ECS 289M Lecture 16

May 5, 2006

Why Didn't They Work?

- For compositions to work, machine must act same way regardless of what precedes low level input (high, low, nothing)
- dog does not meet this criterion
 - If first input is *stop_count*, *dog* emits 0
 - If high level input precedes stop_count, dog emits 0 or 1

State Machine Model

- 2-bit machine, levels *High*, *Low*, meeting 4 properties:
- 1. For every input i_k , state σ_j , there is an element $c_m \in C^*$ such that $T^*(c_m, \sigma_j) = \sigma_n$, where $\sigma_n \neq \sigma_j$

 $-T^*$ is total function, inputs and commands always move system to a different state

May 5, 2006

ECS 289M, Foundations of Computer and Information Security Slide 3

Property 2

- There is an equivalence relation = such that:
 - If system in state σ_i and high level sequence of inputs causes transition from σ_i to σ_i , then $\sigma_i = \sigma_i$
 - If $\sigma_i \equiv \sigma_j$ and low level sequence of inputs $i_1, ..., i_n$ causes system in state σ_i to transition to σ'_i , then there is a state σ'_j such that $\sigma'_i \equiv \sigma'_j$ and the inputs $i_1, ..., i_n$ cause system in state σ_i to transition to σ'_i
- = holds if low level projections of both states are same

Property 3

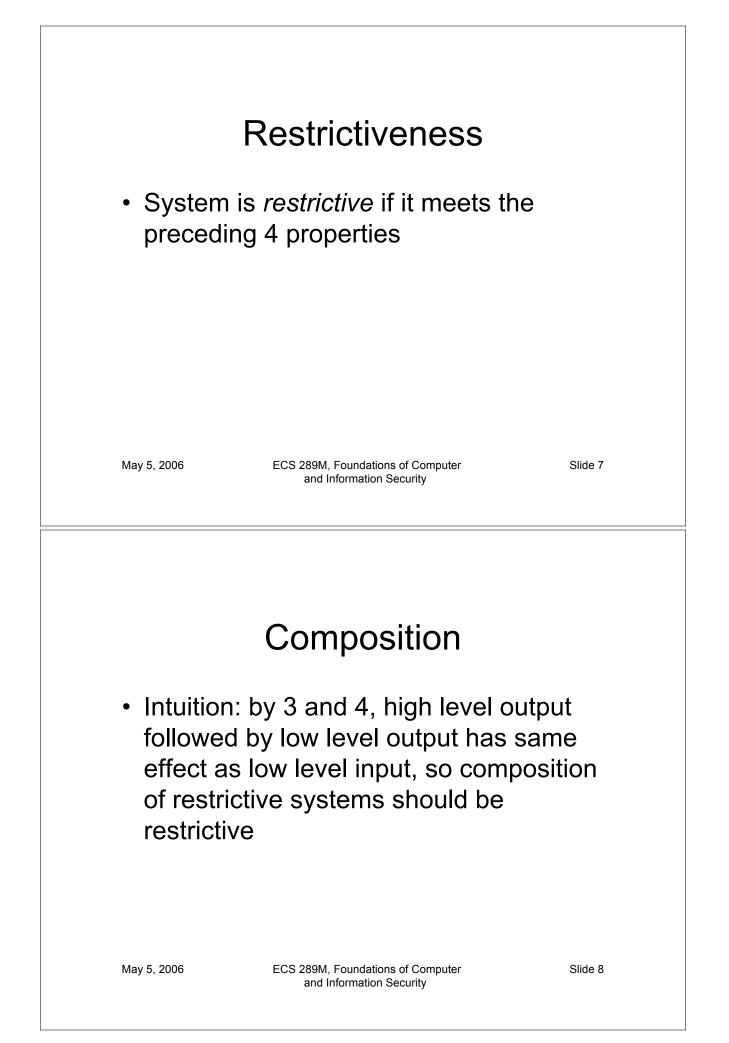
- Let σ_i ≡ σ_j. If high level sequence of outputs o₁, ..., o_n indicate system in state σ_i transitioned to state σ_i', then for some state σ_j' with σ_j' ≡ σ_i', high level sequence of outputs o₁', ..., o_m' indicates system in σ_i transitioned to σ_j'
 - High level outputs do not indicate changes in low level projection of states

May 5, 2006

ECS 289M, Foundations of Computer and Information Security Slide 5

Property 4

- Let σ_i = σ_j, let c, d be high level output sequences, e a low level output. If ced indicates system in state σ_i transitions to σ_i', then there are high level output sequences c' and d' and state σ_j' such that c'ed' indicates system in state σ_i transitions to state σ_i'
 - Intermingled low level, high level outputs cause changes in low level state reflecting low level outputs only



Composite System

- System M_1 's outputs are M_2 's inputs
- μ_{1i} , μ_{2i} states of M_1 , M_2
- States of composite system pairs of M₁, M₂ states (μ_{1i}, μ_{2i})
- e event causing transition
- e causes transition from state (μ_{1a}, μ_{2a}) to state (μ_{1b}, μ_{2b}) if any of 3 conditions hold

May 5, 2006

ECS 289M, Foundations of Computer and Information Security Slide 9

Conditions

- 1. M_1 in state μ_{1a} and e occurs, M_1 transitions to μ_{1b} ; e not an event for M_2 ; and $\mu_{2a} = \mu_{2b}$
- 2. M_2 in state μ_{2a} and e occurs, M_2 transitions to μ_{2b} ; e not an event for M_1 ; and $\mu_{1a} = \mu_{1b}$
- 3. M_1 in state μ_{1a} and *e* occurs, M_1 transitions to μ_{1b} ; M_2 in state μ_{2a} and *e* occurs, M_2 transitions to μ_{2b} ; *e* is input to one machine, and output from other

Intuition

- Event causing transition in composite system causes transition in at least 1 of the components
- If transition occurs in exactly one component, event must not cause transition in other component when not connected to the composite system

May 5, 2006

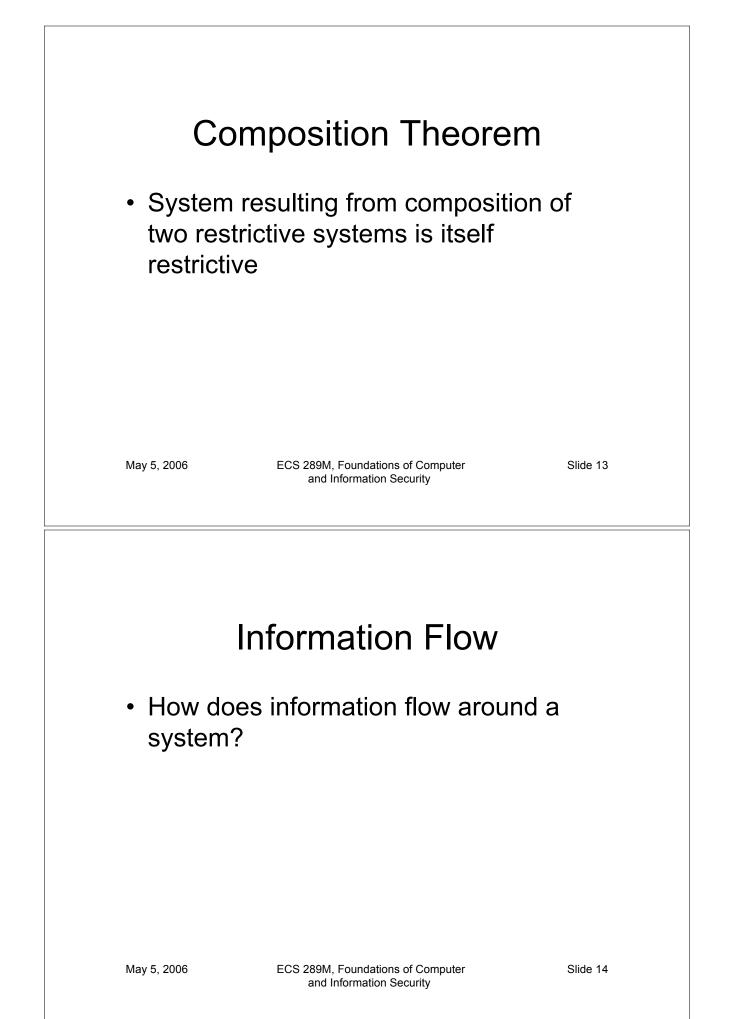
ECS 289M, Foundations of Computer and Information Security Slide 11

Equivalence for Composite

 Equivalence relation for composite system

 $(\sigma_a, \sigma_b) \equiv_C (\sigma_c, \sigma_d) \text{ iff } \sigma_a \equiv \sigma_c \text{ and } \sigma_b \equiv \sigma_d$

 Corresponds to equivalence relation in property 2 for component system



Detour: Entropy

- Random variables
- Joint probability
- Conditional probability
- Entropy (or uncertainty in bits)
- Joint entropy
- Conditional entropy
- Applying it to secrecy of ciphers

May 5, 2006

ECS 289M, Foundations of Computer and Information Security Slide 15

Random Variable

- · Variable that represents outcome of an event
 - X represents value from roll of a fair die; probability for rolling n: p(X = n) = 1/6
 - − If die is loaded so 2 appears twice as often as other numbers, p(X = 2) = 2/7 and, for $n \neq 2$, p(X = n) = 1/7
- Note: *p*(*X*) means specific value for *X* doesn't matter
 - Example: all values of X are equiprobable

Joint Probability

 Joint probability of X and Y, p(X, Y), is probability that X and Y simultaneously assume particular values

- If X, Y independent, p(X, Y) = p(X)p(Y)

-p(X = 3, Y = heads) = p(X = 3)p(Y = heads)= 1/6 × 1/2 = 1/12

May 5, 2006

ECS 289M, Foundations of Computer and Information Security Slide 17

Two Dependent Events

• X = roll of red die, Y = sum of red, blue die rolls

p(Y=2) = 1/36	p(Y=3) = 2/36	p(Y=4) = 3/36	p(Y=5) = 4/36
p(Y=6) = 5/36	p(Y=7) = 6/36	p(Y=8) = 5/36	p(Y=9) = 4/36
p(Y=10) = 3/36	<i>p</i> (<i>Y</i> =11) = 2/36	<i>p</i> (<i>Y</i> =12) = 1/36	

- Formula if events independent:
 p(X=1,Y=11) = p(X=1)p(Y=11) = (1/6)(2/36) = 1/108
- But in reality, Y = 11 is possible *only* when X = 5 and blue die is 6, so:

$$p(X=1, Y=11) = 0$$

Conditional Probability

 Conditional probability of X given Y, *p*(X|Y), is probability that X takes on a particular value given Y has a particular value

-p(Y=7|X=1) = 1/6-p(Y=7|X=3) = 1/6

May 5, 2006

ECS 289M, Foundations of Computer and Information Security Slide 19

Relationship

- p(X, Y) = p(X | Y) p(Y) = p(X) p(Y | X)
- Example:
 - -p(X=3, Y=8) = p(X=3|Y=8) p(Y=8) =(1/5)(5/36) = 1/36
- Note: if *X*, *Y* independent:

$$-\,p(X|\,Y)=p(X)$$

Slide 20

Entropy Uncertainty of a value, as measured in bits • Example: X value of fair coin toss; X could be heads or tails, so 1 bit of uncertainty - Therefore entropy of X is H(X) = 1• Formal definition: random variable X, values $x_1, ..., x_n$; so $\Sigma_i p(X = x_i) = 1$ $H(X) = -\sum_{i} p(X = x_{i}) \log p(X = x_{i})$ May 5, 2006 ECS 289M. Foundations of Computer Slide 21 and Information Security

Heads or Tails?

- $H(X) = -p(X=heads) \lg p(X=heads)$ - $p(X=tails) \lg p(X=tails)$ = $-(1/2) \lg (1/2) - (1/2) \lg (1/2)$ = -(1/2) (-1) - (1/2) (-1) = 1
- Confirms previous intuitive result

n-Sided Fair Die

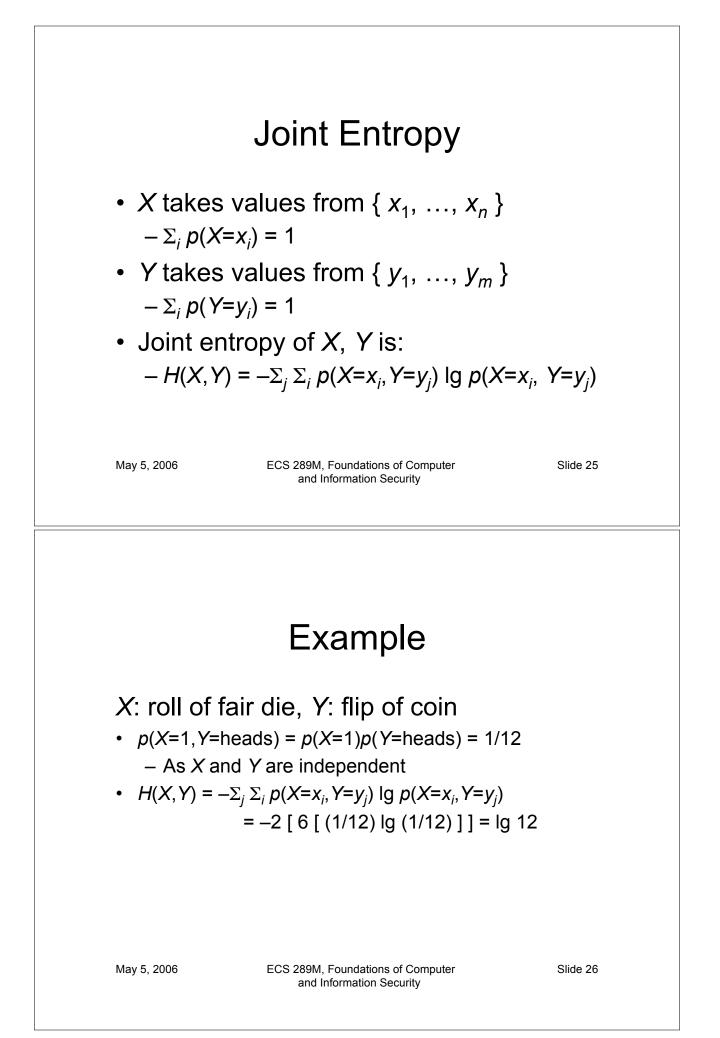
$$\begin{aligned} H(X) &= -\sum_{i} p(X = x_{i}) \log p(X = x_{i}) \\ \text{As } p(X = x_{i}) &= 1/n, \text{ this becomes} \\ H(X) &= -\sum_{i} (1/n) \log (1/n) = -n(1/n) (-\lg n) \\ \text{so} \\ H(X) &= \lg n \\ \text{which is the number of bits in } n, \text{ as} \\ \text{expected} \end{aligned}$$

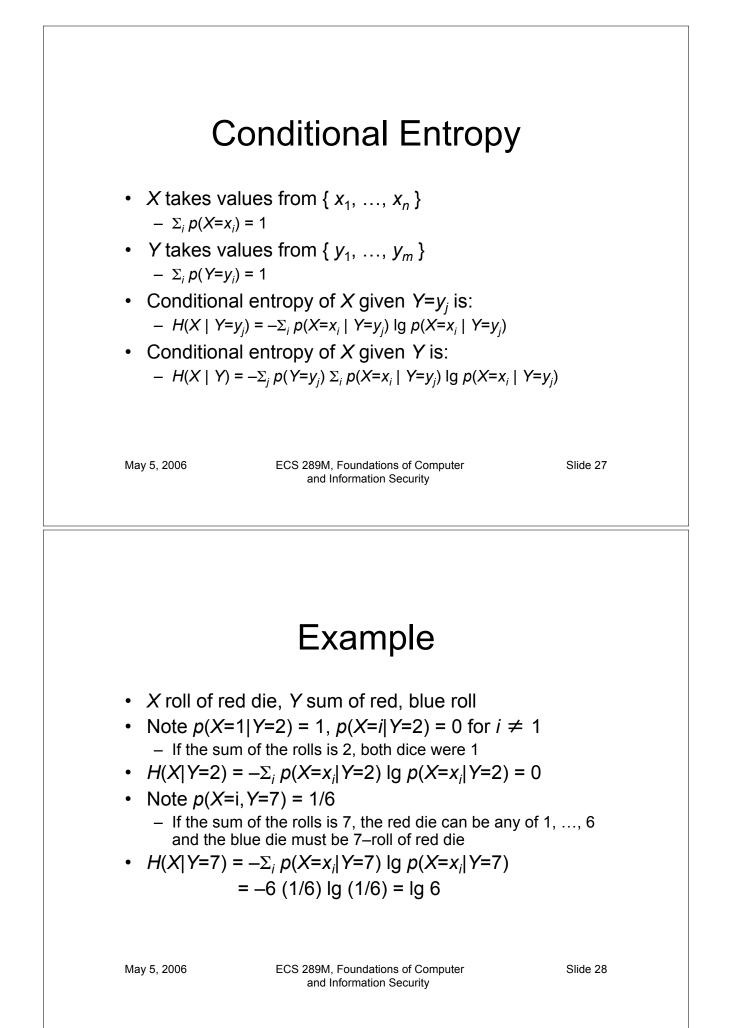
May 5, 2006

ECS 289M, Foundations of Computer and Information Security Slide 23

Ann, Pam, and Paul

Ann, Pam twice as likely to win as Paul *W* represents the winner. What is its entropy? $-w_1 = \text{Ann}, w_2 = \text{Pam}, w_3 = \text{Paul}$ $-p(W=w_1) = p(W=w_2) = 2/5, p(W=w_3) = 1/5$ • So $H(W) = -\sum_i p(W = w_i) \text{ Ig } p(W = w_i)$ = -(2/5) Ig (2/5) - (2/5) Ig (2/5) - (1/5) Ig (1/5) $= \text{ Ig } 5 - (4/5) \text{ Ig } 2 = \text{ Ig } 5 - (4/5) \approx 1.52$ • If all equally likely to win, H(W) = Ig 3 = 1.58





Perfect Secrecy

- Cryptography: knowing the ciphertext does not decrease the uncertainty of the plaintext
- $M = \{ m_1, ..., m_n \}$ set of messages
- *C* = { *c*₁, ..., *c_n* } set of ciphers
- Cipher c_i = E(m_i) achieves perfect secrecy if H(M | C) = H(M)

May 5, 2006

ECS 289M, Foundations of Computer and Information Security Slide 29



- Bell-LaPadula Model embodies
 information flow policy
 - Given compartments *A*, *B*, info can flow from *A* to *B* iff *B* dom *A*
- Variables x, y assigned compartments
 <u>x</u>, <u>y</u> as well as values
 - If $\underline{x} = A$ and $\underline{y} = B$, and A dom B, then y := x allowed but not x := y

Entropy and Information Flow

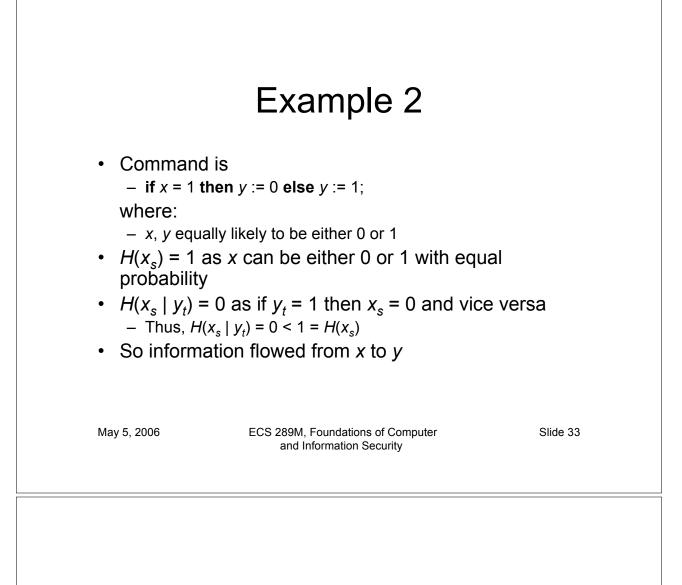
- Idea: info flows from x to y as a result of a sequence of commands c if you can deduce information about x before c from the value in y after c
- Formally:
 - -s time before execution of *c*, *t* time after
 - $-H(x_s \mid y_t) < H(x_s \mid y_s)$
 - If no *y* at time *s*, then $H(x_s | y_t) < H(x_s)$

May 5, 2006

ECS 289M, Foundations of Computer and Information Security Slide 31

Example 1

- Command is x := y + z; where:
 - $0 \le y \le 7$, equal probability
 - -z = 1 with prob. 1/2, z = 2 or 3 with prob. 1/4 each
- s state before command executed; t, after; so
 - $H(y_s) = H(y_t) = -8(1/8) \lg (1/8) = 3$
 - $H(z_s) = H(z_t) = -(1/2) \lg (1/2) 2(1/4) \lg (1/4) = 1.5$
- If you know x_t , y_s can have at most 3 values, so $H(y_s | x_t) = -3(1/3) \lg (1/3) = \lg 3$



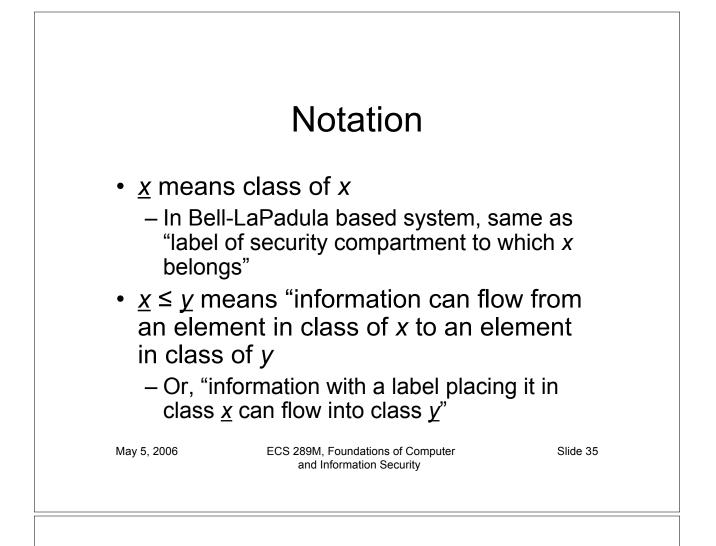
Implicit Flow of Information

- Information flows from x to y without an explicit assignment of the form y := f(x)
 f(x) an arithmetic expression with variable x
- Example from previous slide:

```
- if x = 1 then y := 0
```

```
else y := 1;
```

 So must look for implicit flows of information to analyze program



Information Flow Policies

Information flow policies are usually:

- reflexive
 - So information can flow freely among members of a single class
- transitive
 - So if information can flow from class 1 to class 2, and from class 2 to class 3, then information can flow from class 1 to class 3

Non-Transitive Policies

- Betty is a confident of Anne
- · Cathy is a confident of Betty
 - With transitivity, information flows from Anne to Betty to Cathy
- Anne confides to Betty she is having an affair with Cathy's spouse
 - Transitivity undesirable in this case, probably

May 5, 2006

ECS 289M, Foundations of Computer and Information Security Slide 37

Non-Lattice Transitive Policies

- 2 faculty members co-PIs on a grant
 Equal authority; neither can overrule the other
- Grad students report to faculty members
- Undergrads report to grad students
- Information flow relation is:
 - Reflexive and transitive
- But some elements (people) have no "least upper bound" element
 - What is it for the faculty members?

Confidentiality Policy Model

- Lattice model fails in previous 2 cases
- Generalize: policy $I = (SC_I, \leq_I, join_I)$:
 - SC₁ set of security classes
 - \leq_{I} ordering relation on elements of SC_{I}
 - $-join_l$ function to combine two elements of SC_l
- Example: Bell-LaPadula Model
 - SC₁ set of security compartments
 - $-\leq_l$ ordering relation *dom*
 - *join*, function *lub*

May 5, 2006

ECS 289M, Foundations of Computer and Information Security Slide 39