。電子署名や鍵付き MAC を使用するのであれば、鍵管理も必要である。
3. 電子透かし
   オブジェクトが画像、音声、動画のとき、不正コピーの抑止と証拠を残すために、電子透かしを埋め込む場合がある。
4. ログ
   各オブジェクトのメソッドを実行する際には、ログを取る必要がある。これにより、不正使用を早期に検知することができ、また後々の検証時に証拠とすることができる。
5. 時間情報の保証
   更新時間が正確であることを保証する必要がある。また、これを保証することにより、異常な更新時間の発見から不正使用を検知することができる。

3 セキュリティ操作に適したデータベースの構造

多様な構造に応じたセキュリティ操作を可能にするデータベース構造を考えるために、本稿では各セ

キュリティ操作をメソッドとして定義する。また、セキュリティ操作のためのルールを客体の一つとして扱う。これにより、客体の形態に応じた柔軟なセキュリティ操作を行うことができる。このような構造を前提として、各種セキュリティ操作に適したデータベースの構造について考察する。

3.1 セキュリティ対象となるコンテンツの扱い

図 1 の例において、Product_Info オブジェクトは Contents という属性をもつ。この属性は画像型とテキスト型をもっている。両者は単一の対象を示すものであるが、別の形態になっている。そのため、要求されるセキュリティ操作は異なる。柔軟なセキュリティ操作を可能にするためには、これらを別のオブジェクトとして表す方法 (図 2) と、別の属性として表す方法 (図 3) が考えられる。

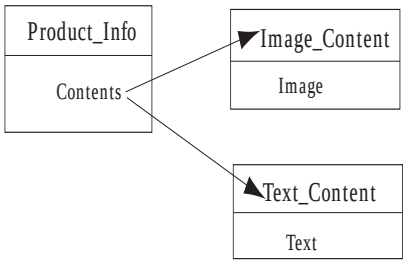

図 2: オブジェクト定義

図 3: 属性定義

3.1.1 オブジェクト定義

異なる形態の対象を別のオブジェクトとして定義するとき、オブジェクトの増加が問題となる。形態が増加すれば、それだけオブジェクトも増加するからである。さらに、セキュリティ操作を考慮すると、形態の数だけでは収まらない。例えば、テキスト型のコンテンツに対し、暗号化と認証子を定義することを考える。このとき、「暗号化コンテンツ」と「認証子付きコンテンツ」という二種類のオブジェクトが定義されることになる (図 4)。これを一つにしてしまうのでは、柔軟なセキュリティ操作に適しているとはいえない。よって、オブジェクト定義の場合、オブジェクトの増加が問題となる。

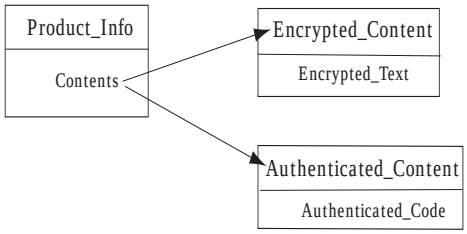

図 4: オブジェクトの増加

しかし、セキュリティ操作をメソッドで定義するため、各オブジェクトのメソッド数は増加しない。また、セキュリティ操作を増加させるとき、他のオブジェクトに影響を与えることなく定義することができる。

欠点としては、図 4 に見られるように、セキュリティ対象となるコンテンツが分散してしまうことである。

3.1.2 属性定義

異なる形態の対象を別の属性として定義するとき、3.1.1 の欠点を補うことができる。図 3 に見られるように、セキュリティ対象となるコンテンツを一元的に管理することができる。また、オブジェクトが増加することもない。セキュリティ操作を増加させたとしても、オブジェクト内の属性が増加するだけである (図 5)。

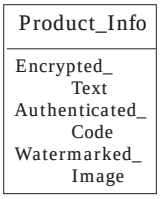

図 5: 属性の増加

しかし、逆にメソッドが増加することになる。セキュリティ操作をメソッドで定義するため、それぞ

れの操作に対する定義が膨大になってしまう。さらに、セキュリティ操作を増加させるときにも、メソッドと属性を改めて定義する必要があり、オブジェクト全体に影響を与えることになる。

3.1.3 比較と分析

以上の特徴を踏まえ、両者を比較する。第一に、セキュリティ操作を増加させたときのオブジェクト数とメソッド数について比較する。オブジェクト定義のとき、オブジェクト数とメソッド数は必要なセキュリティ操作の数に等しい。一方、属性定義のとき、オブジェクト数は常に一つであるが、メソッド数はセキュリティ操作の数に等しい。ただし、ここでは一つのセキュリティ操作に対しメソッド数を一つと仮定する。セキュリティ操作の数が n 倍になったときの比例係数を表 1 に示す。

定義形態	オブジェクト数	メソッド数
オブジェクト	n	n
属性	1	n

表 1: オブジェクト数とメソッド数の比較

また、オブジェクト定義のとき、オブジェクト数対メソッド数の比は 1 : 1 であるが、属性定義のときは 1 : n となる。オブジェクト定義においてセキュリティ操作数を増やしたとき、オブジェクト数は増加するが、メソッドは増加させたオブジェクトに対してのみ追加すればよい。一方属性定義のとき、オブジェクト数に変化はないが、そのオブジェクトのメソッド数が増加し、一つのオブジェクトに対する負荷が増加する。この結果から、オブジェクト数とメソッド数に関してはどちらが優れているとはいえない。アプリケーションの方針によって決定することになる。

次に、セキュリティ管理面について更新作業の点から比較する。更新作業の回数が増えると、対象が危険な状態にある回数も増える。更新作業中に不正使用される可能性もあるということである。このとき、属性定義のような構造であれば更新する必要のない属性が脅威にさらされることもあり得る。図 5 の例でいうと、Encrypted_Text の更新時にはその他の属性も危険な状態になる。また、更新を実行する主体が必要以上の権限をもってしまう。オブジェクト定義であれば、更新対象オブジェクト以外のオブジェクトには影響を与えずに更新作業を行える。よって、更新作業中の不正使用を防ぐことができる。したがって、セキュリティ管理面からはオブジェクト定義が優れているといえる。

以上のことから、多様な形態をもつオブジェクト指向データベースには、セキュリティ操作の数だけオブジェクトを定義する構造が適していると考えられる。

3.2 セキュリティ操作に関するルールの記述

本稿で対象としているデータベースでは、セキュリティ操作はメソッドで定義する。また、前述したように、セキュリティ操作を実行するためのルールを客体の一つとしている。ここでは、このルールを Status と呼ぶ。各種メソッドを実行する際には、必ず Status を参照し、実行する。Status の具体的な内容を以下に挙げる。

- 各種セキュリティ操作を実行するときの方式やプロトコル
- セキュリティ操作に伴う条件

多様なセキュリティ操作を必要とするデータベースにおいては、各セキュリティ操作を独自に行うだけでは十分な管理ができない。いくつかのセキュリティ操作を関連させることによって、必要とされる管理を実行できる。後者はそのための条件である。これには、[6][7] における必須処理の記述を適用することができる。例えば、テキスト型のコンテンツを更新するとき、「データは暗号化し、認証子を更新しなければならない」というルールを設定することにより、データの秘匿性と真正性を保証することができる。また、Status を管理するためには「Status を更新するためのルール」も必要となる。このような内容をもつ Status をどのように定義するかについて考察する。

3.2.1 Status オブジェクト定義

Status をオブジェクトとして定義する例を図 6 に示す。この構造では、ルールを一元管理することがで

きる。Status の内容はここで対象とするデータベースにおいて非常に重要な役割を果たすものであるため、一元管理はセキュリティ管理を容易にすると考えられる。また、他にも Status オブジェクトを定義しておけば、参照先を変更するだけで異なるセキュリティ操作を行うこともできる。つまり、拡張性がある。

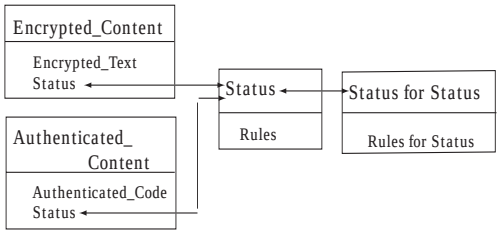

図 6: Status オブジェクト定義

しかし、Status の記述は複雑になってしまう。前述したような、数種のセキュリティ操作を関連付けるための内容を、一つのオブジェクトに記述するのは困難であると考えられる。

3.2.2 Status 属性定義

Status を属性として定義する例を図 7 に示す。属性定義にすると、ルールが分散してしまうが、セキュリティ対象となるデータの形態に応じたセキュリティ操作が容易である。また、それぞれの記述も容易になる。

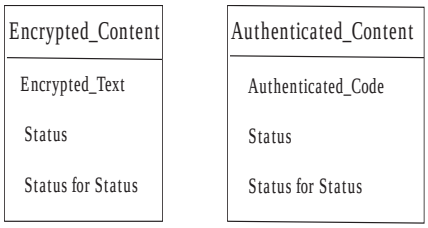

図 7: Status 属性定義

しかし、数種のセキュリティ操作の関連付けを考えると、属性定義は困難である。セキュリティ操作を関連付けるためには、適切な Status の記述が必要であるからだ。例えば、図 7 において、Encrypted_Text の更新を認められている主体が Authenticated_Code の更新を認められていなければ、このコンテンツの真正性を保証することはできない。したがって、分散されている Status が適切なものであるかどうか、常に確認する必要があり、管理が困難となる。

3.2.3 比較と分析

以上のことを踏まえ、Status のオブジェクト定義と属性定義を比較・分析する。第一に、データ更新への対応について考える。対象とするデータベースにおいて、更新量が最も多いと考えられる客体は、Encrypted_Text のような普通のコンテンツである。逆に、Status for Status は更新の可能性が低い。Status は更新する可能性はあっても頻繁ではない。このような更新量を仮定するとき、Status を属性定義すると、普通のコンテンツの更新作業中に Status が漏洩する可能性がある。Status が漏洩すると、無差別改ざんなどの脅威にさらされることも考えられる。したがって、Status を普通のデータと同レベルで管理することは、Status の管理に適していないと考えられる。Status は対象とするデータベースにおけるセキュリティ操作に対して重要な役割を果たすため、属性定義ではセキュリティ管理に不適当である。

第二に、アクセス要求時の処理について考える。アクセス要求時には必ず Status を参照することを仮定する。Encrypted_Text の更新を要求すれば、Status を参照して「暗号化と認証子の更新」が求められることになる。ここで、不適切な Status があると、要求されるセキュリティ操作を満たすことができない。オブジェクト定義であっても、属性定義であっても、不適切な Status があることは、セキュリティ管理において大きな欠陥となり得る。しかし、前述したように両者とも Status の記述が複雑になることは不可避である。よって、不適切な Status を早期に発見できるような構造が求められる。Status を一元管理することは、不適切な Status の発見を容易にするため、オブジェクト定義が適当な構造である。

第三に、Status の拡張性について考える。属性定義のとき、Status を変更・拡張する際に、不整合が生じないようにする必要がある。そのため、一つの Status を変更するごとに、全ての Status を変更しなければならない、という状況も考えられる。これにより、Status が漏洩してしまったり、新たな不適切な Status が生じるなど、セキュリティ管理上不適当な状態になることもあると考えられる。一方、オブジェクト定義では、数種の Status オブジェクトを定義しておけば、参照先を変更するだけで内容を更新

することができる。よって、多様なセキュリティ操作に対応することができる。

以上の比較結果から、Status はオブジェクト定義にしたほうが多様なセキュリティ操作に対応できると考えられる。

4 おわりに

多様なセキュリティ操作に適したデータベースの構造について考察し、以下の二点の結論を得た。(1) 要求されるセキュリティ操作数のオブジェクトを定義し、コンテンツの各形態を独自に保護する構造が必要である。更新作業中に関係のないコンテンツを保護するということからも、セキュリティ操作に応じたオブジェクトの定義が必要である。(2) セキュリティ操作に関するルールを記述する Status は非常に重要な役割を果たすため、オブジェクトとして定義する。これにより、普通のコンテンツの更新時における Status の漏洩を防止することができる。また、多階層的なコンテンツを保護するために複数のセキュリティ操作を関連付けると、Status は複雑な記述になりがちであるが、Status オブジェクト定義で一元管理しておけば不適切な記述を早期に発見することができる。以上の二点が本稿における結論である。今後の課題は、Status の記述方法と記述内容についての考察である。

参考文献

[1] Eduardo B. Fernandez, Ehud Gudes, and Haiyan Song: “A Model for Evaluation and Administration of Security in Object-Oriented Databases,” IEEE Transactions on Knowledge and Data Engineering, Vol.6, No.2, pp275-292. IEEE, April, 1994.

[2] Sushil Jajodia, Pierangela Samarati, V.S.Subrahmanian, and Elisa Bertino: “A Unified Framework for Enforcing Multiple Access Control Policies,” Proceedings of International Conference on Management of Data, pp.474-486. ACM SIGMOD, May, 1997.

[3] Dirk Jonscher and Klaus R. Dittrich: “Argos–A Configurable Access Control System for Interoperable Environments,” Database security IX : status and prospects : proceedings of the Ninth Annual IFIP TC11 Working Conference on Database Security, pp.43-60. Chapman & Hall, 1996.

[4] 木下宏揚: “データベースのアクセス制御に関する考察, ” 1996 年暗号と情報セキュリティシンポジウム予稿集, 電子情報通信学会, SCIS’96-10D, 1996 年 1 月.

[5] 多田共行, 今井秀樹: “オブジェクト指向アクセス制御モデルの考察,” 1998 年暗号と情報セキュリティシンポジウム予稿集, 電子情報通信学会, SCIS’98-4.3.D, 1998 年 1 月.

[6] 工藤道治, 羽田知史: “必須処理付きセキュリティポリシーのためのアクセス制御モデル,” コンピュータセキュリティ研究会予稿集, 情報処理学会, pp.149-156. 2001 年 7 月.

[7] Michiharu Kudo and Satoshi Hada: “Access Control Model with Provisional Actions,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, E84-A, pp295-302. January, 2001.

[8] 石塚圭樹: オブジェクト指向データベース, アスキー, 1996 年.