Anna FELKNER[1], Adam KOZAKIEWICZ[1]

Naukowa i Akademicka Sieć Komputerowa

doi:10.15199/48.2016.02.46

Rozproszona kontrola dostępu w praktyce

Streszczenie. *Tematem pracy jest rodzina języków zarządzania zaufaniem (Role-based Trust management), która służy do reprezentacji polityk bezpieczeństwa i poświadczeń w zdecentralizowanych, rozproszonych systemach kontroli dostępu do zasobów i usług. Artykuł pokazuje, jak wprowadzone rozszerzenia (ograniczenia czasowe poświadczeń, poświadczenia warunkowe i ustalanie porządku) mogą wpłynąć na zwiększenie stosowalności takiego rozwiązania w politykach bezpieczeństwa.*

Abstract. *The topic of this paper is the family of Role-based Trust management languages (RT) which is used for representing security policies and credentials in decentralized, distributed, large scale access control systems. The goal of this paper is to explore the potential of RT languages. It shows how security policies can be made more realistic by including timing information, maintaining the procedure or parameterizing the validity of credentials.* ***(Distributed access control in practice)***

Słowa kluczowe: polityka bezpieczeństwa, kontrola dostępu, zarządzanie zaufaniem, poświadczenia
Keywords: security policy, access control, trust management, credentials

Wstęp

Zasoby poufne, zarówno materialne jak i niematerialne, oferowane przez systemy komputerowe w sektorach rządowych, wojskowych, jak również handlowych i przemysłowych, potrzebują szczególnej ochrony. Co za tym idzie powinny być one udostępniane tylko autoryzowanym użytkownikom. 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, a w szczególności problemowi kontroli dostępu do zasobów i usług systemu. Kontrola dostępu polega na logicznej lub fizycznej weryfikacji uprawnień zaprojektowanej w celu ochrony przed nieautoryzowanym wejściem do systemu lub przed jego użyciem, jak również na zagwarantowaniu, że osoby uprawnione uzyskają należny dostęp. Sprawuje nadzór nad tym, którzy uczestnicy w jakim czasie mają dostęp do poszczególnych zasobów, na czym ten dostęp polega, w jaki sposób korzystają ze wspólnych danych, itp. Jest to niezwykle istotny składnik każdego rozwiązania związanego z bezpieczeństwem systemu, którego zadaniem jest zapewnienie, że zasób jest używany przez odpowiednich użytkowników w uprawniony sposób, w odpowiednim czasie i miejscu. Kontrola taka dotyczy wszelkiego rodzaju sieci i systemów, w których należy chronić dostępu (komputerowych, przemysłowych, elektrycznych, elektroenergetycznych i innych).

Rozproszona kontrola dostępu

Tradycyjnym podejściem do kwestii zapewnienia kontroli dostępu do zasobów i usług są modele obowiązkowej kontroli dostępu (MAC), uznaniowej kontroli dostępu (DAC) i kontrola dostępu oparta na rolach (RBAC), w których użytkownicy są identyfikowani, a dostęp jest im udzielany lub zabraniany, na podstawie ich tożsamości. Oznacza to, że tożsamość wnioskodawcy musi być znana właścicielowi zasobów, czyli dostęp do zasobów może być udzielany tylko w gronie znających się nawzajem użytkowników. Podejście to dobrze się sprawdza w zamkniętych, scentralizowanych systemach, w których tożsamość użytkowników jest a priori znana, natomiast nie nadaje się do zastosowania w otwartych, zdecentralizowanych systemach, gdzie wnioskodawca i dostawca usługi nie znają siebie nawzajem, mogą często się zmieniać w czasie, jak również jest wiele podmiotów uwierzytelniających i wystawiających poświadczenia upoważniające.

Na przykład, rozważmy politykę księgarni, w której studenci, którzy często robią w niej zakupy, otrzymują rabat. Jednakże, gdy dana osoba wchodzi do księgarni i mówi, że nazywa się Jan Kowalski, to sama jego tożsamość nie upoważnia go do rabatu. W takiej sytuacji potrzebujemy informacji, czy jest on studentem i czy często robi zakupy w danej księgarni (czyli czy posiada kartę stałego klienta wydaną przez księgarnię). W tym celu potrzebujemy dwóch dokumentów: legitymacji studenckiej i karty stałego klienta księgarni. Jak widzimy, w tej sytuacji do podjęcia decyzji potrzebujemy dokumentów stwierdzających posiadane przez daną osobę przywileje, a nie mówiących o jego tożsamości. Oznacza to, że potrzebujemy nowego podejścia do kontroli dostępu. Otwartość systemu wiąże się z jego zdolnością do rozszerzania różnymi sposobami, a to z kolei wiąże się z ciągłymi zmianami. Problemem jest wtedy też brak jednego źródła, do którego można się odwołać w celu uzyskania wszystkich dokumentów zawierających aktualne uprawnienia.

Wychodząc naprzeciw tym ograniczeniom, zostały zaproponowane modele zarządzania zaufaniem, czyli uporządkowane podejście do specyfikowania i interpretowania polityk bezpieczeństwa, poświadczeń oraz relacji związanych z zaufaniem. Jest to podejście do rozproszonej kontroli dostępu, w których decyzje bazują na orzeczeniach wydawanych przez wiele podmiotów. Poświadczenie jest dokumentem, wystawionym i podpisanym przez uprawniony podmiot, zawierającym dane niezbędne do uzyskania uprawnień przez inny podmiot (może to być na przykład prawo jazdy, legitymacja ubezpieczeniowa, nazwa użytkownika wraz z hasłem, certyfikat lub też legitymacja, czy karta stałego klienta, jak w powyższym przykładzie). Prawo podmiotu do wykonania danej operacji zależy od przyznanych mu poświadczeń, które może on przekazywać innym podmiotom, co jest nazywane delegacją. Polityka bezpieczeństwa określa, jak konkretne upoważnienia wynikają z posiadanych poświadczeń, a relacje związane z zaufaniem definiują kto ma prawo wystawiać określone poświadczenia.

Przykład zarządzania zaufaniem

Załóżmy, że jeden z aspektów polityki księgarni mówi, że każdy kto jest studentem oraz posiada kartę stałego klienta księgarni, może dostać rabat. W tym celu należy zapewnić:

- możliwość oświadczenia, że każdy kto jest studentem oraz posiada kartę stałego klienta księgarni, może dostać rabat (*polityka*),
- możliwość udowodnienia przez studenta, że faktycznie jest studentem oraz że posiada kartę stałego klienta księgarni (*poświadczenie*),
- możliwość określenia, kto może wystawić takie poświadczenie (*relacje związane z zaufaniem*).

Zarządzanie zaufaniem [1], [2] jest istotnym składnikiem bezpieczeństwa w systemach rozproszonych. Jest filarem

decentralizacji decyzji związanych z bezpieczeństwem, pomaga w znalezieniu odpowiedzi na pytanie „dlaczego" zaufanie jest udzielone. Do zdefiniowania systemu zarządzania zaufaniem jest potrzebny język opisu podmiotów, poświadczeń i ról, które te podmioty pełnią w systemie. W tym celu w [3] została zdefiniowana rodzina języków Role-based Trust management (RT), która łączy zalety kontroli dostępu opartej na rolach i modeli zarządzania zaufaniem bazujących na poświadczeniach.

Rodzina języków Role-based Trust management

Role-based Trust management jest rodziną języków o różnym poziomie złożoności i ekspresyjności. W jej skład początkowo wchodziły języki RT_0, RT_1, RT_2, RT^T i RT^D. Najbardziej podstawową częścią RT jest RT_0, który został przedstawiony w [4]. Pozwala on na wyrażenie dosyć szerokiego zakresu polityk bezpieczeństwa. Dokonywane jest to za pomocą zbioru możliwych do wyrażenia za pomocą RT_0 poświadczeń. Pierwszym rozszerzeniem jest RT_1, który dodaje do RT_0 sparametryzowane role, które mogą wyrażać pola atrybutów. Na przykład dla studenta jego atrybutem może być rok studiów. RT_2 dalej rozszerza RT_1 wprowadzając pojęcie obiektów logicznych, które mogą grupować ze sobą logicznie powiązane obiekty w taki sposób, że pozwolenia na korzystanie z nich mogą być przydzielane razem. Obiektami takimi mogą być na przykład zasoby lub sposób dostępu do nich. Najbardziej interesującym językiem z rodziny RT, a zarazem głównym tematem tej pracy, jest RT^T, w którym zostało wprowadzone pojecie ról wielorakich, które rozszerzają pojęcie roli poprzez pozwolenie na to, aby członkowie roli byli zbiorem podmiotów, a nie pojedynczym podmiotem. Role wielorakie umożliwiają zamodelowanie progowania i polityki podziału obowiązków. Struktury progu wymagają porozumienia kilku podmiotów ze zbioru, żeby potwierdzić jakiś fakt. Polityka podziału obowiązków wymaga, aby za wykonanie zadania odpowiedzialnych było dwóch lub więcej użytkowników pochodzących z różnych zbiorów podmiotów. Na przykład załóżmy, iż polityka banku mówi, że umowę kredytową musi podpisać co najmniej dwóch kasjerów. W tym przypadku rolą wieloraką będzie zbiór tych kasjerów. Żaden pojedynczy kasjer nie może pełnić tej roli samodzielnie. RT^T jako jedyny jest w stanie wyrazić politykę, w której warunkiem wykonania zadania jest uczestnictwo dwóch lub więcej użytkowników przypisanych do dwóch i więcej ról. Można też w nim narzucić, aby ta sama osoba nie mogła wystąpić w dwóch rolach jednocześnie. Ostatnim językiem z rodziny RT jest RT^D, który dostarcza mechanizm do opisu delegacji (praw) aktywacji roli, za pomocą którego można wyrazić wybiórcze użycie potencjału możliwości tej roli i delegację tego potencjału.

Języki z rodziny Role-based Trust management rozwijają się w celu umożliwienia im zastosowania w jak najszerszym zakresie systemów, aby za ich pomocą można było najwierniej odzwierciedlić rzeczywistość.

Składnia języka RT^T

Podstawowymi elementami języka są podmioty, nazwy ról, role i poświadczenia. *Nazwa podmiotu* rozpoczyna się (lub po prostu jest) wielką literą. *Podmiot* w RT, nazywany też jednostką, jest pojedynczym „aktorem", jednoznacznie identyfikowalnym bytem (np. użytkownik), który może wystawiać poświadczenia i przesyłać prośbę (żądanie) dostępu do zasobów. Podmiot może też wystawiać role i definiować członków swoich ról. Może ich tworzyć dowolną liczbę. Przykładem nazwy podmiotu jest *Wydział*. *Nazwa roli* rozpoczyna się (lub po prostu jest) małą literą. Nazwa danej roli jest jej identyfikatorem. Przykładem nazwy roli jest *student*. *Rola* oznaczana jest przez nazwę podmiotu i nazwę roli oddzielonych kropką. Zapis $A.r$ oznacza rolę r wystawioną przez podmiot A. Rola jest zbiorem podmiotów, którym uprawnienia przydzielane są wspólnie. To znaczy, że przydzielenie konkretnego uprawnienia danej roli skutkuje otrzymaniem tego uprawnienia przez każdego z członków tej roli. Na przykład rola: *Wydział.student* może określać zbiór wszystkich studentów Wydziału. *Poświadczenia* definiują role poprzez wskazanie nowych członków ról lub poprzez delegację uprawnień do członków innych ról. Poświadczenia RT^T przyjmują jedną z sześciu poniższych postaci:

- $A.r \leftarrow B$ – *proste członkostwo*: podmiot B należy do roli $A.r$.
- $A.r \leftarrow B.s$ – *zawieranie proste*: rola $A.r$ zawiera wszystkich członków roli $B.s$. Jest to rodzaj delegacji uprawnień nad r z A do B.
- $A.r \leftarrow B.s.t$ – *zawieranie łączone*: wszyscy członkowie roli $C.t$ należą do roli $A.r$, dla każdego C należącego do roli $B.s$. Jest to rodzaj delegacji uprawnień z A do wszystkich członków roli $B.s$.
- $A.r \leftarrow B.s \cap C.t$ – *zawieranie części wspólnej*: tylko członkowie obu ról: $B.s$ i $C.t$ jednocześnie należą do roli $A.r$. Jest to częściowa delegacja z A do B i C.
- $A.r \leftarrow B.s \odot C.t$ – do roli $A.r$ należy jeden członek roli $B.s$ i jeden członek roli $C.t$ jednocześnie. Najmniejszym możliwym zbiorem jest tu zbiór składający się z jednej jednostki, która jest członkiem przecięcia roli $B.s \cap C.t$.
- $A.r \leftarrow B.s \otimes C.t$ – do roli $A.r$ należy jeden członek roli $B.s$ i jeden członek roli $C.t$ jednocześnie, przy czym są to różniący się członkowie ról.

RT^T jest najobszerniejszym językiem z rodziny RT, obejmującym wszystkie reguły składniowe całej rodziny.

Języki z rodziny Role-based Trust management mają zastosowanie w bardzo złożonych systemach, jednakże w pracy przedstawione są bardzo proste przykłady, w celu zrozumienia idei. Przykłady, mimo prostoty, są bardzo realistyczne i pokazują ekspresyjność oraz użyteczność opisanego tu rozwiązania.

Przykład 1

Rozpatrzmy przykład, w którym polityka uczelni mówi, że potrzebujemy co najmniej dwóch z czterech studentów oraz doktoranta, który również może (ale nie musi) być studentem, aby uruchomić przedmiot. Polityka uczelni może być przedstawiona za pomocą poniższych poświadczeń:

(1) $$U.studenci \leftarrow U.student \otimes U.student$$

(2) $$U.przedmiot \leftarrow U.doktorant \odot U.studenci$$

Przyjmując poszczególne role:

(3) $$U.student \leftarrow \{Ala\}$$

(4) $$U.student \leftarrow \{Ola\}$$

(5) $$U.student \leftarrow \{Ela\}$$

(6) $$U.student \leftarrow \{Ula\}$$

(7) $$U.doktorant \leftarrow \{Ola\}$$

(8) $$U.doktorant \leftarrow \{Jan\}$$

możemy wyznaczyć, zgodnie z polityką uczelni, że każda para ze zbioru $\{Ala, Ola, Ela, Ula\}$ spełnia wymóg bycia parą dwóch różnych studentów, czyli na przykład $\{Ola, Ula\}$. Natomiast, aby uruchomić przedmiot, w podanej parze musi zawierać się Ola lub też do podanej pary musimy dodatkowo dodać $Jana$, czyli na przykład $\{Ola, Ula\}$ lub $\{Ala, Ela, Jan\}$.

Semantyka języków RT

Definicja języka obejmuje jego składnię i semantykę. Semantyka teoriomnogościowa, która mapuje role w zbiory podmiotów pełniących te role dla języka RT^T przedstawiona została w [5]. W praktyce większe zastosowanie ma semantyka operacyjna przedstawiona za pomocą systemu wnioskowania ([6]), czyli nowe poświadczenia są uzyskiwane ze zbioru posiadanych poświadczeń za pomocą reguł wnioskowania. Dzięki temu wyznaczamy tylko te zbiory podmiotów, których w danym momencie potrzebujemy, a nie wszystkie podmioty w poświadczeniach, jak w przypadku poprzednich semantyk, co zwiększa wydajność, jak również użyteczność takiej semantyki.

Poniżej zostaną przedstawione trzy różne rozszerzenia wprowadzone do języka RT^T, które pozwolą na łatwe zastosowanie tego języka w rzeczywistych systemach. Poniższe rozdziały pokażą jak niewielkie zmiany wprowadzone do języków RT mogą mocno rozszerzyć ich zastosowanie. Pokażą jak wprowadzenie ograniczeń czasowych, ustalonego porządku i sparametryzowanie ważności poświadczeń, mogą uczynić polityki bezpieczeństwa łatwe do zastosowania w praktyce.

Poświadczenia ograniczone czasowo

W artykule [7] zostało zaprezentowane rozszerzenie języka RT_0 o ograniczenia czasu dostępności poświadczeń (poświadczenia są ważne tylko w określonym przedziale czasowym). Takie ograniczenie ważności ma często miejsce w rzeczywistości. W artykule [8] można znaleźć podobne rozszerzenie wprowadzone do języka RT^T.

Poświadczenia ograniczone czasowo można zapisać w formie: c **in** v, co oznacza „poświadczenie c jest dostępne w czasie v". Skończony zbiór dostępnych poświadczeń uzależnionych od czasu oznaczymy jako $\mathcal{CP}$, a nowy język RT^T uzależniony od czasu został nazwany RT^T_+. Aby uprościć notację, zapisujemy po prostu c, gdy chcemy przedstawić „c **in** $(-\infty, +\infty)$". Ograniczenia czasowe mogą w pewnym stopniu zaspokajać potrzeby, które w systemach niemonotonicznych obsługuje negacja, bez konieczności poświęcania monotoniczności systemu. Ważność czasowa może być wyrażona na różne sposoby, w szczególności:

$$[\tau_1, \tau_2]; [\tau_1, \tau_2); (\tau_1, \tau_2]; (\tau_1, \tau_2); (-\infty, \tau]; (-\infty, \tau);$$
$$[\tau, +\infty); (\tau, +\infty); (-\infty, +\infty); v_1 \cup v_2; v_1 \cap v_2; v_1 \backslash v_2$$

gdzie τ oznacza stałą czasową.

Przykład 2.

Załóżmy, że dwa pierwsze poświadczenia są niezależne od czasu, czyli są polityką uczelni zdefiniowaną odgórnie, niezależną od innych warunków, ważną do odwołania. Jednak prawdopodobne jest, że czas, w którym dana osoba jest studentem czy też doktorantem jest w pewien sposób określony, czyli przynależność do ról jest ograniczona. Zatem poświadczenia definiujące zbiór osób niezbędnych do uruchomienia przedmiotu zostają niezmienione, natomiast poświadczenia przypisujące rolom podmioty przyjmują zmienioną postać pokazaną poniżej:

(9) $$U.student \leftarrow \{Ala\} \textbf{ in } v_1$$

(10) $$U.student \leftarrow \{Ola\} \textbf{ in } v_2$$

(11) $$U.student \leftarrow \{Ela\} \textbf{ in } v_3$$

(12) $$U.student \leftarrow \{Ula\} \textbf{ in } v_4$$

(13) $$U.doktorant \leftarrow \{Ola\} \textbf{ in } v_5$$

(14) $$U.doktorant \leftarrow \{Jan\} \textbf{ in } v_6$$

i oznaczają, że poszczególne osoby, a mianowicie Ala, Ola, Ela, Ula i Jan pełnią poszczególne funkcje w czasie v_1, v_2, v_3, v_4, v_5 i v_6 odpowiednio. Na podstawie powyższych poświadczeń możemy wyznaczyć zbiory osób pełniące określone funkcje w określonym czasie. Jeśli chcemy wyznaczyć zbiór osób, które pełnią łącznie funkcję dwóch różnych studentów, to będzie to na przykład $\{Ala, Ola\}$ w czasie $v_1 \cap v_2$. Jeśli natomiast potrzebujemy wyznaczyć zbiór osób, które mogą wspólnie uruchomić przedmiot to będzie to $\{Ala, Ola\}$ czasie $v1 \cap v2 \cap v5$ lub $\{Ela, Ula, Jan\}$ w czasie $v_3 \cap v_4 \cap v_6$. Można też oczywiście ograniczyć ważność reguł definiujących ogólna politykę uczelni, czyli na przykład zamieniamy poświadczenie definiujące zbiór osób niezbędnych do uruchomienia przedmiotu na poniższe:

(15) $$U.przedmiot \leftarrow U.doktorant \odot U.studenci \textbf{ in } v_A$$

Teraz członkowie roli $U.przedmiot$ mogą przyczynić się do uruchomienia przedmiotu w czasie, który jest iloczynem czasu wyliczonego powyżej i v_A. Tak więc na przykład zbiór $\{Ala, Ola, Ela\}$ jest w stanie umożliwić uruchomienie przedmiotu w czasie $v_1 \cap v_2 \cap v_5 \cap v_A$ natomiast zbiór $\{Ela, Ula, Jan\}$ może pozwolić na uruchomienie przedmiotu w czasie $v_3 \cap v_4 \cap v_6 \cap v_A$.

Rozszerzenie to daje nam dużo możliwości. Możemy wyznaczyć różne zbiory osób, które pełnią określone funkcje w określonym czasie. Możemy też definiować politykę bezpieczeństwa na określony czas. Nie mamy potrzeby nakładania niemonotoniczności (która odebrałaby nam wiele możliwości modelowania), a jednak w pewnym stopniu ją uzyskujemy.

Poświadczenia warunkowe

Inną niezwykle istotną możliwością dodaną do RT^T, za pomocą której można zamodelować więcej realnych polityk bezpieczeństwa, jest możliwość sparametryzowania ważności poświadczeń na podstawie dostępności/niedostępności innych poświadczeń w kontekście wykonania. Poświadczenie warunkowe możne przyjąć następującą postać:

$$\textbf{if } podmiot \in / \notin rola \textbf{ then } poswiadczenie$$

Czyli na przykład za pomocą poniższego poświadczenia:

$$\textbf{if } Ala \notin Firma.pracownik \textbf{ then}$$
$$Ala.finanse \leftarrow Firma.asystentFinansowy$$

możemy opisać sytuację, w której sprawdzamy, że jeśli Ala nie jest w tej chwili aktywnym pracownikiem firmy (na przykład jest na urlopie, zwolnieniu, czy też nie jest już na stałe pracownikiem firmy), jej asystent finansowy może zajmować się wszystkimi kwestiami związanymi z finansami, którymi zajmowala się Ala. Status Ali jest opisany w kontekście wykonywania, kiedy poświadczenie jest używane. Poniższe poświadczenie:

$$Ala.finanse \leftarrow Firma.asystentFinansowy$$

jest dostępne w systemie, jeśli ze wszystkich poświadczeń w danym kontekście wykonania nie jesteśmy w stanie wywnioskować poświadczenia postaci:

$$Firma.pracownik \leftarrow Ala$$

Jeśli na przykład w danym zadaniu kontekst wykonania dostarcza nam poświadczenie:

$$Firma.pracownik \leftarrow Ala \textbf{ in } v$$

gdzie v jest określonym przedziałem czasowym, wtedy poświadczenie:

$$Ala.finanse \leftarrow Firma.asystentFinansowy$$

jest dostępne w każdym czasie, który nie zawiera przedziału czasowego v. Możemy w ten sposób łatwo przedstawić kwestię zastępstw i czasowej delegacji uprawnień.

Jest to też użyteczne w przypadku, kiedy chcemy sprawdzić czy dana osoba jest przypisana do konkretnej roli, czyli, czy B jest członkiem roli $A.r$, a jeśli nie jest to ją do tej roli automatycznie dodać.

$$\textbf{if } B \notin A.r \textbf{ then } A.r \leftarrow B$$

W ten sposób możemy spełnić zapytanie o osobę, która może wykonać daną czynność. W przypadku nie znalezienia takiej osoby, za pomocą powyższego poświadczenia uaktywniamy konkretną, potrzebną nam w danej chwili osobę, w danej roli. W odniesieniu do naszego przykładu może to być na przykład poświadczenie:

$$\textbf{if } Ala \notin Firma.pracownik \textbf{ then}$$
$$Firma.pracownik \leftarrow Ala$$

które dodaje Alę do roli $Firma.pracownik$, czyli czyni ją aktywnym pracownikiem firmy.

Jest to częsty przypadek zastępstwa warunkowego, który jest szczególnie przydatny w języku RT, gdyż często stosuje się zasadę, że pojedynczą osobę w ważnej roli powinna zastępować para (lub więcej) zastępców działających zgodnie. Czyli na przykład poświadczenie:

$$\textbf{if } Ala \notin Firma.pracownik \textbf{ then}$$
$$Ala.projekty \leftarrow Firma.asystentProjektowy \cap Firma.asystentFinansowy$$

mówi nam, że jeśli Ala nie jest aktywnym pracownikiem w firmie, to jej działalność związana z projektami musi być zaakceptowana przez dwóch asystentów - projektowego i finansowego.

Uporządkowanie w poświadczeniach

Kolejnym rozszerzeniem bardzo przydatnym z punktu widzenia zastosowań praktycznych, jest ustalenie porządku, w jakim członkowie ról lub zbiory jednostek mogą się pojawiać w roli czy poświadczeniu. W celu możliwości ustalenia porządku, musimy dodać dwa nowe poświadczenia już na poziomie składni. Są to:

- $A.r \leftarrow B.s \overset{\odot}{\rightarrow} C.t$ – do roli $A.r$ należy jeden członek roli $B.s$ i jeden członek roli $C.t$ jednocześnie, w takiej właśnie kolejności.
- $A.r \leftarrow B.s \overset{\otimes}{\rightarrow} C.t$ – do roli $A.r$ należy jeden członek roli $B.s$ i jeden członek roli $C.t$ jednocześnie, w takiej właśnie kolejności, przy czym są to różniący się członkowie ról.

Zachowanie porządku może mieć bardzo duże znaczenie w przypadku dużych systemów, w szczególności wtedy, gdy jedna osoba pełni kilka ról jednocześnie i chcemy, aby pełniła ona w danej chwili jedną rolę, a w kolejnej inną. Możemy wymusić kolejność występowania danej jednostki w kontekście wykonania w momencie użycia poświadczenia.

Przykład 3.

Przypuśćmy, że w przypadku awarii maszyny na linii produkcyjnej obowiązuje polityka, która nakazuje wypełnić kartę oceny zgodności maszyny z wymaganiami bezpieczeństwa. W takim przypadku na karcie oceny musi podpisać się operator maszyny, który zauważył jej awarię. Następnie podpisać się musi serwisant, którego zadaniem jest naprawa tej maszyny. Efekt jego pracy jest oceniany przez kierownika zmiany. W kolejnym kroku operator musi uruchomić maszynę aby wyprodukować niewielką partię produktu gotowego i wykonać badania/pomiary produktu niezbędne do ustalenia czy dany wyrób jest zgodny z określonymi wymaganiami i normami dopuszczającymi produkt na rynek. Jeśli tak, składa ponownie podpis na karcie oceny i przekazuje maszynę do oceny przez pracownika działu jakości, który nakazuje ponowne wyprodukowanie partii produktu, ocenia go i w przypadku akceptacji, składa podpis na dokumencie. Aby maszynę dopuścić do użytku musi też na karcie oceny zgodności podpisać się kierownik działu technicznego a następnie kierownik linii produkcyjnej. Reasumując, kartę oceny zgodności maszyny z wymaganiami bezpieczeństwa musi podpisać operator (dwukrotnie), serwisant, kierownik zmiany ($M.kZmiany$), pracownik działu jakości ($M.jakosc$), kierownik działu technicznego ($M.kTechniczny$) i kierownik linii produkcyjnej ($M.kLinii$). W takiej sytuacji możemy sobie wyobrazić kilka możliwych scenariuszy jeśli chodzi o narzucenie porządku w kolejności składania podpisów na karcie oceny zgodności przez poszczególne osoby pełniące odpowiednie role.

W przypadku gdy kolejność podpisów nie jest ważna poświadczenie określające ten aspekt polityki bezpieczeństwa wyglądałoby następująco:

$$M.ocena \leftarrow M.operator \odot M.serwisant \odot M.kZmiany \odot M.operator \odot M.jakosc \odot M.kTechniczny \odot M.kLinii$$

Natomiast często wymagane jest, żeby podpisy musiały być składane w określonym (całkowitym lub też częściowym) porządku. W takim przypadku narzucenia kolejności podpi-

sów, poświadczenie musiałoby wyglądać w ten sposób.

$$M.ocena \leftarrow M.operator \overset{\odot}{\rightarrow} M.serwisant \overset{\odot}{\rightarrow} M.kZmiany \overset{\odot}{\rightarrow} M.operator \overset{\odot}{\rightarrow} M.jakosc \overset{\odot}{\rightarrow} M.kTechniczny \overset{\odot}{\rightarrow} M.kLinii$$

Teraz, gdy założymy, że wykonawcami poszczególnych ról są:

$$M.operator \leftarrow \{Jan\} \tag{16}$$

$$M.serwisant \leftarrow \{Jakub\} \tag{17}$$

$$M.kZmiany \leftarrow \{Jakub\} \tag{18}$$

$$M.jakosc \leftarrow \{Piotr\} \tag{19}$$

$$M.kTechniczny \leftarrow \{Jan\} \tag{20}$$

$$M.kLinii \leftarrow \{Jakub\} \tag{21}$$

widzimy, że do wypełnienia karty oceny bezpieczeństwa potrzebujemy łącznie siedmiu podpisów złożonych przez trzy osoby. $Jana$ pełniącego funkcję operatora (dwukrotnie) i kierownika działu technicznego, $Jakuba$ będącego serwisantem, kierownikiem zmiany i kierownikiem linii produkcyjnej, oraz $Piotra$ pełniącego funkcję pracownika działu jakości.

Każda z powyższych ról może być pełniona przez wiele innych osób. W zależności od tego, które poświadczenia w danej chwili mamy (bo nie wszystkie czasem są dostępne), możemy wyznaczyć zbiory składające się z różnych osób, które mogą wspólnie złożyć zbiór podpisów w celu złożenia danego wniosku projektowego.

W niektórych sytuacjach ważne jest ścisłe przestrzeganie porządku składanych podpisów, ale w wielu aplikacjach może to nie być niezbędne, a nawet być nieefektywne. W takiej sytuacji mogą powstać następujące scenariusze:

1. Przestrzeganie porządku podpisów jest istotne, należy więc podpisy uzyskiwać w ściśle założonym porządku. Naturalnym jest, że pierwszy podpis składa operator. W takim przypadku istotne jest aby kierownik zmiany złożył podpis dopiero po uzyskaniu podpisu od serwisanta, a następnie może ponownie się podpisać operator, po nim może złożyć podpis pracownik działu jakości, dalej kierownik działu technicznego i na końcu kierownik linii produkcyjnej. W przypadku, gdy dana osoba pełni więcej niż jedną rolę nie upoważnia to jej do złożenia podpisu w innej kolejności niż narzucona. Oznacza to, że w pierwszym kroku Jan musi podpisać się pod dokumentem jako operator. Następnie podpis składa serwisant czyli $Jakub$. Po czym, w kolejnym kroku, podpisuje się jako kierownik zmiany. Po nim podpisuje się ponownie operator, następnie pracownik działu jakości czyli $Piotr$. Dopiero teraz Jan może złożyć podpis jako kierownik działu technicznego i na koniec może podpisać się kierownik linii produkcyjnej czyli ponownie $Jakub$. Porządek podpisów składanych przez poszczególnym pracowników przedstawiony w powyższym scenariuszu przedstawia tabela 1.
2. Możemy pozwolić na podpisanie dokumentu przez daną osobę, która pełni więcej niż jedną rolę, w tym samym czasie pod warunkiem, że są to bezpośrednio po sobie następujące role. W naszym przypadku znaczyłoby to, że pozwalamy $Jakubowi$ na złożenie jednocześnie podpisu jako serwisant i kierownik zmiany. Opisany przypadek został zaprezentowany krok po kroku w tabeli 2.
   W tak prostym przykładzie widzimy już, że oszczędzamy jeden krok. Pokazuje to, jak dużo możemy oszczędzić i jak istotna może to być zmiana w przypadku dużych aplikacji.
3. W naszym trzecim scenariuszu możemy założyć iż pozwalamy na to aby dana osoba, która pełni więcej niż jedną rolę w poświadczeniu, mogła złożyć swoje podpisy jednocześnie. Może to być niezwykle użyteczne w systemach automatycznych. W naszym przykładzie oznacza to, że Jan może złożyć podpis jako operator w obu miejscach i kierownik działu technicznego jednocześnie, a $Jakub$ jako serwisant, kierownik zmiany i kierownik linii produkcyjnej. Trzeci scenariusz został przedstawiony w tabeli 3.
   W takiej sytuacji zakładamy, że Jan akceptuje swój drugi podpis jako operator, jeśli $Jakub$ podpisze kartę oceny jako serwisant, ale też jako kierownik zmiany oraz akceptuje swój podpis jako kierownik działu technicznego, jesli otrzyma podpis $Piotra$, jako pracownika działu jakości. Natomiast $Jakub$ zatwierdza swój podpis jako kierownika linii produkcyjnej, jeśli uzyskane zostaną podpisy $Jana$ jako operatora po raz drugi i kierownika działu technicznego oraz $Piotra$ jako pracownika działu jakości. W takim przypadku musimy mieć możliwość zaakceptowania lub też odrzucenia podpisu, który zależy od podpisu innej osoby. Jednakże jest to już kwestia implementacyjna danej aplikacji.
   Na tym prostym przykładzie widzimy, jaką oszczędność daje nam możliwość narzucenia porządku w kolejności składania podpisów na karcie oceny bezpieczeństwa. Zastosowanie powyższego rozwiązania w bardziej złożonym systemie pozwoli na uzyskanie oszczędności na o wiele wyższym poziomie.
4. Kolejny przypadek jaki możemy sobie wyobrazić to taki, w którym jednostka może pojawić się w konkretnej roli w kontekście wykonania dokładnie w momencie pojawienia się poświadczenia. Możemy wtedy wprowadzić kolejny typ roli i oznaczyć go za pomocą podkreślonego identyfikatora (na przykład r, s, t). W takiej sytuacji, kiedy zmienimy nasze poświadczenie z:

   $$M.ocena \leftarrow M.operator \overset{\odot}{\rightarrow} M.serwisant \overset{\odot}{\rightarrow} M.kZmiany \overset{\odot}{\rightarrow} M.operator \overset{\odot}{\rightarrow} M.jakosc \overset{\odot}{\rightarrow} M.kTechniczny \overset{\odot}{\rightarrow} M.kLinii$$

   na:

   $$M.ocena \leftarrow M.operator \overset{\odot}{\rightarrow} M.serwisant \overset{\odot}{\rightarrow} M.kZmiany \overset{\odot}{\rightarrow} M.operator \overset{\odot}{\rightarrow} M.jakosc \overset{\odot}{\rightarrow} M.kTechniczny \overset{\odot}{\rightarrow} M.\underline{kLinii}$$

   w naszym trzecim scenariuszu będziemy mieli sytuację opisaną w tabeli 4, to znaczy, że $Jakub$ będzie mógł złożyć swój podpis jako kierownik linii produkcyjnej dopiero w momencie, kiedy pozostałe osoby złożą wszystkie niezbędne podpisy.

Powyższe rozszerzenia pokazują jak możemy zwiększać użyteczność języków z rodziny RT w praktycznym zastosowaniu, a mianowicie, jak uzyskać większą możliwość zastosowania polityk bezpieczeństwa poprzez zastosowanie ograniczeń czasowych, poświadczeń warunkowych czy też ustalonego porządku występujących poświadczeń.

Tablica 1. Porządek podpisów w pierwszym scenariuszu

Krok	operator	serwisant	kZmiany	operator	jakosc	kTechniczny	kLinii
1	Jan	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ
2	Jan	$Jakub$	ϕ	ϕ	ϕ	ϕ	ϕ
3	Jan	$Jakub$	$Jakub$	ϕ	ϕ	ϕ	ϕ
4	Jan	$Jakub$	$Jakub$	Jan	ϕ	ϕ	ϕ
5	Jan	$Jakub$	$Jakub$	Jan	$Piotr$	ϕ	ϕ
6	Jan	$Jakub$	$Jakub$	Jan	$Piotr$	Jan	ϕ
7	Jan	$Jakub$	$Jakub$	Jan	$Piotr$	Jan	$Jakub$

Tablica 2. Porządek podpisów w drugim scenariuszu

Krok	operator	serwisant	kZmiany	operator	jakosc	kTechniczny	kLinii
1	Jan	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ
2	Jan	$Jakub$	$Jakub$	ϕ	ϕ	ϕ	ϕ
3	Jan	$Jakub$	$Jakub$	Jan	ϕ	ϕ	ϕ
4	Jan	$Jakub$	$Jakub$	Jan	$Piotr$	ϕ	ϕ
5	Jan	$Jakub$	$Jakub$	Jan	$Piotr$	Jan	ϕ
6	Jan	$Jakub$	$Jakub$	Jan	$Piotr$	Jan	$Jakub$

Tablica 3. Porządek podpisów w trzecim scenariuszu

Krok	operator	serwisant	kZmiany	operator	jakosc	kTechniczny	kLinii
1	Jan	ϕ	ϕ	Jan	ϕ	Jan	ϕ
2	Jan	$Jakub$	$Jakub$	Jan	ϕ	Jan	$Jakub$
3	Jan	$Jakub$	$Jakub$	Jan	$Piotr$	Jan	$Jakub$

Tablica 4. Porządek podpisów w czwartym scenariuszu

Krok	operator	serwisant	kZmiany	operator	jakosc	kTechniczny	kLinii
1	Jan	ϕ	ϕ	Jan	ϕ	Jan	ϕ
2	Jan	$Jakub$	$Jakub$	Jan	ϕ	Jan	ϕ
3	Jan	$Jakub$	$Jakub$	Jan	$Piotr$	Jan	ϕ
4	Jan	$Jakub$	$Jakub$	Jan	$Piotr$	Jan	$Jakub$

Podsumowanie

Zadaniem artykułu jest pokazanie, jak pewne modyfikacje wprowadzone do istniejących rozwiązań w rozproszonej kontroli dostępu, mogą przyczynić się do zwiększenia możliwości zastosowania ich w rzeczywistych systemach. Stworzony system wnioskowania wydaje się być niezbędny zwłaszcza w przypadku dużych rozproszonych systemów, gdzie użytkownicy mogą posiadać tylko częściową wiedzę na temat swoich aktualnych możliwych poświadczeń. Ograniczenia czasowe wprowadzone do poświadczeń pozwalają na zamodelowanie szerokiego zakresu polityk bezpieczeństwa, gdyż często osoba pełniąca różne role może mieć możliwość pełnienia jednej roli pod warunkiem, że w tym samym czasie nie pełni innej. Role też często są przydzielane na pewien okres czasu. Możliwość sparametryzowania ważności poświadczeń na podstawie dostępności/niedostępności innych poświadczeń w kontekście wykonania pokazuje, jak można rozwiązać kwestię zastępowalności i czasowej delegacji uprawnień. Niezwykle przydatne może też być ustalenie porządku, w jakim członkowie ról mogą się pojawiać w roli czy też poświadczeniu. Umożliwia to zamodelowanie sekwencji czy też ustalonego porządku.

Prace były współfinansowane przez NCBiR w ramach projektu „System zapewnienia bezpiecznej komunikacji IP w obszarze zarządzania siecią elektroenergetyczną”.

***Autorzy**: dr inż. Anna Felkner, dr inż. Adam Kozakiewicz Naukowa i Akademicka Sieć Komputerowa ul. Wąwozowa 18 02-796 Warszawa, Polska, email: anna.felkner@nask.pl, adam.kozakiewicz@nask.pl*

LITERATURA

[1] Blaze M., Feigenbaum J., Lacy J.: Decentralized Trust Management, Proc. 17th IEEE Symposium on Security and Privacy, 1996, pp. 164–173.

[2] Ruohomaa S., Kutvonen L.: Trust management survey, Proc. iTrust 3rd International Conference on Trust Management, Rocquencourt, pp. 77–92, LNCS 3477/2005, 2005.

[3] Li N., Mitchell J.: RT: A Role-based Trust Management Framework, Proc. 3rd DARPA Information Survivability Conference and Exposition, 2003, pp. 201–212

[4] Li N., Winsborough W. H., Mitchell J. C.: Distributed credential chain discovery in trust management, Journal of Computer Security, 11(1), pp. 35–86, 2003.

[5] Felkner A., Sacha K.: The Semantics of Role-based Trust Management Languages, Advances in Software Engineering Techniques, T. Szmuc, M. Szpyrka, and J. Zendulka, Eds. LNCS, Heidelberg: Springer, 2012, vol. 7054, pp. 179–189.

[6] Felkner A., Kozakiewicz A.: Time Validity in Role-based Trust Management Inference System, Secure and Trust Computing, Data Management, and Applications Communications in Computer and Information Science, vol. 187, 2011, pp. 7–15.

[7] Gorla, D., Hennessy, M., Sassone, V.: Inferring Dynamic Credentials for Role-Based Trust Management, Proc. 8th ACM SIGPLAN PPDP'06, pp. 213–224, (2006)

[8] Felkner A., Kozakiewicz A.: More Practical Application of Trust Management Credentials, IEEE Conference: Federated Conference on Computer Science and Information Systems, 2015, pp. 1125–1134,