12. Hierarchie dostępu

12.1 Motywy dostępu:

- wspólne użytkowanie systemów informatycznych

- współdzielenie zasobów

- administrowanie zasobem

- kontrola dostępu

- praktyczność struktur hierarchicznych

Def.

(V, ≤) nazywamy porządkiem częściowym na V jeśli ≤ jest relacja dwuczynnikową na która jest zawarta przechodniość antysmetryczność tj. V ≤ V

$\forall v \in V$

$V_1 \leq V_2 \_ i \_ V_2 = V_3 \Rightarrow V_1 = V_3$

$V_1 \leq V_2 \_ i \_ V_2 \leq V_1 \Rightarrow V_1 = V_2$

Polityka przepływu interakcji wg. modelu BELL-LA-PADULA. W takim modelu x ≤ y oznacza, że informacja może przepływać od x d y (w górę hierachii)

Def.

Polityka bezpieczeństwa to piątka (V_1, ≤, P, O, λ):
V- zbiór klas bezpieczeństwa V={v_1,v_2,v_3,…,v_n}
P- zbiór podmiotów (użytkowników)
O- zbiór obiektów
$\lambda : O \cup P \rightarrow V - funkcje \_ kontroli \_ bezpieczenstwa$

Operacja (przykładowa):
- czytanie informacji zawartych w obiekcie

- zapis do obiektu
- dopisywanie do obiektu
- wykonywanie programu

Ustalając operacje powiemy, że jest ona dowolnie dla pary (P, O) wtedy i tylko wtedy, gdy $\lambda(p) \geq \lambda(o)$

Każdemu obiektowi przypisany jest jeden klucz bezpieczeństwa $\lambda(a)$

Każdemu podmiotowi przypisane są dwa podmioty (elementy zb. Klas bezpieczeństwa) $\lambda(p) = \lambda_{bież}(p)$ oraz
$\lambda(p) = \lambda_{dow}(p)$

Jeżeli n jest zadanym wprowadzeniem to trójka (p, o, n) opisuje uprawnienie podmiotu p do obiektu o w danym momencie tzn. przy uprawnieniu do obiektu $O \Leftrightarrow \lambda(p) \geq \lambda(a)$

W danym ciągu politykę bezpieczeństwa dostępu do danych $(V_1, \leq, P, O, \lambda)$ będziemy ograniczać do pary (V,E).

Przykład:

$V=(v_1,v_2,v_3,v_4)$

v_1- poziom jawny

v_2- informacje poufne

v_3- informacje tajne

v_4- informacje ściśle tajne

12.2 Grafy skierowane

W dalszym ciągu będziemy patrzeć na reprezentacje wyjściowego porządku (V,≤) jako zadane przez graf (acykliczny i skierowany) G=(V, E)

Polityka bezpieczeństwa realizuje się przez przypisanie każdej klasie $v \in V$ klucza k od V, który ma strzec „dostępu" do obiektu w klasie v tj. takich, że $\lambda(o)=v$

Oznaczenia:

$G^{-}(v)$={w ϵ V; istnieje ścieżka w do V}

$G^{+}(v)$={w ϵ V; istnieje ścieżka v do w}

D(v)- zbiór bezpośrednich potomków w (dzieci)

R(v)- zbiór bezpośrednich przodków (rodziców)

Def.

Schematem przedziału kluczy nazywamy parę dwóch algorytmów wielomianowych (sat, Derive) określonych jak następuje

Set(1^l,G) jest randomizowanym algorytmem, który na wejściu dostaje parametr bezpieczeństwa 1^l a na wyjściu dwa odwzorowania:

- publiczne Pub: $V \cup E \rightarrow \{0,1\}^*$,który przypisuje wierzchołkom w grafie pewne parametry publiczne oraz krawędzie (v_i, v_j) pewne etykiety publiczne.

- tajny sec: $V \rightarrow \{0,1\}^* * \{0,1\}^l$, który przypisuje każdemu wierzchołkowi pewna prywatna informacje z funkcją s= s(v) oraz klucz k= k(v), v ϵ V

-Derive (G, Pub, v_i, v_j, s_i) jest deterministycznym algorytmem, który na wyjściu pobiera publiczna (graf G) informację. Pub wygenerowane przez algorytm set wierzchołek barwiony v_i dowolny, vj którego klucz chcemy obliczyć oraz tajna informacje s_i wierzchołka w_i. Algorytm zwraca klucz $k_j = k_i(v_j)$ dla wierzchołka v_j o ile $v_j \in G_r^{+}(V_i)$ lub specjalny symbol $\perp$ w przeciwnych przypadkach

Dla poprawności, algorytmy: Set i Derive musza spełniać następujący warunek poprawności dla każdego

$$\forall v \in V \_\_\_ \forall v_j \in G_v^{+}(V_i) \_\_\_ \Pr\begin{bmatrix} k_j Derive(G, Pub, v_i, v_j, s_i) & \_\_(PUB, set) = set(1^l, G) \\ (s_i, k_i) = Sec(v_i) & (s_j, k_j) = Sec(v_j) \end{bmatrix}$$

Przy losowych wyborach algorytmu set