中華民國八十四年全國計算機會議論文集
Proceedings of National Computer Symposium 1995

商業化WWW應用之安全存取控制策略
A SECURE ACCESS CONTROL STRATEGY FOR COMMERCIAL WWW APPLICATIONS

陳奕明 Yi-Ming Chen
曾黎明* Li-Ming Tseng

國立中央大學資訊管理學系所
Department of Information Management, National Central University
Chung-Li, Taiwan 32054, R.O.C

*國立中央大學資訊工程學系所
Department of Information Engineering, National Central University
Chung-Li, Taiwan 32054, R.O.C

摘要

WWW(World Wide Web)因爲能處理各式資料型態且在Internet上擁有各種用戶端界面，所以具有極大的商業應用潛力。但是要將WWW作商業化應用的最大挑戰就是保証該WWW伺服器的存取控制（Access control）功能能滿足商業環境的需求，包括使用者的分類、文件的分級及存取權的動態給予與收回等。目前WWW伺服器的存取控制多採用傳統的任意裁決存取控制（Discretionary access control）策略，該策略因缺乏一套正規化的方法(Formal method)來保証每一物件的存取都在授權範圍內，所以若要滿足商業環境的應用需求，在系統實現和管理上將增加非常大的負擔而造成WWW商業化應用的阻力。爲解決此問題，本文提出一個以角色分類(Role classification)爲基礎的存取控制策略。此策略將使用者依其瀏覽WWW文件的授權權限不同而分成不同角色並將WWW管理的文件以物件導向(Object-Oriented)的方式予以描述；經由一套正規化方法，我們可以明確地定義每個使用者對每份WWW文件的授權政策（Authorization policy），同時我們也發展一套安全存取控制的演算法，可讓大部分的安全檢查動作交給系統去做而免除人工管理可能造成的失誤和負擔。文中除介紹角色分類的存取控制策略外，對於如何應用與實現此控制策略在文中也將加以說明。

關鍵詞：存取控制策略，授權政策，物件導向，角色分類

Abstract

This paper addresses the access control problem of WWW(World Wide Web) service. Currently, most WWW services adopt the discretionary access control strategy which lacks a formal model to ensure that every access of objects obeys the access authorization policy. As a result, the access control is hard coded in the implementation of application software and is hard to maintain and verify the correctness of codes. In this paper, we propose a new access control strategy based on the role classification concept to overcome this disadvantage. This new strategy divides users into an uncycled role graph and describes the documents within the WWW server by an object-oriented model. As being able to formally describe the authorization policy between roles and documents, our strategy has the following advantages: (1)the change of users' access rights won't affect the access policies, only the associations among users and roles needed to be modified. This advantage will save many management overheads on the maintenance of authorization policies. (2)the system can automatically perform the consistency check of authorization policies. This capability will reduce the security flaw due to the mistake of authorization policies.

key words: access control, authorization policy, object-oriented, role-classification

1. 簡介

自從美國柯林頓總統提出國家資訊基礎建設(National Information Infrastructure, NII) 的計劃以提升美國經濟競爭實力以來，國內各界近年來也全力推動類似計劃，並積極發展相關的網路應用，以期充分利用到 NII 建設的成果。在各種廣域網路的應用中， WWW(World Wide Web)因爲能處理各式資料型態且在 Internet 上擁有各種用戶端界面，具有極大的商業應用潛力而備受矚目[1][2][3]。然而要將目前在 Internet 上使用的 WWW 作商業化應用，最大的挑戰就在於其安全上的考量[2]：例如某公司一方面將其產品型錄放在 WWW 伺服器上歡迎網路上的所有人瀏覽該產品資訊，另一方面對於若干敏感性資料（如產品報價）卻又希望只讓某些特定人士看到（如經銷商），這時就需要 WWW 伺服器具有區別敏感和非敏感性資料，並對不同的使用者群體給予差別存取權利（ Access rights ）的能力。然而過去的 WWW 主要用於學術研究環境，對安全性的需求不高，所以任何一個網路上的使用者只要連上某個 WWW 伺服器，幾乎都能看到該伺服器的所有資料，明顯地無法滿足商業化應用的需求。

近年來，在 Internet 商業化的浪潮下，越來越多的研究者希望提供更嚴密的安全機制供 WWW 使用[4][5][6]，如 EIT 公司提出的 Secure Mosaic 與 SHTTP 及 Netscape 公司提出 SSL(Secure Socket Layer)都是爲了加強 WWW 的安全設計。可是這些研究大都偏重在使用者身分的鑑定（ Authentication ）及資料傳送的隱秘性（ Confidence ）上面，對於存取控制（ Access control ）問題- 亦即保証每一個使用者都在系統授權的範圍內使用系統資源[7][8]，卻多沿用傳統的任意裁決存取控制（ Discretionary access control ）策略[7]。此策略允許客體(Object) 的擁有者自行決定該客體可被那些主體(Subject)存取，對存取控制提供很大的彈性，但卻缺乏一套正規化的方法來保証每一物件的存取都在授權範圍內，所以若要滿足商業環境的應用需求，在系統實現和管理上將增加非常大的負擔而造成 WWW 商業化應用的阻力。爲解決此問題，本文提出一個以角色分類(Role classification)爲基礎的存取控制策略。此策略將使用者依其瀏覽 WWW 文件的授權權限而分成不同角色並將 WWW 管理的文件以物件導向(Object-Oriented) 的方式予以描述，因此在此策略中，我們不但可以明確地定義每個使用者對每一份 WWW文件的授權政策（ Authorization policy ），同時具有下列優點：（ 1 ）使用者的加入/退出或權限異動都只須更改使用者和角色之間的對應關係，不須更動任何授權政策，可減少許多管理上的虛耗（ Overheads ），（ 2 ）系統可對授權政策自動作一致性檢查（ Consistency check ），避免因一時的設定錯誤而造成安全漏洞。

本文分成六節。第二節首先說明商業化 WWW 應用所需的安全存取控制功能與傳統存取控制策略的限制，同時提出我們的解決方案。第三節介紹使用者角色分類模式。第四節說明以物件導向爲基礎的 WWW 文件架構。第五節介紹基於上述角色分類模式與物件導向文件架構的安全存取控制演算法。最後我們在第六節作一結論。

2.問題陳述

2.1 為何商業化WWW應用需要安全存取控制功能？

對於一個商業應用的 WWW 伺服器而言，其需要做到存取控制的理由約有下列數項：

（ 1 ）一方面希望將公司的所有資料（如產品資訊、財務報表、人事資料等）都放在同一個伺服器上以方便所有人去取得所需的資訊，但另一方面有些資料卻只適合讓部份人瀏覽（如人事資料只限於公司內部少數人存取），因此須對資料加以分級並對資料的存取作控制。

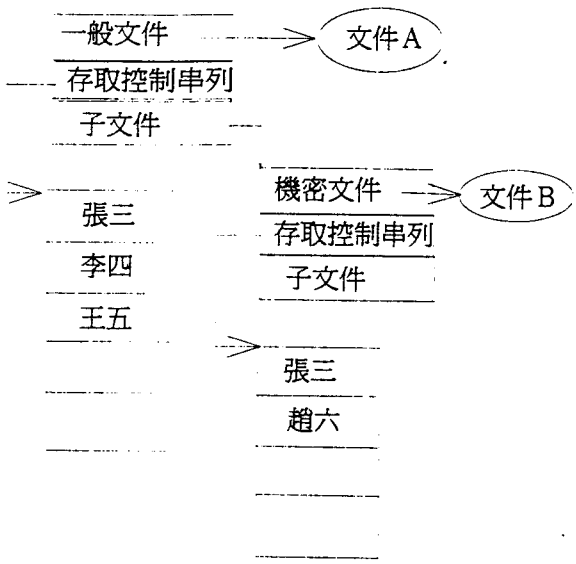
（ 2 ） WWW 不只提供瀏覽（ Browse ）功能，還可以作爲資訊服務工具。例如某出版公司將新書資訊放在 WWW 伺服器供其讀者俱樂部的會員瀏覽時，可先將會員依加入俱樂部的時間長短或繳交會費多寡之不同區分成不同等級，然後讓不同等級的會員看到不同的資訊（如某些人只能看到新書目次，有些人則能看到精彩摘要等）。

（ 3 ）目前每個使用者進入 WWW 伺服器後看到的畫面都一樣，如此特性使得 WWW 服務提供者（ Service provider ）不易設計一個同時適合各種不同程度、背景的使用者的首頁(Home page)。若能允許不同的使用者看到不同的首頁或以不同的資訊排列方式瀏覽，不但方便 WWW 伺服器內資訊的安排，從使用者觀點，在瀏覽 WWW 文件時系統若能自動過濾不相干的資訊將可免於迷失在資訊海中[3]。

由以上舉例可知，要做到安全存取控制除了使用者身份的辨識與認證外，最重要的就是建立使用者身份和 WWW 伺服器內所存文件之間的對應關係並保證此對應關係不會被違反。

2.2 為何傳統的存取控制方法不適用？

目前 WWW 伺服器（如 CERN 3.0 版的 http server ）的存取控制多採用任意裁決存取控制策略。其作法如圖一所示。



圖一：任意裁決存取控制例

圖一作法存在下列缺點：

（1）增加/刪除使用者或使用者權限異動時須找出所有和該使用者有關之文件的存取控制串列(Access control list)並修改之。例如張三由專職員工改成兼職員工而縮小其授權範圍時，須找出所有和張三有關之文件的存取控制串列並于以修改。

（2）各文件的存取控制串列內容重複設定（如張三的名字重複出現在文件 A 和文件 B 的存取控制串列中），造成儲存空間的浪費。

（3）存取權之一致性（Access rights consistency）不易檢查。例如張三不被授權去存取機密文件，但在上述機制中，一旦存取控制串列內容設定錯誤，系統不會自動檢查出來。

爲了改善上述缺點，我們提出一個以角色分類爲基礎的存取控制策略。其系統架構如圖二所示，系統動作流程爲：

（i）首先由 WWW 服務提供者將使用者分類成不同角色，完成角色結構圖(Role structure graph)，該圖說明使用者與角色及角色與角色間的關係（模組一），

（ii）提供文件供人瀏覽者，根據角色結構圖，決定好角色和文件間的授權關係並將此授權關係編輯在文件裡（模組二），

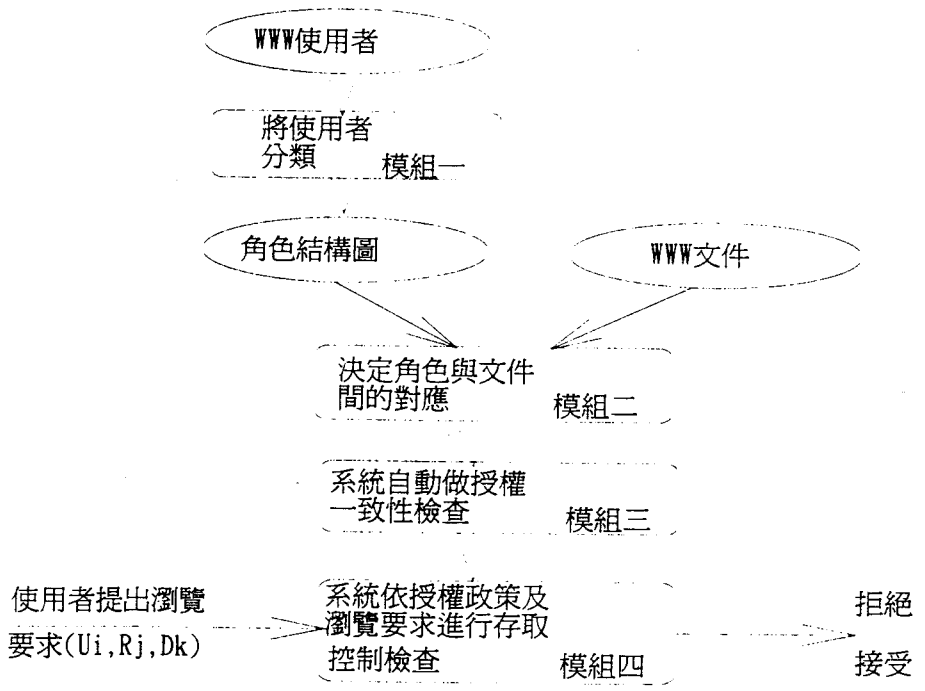
（iii）系統收到文件後自動做安全性檢查(模組三），

（iv）使用者要瀏覽此 WWW 文件時，除使用者識別碼外尚須聲明其角色識別碼及所欲瀏覽的文件，之後由系統自動做存取控制（模組四）。

以上系統架構之特點爲：

（a）通常角色與角色間的關係一如公司組織圖一樣，一經建立好後通常不會輕易改變，所以增加或刪除使用者只要改變使用者與角色間的對應關係即可（在模組一中），不會影響到系統其他部份，

（b）授權政策由文件提供者來決定並由系



圖二：WWW安全存取控制系統示意圖

統自動做安全性檢查，不但可讓文件提供者確信其授權未被扭曲，且 WWW 服務提供者可免於做存取權一致性檢查等繁瑣工作。

以上系統動作流程及上述兩項特點的實現將在下面文章中作詳細說明。

3. 使用者角色分類模型

本模型旨在將使用者依其瀏覽 WWW 文件的不同授權程度，區分成不同的角色以方便存取控制。爲說明方便起見，我們定義了下列名詞：

[定義一]：使用者（ **User, U** ）

使用者意指瀏覽 WWW 文件的個人，以 U_i 表示，$i \in \{1,2,...,M\}$, M 表示此 WWW 伺服器所能接受的最大的使用者個數。使用者可分成二類：長期的和臨時的，前者例如公司內職員，後者例如由網路上進來的使用者。不管是哪一種，系統都會給予一個唯一的識別碼。對長期使用者而言，其識別碼是經由向 WWW 服務提供者註冊而得。臨時性的使用者則是進入 WWW 伺服器時才由伺服器給予一個識別碼，該識別碼在使用者離開 WWW 後便由系統收回。

[定義二]：角色（ **Role, R** ）

角色代表一個使用者去瀏覽 WWW 文件時的一種身分。例如某公司將其內部資訊及產品型錄建成 WWW 文件供公司內外人員瀏覽時，這時使用者的角色就可分爲業務經理、業務代表、業務部門人員、人事課長、總經理、公司內部人員、經銷商及消費者等等（參見圖三之角色分類例子）。

值得注意的是，角色的定義隱含下列兩層意義：

（ 1 ）一個人可能扮演一種以上的角色。譬如張三可能兼任總經理和業務經理。所以當一個扮演多種角色的人進入系統時，必須聲明他是以何種角色進入。

（ 2 ）角色可能由其他多種角色構成，例如公司內部人員這個角色包括業務部門人員、人事部門人員和總經理等角色。爲了釐清不同角色間的關係，我們將角色分成直接角色和間接角色兩大類：

[定義三]：直接角色（ **Direct Role, DR** ）

$DR=\{U_i\}$, $i \in \{1,2,...,M\}$。

若 $DR_i=\{U_1,U_2\}$, 則稱 U_1,U_2 屬於 DR_i 且以 $USER(DR_i)=\{U_1,U_2\}$ 表示。

[定義四]：間接角色（ **Indirect Role, IR** ）

$IR=\{DR\}^* \cup \{IR'\}^*$, *表示該集合內的元素出現 0 次或 0 次以上，但{DR}和{IR'}不能同時爲空集合。

若 $IR_i=DR_j \cup IR_k$，則 $USER(IR_i)=USER(DR_j) \cup USER(IR_k)$。

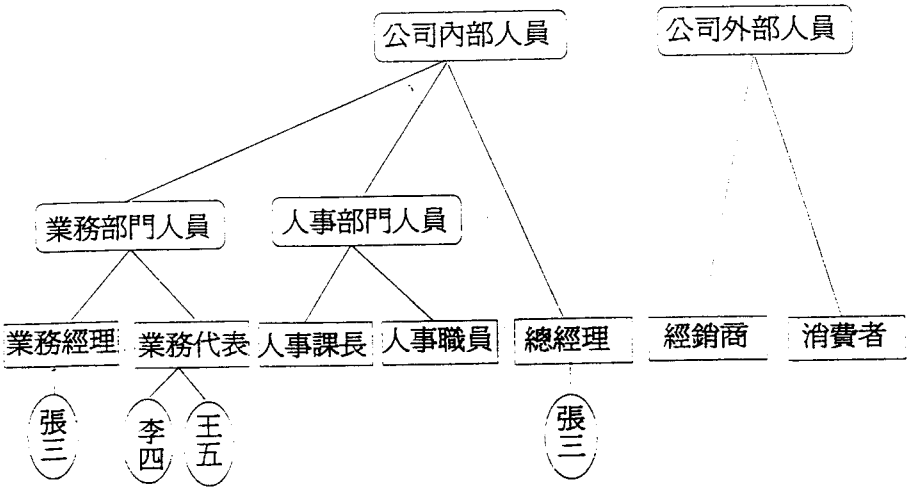
由以上兩個定義可知直接角色代表一種最特別的身分（例如業務代表），間接角色代表其各組成角色（ Component role ）的共同身分（例如業務部門人員和總經理皆爲公司內部人員）。

[定義五]：共通性指標（ **Commonnesss Index, CI** ）

CI 爲一函數，其輸入爲角色，輸出爲包含 0 的正整數。其定義爲：

$CI(DR)=0$

$CI(IR_i)= \max CI(IR_j)+1$



圖三：角色結構圖之例

where $IR_j \in \{\text{Components of } IR_i\}$

由以上定義可知 CI 值愈大者，其包含其他角色的共通性愈高。

[定義六]：獨立與普化/殊化關係（ In-dependent and Generalized/Specialized Re-lationship ）

若 $CI(R_i)=CI(R_j)$ 或 $USER(R_i) \cap USER(R_j)=\varnothing$，則稱 R_i,R_j 互相獨立；否則若 $CI(R_i)>CI(R_j)$ 且 $USER(R_j) \cap (USER(R_j) \neq \varnothing$，則意指角色 R_i 具有較角色 R_j 更普遍化的身份，亦即屬於 R_j 的使用者一定也屬於 R_j，$USER(R_i) \supset USER(R_j)$，但反之不成立。我們稱 R_i 爲 R_j 的較普化角色(generalized role)，R_j 爲 R_i 的較殊化角色(specialized role)。

4.以物件導向為基底的WWW文件

此處文件泛指 WWW 所處理的各種型態的資料，爲了簡單起見皆以文件稱之並且在以下的討論中僅考慮和存取控制有關的部份。因爲物件導向模式具有繼承性(inheritance)等優點[3][9][10]，所以我們將 WWW 伺服器內的文件組成物件導向架構。每一個文件可包含其他較底層的文件，其定義如下：

[定義七]：文件（ Document, D ）

每一文件可由下列四項屬性定義： D={DID, p.D, c.D, ARS}，其中 DID 表該文件的識別碼， p.D 代表 D 的父層文件(Parent document)， c.D 代表所屬的子層文件(Child document)， ARS (Access Role Set)代表被允許瀏覽該文件的角色集合(Role set)。每一文件的 ARS 都可能不同，文件和 ARS 的配對稱爲授權政策(Authorization policy)。

爲了簡化系統的管理及提高系統的安全起見，授權政策的指定還須遵守下列三項規則：

[規則一]：存取權的繼承規則（ Inheritance Rule ）

令 D_i 表文件 i， $ARS(D_i)$表 D_i 的 ARS。此繼承規則的規定如下：

若 $R_i \in ARS(D_i)$, 則 $R_i \in ARS(c.D_i)$，除非存在一 R_j, $CI(R_j)<CI(R_i)$ 且 $R_j \in ARS(c.D_i)$

以上規則之意義爲：除非在 D_i 的子層文件中需要更殊化的角色才能存取（譬如不只是公司內部成員而且還必須是人事經理），否則能存取 D_i 者也能存取 $c.D_i$。

[規則二]：授權的一致性規則（ Consistency Rule ）

若 $CI(R_i) < CI(R_j)$ 且 $R_i \notin ARS(D_i)$ 則 $R_j \notin ARS(D_i)$

此規則保証若某一角色不被允許去瀏覽 D_i，則較其普化的角色也不應被允許。例如若人事部門人員不被允許瀏覽產品報價資料，則其較普化角色（如公司內部人員）也應禁止瀏覽該份資料。

[規則三]：授權的簡潔規則(Non-Redundency Rule ）

若 R_i 與 R_j 皆屬於 $ARS(D_i)$，則 R_i, R_j 互相獨立

此規則意指若兩個或兩個以上的角色皆可瀏覽 D_i 且有普化/殊化關係，則僅須將 CI 值較高者放到 $ARS(D_i)$ 中以減少 ARS 的冗餘(Redundency)。

利用以上較正規化的模式描述，我們可以發展一套 WWW 伺服器的存取控制機制，使得系統能依使用者進入伺服器內時扮演的角色及文件授權政策的宣告來自動做安全存取控制檢查。下面一節即說明安全存取控制的演算法。

5.安全存取控制演算法

根據前面兩節的定義，我們可將以角色分類爲基礎的 WWW 安全存取問題表達如下：（參見圖二）

已知：（ 1 ）已註明 ARS 宣告的 WWW 文件
（ 2 ）角色結構圖

輸入：(U_i,R_j,D_k)

輸出：拒絕或接受該瀏覽要求；若接受，還須傳回 D_k 的子層文件（亦即 $c.D_k$ ）。

使用者輸入其身分識別碼 U_i 、所欲扮演的角色 R_j 及欲瀏覽的文件 D_k，系統首先檢查該使用者是否可以扮演其所宣稱的角色（依據角色結構圖來判斷）；若可，則進一步檢驗該使用者是否有權瀏覽該文件？若無立刻予以拒絕，否則表示接受並進一步檢查有那些子層文件可被該使用者瀏覽並傳回。整個的存取控制動作可以圖四的演算法來說明。

以上演算法的關鍵在於 Step3 和 Step4 。在 Step3 ，若角色 R_i 已在 $ARS(D_k)$中，瀏覽要求成立；但要注意的是 R_i 即使不在 $ARS(D_k)$中，其要求也不應立刻拒絕，因爲此時 R_i 和 $ARS(D_k)$中的各元素（例如 R_j ）的關係可能爲（ 1 ） R_i 是 R_j 的較普化角色（ 2 ） R_i,R_j 互相獨立，或（ 3 ） R_j 是 R_i 的較普化角色。若屬於前兩種情形，則拒絕該要求；否則表示 R_i 雖不在 $ARS(D_k)$內，但其條件已超過 $ARS(D_k)$之要求，故應于以接受（譬如總經理雖不屬於人事部門成員，但也應被允許去瀏覽公司人事資料）。

6. 結論

Step1: Set Reject=TRUE

Step2: Check if $U_i \in USER(R_j)$? If the answer is NO, then goto Step 5, else

Step3: Check if $R_j \in ARS(D_k)$? If the answer is YES, then call Accept(R_j,D_k) and goto Step 5, else

Step4: If there exists $R_j \in ARS(D_k)$ and R_j is the generalized role of R_i, then for each R_j, call Accept(R_j, D_k),

Step5: If the Reject=TRUE, then reject the brwosing request, else display c.D_k, except the components indicated in the Refuse-Component array.

```
{ Procedure Accept(Ri,Di)

      for each Dj ∈ c.Di, check if there exists Rj,
      where Rj is the specialized role of Ri , and
      Rj∈ ARS(c.Di)? If the answer is yes, then put
      Dj to the Refuse-component array.
      set Reject=FALSE and return the Refuse-
      component array
      }
```

圖四：安全存取控制演算法

安全是企業界在導入 Internet 應用時的首要考量。 Internet 上的 WWW 要轉換成商業化應用，有許多安全上的功能待加強，存取控制是其中核心的一項。傳統 WWW 以任意裁決存取控制策略實現的存取控制機制，不易對授權政策的設定做一致性的檢查，而且使用者加入或退出都會造成系統管理者很大的負擔。本文提出一個以角色分類爲基礎的存取控制方法，此方法不但保有任意裁決存取控制方法的彈性，且其將使用者分成不同角色，再依角色來決定使用者存取權的作法，不但可減少使用者加入或退出使用群及使用者授權異動所造成的管理負擔，並且在管理方面，透過角色名稱我們可以更容易檢查出授權政策的指定是否合理，這些都是加強商業化 WWW 安全性所亟需的功能。目前我們除了正在建立一雛形系統以驗證本文觀念的可行性與其效能外，並計劃和 SHTTP的身分辨識與認證等功能結合以構建更完整的 WWW 安全機制。

[參考文獻]

[1] L. Press, "Commercialization of the Internet", *Communications of ACM*, Nov. 1994, 37(11), pp.17-21.

[2] B.A. Nejmeh, "Internet: A Strategic Tool for the Software Enterprise", *Communications of ACM*, Nov.1994, 37(11), pp.23-27.

[3] K.H. Smith, Jr. "Accessing Multimedia Network Services", *IEEE Communications Magazine*, May.1992, pp.72-80.

[4] "The SSL Protocol", by Netscape Communications, Inc., *Documents available from the Internet*, 1995

[5] "The Enhanced Mosaic Security Framework", by Spyglass, Inc. *Documents available from the Internet*, 1994

[6] E. Rescorla and A. Schiffman, "The Secure Hyper Text Transfer Protocol", *Document available from the Internet*, June, 1994

[7] D.E.R. Denning, *Cryptography and Data Security*, Addison-Wesley Publishing Company, Inc. 1982

[8] R.S. SandHu and P Samarati, "Access Control:Principle and Practice", *IEEE Commmunications Magazine*, Sept. 1994, pp.40-48.

[9] E.B. Fernandez, E. Gudes and H. Song, "A Model for Evaluation and Adminstration of Security in Object-Oriented Databases", *IEEE Trans. on Knowledge and Data Engineering*, April 1994, 6(2), pp.275-291.

[10] P.B. Davis, D. Judhope, C. Taylor and C. Jones, " A Semantic Database Approach to Knowledge-Based Hypermedia Systems", *Information and Software Technology*, 1994, 36(6), pp.323-329