

Lecture 14 Outline

Reading: *text*, §8.2–8.2

1. Deterministic noninterference
 - a. Model of system
 - b. Example
 - c. Relationship of output to states
 - d. Projections and purge functions
 - e. Example
 - f. Noninterference
2. Alternative definition of security policy
 - a. Output-consistent
 - b. Security policy
 - c. Alternate projection function
 - d. Noninterference-secure with respect to the policy r
3. Unwinding Theorem
 - a. Locally respects
 - b. Transition-consistent
 - c. Unwinding theorem
4. Access Control Matrix interpretation
 - a. Model
 - b. ACM conditions
 - c. Policy conditions
 - d. Result
5. Policies that change over time
 - a. Generalization of noninterference
 - b. Example
6. Composing deterministic, noninterference-secure systems

Table of Notation

<i>notation</i>	<i>meaning</i>
C	set of commands (s, z) , where s executes operation z
C^*	set of sequences of commands
π'	generalized noninterference analogue to the purge function $\pi_{G,A}$
ϵ	empty string
c_s	sequence of commands
$P(c, \sigma_i)$	output from command c being executed in state σ_i
$P^*(c_s, \sigma_i)$	outputs when command sequence c_s is executed in state σ_i
$proj(s, c_s, \sigma_i)$	set of outputs in $P^*(c_s, \sigma_i)$ that subject s is authorized to see
w	sequence of elements of C leading up to current state
$cando(w, s, z)$	true if s can execute z in current state
$pass(s, z)$	give s right to execute z
w_n	v_1, \dots, v_n where $v_i \in C^*$
$prev(w_n)$	w_{n-1}
$last(w_n)$	v_n
π_L	projection function deleting all <i>High</i> inputs from trace