Anna FELKNER[1]

ROZPROSZONA KONTROLA DOSTĘPU W SYSTEMACH INFORMATYCZNYCH

Streszczenie

Artykuł dotyczy problemu bezpieczeństwa rozproszonych systemów informatycznych, w tym zwłaszcza ważnego problemu kontroli dostępu do zasobów i usług systemu. Jest wprowadzeniem w zagadnienia zarządzania zaufaniem w systemach informatycznych, rozumianym jako pewna forma zdecentralizowanej kontroli dostępu, gdzie decyzje związane z nadzorem wynikają z polityki, którą tworzy wiele podmiotów.

Abstract

The article presents the problem of distributed computer systems security. Special attention was focused on an important problem of the access control to resources and system services. It is an introduction to the issues of the trust management in IT systems, understood as a certain form of a decentralised access control in situations where decisions are made by many entities.

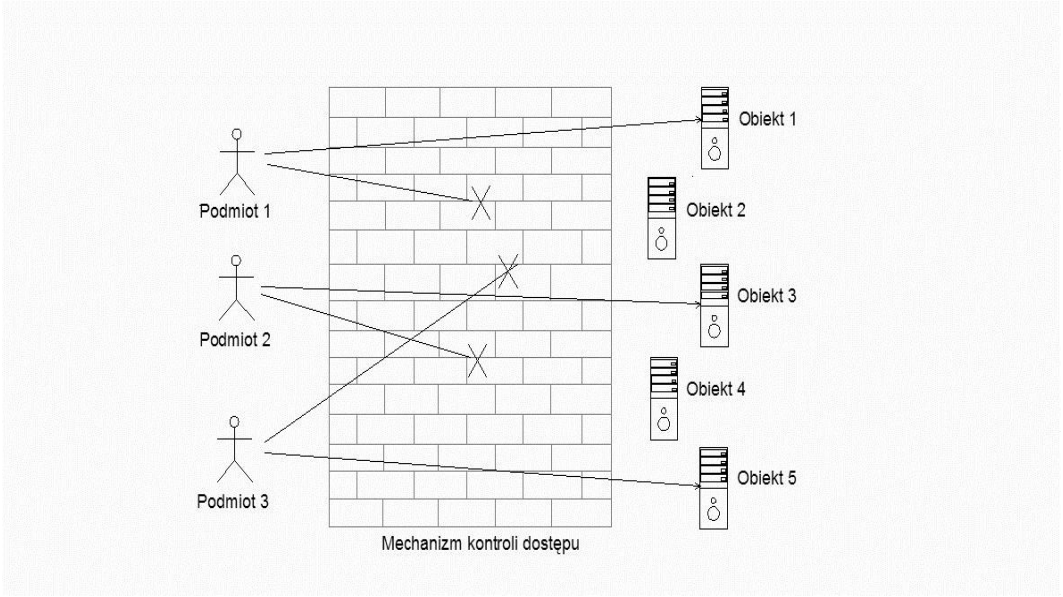
1 WPROWADZENIE

We współczesnym świecie niezbędne jest zapewnienie pełnej ochrony posiadanym zasobom, zarówno materialnym, jak i niematerialnym. W celu ochrony zasobów przed nieautoryzowanym dostępem należy opracować i zastosować odpowiednie polityki bezpieczeństwa, których istotną częścią jest kontrola dostępu. Niniejszy artykuł jest poświęcony kwestii bezpieczeństwa rozproszonych systemów informatycznych, w tym zwłaszcza ważnemu problemowi kontroli dostępu do zasobów i usług systemu.

Kontrola dostępu polega na logicznej lub fizycznej weryfikacji uprawnień (mechanizmie weryfikacyjnym) zaprojektowanej (zaplanowanej) w celu ochrony przed nie autoryzowanym wejściem do systemu lub przed jego użyciem, a także na zagwarantowaniu, że osoby uprawnione uzyskają dostęp. Zapewnia bezpieczeństwo zarówno danych, jak i systemów. Kontrola fizyczna wiąże się z zabezpieczeniem budynków, sejfów i serwerowni. Natomiast kontrola logiczna to między innymi zabezpieczenie komputerów, sieci komputerowych, wszelkiego rodzaju sprzętu komputerowego, danych, informacji, itd.

[1] Dr inż. Anna Felkner jest adiunktem w Naukowej i Akademickiej Sieci Komputerowej.

Kontrola dostępu obejmuje sprawowanie nadzoru nad tym, którzy uczestnicy (osoby, procesy, maszyny, itd.) i w jakim czasie mają dostęp do poszczególnych zasobów systemu komputerowego, na czym ten dostęp polega, w jaki sposób korzystają ze wspólnych danych, itp. Kontrola dostępu działa na kilku poziomach: aplikacji, warstwy pośredniej (ang. *middleware*), systemu operacyjnego i sprzętu. Jest to kluczowy składnik każdego rozwiązania związanego z bezpieczeństwem systemu, którego zadaniem jest zapewnienie, że zasób jest używany przez odpowiednich odbiorców w uprawniony sposób, w odpowiednim miejscu i czasie. Poniżej (rysunek 1) zamieszczono poglądowy rysunek przedstawiający działanie mechanizmu kontroli dostępu.



Rys. 1. Działanie mechanizmu kontroli dostępu

2 TRADYCYJNE MODELE

Tradycyjne modele kontroli dostępu dzielimy na trzy podstawowe grupy: uznaniowa (*Discretionary Access Control*, DAC), obowiązkowa (*Mandatory Access Control,* MAC) oraz kontrola dostępu oparta na rolach (*Role-Based Access Control,* RBAC).

2.1 Uznaniowa kontrola dostępu

W modelu *uznaniowej kontroli dostępu* (DAC) [7] środki ograniczania dostępu do obiektów oparte są na pojęciu własności obiektu. Podmiot posiadający prawo własności do obiektu decyduje o przyznawaniu i odbieraniu praw dostępu innym podmiotom oraz

przekazywaniu im uprawnień. W większości systemów DAC prawo własności przysługuje twórcy obiektu.

Wadą tego modelu jest to, że nie daje on pewności, co do spełnienia wymagań ochrony. Użytkownik, który jest uprawniony do podjęcia działań na konkretnym obiekcie, może to uprawnienie przekazać innemu użytkownikowi, który nie jest do tego uprawniony. Wszelkie tego rodzaju działania mogą odbywać się bez wiedzy właściciela obiektu, na którym te operacje są wykonywane.

2.2 Obowiązkowa kontrola dostępu

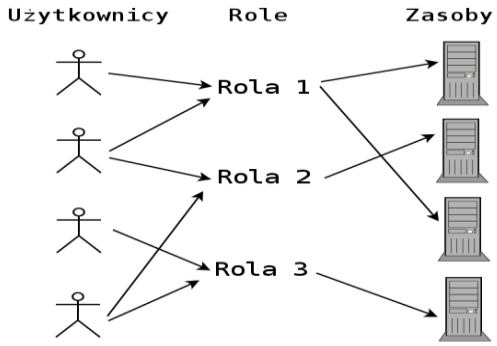
W systemach, w których konieczna jest stała kontrola nad rozprzestrzenianiem się informacji, często stosowana jest polityka *obowiązkowej kontroli dostępu* (MAC) [9]. Ideą leżącą u podstaw tego typu modeli jest założenie, że informacja może przepływać tylko w jednym kierunku, tzn. od obiektów o niższej klasyfikacji tajności do obiektów o wyższym stopniu tajności. Decyzje odnośnie uprawnień obiektu są ustalane przez administratora systemu, a użytkownik nie ma możliwości modyfikacji swoich uprawnień. W systemach wykorzystujących model MAC, wszystkie podmioty i obiekty muszą mieć przypisane etykiety wrażliwości. Etykieta wrażliwości podmiotu ustala jego poziom zaufania, a etykieta wrażliwości obiektu określa poziom zaufania, który jest wymagany, aby móc otrzymać do niego dostęp. Gdy podmiot chce otrzymać dostęp do obiektu, jego etykieta wrażliwości musi być równa lub wyższa niż etykieta wrażliwości obiektu, do którego chciałby otrzymać dostęp. Najważniejszą własnością tego modelu, odróżniającą go od modelu uznaniowej kontroli dostępu jest to, że użytkownik nie może w pełni kontrolować zasobów, które sam stworzył, nie ma też wpływu na działanie mechanizmów kontroli dostępu.

2.3 Kontrola dostępu oparta na rolach

W *kontroli dostępu opartej na rolach* (RBAC) [4, 8] uprawnienia dostępu do zasobów zamiast podmiotom przypisane są rolom, które dane podmioty pełnią. Podmiot może pełnić różne role, ale w zależności od tego, jaką rolę pełni aktualnie, posiada tylko takie uprawnienia, które są mu niezbędne do jej pełnienia. Są one przydzielane mu ze względu na jego przynależność do roli, a nie tożsamość. Upraszcza to zarządzanie uprawnieniami, gdyż rozdziela etap przydzielania praw potrzebnych do wykonywania określonych zadań, od określenia, kto ma być wykonawcą tych zadań. Dlatego też model ten jest często stosowany w dużych scentralizowanych systemach z wieloma użytkownikami.

Główną przyczyną powstania modelu RBAC jest fakt, że najczęściej uprawnienia, jakie są przydzielane poszczególnym pracownikom w organizacji są ściśle związane z funkcją (czyli rolą), jaką pełnią. Rzadko są one nadawane konkretnej osobie. Dostęp użytkowników do zasobów jest regulowany poprzez ich przynależność do konkretnej roli. Uprawnienia, które są przydzielane danej osobie, zmieniają się znacznie częściej (np. zmiana stanowiska pracy) niż uprawnienia, które są przydzielone danej roli. Zatem prawa dostępu do zasobów powinny raczej zależeć od funkcji, jaką pełni pracownik niż od jego tożsamości. Związanie uprawnień z rolami i przyporządkowanie poszczególnych użytkowników odpowiednim rolom upraszcza częste operacje w przedsiębiorstwie, takie jak dodawanie użytkownika albo zmiana działu przez tego użytkownika. RBAC jest mechanizmem bezpieczeństwa, który może niezwykle obniżyć koszty i złożoność administracji bezpieczeństwa w dużych systemach. Dzięki połączeniu uprawnień z użytkownikami za pomocą pojęcia roli wzrasta elastyczność oraz ekspresyjność tego modelu.

Model kontroli dostępu opartej na rolach jest najbardziej elastycznym typem polityki kontroli dostępu, często spotykanym w systemach informatycznych przedsiębiorstw. W tym modelu można symulować zarówno uznaniową jak i obowiązkową kontrolę dostępu, być wykonawcą tych zadań. Poglądowe przedstawienie modelu RBAC jest pokazane na rysunku 2.



Rys. 2. Model kontroli dostępu opartej na rolach

3 ROZPROSZONA KONTROLA DOSTĘPU

Tradycyjne modele kontroli dostępu, takie jak DAC, MAC i RBAC, nadają się do kontrolowania dostępu do zasobów dla ustalonej znanej grupy użytkowników. Modele te podejmują decyzje związane z kontrolą dostępu na podstawie tożsamości wnioskującego i często nie nadają się do zastosowania w środowiskach rozproszonych, gdzie jest wiele podmiotów uwierzytelniających, jak również wystawiających certyfikaty, oraz gdzie

użytkownicy zmieniają się dynamicznie, a ich tożsamość nie jest a priori znana. Właściciel zasobu i podmiot zgłaszający żądanie dostępu nie znają siebie nawzajem. Problemem jest również brak jednego źródła, do którego można się odwołać w celu uzyskania wszystkich dokumentów zawierających aktualne uprawnienia. Wychodząc naprzeciw tym ograniczeniom, zostały zaproponowane modele rozproszonego zarządzania zaufaniem (m.in. PolicyMaker [2, 3], KeyNote [1] i RT [5, 6]).

3.1 Zarządzanie zaufaniem

Pojęcie zarządzania zaufaniem zostało wprowadzone po raz pierwszy w 1996 roku przez członków AT&T Labs – Research (Blaze, i Lacy) i zostało zdefiniowane jako uporządkowane podejście do zarządzania politykami bezpieczeństwa, poświadczeniami oraz relacjami związanymi z zaufaniem. W systemach tych decyzje związane z kontrolą dostępu bazują na atrybutach pytającego, a nie na jego tożsamości. Atrybuty są przedstawione w postaci poświadczeń. Poświadczenie jest dokumentem przekazywanym od jednego podmiotu do innego, używanym w celu ustalenia praw dostępu podmiotu. Za ich pomocą przydzielane są uprawnienia. Poświadczenia są wystawiane i podpisane przez podmioty do tego uprawnione, zawierające dane niezbędne do uzyskania uprawnień do wykonywania czynności, których nie moglibyśmy wykonywać nie posiadając takiego dokumentu. Każde poświadczenie w systemach zarządzania zaufaniem deleguje konkretne pozwolenie od wystawcy do danego podmiotu. Poświadczeniem może być prawo jazdy, dowód członkostwa w konkretnej grupie, karta biblioteczna czy też karta kredytowa. To, czy dany podmiot może wykonać daną operację, czy ma do czegoś prawo, zależy od przyznanych mu poświadczeń, które może on otrzymywać od wielu podmiotów. Może je też przekazywać innym podmiotom. Przekazanie uprawnień do jednego czy większej ilości zasobów innemu użytkownikowi, nazywamy *delegacją*. Delegacja jest ważnym mechanizmem usprawniającym skalowalność i elastyczność systemów zarządzania zaufaniem. Na przykład kierownik, który wyjeżdża na urlop, przekazuje część swoich uprawnień swemu zastępcy. Relacje związane z zaufaniem pomiędzy dwoma podmiotami opierają się tylko na weryfikacji poświadczenia i są używane do kontroli dostępu do usług i zasobów.

3.2 Przykład zarządzania zaufaniem

Przypuśćmy, że jeden z aspektów polityki księgarni internetowej mówi, że każdy, kto jest studentem Wydziału Politechniki oraz posiada kartę stałego klienta wydawnictwa, z którym księgarnia współpracuje, może dostać rabat. W tym celu należy zapewnić:

- możliwość oświadczenia, że każdy, kto jest studentem Wydziału Politechniki oraz posiada kartę stałego klienta wydawnictwa, z którym księgarnia współpracuje, może dostać rabat (**polityka**),
- możliwość udowodnienia przez studenta, że faktycznie jest studentem Wydziału i że posiada kartę stałego klienta wydawnictwa oraz możliwość udowodnienia przez Wydział, że jest integralną częścią Politechniki (**poświadczenia**),
- możliwość określenia, kto może wystawić takie poświadczenia (**relacje związane z zaufaniem**).

3.3 Rodzina języków Role-based Trust management

Rodzina języków Role-based Trust management (RT) łączy siłę kontroli dostępu opartej na roli i systemów zarządzania zaufaniem. Jest to rodzina języków różniących się właściwościami. Najbardziej podstawowa część RT to RT_0. Jest to język bazowy, który pozwala na wyrażenie dosyć szerokiego zakresu polityk bezpieczeństwa.

Pierwszym rozszerzeniem tego języka jest RT1, który dodaje do języka RT0 sparametryzowane role, które mogą wyrażać pola atrybutu. RT2 zwiększa RT1 o obiekty logiczne, które mogą grupować logicznie powiązane obiekty, tak że ich pozwolenia mogą być przydzielane razem. Natomiast w RTT zostało wprowadzone pojęcie ról wielorakich, za pomocą których można wyrazić strukturę progu i politykę podziału obowiązków. Struktury progu wymagają porozumienia wśród kilku jednostek ze zbioru. Polityka podziału obowiązku wymaga, aby za wykonanie zadania odpowiedzialnych było dwóch lub więcej użytkowników. Umożliwia to nałożenie warunku na wykonanie zadania przez więcej niż jedną rolę, co w znacznym stopniu poszerza zakres możliwości modelu. Ostatnim językiem jest RT^D, który dostarcza delegacji aktywacji roli, które wyrażają wybiórcze użycie potencjału możliwości i delegacji tego potencjału. Oznacza to, że za jego pomocą można przekazać część uprawnień w celu wykorzystania przez inny podmiot.

Największe znaczenie praktyczne ma język RT^T, który jako jedyny jest w stanie wyrazić politykę, w której warunkiem wykonania zadania jest to, aby za to zadanie odpowiedzialne były co najmniej dwa podmioty, przypisane do dwóch i więcej ról. Można w nim na przykład wyrazić politykę banku, w której warunkiem udzielenia wysokiego kredytu jest zatwierdzenie go przez określone osoby. Co więcej mogą to być osoby pełniące kilka różnych ról. Można też wymagać, żeby nie była to jedna osoba pełniąca kilka ról jednocześnie, czyli na przykład dwóch różnych kasjerów i kierownik banku, który może, ale nie musi być kasjerem.

Podsumowanie najważniejszych własności i możliwości języków z rodziny role-based trust management jest zebrane w tabeli 1. Dokładny opis całej rodziny języków RT można znaleźć w [6].

Tabela 1. Właściwości poszczególnych języków z rodziny RT

Język RT	Właściwości
RT_0	• lokalne upoważnienia nad rolami, • hierarchie ról, • przekazywanie uprawnień między rolami, • przekazywanie uprawnień oparte na atrybutach, • przecięcia ról.
RT_1	właściwości RT_0 plus: • sparametryzowane role.
RT_2	właściwości RT_1 plus: • obiekty logiczne.
RT^T	właściwości RT_0 plus: • wielorakie role, • struktura progu, • polityka podziału obowiązków.
RT^D	właściwości RT_0 plus: • delegacja aktywacji roli, • dynamiczna delegacja poświadczeń.

3.3.1 Składnia języków z rodziny RT

Podstawowe elementy języków RT to podmioty, nazwy ról, role i poświadczenia. *Podmiot* w RT jest jednoznacznie identyfikowalnym bytem (np. użytkownik lub proces), który może wystawiać role, poświadczenia i przesyłać żądanie dostępu do zasobów. *Nazwa roli* jest jej identyfikatorem. *Rola* jest zbiorem podmiotów, którym uprawnienia przydzielane są wspólnie, co oznacza, że przydzielenie konkretnego uprawnienia danej roli skutkuje otrzymaniem tego uprawnienia przez każdego z członków tej roli. *Poświadczenia* definiują role poprzez wskazanie nowych członków ról lub poprzez delegację uprawnień do członków innych ról. *Nazwa podmiotu* rozpoczyna się (lub po prostu jest) wielką literą, a n*azwa roli* małą literą. *Rola* oznaczana jest przez *nazwę podmiotu* i *nazwę roli* oddzielonych kropką. *Poświadczenia* przyjmują jedną z sześciu poniższych postaci:

$A.r \leftarrow B$	podmiot B należy do roli $A.r$,
$A.r \leftarrow B.s$	wszyscy członkowie roli $B.s$ należą również do roli $A.r$,
$A.r \leftarrow B.s.t$	wszyscy członkowie roli $C.t$ również należą do roli $A.r$, dla każdego C należącego do roli $B.s$,
$A.r \leftarrow B.s \cap C.t$	tylko członkowie obu ról: $B.s$ i $C.t$ jednocześnie należą do roli $A.r$,
$A.r \leftarrow B.s \bullet C.t$	do roli $A.r$ należy jeden członek roli $B.s$ i jeden członek roli $C.t$ jednocześnie,
$A.r \leftarrow B.s \otimes C.t$	jak w poprzednim przypadku, przy czym są to różniący się członkowie ról.

3.3.2 Przykłady użycia języków z rodziny RT

Przykład 1. *Rozpatrzmy przykład, w którym polityka uczelni mówi, że dana osoba może uczestniczyć w przedmiocie, gdy jest studentem wydziału na uczelni U, na której przedmiot jest prowadzony. Aby móc pełnić rolę wydziału należy być jednostką organizacyjną oraz prowadzić działalność naukową i dydaktyczną. Jaś jest studentem na wydziale WE, który jest jednostką organizacyjną oraz prowadzi działalność naukową i dydaktyczną. Tak więc Jaś ma prawo uczestniczyć w przedmiocie, co pokazują następujące poświadczenia:*

U.przedmiot ← U.wydział.student
U.wydział ← U.jednostkaOrganizacyjna ∩ U.prowadziDziałalność
U.jednostkaOrganizacyjna ← {WE}
U.prowadziDziałalność ← {WE}
WE.student ← {Jaś}

Przykład 2. *Przypuśćmy teraz, że w powyższym przykładzie postawiono również warunek, że aby dany przedmiot mógł być uruchomiony w danym semestrze, musi na niego być zapisane przynajmniej 2 z 3 osób będących studentami danego roku. Gdybyśmy chcieli to wyrazić za pomocą języka RT_0 musielibyśmy wypisać wszystkich studentów na tym roku (w tym przypadku 3) i za pomocą przecięć wybrać z nich dwóch. Natomiast*

w języku RT^T wystarczy do tego tylko jedno poświadczenie. Dalej przypuśćmy, iż chcielibyśmy wprowadzić ponadto wymaganie na zapisanie się na dany przedmiot dwóch studentów, ale też doktoranta, który nie może być wśród tych dwóch pozostałych wybranych studentów. Musi to być inna osoba. Jest to niemożliwe do wykonania za pomocą żadnego z poświadczeń dostępnych w poprzednich językach z rodziny RT. Jest to natomiast możliwe do wykonania przy pomocy operatorów wprowadzonych w RT^T. Zbiór poświadczeń dla tego przykładu wyglądałby wtedy w sposób następujący:

U.przedmiot ← U.wydział.student

U.wydział ← U.jednostkaOrganizacyjna ∩ U.prowadziDziałalność

U.jednostkaOrganizacyjna ← {WE}

U.prowadziDziałalność ← {WE}

WE.studenci ← WE.student • WE.student

W E.studenciDoktoranci ← WE.studenci ⊗ WE.doktorant

Przyjmując poszczególne role:

WE.student ← {Jaś}

WE.student ← {Staś}

WE.student ← {Zosia}

WE.doktorant ← {Jaś}

WE.doktorant ← {Kasia}

możemy wyznaczyć wszystkie minimalne zbiory osób, których zapisanie się na przedmiot spowoduje jego uruchomienie. Są to na przykład: {Jaś, Staś, Kasia}, {Jaś, Zosia, Kasia} lub {Staś, Zosia, Kasia}.
Dodając kolejną regułę:

WE.studenci ← WE.studenci • WE.student

jesteśmy w stanie wyznaczyć wszystkie - nie tylko minimalne - zbiory potrzebne do uruchomienia wykładu. Dodatkowy zbiór, który byśmy otrzymali dzięki tej regule to: {Jaś, Staś, Zosia, Kasia}.

W przypadku, gdybyśmy chcieli nałożyć warunek na uruchomienie tego przedmiotu w danym semestrze poprzez zapisanie się na niego przynajmniej 20 z 30 osób będących studentami danego roku, zapis w RT_0 stałby się bardzo skomplikowany, gdyż znów trzeba by było wypisać wszystkich studentów (w tym przypadku 30) i wybrać z nich wszystkie

możliwe kombinacje, co wymaga sformułowania dużej ilości poświadczeń. Za każdym razem, gdy zmienialiby się studenci, należałoby dokonywać zmian w polityce uczelni. Natomiast dzięki poświadczeniu, jakie nam umożliwia operator wprowadzony w RT^T, jedyną zmianą byłoby rozszerzenie powyższego poświadczenia z dwóch studentów na dwudziestu. Niewątpliwie jest to duży zysk.

Powyższy przykład pokazuje, jak duże usprawnienie zostało osiągnięte dzięki poświadczeniom, które oferuje nam język RT^T. Pokazuje nam, że potrzeba tworzenia bardziej złożonych i skomplikowanych polityk wymusiła wprowadzenie dodatkowych możliwości w tym języku.

4 PODSUMOWANIE

Systemy zarządzania zaufaniem, z powodu możliwości delegowania uprawnień, zapewniają znaczną skalowalność administracji bezpieczeństwa w dużych systemach informatycznych. Umożliwia to zastosowanie ich w otwartych i rozproszonych środowiskach, gdzie mamy do czynienia z dużą liczbą często zmieniających się użytkowników uzyskujących dostęp do zasobów, które powinny być zabezpieczone. Z wyżej wymienionych powodów autorka artykułu pracuje nad rozszerzeniem możliwości języków z rodziny RT.

Literatura

1. M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis. *The KeyNote Trust-Management System*, Version 2. RFC 2704, 1999.
2. M. Blaze, J. Feigenbaum, and J. Lacy: *Decentralized Trust Management*. In Proc. 17th Symposium on Security and Privacy, 1996.
3. M. Blaze, J. Feigenbaum, and M. Strauss: *Compliance Checking in the PolicyMaker Trust-Management System*. In Proc. of the Financial Cryptography, 1998.
4. D. Ferraiolo i in.: *Proposed NIST Role-Based Access control, ACM Transactions on Information and Systems Security*, 2001.
5. N. Li and J. C. Mitchell. RT: *A Role-based Trust-management Framework*. In Proceedings of the 3rd DARPA Information Survivability Conference and Exposition, 2003.
6. N. Li, J. C. Mitchell, and W. H. Winsborough: *Design of a Role-Based Trust Management Framework*. In Proceedings of the 2002 IEEE Symposium on Security and Privacy, 2002.

7. National Computer Security Center. *A guide to understanding discretionary access control in trusted systems*, NCSC-TG-003. 1987.
8. R. S. Sandhu i in.: *Role-Based Access Control Models*; IEEE Computer, 1996.
9. X. Qian, T. F. Lunt. *A MAC Policy Framework for Multilevel Relational Databases*. IEEE Transactions Knowledge and Data Engineering, vol. 8(1), 1996.