Introduction to Computer Security

September 2, 2004

IS 2935 / TEL 2810

- The objective of the course is to cover the fundamental issues of information system security and assurance.
  - Develop broad understanding of diverse issues
- Core course for (Security Assured Information Systems Track) SAIS
- Certified by NSA

Course Outline

- Security Basics (1-8)
  - General overview and definitions
  - Security models and policy issues
- Basic Cryptography and Network security (9-12, 26)
  - Crypto systems, digital signature, authentication, PKI
  - IPSec, VPN, Firewalls
- Systems Design Issues and Information assurance (13-21, 24)
  - Design principles
  - Security Mechanisms
  - Auditing Systems
  - Risk analysis
  - System verification
- Intrusion Detection and Response (23, 25, ..)
  - Attack Classification and Vulnerability Analysis
  - Detection, Containment and Response/Recovery
- Legal, Ethical, Social Issues
- Evaluation, Certification Standards
- Miscellaneous Issues (22, ..)
  - Malicious code, Mobile code
  - Digital Rights Management, Forensics
  - Watermarking, Trust Management
  - E/M-commerce security, Multidomain Security
- Implementations - Java Security

Course Material

- Textbook
  - Computer Security: Art and Science, Matt Bishop, Addison- Wesley, 2003
    - Will follow the book mostly
    - Will be supplemented by other material (references and papers)
    - Errata URL: http://nob.cs.ucdavis.edu/~bishop/
- Recommended
  - Inside Java 2 Platform Security, 2nd Edition, L. Gong, G. Ellision, M. Dageforde
- Other References
  - Security in Computing, 2nd Edition, Charles P. Pfleeger, Prentice Hall
  - Security Engineering: A Guide to Building Dependable Distributed Systems, Ross Anderson, Wiley, John & Sons, Incorporated, 2001
  - Building Secure Software: How to avoid the Security Problems the Right Way, John Viega, Gary McGraw, Addison-Wesley, 2002
- Papers
  - List will be provided as supplemental readings and review assignments

Prerequisites

- Assumes the following background
  - Programming skill
  - Working knowledge of
    - Operating systems, algorithms and data structures, database systems, and networks
  - Basic Mathematics
- Not sure?   SEE ME

Grading

- Lab + Homework/Quiz/Paper review 35%
- Paper/Project 15%
  - List of suggested topics will be posted;
  - Encouraged to think of a project/topic of your interest
- Exams 40% includes
  - Midterm 20%
  - Comprehensive Final 20%
- Remaining 10 %
  - LERSAIS-SIG (Student Interest Group)
  - Seminar and participation

Contact

- James Joshi
- 721, IS Building
- Phone: 412-624-9982
- E-mail: jjoshi@mail.sis.pitt.edu
- Web: www2.sis.pitt.edu/~jjoshi/INFSCI2935
- Office Hours:
  - Fridays: 2.00 – 4.00 p.m.
  - By appointments
- GSA: will be announced later

Course Policies

- Your work MUST be your own
  - Zero tolerance for cheating
  - You get an F for the course if you cheat in anything however small – NO DISCUSSION
- Homework
  - There will be penalty for late assignments (15% each day)
  - Ensure clarity in your answers – no credit will be given for vague answers
  - Solutions will be posted in the library OR Webpage
- Check webpage for everything!
  - You are responsible for checking the webpage for updates

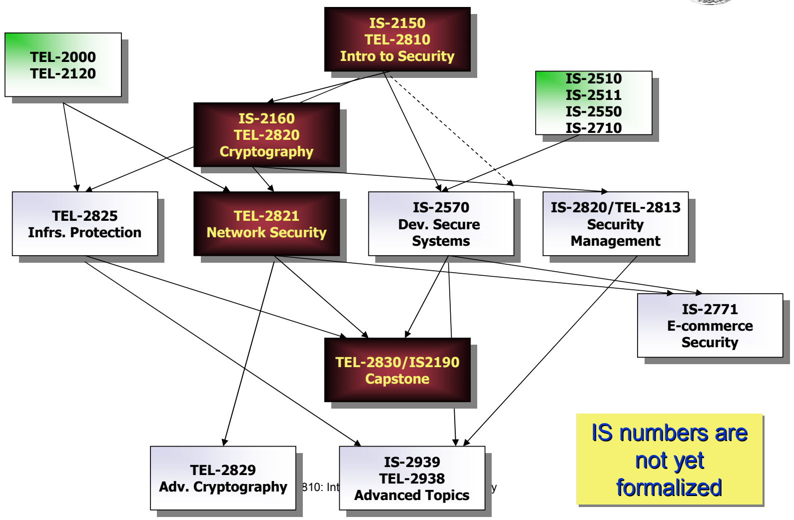
MSIS
Security Assured Information Systems Track

Foundations (3 credits)	Cognitive Systems (6 credits)	Systems and Technology (12 credits)	SAIS Track Core (12 credits)	SAIS Track Electives (3 credits)
IS-2000 Intro to IS	IS-2300 Human Information Processing IS-2470 Interactive System Design	IS-2511 Advanced Analysis & Design TEL-2000 Intro To Telecom IS-2550 Client-Server IS-2710 DBMS	IS-2150 Intro To Security IS-2160 Cryptography TEL-2821 Network Security TEL 2830/IS-2190 Capstone Course in Security	IS-2570 Developing secure Systems IS-2771 Security in E-Commerce IS-2820/TEL-2813 Security Management LIS-2194 Information Ethics LIS-2184 Legal issues in Handling Information

MST
Security Assured Information Systems Track

Core Required (9 credits)	Human Comm Mgmt/Policy (6 credits)	Protocols and Design (6 credits)	SAIS Track Core (12 credits)	SAIS Track Electives (3 credits)
TEL-2210 Electronic Comm II TEL-2120 Network Performance TEL-2310 Computer Networks	IS-2300 Human Information Processing TEL-2510 US Telecom Policy OR TEL-2511 Intl. Telecom Policy OR LIS-2194 Information Ethics	TEL-2110 Network Design TEL-2121 Network Mgt. TEL-2320 LANs TEL-2321 WANs TEL-2720 Cellular Radio and PCS TEL-2721 Mobile Data Networks	TEL-2810 Intro To Security TEL-2820 Cryptography TEL-2821 Network Security TEL-2830 Capstone Course in Security	TEL-2825 Infrs. Protection IS-2771 Security in E-Commerce IS-2820/TEL-2813 Security Management TEL-2829 Adv. Cryptography OR Other Electives

Expected Pre-requisite Structure



National Center of Academic Excellence in Information Assurance Education

(2004-2007)

Certified for

NSTISS 4011
Information Security Professionals
NSTISS 4012
Designated Approving Authority (DAA)
NSTISS 4013
System Administrator in Information Systems Security

The Department of Information Science and Telecommunication's
Laboratory of Education and Research on Security Assured Information Systems
(LERSAIS),
a National Center of Academic Excellence in Information Assurance Education (2004-2007),
hereby certifies that

Mr. John Smith

has successfully completed the requirements for the DIST's IA certification in Fall 2004

The DIST's IA certification requires a student to demonstrate competence in the following three IA courses
TELCOM 2810 Introduction to Computer Security;
TELCOM 2820 Cryptography
TELCOM 2821 Network Security

These three courses have been certified by the National Security Agency (NSA) as meeting the following IA education standards set by the Committee on National Systems Security (CNSS)
NSTISSI No. 4011, Information Systems Security Professionals
NSTISSI No. 4012, Designated Approving Authority
NSTISSI No. 4013, System Administrators in Information Systems Security

SAMPLE

Ronald Larsen
(Dean, School of Information Sciences)

Other Important Information

- In the process of setting up scholarships for IA education (DoD and/or NSF)
  - 2-years support (MS degree, 2 years of PhD)
  - US Citizens only
  - Requires 2 years work with federal agency
  - Expected to start next Fall (check LERSAIS URL: http://www.sis.pitt.edu/~lersais/
- NSA people visiting DIST next Thursday
  - Discuss internship/job opportunities

Introduction to Security

Overview of Computer Security

Information Systems Security

- Deals with
  - Security of (end) systems
    - Examples: Operating system, files in a host, records, databases, accounting information, logs, etc.
  - Security of information in transit over a network
    - Examples: e-commerce transactions, online banking, confidential e-mails, file transfers, record transfers, authorization messages, etc.

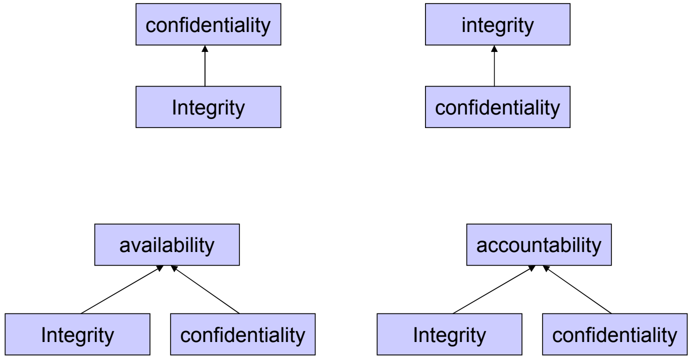
"Using encryption on the internet is the equivalent of arranging an armored car to deliver credit card information from someone living in a cardboard box to someone living on a park bench" –
Gene Spafford

Basic Components of Security

- Confidentiality
  - Keeping data and resources secret or hidden
- Integrity
  - Ensuring authorized modifications;
  - Includes correctness and trustworthiness
  - May refer to
    - Data integrity
    - Origin integrity
- Availability
  - Ensuring authorized access to data and resources when desired

  (Additional from NIST)
- Accountability
  - Ensuring that an entity's action is traceable uniquely to that entity
- Security assurance
  - Assurance that all four objectives are met

Interdependencies



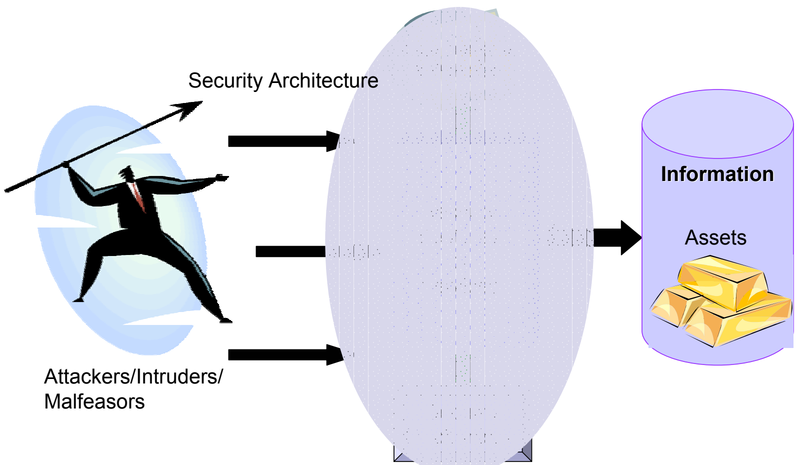
Information Security 20 years back

- Physical security
  - Information was primarily on paper
  - Lock and key
  - Safe transmission
- Administrative security
  - Control access to materials
  - Personnel screening
  - Auditing

Information security today

- Emergence of the Internet and distributed systems
  - Increasing system complexity
- Digital information needs to be kept secure
  - Competitive advantage
  - Protection of assets
  - Liability and responsibility
- Financial losses
  - The FBI estimates that an insider attack results in an average loss of $2.8 million
  - There are reports that the annual financial loss due to information security breaches is between 5 and 45 billion dollars
- National defense
  - Protection of critical infrastructures:
    - Power Grid;
    - Air transportation
  - Interlinked government agencies
    - Bad Grade for most of the agencies
    - Severe concerns regarding security management and access control measures (GAO report 2003)

Terminology



Attack Vs Threat

- A threat is a “potential” violation of security
  - The violation need not actually occur
  - The fact that the violation *might* occur makes it a threat
  - It is important to guard against threats and be prepared for the actual violation
- The actual violation of security is called an attack

Common security attacks

- Interruption, delay, denial of receipt or denial of service
  - System assets or information become unavailable or are rendered unavailable
- Interception or snooping
  - Unauthorized party gains access to information by browsing through files or reading communications
- Modification or alteration
  - Unauthorized party changes information in transit or information stored for subsequent access
- Fabrication, masquerade, or spoofing
  - Spurious information is inserted into the system or network by making it appear as if it is from a legitimate entity
- Repudiation of origin
  - False denial that an entity created something

Classes of Threats (Shirley)

- Disclosure: *unauthorized access to information*
  - Snooping
- Deception: *acceptance of false data*
  - Modification, masquerading/spoofing, repudiation of origin, denial of receipt
- Disruption: *interruption/prevention of correct operation*
  - Modification
- Usurpation: *unauthorized control of a system component*
  - Modification, masquerading/spoofing, delay, denial of service

Policies and Mechanisms

- A security policy states what is, and is not, allowed
  - This defines “security” for the site/system/*etc*.
  - Policy definition: Informal? Formal?
- Mechanisms enforce policies
- Composition of policies
  - If policies conflict, discrepancies may create security vulnerabilities

Goals of Security

- Prevention
  - To prevent someone from violating a security policy
- Detection
  - To detect activities in violation of a security policy
  - Verify the efficacy of the prevention mechanism
- Recovery
  - Stop policy violations (attacks)
  - Assess and repair damage
  - Ensure availability in presence of an ongoing attack
  - Fix vulnerabilities for preventing future attack
  - Retaliation against the attacker

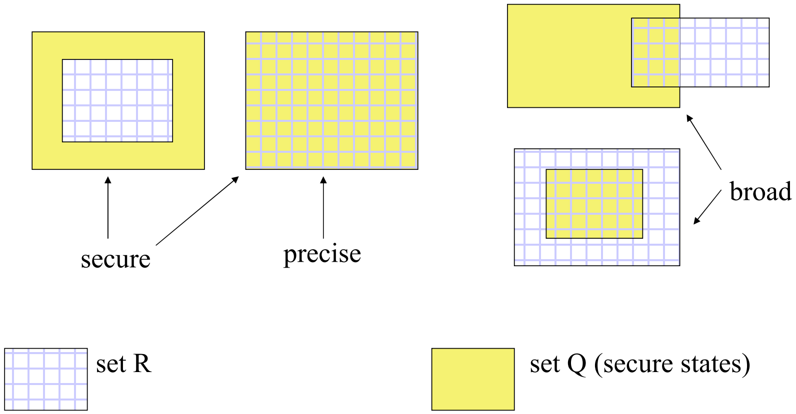
Assumptions and Trust

- Policies and mechanisms have implicit assumptions
- Assumptions regarding policies
  - Unambiguously partition system states into “secure” and “nonsecure” states
  - Correctly capture security requirements
- Mechanisms
  - Assumed to enforce policy; i.e., ensure that the system does not enter “nonsecure” state
  - Support mechanisms work correctly

Types of Mechanisms

- Let P be the set of all the reachable states
- Let Q be a set of secure states identified by a policy: $Q \subseteq P$
- Let the set of states that an enforcement mechanism restricts a system to be R
- The enforcement mechanism is
  - Secure if $R \subseteq Q$
  - Precise if $R = Q$
  - Broad if there are some states in R that are not in Q

Types of Mechanisms



Information Assurance

- *Information Assurance Advisory Council (IAAC)*:
  "Operations undertaken to protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality and non-repudiation"
- National Institute of Standards Technology
  "Assurance is the basis for confidence that the security measures, both technical and operational, work as intended to protect the system and the information it processes"

Assurance

- Assurance is to indicate "how much" to trust a system and is achieved by ensuring that
  - The required functionality is present and correctly implemented
  - There is sufficient protection against unintentional errors
  - There is sufficient resistance to intentional penetration or by-pass
- Basis for determining this aspect of trust
  - Specification
    - Requirements analysis
    - Statement of desired functionality
  - Design
    - Translate specification into components that satisfy the specification
  - Implementation
    - Programs/systems that satisfy a design

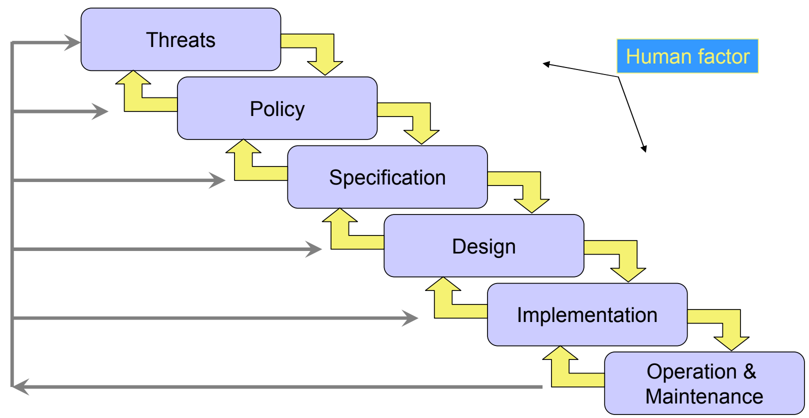
Operational Issues

- Cost-Benefit Analysis
  - Benefits vs. total cost
  - Is it cheaper to prevent or recover?
- Risk Analysis
  - Should we protect something?
  - How much should we protect this thing?
  - Risk depends on environment and change with time
- Laws and Customs
  - Are desired security measures illegal?
  - Will people do them?
  - Affects availability and use of technology

Human Issues

- Organizational Problems
  - Power and responsibility
  - Financial benefits
- People problems
  - Outsiders and insiders
    - *Which do you think is the real threat?*
  - Social engineering

Tying all together: The Life Cycle



Mathematics Review

Propositional logic/calculus

- Atomic, declarative statements (propositions)
  - that can be shown to be either TRUE or FALSE but not both; E.g., “Sky is blue”; “3 is less than 4”
- Propositions can be composed into compound sentences using connectives
  - Negation $\neg p$ (NOT) highest precedence
  - Disjunction $p \vee q$ (OR) second precedence
  - Conjunction $p \wedge q$ (AND) second precedence
  - Implication $p \rightarrow q$ q logical consequence of
- Contradiction: Formula that is always false : $p \wedge \neg p$
- Tautology: Formula that is always false : $p \vee \neg p$
- Construct truth tables??

Predicate/first order logic

- Propositional logic
- Variable, quantifiers, constants and functions
- Consider sentence: *Every directory contains some files*
- Need to capture "every" "some"
  - F(x): x is a file
  - D(y): y is a directory
  - C(x, y): x is a file in directory y
  - Existential quantifiers $\exists$ (There exists)
    - E.g., $\exists$ x is read as There exist some x
  - Universal quantifiers $\forall$ (For all)
- $\forall y \; D(y) \rightarrow (\exists x \; (F(x) \wedge C(x, y)))$ read as
  for every y, *if* y is a directory, then there exists a x such that x *is a file* and *x is in directory y*

Mathematical Induction

- Proof technique - to prove some mathematical property
- E.g. want to prove that M(n) holds for all natural numbers
  - Base case:
    - Prove that M(1) holds – called
  - Induction Hypothesis:
    - Assert that M(n) holds for n = 1 to k
  - Induction Step:
    - Prove that if M(k) holds then M(k+1) holds
- Exercise: prove that sum of first n natural numbers is
  - $1 + \ldots + n = n(n + 1)/2$

Lattice

- Let S, a set
- Cartesian product: S x S
- Binary relation R on S is a subset of S x S
- IF $(a, b) \in R$ we write aRb
  - Example, R is “less than equal to” ($\leq$)
  - If S = {1, 2, 3} then R is {(1, 1), (1, 2), (1, 3), ????)
  - $(1, 2) \in R$ is another way of writing $1 \leq 2$
- Properties of relations
  - Reflexive: is aRa for all $a \in S$
  - Antis-symmetric: if aRb and bRa implies a = b for all $a, b \in S$
  - Transitive: if aRb and bRc imply that aRc for all $a, b, c \in S$
  - Which properties hold for “less than equal to” ($\leq$)?

Lattice

- Total ordering: when the relation orders all elements
  - E.g., “less than equal to” ($\leq$) on natural numbers
- Partial ordering (poset): when the relation orders only some elements not all
  - E.g. “less than equal to” ($\leq$) on complex numbers; Consider (2 + 4i) and (3 + 2i)
- Upper bound ($u, a, b \in S$)
  - u is an upper bound of a and b means aRu and bRu
  - Least upper bound : lub(a, b) *closest upper bound*
- Lower bound ($u, a, b \in S$)
  - l is a lower bound of a and b means lRa and lRb
  - Greatest lower bound : glb(a, b) *closest lower bound*

Lattice

- A lattice is the combination of a set of elements S and a relation R meeting the following criteria
  - R is reflexive, antisymmetric, and transitive on the elements of S
  - For every s, t $\in$ S, there exists a greatest lower bound
  - For every s, t $\in$ S, there exists a lowest upper bound
- What about S = {1, 2, 3} and R = $\leq$?
- What about S = {2+4i; 1+2i; 3+2i, 3+4i} and R = $\leq$?

Access Control Matrix

Protection System

- State of a system
  - Current values of
    - memory locations, registers, secondary storage, etc.
    - other system components
- Protection state (P)
  - A system state that is considered secure
- A protection system
  - Describes the conditions under which a system is secure (in a protection state)
  - Consists of two parts:
    - A set of generic rights
    - A set of commands
- State transition
  - Occurs when an operation (command) is carried out

Protection System

- Subject (S: set of all subjects)
  - Active entities that carry out an action/operation on other entities; Eg.: users, processes, agents, etc.
- Object (O: set of all objects)
  - Eg.:Processes, files, devices
- Right
  - An action/operation that a subject is allowed/disallowed on objects

Access Control Matrix Model

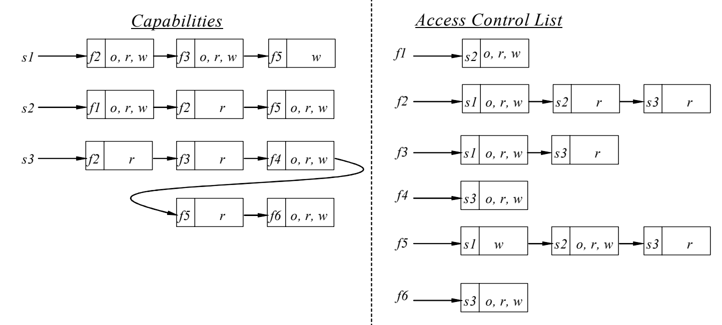
- Access control matrix
  - Describes the protection state of a system.
  - Characterizes the rights of each subject
  - Elements indicate the access rights that subjects have on objects
- ACM is an abstract model
  - Rights may vary depending on the object involved
- ACM is implemented primarily in two ways
  - Capabilities (rows)
  - Access control lists (columns)

Access Control Matrix

o: own
r: read
w:write

Access Matrix

	f1	f2	f3	f4	f5	f6
s1		o, r, w	o, r, w		w	
s2	o, r, w	r			o, r, w	
s3		r	r	o, r, w	r	o, r, w



Access Control Matrix

Hostnames	*Telegraph*	*Nob*	*Toadflax*
Telegraph	*own*	*ftp*	*ftp*
Nob		*ftp, nsf, mail, own*	*ftp, nfs, mail*
Toadflax		*ftp, mail*	*ftp, nsf, mail, own*

	Counter	*Inc_ctr*	*Dcr_ctr*	*Manager*
Inc_ctr	+			
Dcr_ctr	-			
manager		*Call*	*Call*	*Call*

State Transitions

- Let initial state $X_0 = (S_0, O_0, A_0)$
- Notation
  - $X_i \vdash \tau_{i+1} X_{i+1}$: upon transition τ_{i+1}, the system moves from state X_i to X_{i+1}
  - $X \vdash^* Y$: the system moves from state X to Y after a set of transitions
  - $X_i \vdash c_{i+1} (p_{i+1,1}, p_{i+1,2}, \ldots, p_{i+1,m}) X_{i+1}$: state transition upon a command
- For every command there is a sequence of state transition operations

Primitive commands (HRU)

Create subject s	Creates new row, column in ACM;
Create object o	Creates new column in ACM
Enter r into $a[s, o]$	Adds *r* right for subject *s* over object *o*
Delete r from $a[s, o]$	Removes *r* right from subject *s* over object *o*
Destroy subject s	Deletes row, column from ACM;
Destroy object o	Deletes column from ACM

Create Subject

- Precondition: $s \notin S$
- Primitive command: **create subject** *s*
- Postconditions:
  - $S' = S \cup \{ s \}$, $O' = O \cup \{ s \}$
  - $(\forall y \in O')[a'[s, y] = \varnothing]$ (row entries for s)
  - $(\forall x \in S')[a'[x, s] = \varnothing]$ (column entries for s)
  - $(\forall x \in S)(\forall y \in O)[a'[x, y] = a[x, y]]$

Create Object

- Precondition: $o \notin O$
- Primitive command: **create object** *o*
- Postconditions:
  - $S' = S$, $O' = O \cup \{ o \}$
  - $(\forall x \in S')[a'[x, o] = \varnothing]$ (column entries for *o*)
  - $(\forall x \in S)(\forall y \in O)[a'[x, y] = a[x, y]]$

Add Right

- Precondition: $s \in S$, $o \in O$
- Primitive command: enter *r* into $a[s, o]$
- Postconditions:
  - $S' = S$, $O' = O$
  - $a'[s, o] = a[s, o] \cup \{ r \}$
  - $(\forall x \in S' - \{ s \})(\forall y \in O' - \{ o \})$ $[a'[x, y] = a[x, y]]$

Delete Right

- Precondition: $s \in S$, $o \in O$
- Primitive command: **delete** r **from** $a[s, o]$
- Postconditions:
  - $S' = S$, $O' = O$
  - $a'[s, o] = a[s, o] - \{ r \}$
  - $(\forall x \in S' - \{ s \})(\forall y \in O' - \{ o \})$ $[a'[x, y] = a[x, y]]$

Destroy Subject

- Precondition: $s \in S$
- Primitive command: **destroy subject** s
- Postconditions:
  - $S' = S - \{ s \}$, $O' = O - \{ s \}$
  - $(\forall y \in O')[a'[s, y] = \varnothing]$ (row entries removed)
  - $(\forall x \in S')[a'[x, s] = \varnothing]$ (column entries removed)
  - $(\forall x \in S')(\forall y \in O')$ $[a'[x, y] = a[x, y]]$

Destroy Object

- Precondition: $o \in o$
- Primitive command: **destroy object** *o*
- Postconditions:
  - $S' = S$, $O' = O - \{ o \}$
  - $(\forall x \in S')[a'[x, o] = \varnothing]$ (column entries removed)
  - $(\forall x \in S')(\forall y \in O')\ [a'[x, y] = a[x, y]]$