Course Notes for CS3600

Introduction to Computer Security

NPS CISR

Naval Postgraduate School
Center for Information Systems Security
Studies and Research

December 14, 1998

Computer Science Department

Naval Postgraduate School
Monterey, CA 93943-5118

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Section 1

An Introduction to Computer Security

Copyrights and Permissions

The information contained in these documents is subject to change without notice.

The Naval Postgraduate School Center for Information Systems Security Studies and Research (NPS CISR) Introduction to Computer Security course notes and documentation are Copyright (c) 1998 by the Naval Postgraduate School Center for Information Systems Security Studies and Research.

Permission to use, copy, and modify NPS CISR course notes or their documentation for any non-commercial purpose is hereby granted without fee, provided that (i) the above copyright notices and the following permission notices appear in ALL copies of the software and related documentation, and (ii) The Naval Postgraduate School Center for Information Systems Studies and Research be given written credit in your course materials written documentation and be given graphical credit on any Web-based or alternative presentation of those materials.

Do not redistribute NPS CISR course materials without express written consent of NPS CISR. (E-mail communication qualifies as written permission.) These restrictions help justify our research efforts to the sponsors who fund our research.

NPS CISR course materials, associated software, and documentation were designed, written, and implemented at U.S. Government expense by employees of the U.S. Government. It is illegal to charge any U.S. Government agency for its partial or full use.

THESE MATERIALS ARE PROVIDED "AS IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Neither NPS CISR nor the Naval Postgraduate School shall be liable for errors contained herein or for incidental consequential damages in connection with the furnishing, performance, or use of this material.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Course Overview

Sections

1. *Introduction to Computer Security*

- Computer Security definition, laws, historical perspective

2. *Access Control I*

- Identification and Authentication and Discretionary Access Control

3. *Access Control II*

- Mandatory Access Control and Supporting Policies

4. *Building Secure Systems I*

- Design and implementation concepts that support assurance

5. *Malicious Software and Intrusion Detection*

- Trojan Horses, viruses, worms, etc.

6. *Accreditation, Certification and Disaster Planning*

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Course Overview

7. *Cryptography Basics*

- Private key, public key and hashing schemes

8. *Cryptographic Protocols*

- Cryptographic protocols for providing secrecy, integrity and authentication

9. *Network Security I*

- Special considerations, combining access control and cryptography

10. *Network Security II*

- TCP/IP, Internet and firewalls

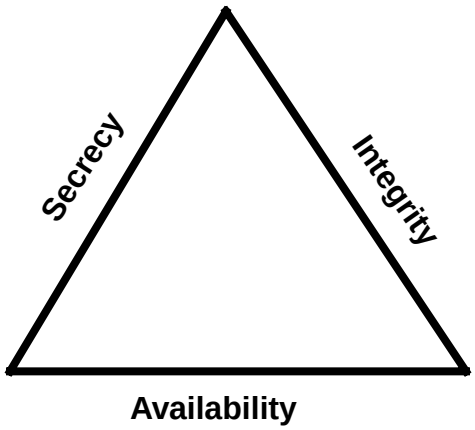
11. *Network Security III*

- Key management, PKI, e-commerce

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Aspects of Computer Security

The Golden Triangle of COMPUSEC



Broad definition also includes:

- Physical Security
- Emissions Security
- Personnel Security

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Who Should be Concerned?

As a Member of DoD

- You are required to safeguard classified material.
- You are required to safeguard sensitive but unclassified material such as privacy act data.

As a Private Citizen

Questions you should be asking about information stored about you:

- Who has it?
- Who is selling it?

Personal data (age, phone, address, SSN, etc.)

Credit information

Medical information

Purchasing history

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Issues for Concern

Why Would Someone Attempt Unauthorized Access?

- Curiosity
- Vandalism
- Financial gain
- Intelligence gathering
- Terrorism
- Warfare

Violations to Data Secrecy

- Wiretaps
- Obtain classified information
- Obtain financial data

Violations to Data Integrity

- Alter bank records
- Alter source e-mail address

Violations Affecting Availability

- Steal Time
- Deny Service

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Legislation Addressing Computer Security Issues

Three types of laws address computer security

- Laws about classified and sensitive but unclassified (SBU) data.
- Computer crime laws.
- Laws regarding privacy issues.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Protection of Classified or Sensitive Information

National Security Decision Directive 145

(NSDD 145) - 1984

- Mandated protection of both classified and unclassified sensitive information.
  - Productivity statistics
  - Census Bureau statistics
  - Air traffic control information
  - Health and financial records
- Gave NSA jurisdiction to encourage, advise and assist in the private sector (controversial to say the least).
- Created the System Security Steering Group
  - Secretaries of Defense, State and Treasury
  - Attorney General
  - Director of OMB
  - Director of CIA
- Revised and reissued in 1990 as NSDD 42
  - Scope narrowed to primarily defense related information.

National Telecommunications and Information Systems Security Publication 2

(NTISSP 2) - 1986

"National Policy on Protection of Sensitive but Unclassified Information in Federal Government Telecommunications and Automated Systems"

> Sensitive, but unclassified information is information the disclosure, loss, misuse, alteration, or destruction of which could adversely affect national security or other federal government interests. National security interests are those unclassified matters that relate to the national defense or the foreign relations of the U.S. government. Other government interests are those related, but not limited to the wide range of government or government derived economic, human, financial, industrial, agricultural, technology, and law enforcement information, as well as the privacy or confidentiality of personal or commercial proprietary information provided to the U.S. government by its citizens.

- Applies to all government agencies and contractors.
- Rescinded in March 1987 due to privacy concerns.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Protection of Classified or Sensitive Information

Computer Security Act

Public Law 100-235 (1987)

- Requires every U.S. government computer system that processes sensitive information to have a customized security plan.
- Requires all users of these systems (federal employees or contractors) to receive computer security training.
- Further defines sensitive information as:

  "...information whose loss, misuse, unauthorized access to, or modification of could adversely affect the national interest, or the conduct of federal programs, or the privacy to which individuals are entitled to under ... the Privacy Act"
- Gave the Institute of Computer Sciences and Technology (branch of NIST) responsibility for assessing the vulnerability of federal computer systems, for developing standards, and for providing technical assistance as well as developing guidelines for the training of personnel

DoD Directive 5200.28

Security Requirements for Automated Information Systems

- Provides mandatory, minimum AIS security requirements.
- Promotes the use of cost-effective, computer-based security for AISs.
- Applies to classified information, sensitive unclassified information and unclassified information.
- Applies to all AISs (stand-alone systems, communications systems, computer network systems, peripheral devices, embedded computer systems, personal computers, word processors, office automation systems, application and operating systems software and firmware).
- Requires all DoD systems to be accredited.
  - Outlines the accrediation process.
  - Specifies accrediation responsibilities, DAA, ISSO, etc.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Computer Crime

18 U.S. Code 1005 (1948)

- Prohibits making false entries in bank records.

18 U.S. Code 1006 (1948)

- Prohibits making false entries in credit institution records.

18 U.S. Code 1362 (1948)

- Prohibits malicious mischief to government property.

18 U.S. Code 2071 (1948)

- Prohibits concealment, removal, or mutilation of public records.

18 U.S. Code 1343 (1952**)**

- Prohibits wire fraud using any interstate communications system.

18 U.S. Code 1029 (1984)

- Prohibits fraudulent use of credit cards, passwords, and telephone access codes.

18 U.S. Code 2701 (1986)

- Prohibits unauthorized access to information that's stored electronically.

18 U.S. Code 2778 (1989)

- Prohibits illegal export of software or data controlled by the DoD

18 U.S. Code 2510 (1989)

- Prohibits the illegal export of software or data controlled by the Dept. of Commerce.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Computer Crime

Computer Fraud and Abuse Act

Public Law 99-474 (1986)

- Prohibits unauthorized or fraudulent access to government computers.
- Prohibits access with intent to defraud.
- Prohibits intentional trespassing.
- Applies to computers containing national defense, banking or financial information.
- Establishes penalties.
  - Fine of $5000 or twice the value of anything obtained.
  - Up to five (5) years in jail.
- Robert T. Morris (first person convicted - 1990)
  - 3 yrs probation, $10,000 fine, 400 hrs community service
  - Supreme Court refused to hear his case
- Wording vague
  - Must show intent to use information to injure U.S. or to provide advantage to foreign nation.
  - No distinction between those who use computers for hacking, crime or terrorism.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Privacy

U.S. Constitution, Bill of Rights

(1791)

- Fourth Amendment guarantees protection against unreasonable search and seizure.

  (Note: The Constitution does not explicitly guarantee an individuals right to privacy!)

Privacy Act

Public Law 93-579 (1974)

- Requires U.S. government to:
  - safeguard personal data processed by federal agency computer systems.
  - provide ways for individuals to find out what information is being recorded on them and a means to correct inaccuracies.

Right of Financial Privacy Act

(1978)

- Establishes that a depositor's bank accounts are private
- Can be accessed only by court order and proper notification

Electronic Funds Transfer Act

(1979)

- Protects the privacy of transmission of funds using electronic funds transfer (EFT)

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Privacy

Electronic Communications Act

(1986)

- Prohibits unauthorized interception of communications regardless of how transmission takes place:
  - wire
  - radio
  - electromagnetic
  - photo-electric
  - photo-optical

Computer Matching and Privacy Protection Act

U.S. Code 552a (1988)

- Protects against privacy violations due to information matching policies of the federal government.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Early Computer Security Efforts

1950's

- Stand alone systems using physical security for systems and terminals.
- Development of first TEMPEST standard.
- Establishment of U.S. Communications Security (COMSEC) Board.

1960's

- Beginning of the age of Computer Security.
- Timesharing systems (multiple users at the same time) and remote terminals created new problems.
- DoD launched first study of threats to DoD computers (1967).
    - Assembled task force under Advanced Research Projects Agency (ARPA).
    - Published findings in 1970 Security Controls for Computer Systems

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Early Computer Security Efforts

1970's

Security Requirements for Automatic Data Processing (ADP) Systems - issued by DoD in 1972
"Classified material contained in an ADP system shall be safeguarded by the continuous employment of protective features in the system's hardware and software design and configuration."

- Tiger Teams were used to test vendor claims of computer security features by attempting to break into the vendor's system.
  - The Tiger Teams showed overwhelming success at breaking in.
  - Patching techniques were used to shore up a system's weaknesses.
  - After patching, systems were still penetratable.
  - The patch and penetrate scheme shown to be inherently flawed. The concept of a Reference Monitor (Security Kernel) is spawned.
  - Most Tiger Teams were sponsored by DoD.
- IBM spends $40 Million to address computer security issues.
- First mathematical model of a multi-level security policy.
  - Developed by David Bell and Leonard LaPadula.
  - Central to development of computer security standards.
  - Laid groundwork for later models.
- Development of first security kernel.
  - USAF develops security Kernel for Multics system.
- Other kernels under development.
  - Mitre's DEC PDP-11/45
  - UCLA's Data Secure UNIX PDP-11/70

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Early NBS and NSA (NCSC) Involvement

National Bureau of Standards (NBS)

(National Institute of Standards and Technology - NIST)

1968 - NBS performed an initial study to evaluate the government's computer security needs.

1972 - NBS sponsored a conference on computer security in collaboration with ACM.

1973 - NBS initiated program aimed at researching development standards for computer security.

1977 - NBS began a series of Invitational Workshops dedicated to the Audit and Evaluation of Computer Systems.

Conclusion in NBS report from 1977 workshop
"...The point is that internal control mechanisms of current operating systems have too low integrity for them to... effectively isolate a user on the system from data that is at a 'higher' security level than he is trusted... to deal with."

National Computer Security Center

1980 - Director of NSA assigned responsibility for trusted information security products.

Response to NBS workshop and public seminars on the DoD Computer Security Initiative.

1981 - DoD Computer Security Center (CSC) was established

1985 - CSC becomes NCSC and assumes scope of responsibility broadens.

1985 - Communications and computer security merge under Deputy Directorate for Information Security Systems (INFOSEC).

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Tasking of the NCSC

- Encourage the widespread availability of trusted computer systems.
- Evaluate the technical protection capabilities of industry and government developed systems
- Provide technical support of government and industry groups engaged in computer security research and development.
- Develop technical criteria for the evaluation of computer systems.
- Evaluate commercial systems.
- Conduct and sponsor research in computer and network security technology.
- Develop and provide access to verification and analysis tools used to develop and test secure computer systems.
- Conduct training in the areas of computer security.
- Disseminate computer security information to other branches of the federal government and to industry.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Why Is Computer Security Difficult?

Some Factors

- Most managers are unaware of the value of their own computing resources.
- Fear of damage to public image.
- Legal definitions are often vague or non-existent.
- Legal prosecution is difficult:
  - Criminal must be traced.
  - No 'hard' evidence.
  - Hard to pin a value to data.
  - "No fingerprints" mentality.
  - Criminals viewed as just curious intellectuals.
- Computer Criminals do not fit a stereotype.
- The Law and Ethics are often unclear.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Computer Security Problems and Crimes

Computer Security Problems

- Most loss or damage is not malicious
  - Ignorance of existing policies.
  - Ignorance of the system on which they work.
- Accidents
  - Anyone can make a mistake!

Computer Security Crimes

Amateurs

- Temptation is there if access is available.
  - You wouldn't ask a stranger to hold your wallet while you went around the corner to move your car.
- Disgruntled employees
  - Oh Yeah! I'll show you!

Crackers and Hackers

- Often the challenge or Curiosity
  - West German group (Cliff Stoll)
  - Desert Shield / Desert Storm

Corporate Raiders

- Trade Secrets
- Inside Information
- Financial predictions

Foreign Intelligence

- West German group (Cliff Stoll)
- Desert Shield / Desert Storm

Terrorists

- No major incidents have occurred yet!
  - This is a potential nightmare waiting to happen.
  - Potential Economic disaster.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Categories of Computer Misuse

Human Error

- Hard to control

Abuse of Authority

- White collar crime

Direct Probing

- Rattling doorknobs

Probing With Malicious Software

- Trojan Horses

Direct Penetration

- Exploiting system bugs

Subversion of Mechanism

- Trap doors

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Is There A Threat?

Bank Theft (1984)

- Branch manager netted $25 million!

HBO Attack (1986)

- Captain Midnight overpowered HBO uplink.
- Part-time uplink operator.
- Displayed brief message to viewers.

Chaos Club

- West German Computer Club.
- In 1987 announced that it had successfully penetrated a United States Government Computer (NASA's).
- Able to store and manipulate information on SDI.
- NASA was unaware of penetration until messages started appearing on the system.
- NASA initially reported no damage.
- Virus later found on system which may have originated during the initial break-in.

Cliff Stoll and the KGB

- West German crackers tried to break in to over 450 computers (1987).
- 30 successful attempts.
- Looking for NBC related information to sell to KGB.
- First prosecution for Computer espionage.

Airline Computers (1988)

- A major airlines discovered its reservation and ticketing system had been penetrated.
- Bogus reservations had been made.
- Illegal tickets issued.

Internet Worm (1988)

- Affected Sun and VAX systems.
- 2100-2600 systems affected.
- Culprit: Robert T. Morris, a Cornell graduate student.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Is There A Threat?

Friday 13th Virus (1988)

- Threatened to erase the hard disks of financial, research and administrative computers.
- Originated at Hebrew University in Jerusalem.

Virus Flambe (1988)

- Infected a computer consulting firm.
- Altered scan rate of IBM monitors.
- Monitor burst into flames.

Satellite Positioning System (1989)

- 14 year old boy using Apple computer:
  - Cracked Air Force SPS.
  - Dialed unauthorized long-distance access codes.
  - Browsed through file of 200 businesses.

Desert Storm/Desert Shield (1990)

- 40 known attempts (6 confirmed successful)!
- How many unknown attempts/successes

Airline traffic control system (Arorua IL) (1995)

- Air traffic delayed for several hours!

Word Macro Virus (1995)

- Currently the most prevalent virus.

ActiveX Trojan Horse (1997)

- Transfers funds from your bank account to the hacker's account.

Erotic Photograph Trojan Horse (1997)

- Reconnects your system through a phone number in Moldova.

There is a problem and it's getting worse!

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Economic Damage from the Internet Worm

INDIRECT COSTS		
	Lost Machine Time	*Lost Access*
Machine hours unable to access network	2,076,880	
User hours unable to access network	8,307,520	
Burdened cost per hour	$20	$3
COST	**$41,537,600**	**$24,922,560**
DIRECT COSTS		
	Programmer Time	*Admin. Time*
Shutdown, monitor and reboot 42,700 machines	64,050	1,000
Initial problem analysis 12,400 machines	49,600	11,000
Identify, isolate, remove, clean, return to operation (6,200)	74,400	2,000
Reinfection, removal from network, shut-down analysis, monitor	62,000	12,000
Create patch, debug, install, test, check-out, monitor and implement	62,000	18,000
Analyze worm, disassemble, document (at each of 1200 networks)	192,000	22,000
Install fix on all UNIX systems, test, checkout and monitor	105,000	6,000
Residual checkup, tech communications conferencing and ripple events	187,000	264,000
TOTAL HOURS	796,050	336,000
Hourly Rate	$22	$42.50
COST	**$17,513,100**	**$14,280,000**
TOTAL COSTS: $98,253,260		

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Information Warfare

Overview

- Practically all of today's vehicles, ships and aircraft use control devices, communications systems and weapons systems that use computer systems, in one form or another. These systems, as in all information systems, rely upon the integrity of programs and the data which they input and output. Therefore, it is reasonable to assume that any attack on the data and programs may render these systems useless.

Hard kill, Soft kill or Imperceptible Degradation

- We frequently think in terms of complete system kills, either from a direct missile hit to the platform (hard kill) or to supporting sensor systems (soft kill) which the platform relies upon for its navigation, communication or targeting. The benefit of course, is that we can achieve total destruction of the system or render it completely useless; however, the major disadvantage is that our adversary is aware of the loss of their system and may engage alternate systems.
- Less obvious is the advantage that can be achieved through imperceptible degradation of the targeted information system. If we can reduce the measure of effectiveness of the system, often referred to as probability of kill (PK), in such a fashion that our adversary is unaware and places his faith in the system then we can effectively increase our force multiplier.
- There are a number of tradition battlefield models which have proven reasonably accurate in measuring overall force effectiveness on actual battlefields. Modification of these models to reflect system degradation to the enemy's control, communication and weapons systems have shown that viruses and worms can have a significant effect on battlefield results.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Insertion Techniques

- Insertion of viral code may take place during the manufacturing or distribution process. However, it is possible that the code could be inserted by field operatives during a time of crisis.
- The code could be triggered remotely or by use of logic bombs that render the system completely ineffective or slightly degrade system performance. The code obviously needs be hardware/software specific; that is, the code must be written to target a specific communications or weapons system.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Information Warfare

Battle Damage Assessment (BDA)

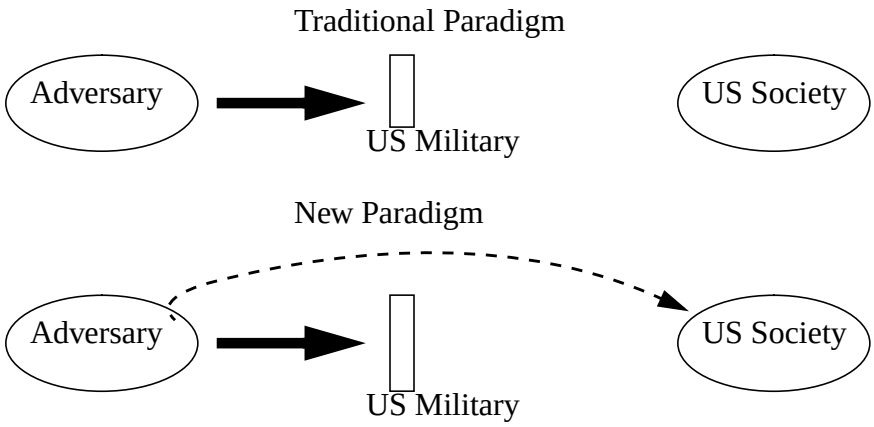
- Battle damage assessment for complete systems kill is challenging enough and frequently relies upon sophisticated surveillance systems to provide the battlefield commander with accurate information. Battle damage assessment for systems which have their PK perturbed slightly cannot be immediately assessed and requires analysis of battlefield results over prolonged periods.
- This factor makes it difficult for commanders to assess the overall effectiveness of such warfare techniques during the fog of war and as a result they may actually be unaware that it has achieved an advantage in their favor.

IW Difficulties

- What constitutes an act of war?
- What is the correct response?
- The civilian sector is not prepared to deal with attacks.

IW issues:

- IW attacks can be perpetrated with very little resources.
- Presents a different attack paradigm.



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

This page is intentionally blank.

Section 2

Access Control I

Identification and Authentication and

Discretionary Access Control Policies

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Access Control Policies

Security Policy:

- Generally speaking, a *security policy* describes how people may access documents or other information.
- A computer's version of a security policy consists of a precise set of rules for determining authorization as a basis for making access control decisions.
- This section and the following section present several security policies that are commonly implemented in computer systems.
- Policies presented include:
  - Access to systems based upon user identification.
  - Access to objects (such as files, directories, etc.) based upon user identification, where owners of objects can, at their discretion, grant access to other users.
  - Access to objects (such as files, directories, etc.) based upon the clearance level of the user.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

System Access Control

Controlling Access to the System Physically

Guards

- need at least 4 for 24 hour coverage
- must recognize someone, or token
- no record of access

Locks

- cheaper than a guard
- no record of access

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Identification and Authentication (I and A)

Controlling Access to the System Using Identification and Authentication

Two Step Process

Identification

- Telling the system who you are.

Authentication

- Proving to the system that you are who you say you are.

Three classic ways of establishing proof.

- Something you know.
- Something you have.
- Something you are.

I and A Benefits

- Can provide a complete log of access and attempted accesses.
- Access privileges granted/removed quickly

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

I and A

Passwords

- Something you know.
- Agreed upon code words entered by user.
- Subject to:
  - Loss
  - Disclosure
  - Attack

Attacks on Passwords

- Brute force attack.
  - Try all words.
- Probable password attack.
  - Try short words.
  - Try common words.
- Probable user password attack.
  - Family names.
  - Birth dates.

Password File

- Conventional encryption.
  - Enter password.
  - Decrypt stored password from table.
  - Compare passwords.
- One way cipher.
  - Enter password.
  - Encrypt password.
  - Compare to encrypted password.

Attacks Using Password File

- Readable password file.
- Backup tapes.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Guessing Passwords

Password Space:

- The password space is the set of all passwords.
- The size of a password space is determined by:
- The length of passwords, denoted by L.
- The size of the password alphabet, denoted by A.
  - If passwords only consist of lower case letters, $A = 26$.
  - If passwords consist of lower and upper case letters and digits, $A = 62$.
- The size of the password space is A^L.

L	26^L	52^L	62^L
4	4.57×10^5	7.31×10^6	1.47×10^7
6	3.09×10^8	1.98×10^{10}	5.68×10^{10}
8	2.09×10^{11}	5.34×10^{13}	2.18×10^{14}
10	1.41×10^{14}	1.44×10^{17}	8.39×10^{17}

Exhaustively trying all passwords:

- On the average, you will need to try half of them.
- If an intruder (using a computer) tries 1 password each second, they can try 60 passwords a minute, or 86,400 passwords a day.
- If passwords are of length 6 and consist of lower case letters, it will take 60 months, on the average.
- If an English word is used as a password, the problem is greatly simplified. There are only 5000 8-letter English words. The intruder can guess one of these in 42 minutes, on average.
- If the intruder steals an encrypted password file and the encryption software, it takes only 10^{-6} seconds to check whether an encrypted string is one of the encrypted passwords.
- Thus, potential passwords can be tested 1,000,000 times faster.
- A 6-letter password can be guessed in 155 seconds, on average.

Internet Worm Password Guesses:

- The following list shows passwords that the Internet worm tried.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Passwords Tried by the Internet Worm

aaa
academia
aerobics
airplane
albany
albatross
albert
alex
alexander
algebra
aliases
alphabet
ama
amorphous
analog
anchor
andromache
animals
answer
anthropegenic
anvils
anything
aria
ariadne
arrow
arthur
athena
atmosphere
aztecs
azure
bacchus
bailey
banana
bananas
bandit
banks
barber
baritone
bass
bassoon
batman
beater
beauty
beethoven
beloved
benz
beowulf
berkeley
berliner
beryl
beverly
bicameral
bob
brenda
brian
bridget
broadway
bumbling
burgess
campanile
cantor
cardinal
carmen
carolina
caroline
cascades
castle
cat
cayuga
celtics
cerulean
change
charles
charming
charon
chester
cigar
classic
clusters
coffee
coke
collins
comrades
computer
condo
cookie
cooper
cornelius
couscous
creation
creosote
cretin
daemon
dancer
daniel
danny
dave
december
defoe
deluge
desperate
develop
dieter
digital
discovery
disney
dog
drought
duncan
eager
easier
edges
edinburgh
edwin
edwina
egghead
eiderdown
eileen
einstein
elephant
elizabeth
ellen
emerald
engine
engineer
enterprise
enzyme
erastz
establish
estate
euclid
evelyn
extension
fairway
felicia
fender
fermat
fidelity
finite
fishers
flakes
float
flower
flowers
foolproof
football
foresight
format
forsythe
fouier
fred
friend
frighten
fun
fungible
gabriel
gardner
garfield
gauss
george
gertrude
ginger
glacier
gnu
golfer
gorgeous
gorges
gosling
gouge
graham
gryphon
guest
guitar
gumption
guntis
hacker
hamlet
handily
happening
harmony
harold
harvey
hebrides
heinlein
hello
help
herbert
hiawatha
hibernia
honey
horse
horus
hutchins
imbroglio
imperial
include
ingres
inna
innocuous
irishman
isis
japan
jessica
jester
jixian

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Passwords Tried by the Internet Worm

johnny	mike	philip	rules	success
joseph	minimum	phoenix	ruth	summer
joshua	minsky	pierre	sal	super
judith	moguls	pizza	saxon	superstage
juggle	moose	plover	scamper	support
julia	morley	plymouth	scheme	supported
kathleen	mozart	polynomial	scott	surfer
kermit	nancy	pondering	scotty	suzanne
kernel	napoleon	pork	secret	swearer
kirkland	nepenthe	poster	sensor	symmetry
knight	ness	praise	serenity	tangerine
ladle	network	precious	sharks	tape
lambda	newton	prelude	sharon	target
lamination	next	prince	sheffield	tarragon
larkin	noxious	princeton	sheldon	taylor
larry	nutrition	protect	shiva	telephone
lazurus	nyquist	protozoa	shivers	temptation
lebesque	oceanography	pumpkin	shuttle	thailand
lee	ocelot	puneet	signature	tiger
leland	olivetti	puppet	simon	toggle
leroy	olivia	rabbit	simple	tomato
lewis	oracle	rachmaninoff	singer	topography
light	orca	rainbow	single	tortoise
lisa	orwell	raindrop	smile	toyota
louis	osiris	raleigh	smiles	trails
lynne	outlaw	random	smooch	trivial
macintosh	oxford	rascal	smother	trombone
mack	pacific	really	snatch	tubas
maggot	painless	rebecca	snoopy	tuttle
magic	pakistan	remote	soap	umesh
malcolm	pam	rick	socrates	unhappy
mark	papers	ripple	sossina	unicorn
markus	password	robotics	sparrows	unknown
marty	patricia	rochester	spit	urchin
marvin	penguin	rolex	spring	utility
master	peoria	romano	springer	vasant
maurice	percolate	ronald	squires	vertigo
mellon	persimmon	rosebud	strangle	vicky
merlin	persona	rosemary	stratford	village
mets	pete	roses	stuttgart	virginia
michael	peter	ruben	subway	warren

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Passwords Tried by the Internet Worm

water
weenie
whatnot
whiting
whitney
will
william
williamsburg
willie
winston
wisconsin
wizard
wombat
woodwind
wormwood
yacov
yang
yellowstone
yosemite
zap
zimmerman

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Password Issues

Password Issues

- Use more than just A-Z.
- Use a password of at least 6-characters
- Avoid actual names or words.
- Choose an unlikely password.
- Change your password regularly.
- Don't write it down.
- Don't tell it to someone else.
- Avoid shoulder-hangers.

Implementation Issues:

- System may actually give away information.
  - Which part of login is incorrect.
  - Which system is being accessed.
- Limit access attempts.
- Enforce password time limits.
- Employ terminal restrictions
- Employ password checking programs.
  - Proactive checkers are best.
  - Ensures adequate password length.
  - Ensures adequate password alphabet (forces the inclusion of capital letters, punctuation, or numbers).
  - Avoids the use of English words.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Authentication Devices

Tokens and Smart cards

- Something you have.
- A token is an object which authenticates its possessor.
- Must be unforgeable and unique.
- Not foolproof since it may be lost or stolen.
- Smart card may compute the response to challenge.
- Smart card may perform encryption.

Personal Characteristic Recognition (Biometric Devices)

- Something you are.
- Retinal scanners.
- Palm/fingerprints.
- Voice pattern recognition.
- Difficult for imposter to duplicate.

Challenge and Response Systems

- Something you have and something you know.
- Passwords are in the clear from time of entry until accepted by host.
- Normal passwords are static.
- Challenge and reply systems create a pseudo one time password system.
- Passwords become dynamic.
- To ensure security:
  - Encryption keys should be changed regularly.
  - Algorithms should be changed occasionally.
- Challenge and reply systems are most appropriate for host-to-host communications because of the computing power available.
- This method affords authentication and identification as well as eliminates the replay problem.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Modem Issues

Automatic Call-Back

- Internal table must be well protected.
- This same technique can be used between two hosts that wish to communicate.

Steps

- User dials a computer system.
- User identifies himself/herself to system.
- System breaks communication.
- System consults internal table.
- System calls back at predetermined telephone number.
- If number specified by user not one of those listed in the computer's directory then a warning is issued to security officer.

Silent Modem

- Carrier tone is suppressed until caller sends the first tone.
- Does not reveal that the telephone line is a modem line.
- No real protection, only forces intruder to take a second step.
- Prevents a computer from dialing randomly in search of another computer.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Login Spoofing

Problem:

- A password grabbing program is malicious software that is left running on a terminal that mimics the normal login prompt.
- After a user enters a login name and password, the program records the name and password and displays the normal incorrect password message and exits.
- The correct system login prompt is displayed and the user logs in again, this time without further problems.
- However, the person that left the spoofing program running can retrieve the login name and password and login under an assumed identity.
- This type of program is a type of Trojan Horse program. Specifically, it is a "spoofing" Trojan Horse program. It is also called a "password grabber".

Solution:

- *The Trusted Path*
- An unforgeable link between the terminal and the system.
- When the trusted path is invoked, all user processes to a terminal are killed and the system trusted path screen or menu is displayed.
- It provides a means where the user can be sure that they are communicating with the REAL system.
- Before logging in, users ALWAYS invoke the trusted path.
- All password management functions, like changing passwords, should use the trusted path.
- As we will see in other sections, other trusted functions should use the trusted path too.

Note:

- Passwords and biometric devices are ONLY good for authenticating the user to the system.

- A *trusted path* is required to authenticate the system to the user.

- I and A consists of both identifying and authenticating the user to the system and identifying and authenticating the system to the user.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Data Access Control

Discretionary Access Control (DAC) is a data access control policy that allows users to grant or deny other users access to their files.

Common implementations

- Permission Bits
- Password Schemes
- Capability Lists
- Access Control Lists (ACLs)

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

DAC

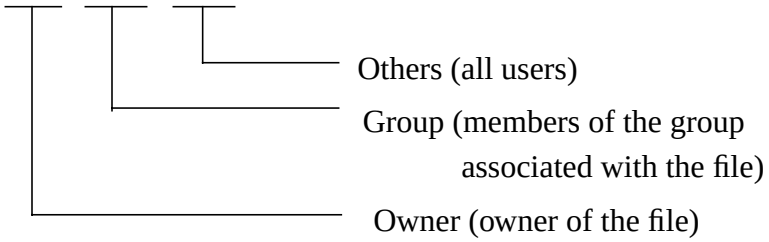
Permission Bits

- Used by Unix, VMS and other systems.
- A user is specified as the owner of each file or directory.
- Each file or directory is associated with a group.
- At any specific time each user is associated with a group.

Example

r w e r w e r w e bits specifying Read, Write or Execute permission

file1 r w - r w - - - - Alice (file owner)
file2 r - e r - e r - e Bob (file owner)
file3 r - - r - - r - - Alice (file owner)



Drawbacks

- Insufficient granularity (how does Alice give ONLY Bob read access to file1?).
- Deny access to a single user? No.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

DAC

Passwords for file / directory access

- A single password for every file.

Example

- file1 password1
- file2 password2
- file3 password3

Drawbacks

- Loss - forgotten.
- Disclosure - loose lips; requires reprotecting the file.
- Revocation - password must be changed and all legitimate users must be notified.
- System Administration nightmare, too many passwords.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

DAC

Capability Lists

General Schema:

- Every object has a unique owner.
- Owner possesses major access rights.
- Owner may declare who has access.
- Owner may revoke access.
- One capability list per user.
- Names all objects user is allowed access to.
- Lists maintained by OS.
- Users cannot access lists directly.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Capability Lists

Example Capability Lists:

Alice's list of capabilities

- file1 (Owner, Read, Write)
- file2 (Read)
- file3 (Execute)

Bob's list of capabilities

- file2 (Owner, Read, Write)
- file5 (Execute)

Trent's list of capabilities

- file3 (Owner, Execute)
- file6 (Read)

Difficulties with Capability Lists Schema:

- Management of large/many lists.
- Revocation of access - must search all user lists to determine if object is on that user's list.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Access Control Lists

General Schema:

- One list for each object.
- Shows all users who have access.
- Shows what access each user has.
- Generally, specifies access based on users and groups.
- Generally, wildcard values are supported to simplify administration.
- Entries are generally listed in order from most specific to least specific and are interpreted in a manner that supports a desired policy. One such policy might be use specific rights over wildcard rights.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Access Control Lists

Example Access Control Lists:

File Alpha

Jones.crypto	rew
Green.*	n
*.crypto	re
.	r

File Beta

Smith.druid	r
.	n

In this example:

- User Jones in group crypto has rew access to file Alpha.
- User Green does not have access to file Alpha.
- All users in group crypto (with the exception of Green) have re access to file Alpha.
- All users, other than Green, have r access to Alpha.
- User Smith in group druid has r access to file Beta.
- No other users have any access to file Beta.

Drawbacks

- Requires a more complicated implementation than permission bits.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

DAC Weakness

Suppose you have a system that:

- correctly enforces an I and A policy,
- correctly enforces a DAC policy,
- stores both Unclassified and Secret information, and
- has both Unclassified and Secret users.
- Also suppose that all Secret users act in accordance with procedures for handling classified information (i.e., they do not set access permissions on files containing Secret information such that Unclassified users can view them).

Question: What can go wrong?

Answer: Malicious software.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

DAC Weakness

Consider the following scenario:

- An unclassified user, Ivan, brings a great Star Trek game into work. The game becomes very popular. Unbeknownst to users the program surreptitiously copies user's files into Ivan's directories with permissions such that Ivan can read them. This type of program is called a Trojan Horse program. It performs a useful function so that users will use it, but it secretly performs other actions.

- How does the program do this? When Alice, a Secret user, runs programs, those programs (text editors, etc.) are able to access all files accessible by Alice, because those programs are running on behalf of Alice.

- When Alice runs the Star Trek program, it too runs on her behalf and can access all files accessible by Alice. Thus, the game program can read all files readable by Alice and make a copies of them into Ivan's directory with permissions on the files set such that they are readable by Ivan.

The gist is, when Alice runs the game program (or any malicious software) it can do any thing that Alice can do.

Conclusion:

DAC mechanisms have an inherent weakness. They are vulnerable to Trojan Horse attacks.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

DAC Weakness

How great is the threat of malicious software?

Consider the following points:

- How much software on your own system did you write?
- How much software on your system can you absolutely vouch for?
- More and more software is written overseas these days.
- It only takes one bad engineer in a group of a thousand good engineers to embed a Trojan Horse in a product.
- If you store information that is worth stealing, the Trojan Horse attack is very attractive
- Are you running a browser that downloads and executes Java applets?

Note:

The users act in accordance with the security policy, it is software that is malicious.

Want to know more?

- A Guide to Understanding Discretionary Access Control in Trusted Systems, NCSC-TG-003

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Section 3

Access Control II

Mandatory Access Control Policies and

Supporting Policies

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Mandatory Access Control Policies (MAC)

Why Do We Need a MAC Policy?

- From Section 2, we know that DAC policies inherently cannot prevent a malicious software (Trojan Horse) attack.
- We need a policy that does address the malicious software problem.
- A MAC Policy is such a policy.

A Mandatory Access Control policy is a policy in which people do not have control over the authorization of people to information.

- Note how this policy differs from a DAC policy.

Within some universe of discourse Mandatory Policies are:

- Global - sensitivity of information does not change relative to its "location" in the system
- Persistent- sensitivity of information does not change from time to time
  - does not state that information is TS on MWF but only C the remaining days of the week

Example MAC Policy

- Military Security Policy

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Mandatory Access Control Policies (MAC)

Mandatory Access Control Policy Definitions

Access Class

- User - Clearance
- Information - Sensitivity
- Clearance and Sensitivity can be mapped to system attributes call *Access Classes*.

Object

- Any passive entity that contains information.
- For the time being, think of this as a file.

Subject

- Active entities operating on behalf of users.
- For the time being, think of this as being associated with a process.

In an implementation of a MAC policy

- Each subject has a label (or access class).
- Each object has a label (or access class).
- The ability of a subject to access an object is based upon a comparison of the subject's label and the object's label.
- Two labels are compared using the "dominance" operator " $\geq$ ".
  - I.e., if label A dominates label B, we write $A \geq B$.
- As an example, consider the set of military classification levels {Top Secret, Secret, Confidential, Unclassified}. Where:
  - Top Secret $\geq$ Secret
  - Top Secret $\geq$ Confidential
  - Top Secret $\geq$ Unclassified
  - Secret $\geq$ Confidential
  - etc.
- Technically Top Secret $\geq$ Top Secret, Secret $\geq$ Secret, etc.

Note

- Object labels and subject labels are a requirement of MAC policy implementations.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Bell and LaPadula Model (BLP)

Bell and LaPadula Model Facts

- The Bell and LaPadula Model is a mathematical description of the DoD Security Policy (a later section discusses the need to have a mathematical description).
- The Bell and LaPadula Model specifies read and write access between a subject and an object based upon the dominance relationship between the subject's label (or access class) and the objects's label (or access class).
- The Bell and LaPadula Model is the most common model for MAC policies.
- Applies only to secrecy (not integrity) of information.
- It includes both discretionary and mandatory access rules
  - Both checks are made upon request for access.
  - We will only look at the MAC aspects of the model since we are using the model to demonstrate a MAC policy.

BLP Mandatory access control

- Lets S be the set of all subjects in a system and O be the set of all objects in a system.
- For each subject *s* in S there exists a label or access class for s called C(s).
- For each subject *o* in O there exists a label or access class for o called C(o).

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Bell and LaPadula Model

Basic Properties of Bell-LaPadula Model
Simple Security Property A subject s may have read access to an object o only if $C(s) \geq C(o)$ (You shall only view objects which are classified at the same level or lower than the level for which you are cleared) **** - Property*** Also called ***Confinement Property*** A subject s may have write access to an object o only if $C(s) \leq C(o)$ (You shall not talk to anyone who is cleared at a level below you)

The first property (The Simple Security Property) is:

- The normal "no read up" policy where
  - Secret users can read Secret, Confidential and Unclassified information (read down allowed)
  - but Secret users cannot read Top Secret (no read up)

The second property (the *-Property, pronounced 'Star Property') is required to prevent malicious software from writing down.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Bell and LaPadula Model

Why the *-Property is needed

- Recall the Star Trek game that contained a Trojan Horse program. If a Secret user uses the program on a system that **does not enforce the *-Property**, the Trojan Horse could read Secret files and write them to Unclassified files, where Ivan (the person who wrote the Star Trek game) (who is an Unclassified user) can read them.

- If, however, a system enforces the *-Property, a Trojan horse cannot write down.

Thus:

- In a computer system, a mandatory policy can protect information in objects from unauthorized access
  even in the face of malicious software.

Restatement of the BLP rules:

- No read up.

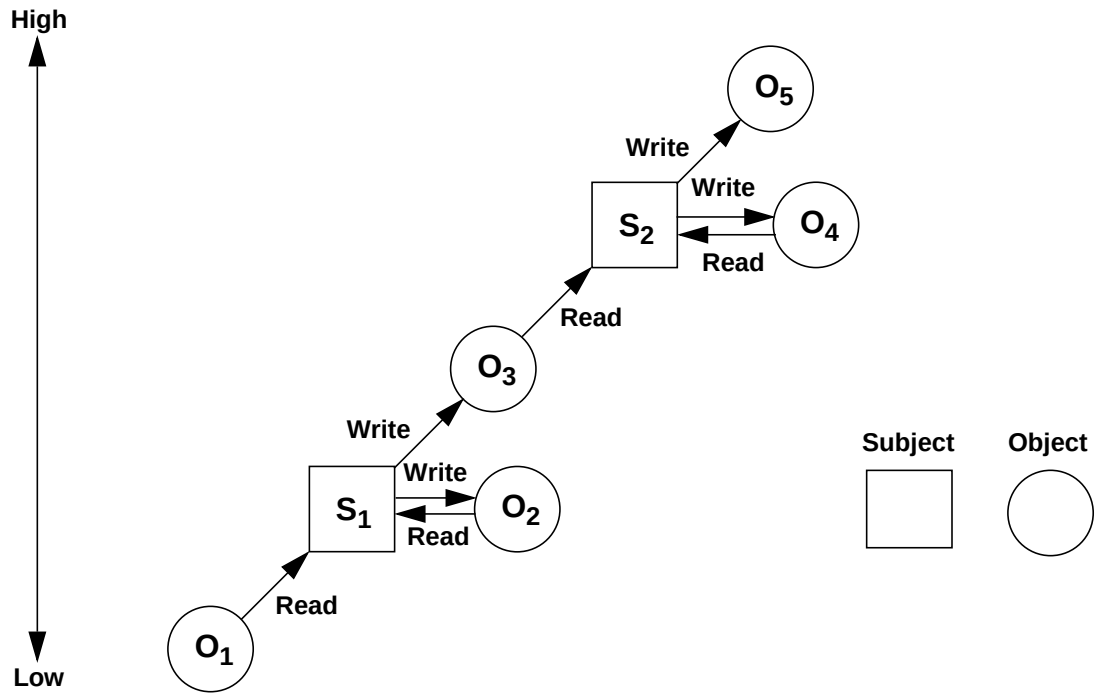
- No write down.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Bell and LaPadula Model

The BLP Model is often described in terms of secure information flows. The Figure below shows such a flow diagram. This is another way of saying that there is "no read up" and "no write down."

Secure Flow of Information (B-LP)



As indicated by the diagram, a subject can only both read and write an object if the object has the same access class value as the subject.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

BLP Example

Consider the following objects and subjects:

- File1 has an access class value of Secret.
- File2 has an access class value of Confidential.
- File3 has an access class value of Top Secret.

- Subject1 has an access class value of Top Secret.
- Subject2 has an access class value of Confidential.

Under the BLP Model the following accesses are allowed:

- Subject1 can read File1, File2 and File3.
- Subject1 can write only File3.

- Subject2 can read File2.
- Subject2 can write File1, File2 and File3.

Can an Unclassified user blindly write to Secret?

- Yes. The model allows it, but most implementations prohibit arbitrary blind write ups.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

MAC Issues

Question:

- How does Alice, a Secret user, write information to an Unclassified file?

Answer:

- Systems that support MAC policies, must also support the notion of a ***session level***.
- When a user logs on they request a session level, which can be any level up to their clearance level.
- If Alice logs on and requests a session level of Secret, a Secret level subject is created on her behalf.This subject can read files at or below Secret and can write files at or above Secret.
- While Alice is logged in, she can re-negotiate a new session level to any other level that she is allowed to operate at. This means if she needs to write and Unclassified file, she must negotiate an Unclassified session.
- Session negotiation should use the trusted path.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Question:

Who puts the access class label values on objects (files)?

Answer:

- When an object (a file) is created (e.g., with a text editor), its access class value is specified as part of the creation process.

- When files are imported into a system (off a floppy disk, from the network, etc.), they are labelled appropriately.
  - If a file is downloaded from an Unclassified network, it is labelled as Unclassified.
  - If a file is downloaded from a Secret network, it is labelled Secret.
  - If a file is imported off an Unclassified Floppy Disk, it is labelled as Unclassified.
  - If a floppy disk contains multilevel data (files at different access class values), then the files on the disk are labelled accordingly and when they are imported into a system the system label value is made the same as the label value of the file on the disk.

Observation:

- Need a consistent set of machinable labels for heterogeneous systems.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Compartments

Use of compartments

- MAC policies often use Compartments in conjunction with access class levels.
- The term category is sometimes used in place of the word compartment.
- Access class levels generally refer to values that are hierarchically ordered with respect to a dominance operator.
- E.g., Top Secret ≥ Secret , Secret ≥ Confidential , Confidential ≥ Unclassfied
- Compartments are not hierarchically ordered values.
- Compartments are set elements where dominance is determined by whether or not a set of compartments is a subset of another set of compartments.

Example:

- Consider a situation where compartment names are fruits.
  - If A = {apples, peaches, apricots} and
  - B = {peaches, apricots} then
  - $A \geq B$ because B is a subset of A.

It is possible to have two sets of compartments C and D, such that C does not dominate D and D does not dominate C.

Example:

- Non-comparable sets of compartments:
  - C = {oranges, apples, peaches} and
  - D = {oranges, peaches, bananas}.

Beware:

- Often DoD usage of the term "need-to-know" to refers to the use of compartments and non-DoD literature often uses the term "need-to-know" to refer to DAC.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Compartments and Levels

Examples:

- Levels = Top Secret, Secret, Confidential
- Compartments = Crypto, Nuclear, Biological, Red, Green

  - Alice is logged in at a session level of "Secret-Nuclear, Red"
  - Tim is logged in at a session level of "Confidential-Crypto, Nuclear, Biological"
  - Anne is logged in at a session level of "Top Secret-Green"

  - File1 has class "Secret-Green"
  - File2 has class "Secret-Red, Green"
  - File3 has class "Confidential-Red"
  - File4 has class "Top Secret-Green"
  - File5 has class "Secret-Nuclear, Red, Green"

Assuming the BLP model of read and write access (where write up is allowed), who has read access to which files and write access to which files?

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Integrity

Note:

- The term integrity is used in two ways in the context of computer security.
- Program or execution integrity refers to a system's ability to provide protected domains of execution.
- Data integrity refers to keeping data free from unauthorized modification.

Secrecy versus Integrity

- Recall from the "Golden Triangle" slide that secrecy and data integrity concerns are distinct.
- Secrecy concerns the prevention of unauthorized disclosure of data or information.
- To re-enforce the orthogonal nature of these concepts, provide examples of the four types of data labeled in the table below:

	High Integrity	Low Integrity
High Secrecy		
Low Secrecy		

Question:

- Where does data integrity fit into a MAC scheme that enforces the BLP Model?

Answer:

- Nowhere.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Biba Integrity Model

Biba Model

- In addition to enforcing a policy for secrecy, we would like systems to enforce a mandatory policy for data integrity too.
- The Biba Integrity Model addresses the unauthorized modification problem by restricting read and write accesses.
- Uses integrity levels and integrity compartments much like sensitivity levels and sensitivity compartments.
- For each subject s in S and each object o in O:
  - Fixed integrity classes I(s) and I(o)
- A high integrity file is one whose contents are created by high-integrity processes.
  - The properties guarantee that the high-integrity file cannot be contaminated by information from low-integrity processes.
  - The high-integrity process that writes the file cannot be subverted by low integrity processes or data.
  - The integrity class label on a file guarantees that the contents came only from sources of at least that degree of integrity.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Biba Integrity Model

Basic Properties of Biba Model
Simple Integrity Property A subject s can modify (have write access to) object o, only if $I(s) \geq I(o)$. (An low integrity subject will not write or modify high integrity data.) **** - Property*** If a subject s can have read access to object o, only if $I(o) \geq I(s)$. (The high integrity subject will not read low integrity data.)

Restatement of the Biba rules:

- No write up.
- No read down.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Biba Example

Consider the following objects, subjects and integrity levels:

- File1 has an access class value of Administrator.
- File2 has an access class value of User.
- File3 has an access class value of Security Administrator.

- Subject1 has an access class value of Security Administrator.
- Subject2 has an access class value of User.

Where

- "Security Administrator" dominates "Administrator"
- "Administrator dominates User"

Under the Biba Model the following accesses are allowed:

- Subject1 can read File3.
- Subject1 can write File1, File2 and File3.

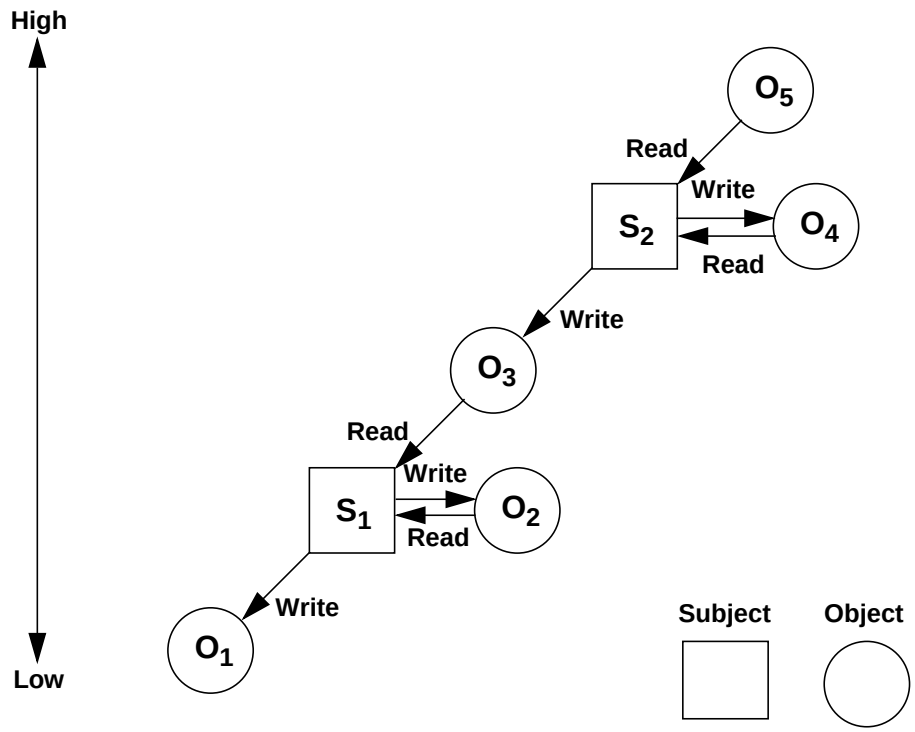
- Subject2 can read File1, File2 and File3.
- Subject2 can write File2.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Biba Integrity Model

The Biba Model is often described in terms of secure information flows. The Figure below shows such a flow diagram. This is another way of saying that there is "no write up" and "no read down."

Integrity Flow (Biba)



As indicated by the diagram, a subject can only both read and write an object if the object has the same access class value as the subject.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Combined BLP and Biba Example

This example demonstrates how read and write accesses are restricted in systems that support the Bell and LaPadula secrecy model and the Biba integrity model.

- Each subject and each object has both a sensitivity label and an integrity label.

- Consider the following sensitivity levels TS, S, C.
- Consider the following integrity levels SecAdmin, Admin, User.

  - File1 TS-User
  - File2 TS-SecAdmin
  - File3 S-Admin
  - File4 C-SecAdmin

  - Subject1 S-User
  - Subject2 S-SecAdmin
  - Subject3 TS-SecAdmin
  - Subject4 TS-User

- Subject1 can read File3 and File4
- Subject1 can write File1
- Subject2, etc.

- The class TS-User is called system-high, because a subject at this class can read every object in the system. An object of this class can be written by every subject in the system.

- The class C-SecAdmin is called system-low, because it can be read by every subject in the system. A subject of this class can write every object in the system.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

MAC Conclusions

Concluding statements about MAC.

- A MAC policy can prevent malicious software (e.g., Trojan Horses) from directly leaking information from high to low.
- Recall that we trust users to not give the store away, but we generally can't say the same thing for software.

- So we build systems that enforce a MAC policy on applications and we don't have to worry about the application software.

  - For example, a subject running at Secret cannot write any information at a level below Secret.

Note that a Trojan Horse can write information between objects at the same security level.

- For example a Trojan horse can read one Secret file and copy it to another Secret file.
- Is this a problem?
- No. Here's why.
- This scenario would require a bad guy (e.g., Ivan) to have a Secret clearance. (So you need personnel security too.)
- He brings in his killer Star Trek game (with an embedded Trojan Horse).
- Sue, a Secret user, plays the Star Trek game and the Trojan Horse copies her Secret files into Ivan's directory. But Ivan is already cleared for Secret information so the Trojan Horse does not get him any information he is not already cleared to see.

- In general, systems that support a MAC policy also support a DAC policy to provide a convenient separation of user's data.

There is still a potential problem with MAC systems.

- Covert Channels can still leak information from high to low in spite of a MAC policy.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Covert Channels

Covert channels are flows of information between access class levels counter to a MAC policy but which are allowed by an implementation.

- Covert channels are a means of leaking information from high to low, one bit at a time.
- If the rate of transmitting bits across the channel (the channel baud rate) is great, this threat is significant.
- Covert channels involve two programs, of which one must be a Trojan Horse. Covert channels are a little complicated to implement.
- However, if information being stored is very valuable, the covert channel threat is real.
- Covert channels come in two varieties. Storage channels and timing channels.
    - Covert storage channels exploit a resource common to both a high subject and a low subject.
    - Automated flow analysis tools can identify every storage channel in a formal specification of a system's interface.
    - Covert timing channels exploit a mechanism where a high subject can affect the timing of low subject.
    - No automatic means exist for identifying every existing timing channel at a system's interface.
    - Timing channels are identified by a examination of the interface.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Covert Storage Channel Example

The classic example of a covert storage channel is the disk exhaustion channel.

- Ivan, (a low user) introduces a Trojan Horse program (e.g., Star Trek game) into the system and somehow gets a high user to execute it.
- When the high user plays the Star Trek game a sub-program is spawned and goes to sleep. The sub-program contains the Trojan Horse and wakes up and starts running at a time when activity on the system is low (e.g., at 0100).
- Ivan starts another program (a low program) that will wake up at 0105, (5 minutes later than the high program). This allows the high program time to initialize the channel.
- The high program finds a high file to copy (fileA).
- The high program initializes the channel by repeatedly creating files until the "disk full" exception is returned.
- The two programs will synchronize with each other by reading a system clock. The high program will signal bits on every even millisecond and the low program will receive bits on every odd millisecond.
- The high program starts reading the bits out of FileA. The following steps are repeatedly performed until the high program is through reading the file.

- The high program does: (on even milliseconds)
  - If a bit is a 0, the high program deletes one file. (Creating room on the disk for a file to be created.)
  - If a bit is a 1, the high program does not delete a file. (So there is no room on the disk to create a file).

- The low program does: (on odd milliseconds)
  - The low program always tries to create a file. If there is room on the disk, the create file call is successful. If the call is successful, the low program writes a 0 into a destination file.
  - If there is no room on the disk, the create file call will fail, with the "disk full" exception. If the call is unsuccessful, the low program writes a 1 into the destination file.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Covert Channels

Storage Channel Example Conclusions:

- The channel baud rate of the previous example is 1 bit every 2 milliseconds.
- This is 500 bits per second, which is 30,000 bits per minute.
- The timing scheme used in the example is very conservative. Much higher baud rates are generally attainable.
- One way to close the disk exhaustion channel is to partition the disk into volumes and allocate each volume to a different security level. For example, volume 0 is for TS files, volume 1 is for S files and volume 2 is for C files.
- Under this partitioning scheme, a C subject cannot tell if the TS volume is full or not. Recall that in the covert channel scenario, the C subject determined if the disk was full by attempting to create a file. Under the partitioning scheme C subjects create files on a separate volume than the TS subjects.

Covert Timing Channels

- Covert timing channels exploit a mechanism where a high subject can affect the timing of a low subject.
- A potential timing channel, which exists on single processor systems, uses the fact that both the high subject and the low subject use the same physical processor.
- To signal a 1, the high subject performs a lengthy operation (e.g., disk I/O) and signals a 0 by performing a short operation.
- When the high subject finishes its operation, the low subject is scheduled to run.
- When the low subject gets scheduled, it reads the system clock and determines how long the high subject operation took.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Multilevel Subjects

Multilevel subjects versus Single-Level subjects

- Up until this point, all subjects in the MAC discussions were single level subjects.
- That is, a subject could both read and write at only one level.
- For example, a Secret subject could only both read and write Secret level objects. (This is required to prevent malicious software from writing high data to low objects.)
- Situations exist (e.g., information downgrading) where a subject needs to be able to both read and write over a range.
- For example, downgrading information from Secret to Confidential would require a subject to read information in Secret objects and write it to Confidential objects.

Thus, MAC implementations must provide some means for multilevel subjects.

- Note that the vast majority of subjects will still be single-level subjects, because multilevel subjects are subject to the Trojan Horse problem.
- Thus, any code that is used in a multilevel subject MUST BE TRUSTED.
- That is, it must be examined to determine that it does not contain a Trojan Horse.
- Often multilevel subjects are call *trusted subjects*.
- Besides downgrading, there are other selected applications that require a multilevel subject.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Applications of Multilevel Subjects

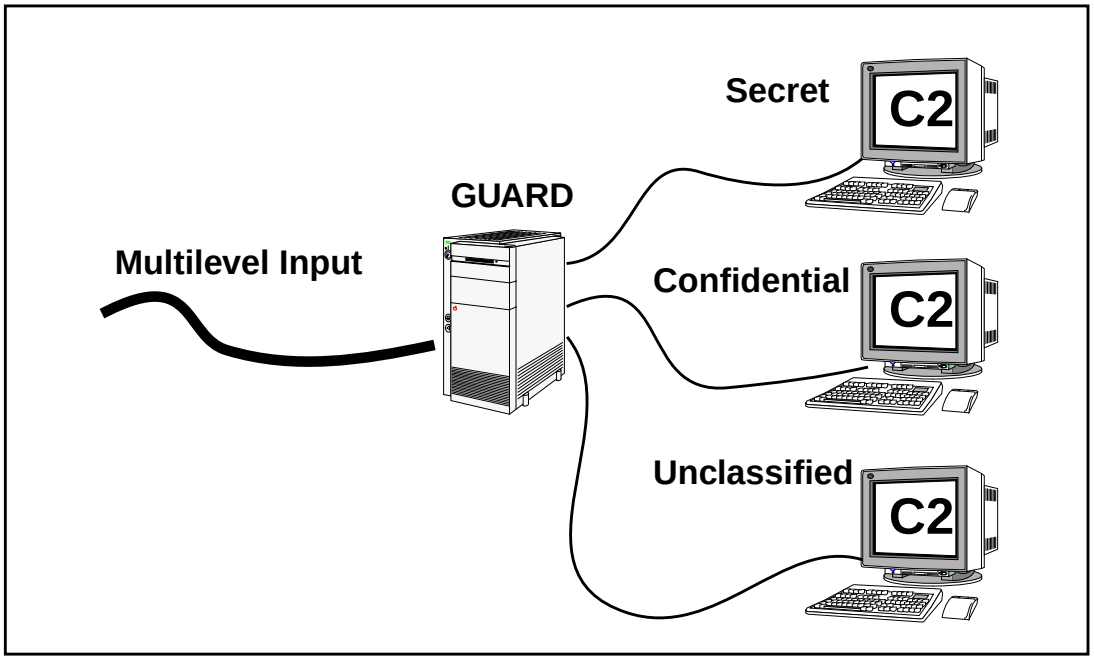
Down grading information

- Need human review - judgement
- Automated sanitization cannot assure that all sensitive information has been removed

Login to a Multilevel System

- users need to set session level
- user has HIGH and LOW range of security levels
- must set read and write class for session

Guards



Guard might consist of four processes:

- one multilevel process, P1 (multilevel),
- three single level processes P2 (S), P3 (C), P4 (U)

- P1 receives labeled information from multilevel input
- P1 inspects label and puts information into a labeled object
- P2, P3, or P4 send information from objects at its level to output system
- Guard is trusted to insure that labeled information is sent to the correct end system.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Supporting Policies

Supplement both mandatory and discretionary security policies

No matter how complex a policy may appear, if sufficient analysis is applied, it will be mandatory, discretionary, or supporting

Identification and Authentication

- associate subjects with users
- authenticate user to system and system to user (trusted path)

Audit - Accountability

Data Consistency Policy

- protects against damage resulting from user or software error
- protects against unauthorized modification or destruction of information
- E.g., Ages are non-negative, department credit card purchases must not be greater than $200.00.

Accountability Policy

- authentication of individuals, thus permitting them to be accountable for the actions
- auditing of individual accesses and access attempts
  - deterrent to misuse
  - detection of security violations

Labeling Policy

- assignment of access labels to information entering and leaving the system
- assignment of access class authorizations to users

Aggregation Policy

- labeling of aggregates more sensitive than individual elements

Sanitization Policy

- release of derived information which is at a lower class than that from which it was derived.
- release of information from aggregates where the individual elements are at a lower class that the aggregate

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Supporting Policies (continued)

Reclassification Policy

- classification changes raising or lowering the sensitivity of information

Applicability of Supporting Policies

Supporting Policy	Mandatory	Discretionary	Dependency
identification and authentication	x	x	classification dependent for I&A on multilevel system
reclassification	x		classification dependent
labeling	x		classification dependent
sanitization	x		classification dependent
aggregation	x		classification dependent
consistency		x	classification independent
accountability	x	x	

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

This Page is intentionally blank

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Section 4

Building Secure Systems

TCBs, Reference Monitors, Protection Domains, Subjects and Objects.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Assurance Versus Policy

Security Policy

- The previous two sections discussed several security policies and supporting policies for computer systems.
- These policies state rules that are enforced by a system's security features.

Assurance

- Assurance, within the context of computer security, is a measure of trust or confidence that a system's security policies are correctly enforced.

Note:

- Security Policies and Assurance are orthogonal. The number and type of policies enforced by a system says nothing about how well the policies are implemented. Assurance addresses the issue of how well a policy is implemented (with respect to correctness of the policy being enforced).

- The amount of effort required to analyze a system's ability to properly enforce its security policies, is dependent on the amount of software, firmware and hardware responsible for implementing the security features of the system.
  - A small amount of software can be analyzed with a reasonable amount of effort.
  - A large amount of software (e.g., an entire Unix operating system) **cannot** be fully understood and analyzed with **any** amount of effort. Large systems are beyond the scope of current analyzing tools and techniques.
- If we want to be able to analyze and fully understand the security features of a system, we either:
  - Build only small systems.
  - Build systems such that the security relevant code is small and separable from the non-security relevant code. Thus, only the small amount of security relevant code needs to be analyzed.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Trusted Computing Bases

Trusted Computing Base Issues

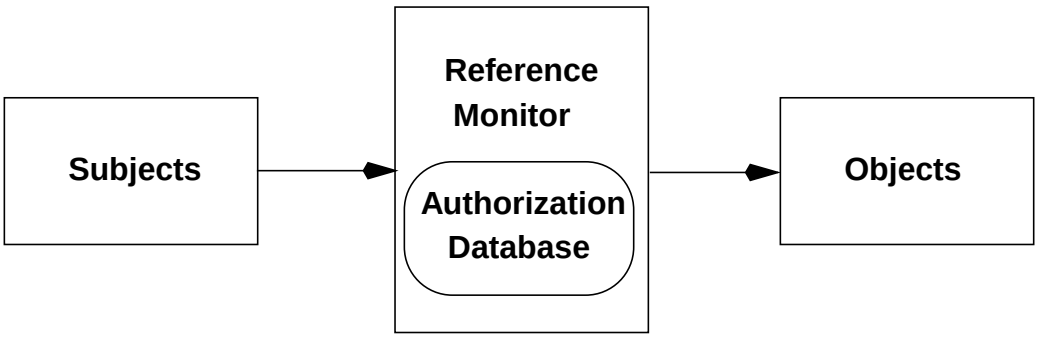
- A Trusted Computing Base (TCB) is the totality of protection mechanisms within a computer system, including hardware, firmware and software. That is, the TCB contains the security relevant software, firmware and hardware.
- An imaginary boundary around the TCB is call the *security perimeter.*
- The TCB contains mechanisms for implementing the various security policies enforced by a system, (MAC, DAC, I & A, Audit, etc.).
- Of these policies the most crucial is MAC (recall that DAC is inherently flawed due to its susceptibility to malicious software).
- Special design and implementation requirements are needed for the portion of the TCB that implements MAC.
- These special design and implementation requirements lead to the *Reference Monitor Concept*.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Reference Monitor Concept

General Schema:

- The Reference Monitor is an abstraction that allows subjects to access objects.
- The Reference Monitor is interposed between all subjects and objects.
- The Reference Monitor makes reference to an authorization database.
- At an abstract level, the Reference Monitor supports two classes of functions:
  - Reference functions - for accessing information
  - Authorization functions - change authorization database



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Reference Monitor

Implementation Requirements

Completeness

- The Reference Monitor must be invoked on every reference of a subject to an object.

Isolation

- The Reference Monitor and its database must be protected from unauthorized alteration.

Verifiability

- The Reference Monitor must be small, well-structured, simple and understandable so that it can be completely analyzed, tested and verified to perform its function properly.

Support Functions

- Reference Monitors often utilize supporting policies.

Identification and Authentication

- identify users to the system - who you are
- authenticate users to system - what you have, know, or are
- reliably identify trusted part of system to users

Audit

- record security relevant operations
- introduction of new objects into a domain
- deletion of objects
- create an **audit trail** composed of **audit records**
  - reference monitor may be source of only some of the audit trail information

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Security Kernel

Security Kernel Facts

- A *Security Kernel* is an implementation of the reference monitor concept.
- It includes hardware, firmware and software.
- It demonstrates
  - Completeness
  - Isolation
  - Verifiability

Conclusion

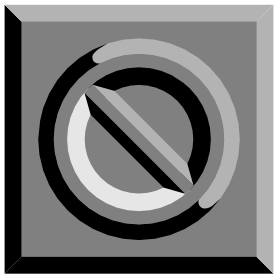
- A system that is built upon a Security Kernel:
  - Lends it self to a tractable analysis to determine *how well the system enforces specific security policies.*

Addressing Computer Misuse

- Don’t need the power of a Security Kernel to address
  - user errors: best countered by user education
  - abuse of authority: countered by audit
  - direct probing: countered by sound management and audit
- Security Kernel particularly suited for addressing several categories of computer misuse:
  - probing with malicious software: countered by MAC policy
  - penetration: countered by high assurance systems
  - subversion: countered by high assurance systems

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Security Kernel from an Existing Operating System?



NO!

Security Kernel Design using Existing Operating System

- Security functions may be diffused throughout system.
- Massive redesign required to
  - isolate security relevant functionality
  - insure modularity
  - insure use of hardware features in support of security kernel objectives

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Determining Assurance

Aspects of Assurance

- Section 11 (Building Secures Systems II) covers several aspects of determining the level of assurance of a particular system.
- The remainder of this section covers:

-- The use of Security Models to help establish assurance.

-- System architecture as it relates to assurance.

Security Model issues:

- A Security Model is a precise and unambiguous statement of a systems security policy.
- A Security Model is an obvious representation of the security policy.
- A Security Model is simple and abstract, and therefore is easy to comprehend.
- An Informal Security Model may be written in formal mathematical notation or in a natural-language.
- A Formal Security Model is written in formal mathematical notation.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Security Models

Two ways a system may be insecure

- flaws in policy
- flaws in implementation

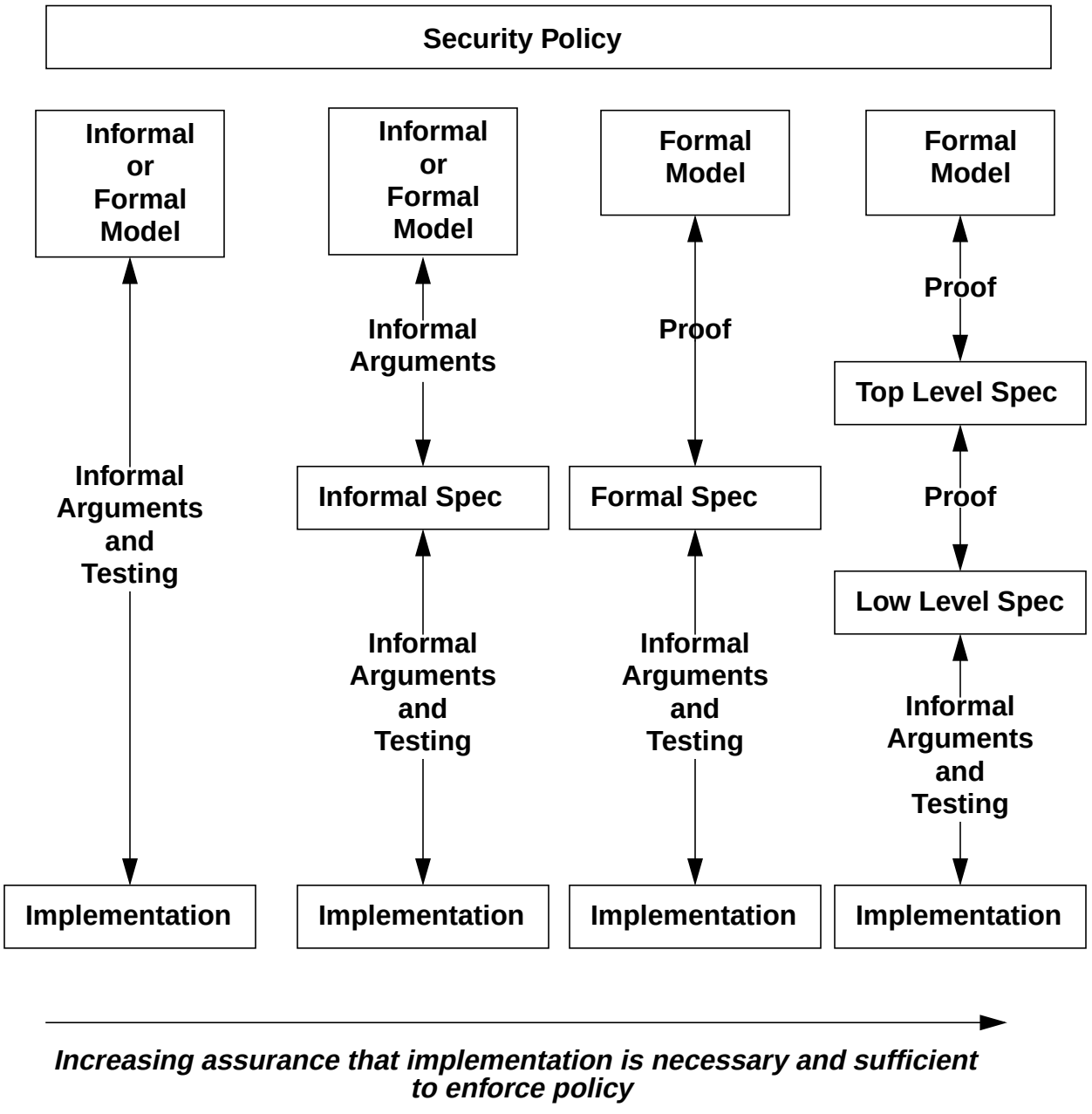
Reasons for using a Security Model

- Security Models can address both issues mentioned above.
- Both Informal and Formal Security Models can be used to establish that a Security Policy is not flawed.
  - We do not want to implement a system that enforces a flawed policy.
  - In the case of Formal Security Models, this proof is made by mathematically showing that the Model is consistent with its axioms.
- Both Informal and Formal Security Models can be used to establish that an implementation faithfully reflects the security policy.
  - An Informal Security Model can be mapped to an implementation or an Informal Specification which can help establish that an implementation faithfully reflects the enforcement of a security policy.
  - A Formal Security Model can be mapped mathematically to a Formal Specification which can help establish that an implementation faithfully reflects the enforcement of a security policy.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Formal Work Improves Verification

Objective:
Demonstrate that the implementation is faithful to the policy.



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

General Characteristics of Security Models

Model constrains design to meet security requirements

- Doesn't constrain how that design might be implemented.
- Guides security relevant behavior of the mechanism.
- Mechanism expressed in a functional specification.

Do you always need a model?
No
Add-ons to an existing system are an example of weak security fixes where modeling would be useless.

Three Major Types of Models

State Machine Models

- State variables represent (security) state of machine.
- Transition functions describe changes to the variables.
- Access Matrix Models are state machine models.
  - Access matrix model shows how matrix changes using transition functions.
- Attribute model shows how security attributes of subjects and objects are compared.

Information Flow Models

- Control flow of information from one object to another.
- Useful for covert channel analysis.

Non-Interference Models

- Subjects operate in different domains and are prevented from affecting each other in ways that would violate the security policy.
- Still active research topic.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

State Machine Modeling Steps

Preliminary Steps

- Define relevant security state variables.
- Define what it means to be in a secure state.
  - (e.g., all Unclassified subjects don't have read access to any Confidential objects.)
  - **This is called the *invariant*.**
- Define the state transition functions.
- Select an initial state for the system.

Proof Steps That Establish the Model is Consistent With its Axioms

- Prove that the initial state is secure.
- Prove that each individual function maintains a secure state (i.e., that they take a secure state to another secure state).

Induction is the basis of the proof.

- Since the initial state is secure and
- since all state transition functions maintain a secure state (take a secure state to another secure state)
- all combinations of transition functions can only result in a secure state.

Conclusion:

If a system starts in a secure state and all transition functions take a secure state to another secure state, the system will always be in a secure state.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Example Access Control Matrix

Abstract Representation of an Authorization Database:

- A table in which:
  - Each row represents a single user (subject).
  - Each column represents an object.
- Access matrix can be represented as a list of triplets to avoid sparse matrix. <*subject, object, rights*>

	Objects							
Subjects	Bibliog	Temp	Test.tmp	Help.txt	C_Compiler	Linker	Sys_Clock	Printer
User_A	RW	RW	RW	R	X	X	R	W
User_B	R	-	-	R	X	X	R	W
User_S	RW	-	R	R	X	X	R	W
User_T	-	-	-	R	X	X	R	W
SysMGR	-	-	-	RW	X	X	RW	-
User_Svcs	-	-	-	-	X	X	R	W

Access Control Matrix

In the example above the rights are

R - read

W - write

X - execute

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Simple State Machine (SSM) Example

More details? See Gasser, *Building a Secure System*

Policy

A person may read a document only if the person's clearance is greater than or equal to the classification of the document.

Policy to Model Translation

Policy terms	Model terms
people/paper world	**computer world**
person	subject
document	object
clearance	access class
classification	access class

Property 1: A subject may read an object only if the access class of the subject is greater than or equal to the access class of the object.

Property 2: A subject may write to an object only if the access class of the object is greater than or equal to the access class of the subject.

- Note that Property 1 is the Bell and LaPadula simple security property and Property 2 is the BLP *-property.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

SSM: Define State Variables

Notation:

- ∅ null set
- { } set notation
- ∪ union
- ∀ for all
- ∈ is an element of
- ’x the value of x in the state after a transition (i.e., in the next state)

S = set of current subjects

O = set of current objects

sclass(s) = access class of subject *s*

oclass(o) = access class of object *o*

A(s,o) = set of access modes equal to one of:

{**r**} subject *s* can read object *o*

{**w**} subject *s* can write object *o*

{**rw**} subject *s* can read and write object *o*

∅ neither read nor write access

contents(o) = contents of object *o*

subj = active subject

state of system at any time is

{*S, O, sclass, oclass, A, contents, subj*}

Define Secure State

Invariant: the system is secure if and only if, ∀ s ∈ *S*, *o* ∈ O,

if **r** ∈ *A(s,o)*, then *sclass(s)* ≥ *oclass(o)*,

if **w** ∈ *A(s,o)*, then *oclass(o)* ≥ *sclass(s)*,

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

SSM: Define Transition Functions

Function	Effect
create_object (o, c)	create object o at class c
set_access (s, o, modes)	set access modes for subject s to object o
write_object (o, d)	write data d into contents of o
create/change_object(o,c)	set class of o to c and create
copy_object(from, to)	copy contents (from) to contents (to)
append_data(o, d)	add data d to contents of o

Functions are defined mathematically and are atomic operations.

create_object *(o, c)*

if $o \notin O$
then $'O = O \cup \{o\}$ and
$'oclass(o) = c$ and
$\forall s \in S, 'A(s, o) = \varnothing$

set_access(s, o, modes)

if $s \in S$ and $o \in O$
and if { [**r** $\in$ *modes* and $sclass(s) \geq oclass(o)$] or **r** $\notin$ *modes*}
and
{ [**w** $\in$ *modes* and $oclass(o) \geq sclass(s)$] or **w** $\notin$ *modes*}
then $'A(s, o) = modes$

Notes

- = means mathematical equality not programming assignment.
- The order of statements not important.
- Transition functions must be atomic.
- If something isn't described in the function then nothing happens to it, everything that changes in the state must be described in the function.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

SSM: Proof of Consistency

Prove Each Transition Function

Invariant and **Function** imply '**Invariant**

***Example:* create_object** proof

$\forall$ s $\in$ *S*, *o* $\in$ O, if **r** $\in$ *A(s,o)*, then *sclass(s)* $\geq$ *oclass(o)*,

if **w** $\in$ *A(s,o)*, then *oclass(o)* $\geq$ *sclass(s)*,

and

if $o \notin O$

then $'O = O \cup \{o\}$ and $'oclass(o) = c$ and $\forall\ s \in S,\ 'A(s, o) = \varnothing$

implies

$\forall$ s $\in$ *'S*, *o* $\in$ 'O, if **r** $\in$ *'A(s,o)*, then *'sclass(s)* $\geq$ *'oclass(o)*,

if **w** $\in$ *'A(s,o)*, then *'oclass(o)* $\geq$ *'sclass(s)*,

Note how the create_object function needs to force nulls in the column of the access matrix for the new object. Needed for the function to be secure.

Define and Prove Secure Initial State

$\{S_0, O_0, sclass_0, oclass_0, A_0, contents_0, subj_0\}$

Simple Initial State

$S_0 = \varnothing$ and $O_0 = \varnothing$

Another Initial State

$\forall\ s \in S_0,\ o \in O_0$, $\text{sclass}_0(s) = c_0$, $\text{oclass}_0(o) = c_0$

$A_0(s, o) = \{\mathbf{r}, \mathbf{w}\}$

Conclusion

After each transition function is proved secure (i.e., each transition function takes a secure state to a secure state) and an initial state is proved secure, the model proof is complete. This means that the model is consistent with its axioms and that the security policy is not flawed.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

SSM: Constraints

Must insure that transitions from state to state are secure

Add constraints - these address values in two consecutive states

- maintain secure relationship between "old" and "new" values
- restrict subjects from invoking certain operations under certain conditions
- control transitions that modify information

Example

change/create_object (*o, c*)

'oclass(o) = *c*; and

if $o \notin O$ then $'O = O \cup \{o\}$; and

$\forall s \in S, 'A(s, o) = \varnothing$

Problem: this function permits the access class of an object to be changed. Information could be downgraded.

Solution: add a new property

Property 3: the access class of an object cannot decrease

We are dealing with a particular type of transition, so add a constraint

constraint: $\forall o \in O$, 'oclass(o) $\geq$ oclass(o)

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Important Models

Computer Security literature often makes reference to the following Security Models:

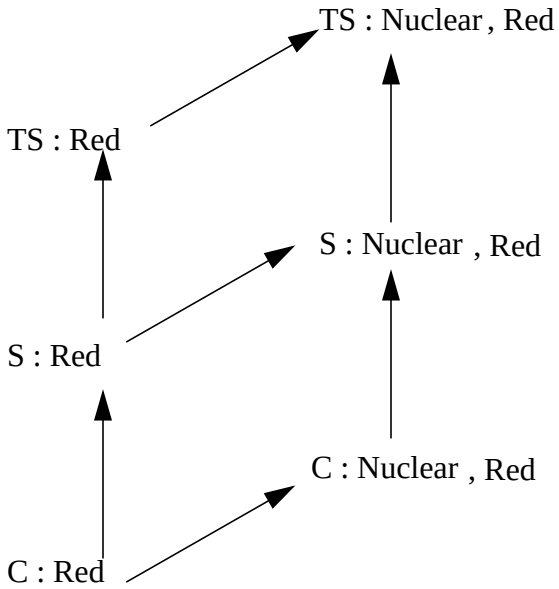
- Lattice Model
  - A lattice model is a generalized model where the elements form a mathematical structure called a lattice.
  - The figure on the following page shows a lattice representation of the legal information flows within the context of military style labels.
  - Many other models satisfy lattice model properties.
  - Models that exhibit lattice properties lend themselves to mathematical analysis.
- Bell and Lapadula Model
  - It can be expressed in terms of a lattice model.
  - It is a confidentiality model.
  - It was the first mathematical model of a multilevel secure computer system,
- Biba Model
  - It can be expressed in terms of a lattice model.
  - It is an integrity model.
  - It is the dual of the BLP model.
- Graham-Denning Model
  - An information flow model.
- Harrison-Ruzzo-Ullman Model
  - An information flow model.
  - A theoretically important model, which facilitates proofs regarding the decidablitiy of subjects gaining rights to objects.
- Clark-Wilson Model
  - It is a commercial model that is transaction oriented.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Example Lattice

Example lattice showing legal information flows in a system that has:

- Secrecy levels of "TS", "S" and "C"
- The compartments "Nuclear" and "Red"



- Notice how information flows from label "A" to label "B" only if label "B" dominates label "A".
- Least Upper Bound (LUB)
  - The LUB of a set of lattice elements (or levels and compartments) is defined to be the "least dominant" element that dominates all elements of the set.
- Greatest Lower Bound (GLB)
  - The GLB of a set of lattice elements (or levels and compartments) is defined to be the "greatest dominant" element that is dominated by each element of the set.
- Examples:
  - The LUB of "S : Red" and "C : Nuclear, Red" is "S : Nuclear, Red"
  - The GLB of "TS : Nuclear, Red" and "S : Red" is "S : Red"
- System-High is the upper bound of all security classes in a system.
- System-Low is the lower bound of all security classes in a system.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

System architecture issues:

System architecture considerations directly address the isolation requirement of Security Kernels. Important specific issues are:

- The use of hardware to support domains of execution.
- The use of hardware to support distinct storage objects.
- The use of Layering, Modularity and Data-Hiding.
  - Layering is the structuring of software into distinct loop-free layers (i.e., layers only call down).
  - This allows software to be analyzed in smaller chunks (one layer at a time) since the correctness of a lower layer is not affected by an upper layer.
  - Modularity is the structuring of software into small understandable single purpose chunks.
  - Data-Hiding is the structuring of data such that it can only be manipulated through a simple high-level well defined module interface.
- The use of the principle of "Least Privilege".

Principle of least Privilege
A subject should have access to the fewest objects needed for the subject to work successfully. (Information is limited by a need-to-know!) *Example*: The system backup program may be allowed to bypass read restrictions on files, but it need not have the ability to modify files. The restore program might be allowed to write files but not read them.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

OS Protection of Memory

An OS may offer protection of memory (system objects) at any of several levels:

- No protection (unrestricted sharing between processes and subjects)
- Isolation (no sharing between processes subjects)
- Restricted sharing between processes and subjects
  - Share via access limitation
  - Share by capabilities

Types of protection:

- Physical Separation
- Temporal Separation
- Cryptographic Separation
- Logical Separation

Hardware can provide support for protection

- registers
- privilege levels
- privileged instructions

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Simple (Early) Protection Schemes

Fences

Fixed Fence

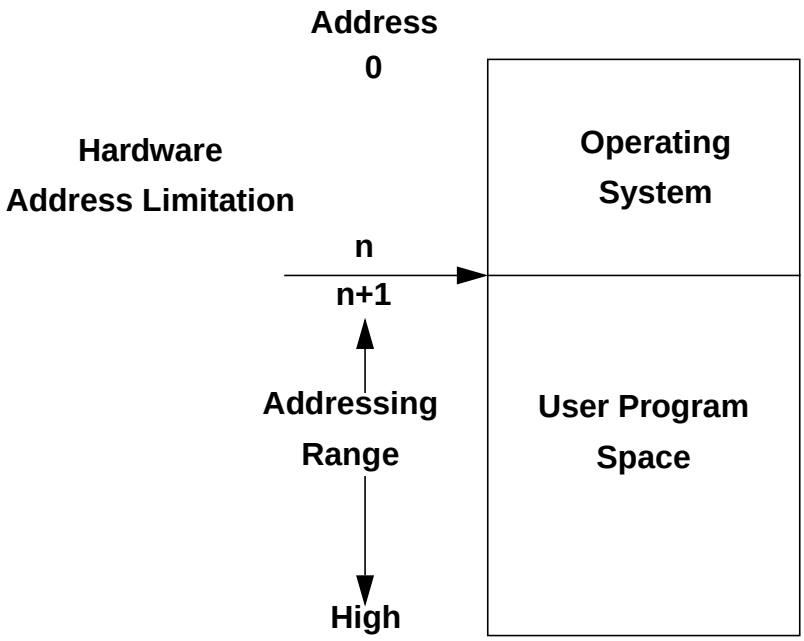
- A method to confine users to one side of a boundary.
- Predefined memory address (fixed).
- Operating system on one side.
- User program on the other side.

Relocation:

- The process of taking a program written as if it began at address 0 and changing all addresses to reflect the actual address at which the program is located in memory.

Fence register:

- Contains address of end of OS.
- Provides means of code relocation.
- Only protects operating system.
- Does not protect one user from another



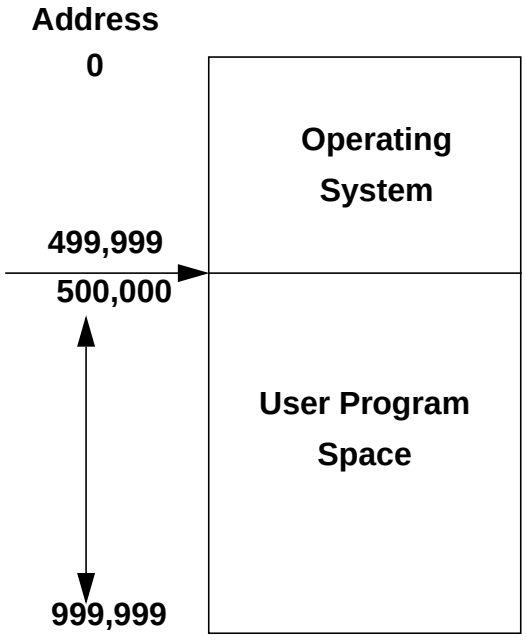
Fixed Fence

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Process / Subject Distinction

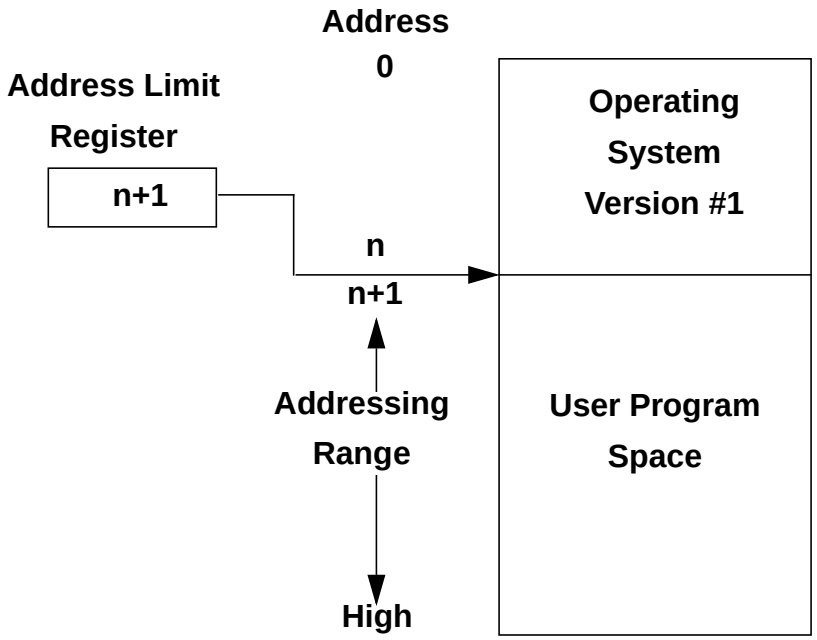
The difference between a Process and a Subject:

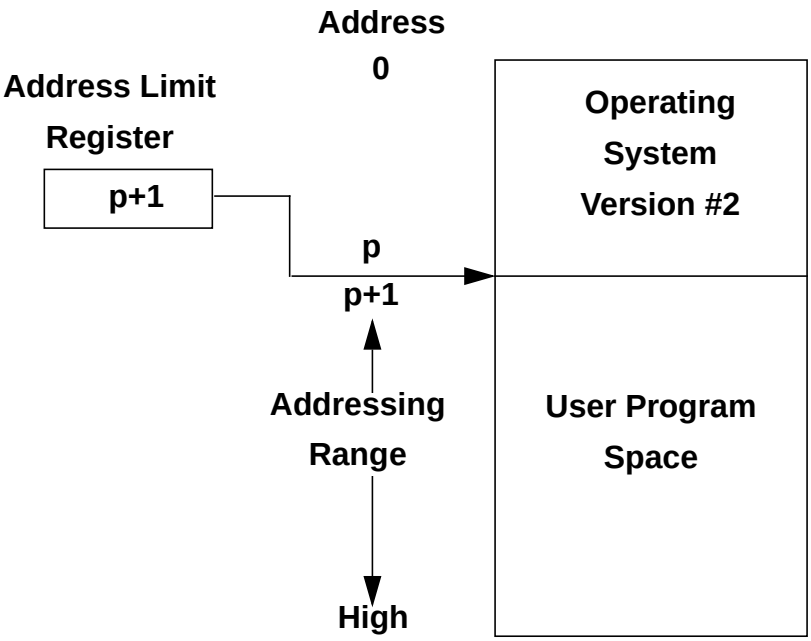
- A Process is a thread of execution.
- A Subject is a Process executing in a domain.
  - A domain is an address space (i.e., the totality of memory locations addressable by a process).
- In the situation shown below there are two domains for a process:
  - The OS domain (memory location 0 to 499,999) and
  - the User Program domain (500,000 to 999,999),
- When process PROC1 is executing a User Program, it is restricted to the memory range 500,000 to 999,999. When the User Program needs an OS service (e.g., writing to a device), PROC1 makes an OS call and PROC1 starts executing in the OS. While PROC1 is executing in the OS, it is restricted to the memory range 0 to 999,999. When PROC1 finishes the requested OS service, PROC1 returns to executing within the User Program.
- The OS subject of PROC1 is when PROC1 is executing in the OS.
- The User subject of PROC1 is when PROC1 is executing a User Program.



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Protection of Memory and Addressing





Variable Fence Register

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Protection of Memory and Addressing

Base/Bounds Registers

Base register:

- All addresses are offset from the base register.
- A variable fence register generally called a base register.

Bounds Register:

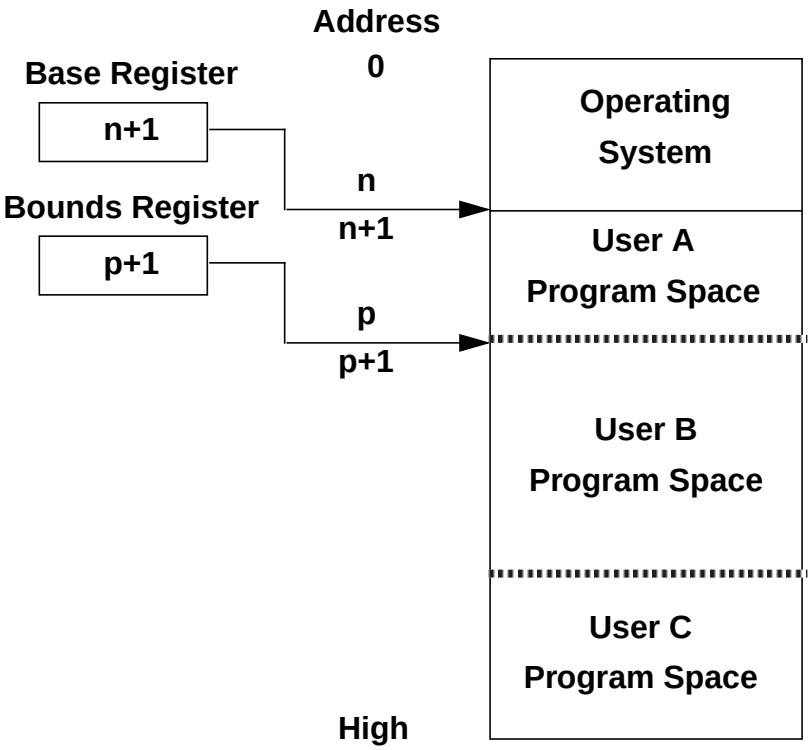
- Provides upper address limit.
- When used with a base register user program is confined.
- Provides one user's program protection from another.
- Does not protect user from himself/herself.
  - Can be achieved through additional registers

Virtual Machine Supervisory Program:

- Generally the only process which can change the contents of these registers.
- Maintains a protected table of all register value pairs (one for each virtual processor).

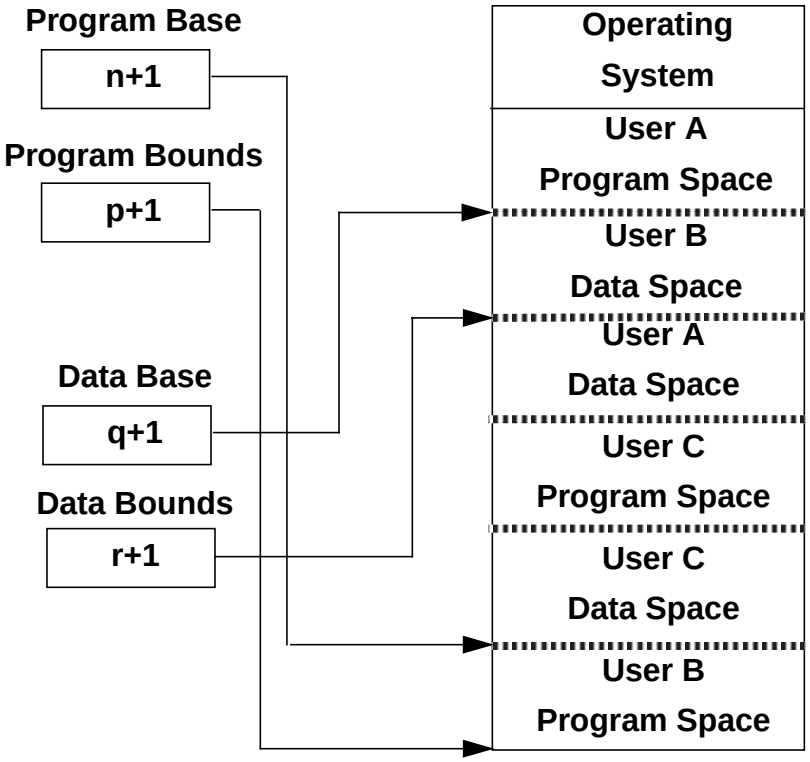
Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Protection of Memory and Addressing



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Protection of Memory and Addressing



Two Pairs of Base/Bounds Registers

- The Program registers specify the range of memory addresses allowed for code.
- The Data registers specify the range of memory addresses allowed for data references.
- A general scheme would also include a memory area for each process stack.
- Given that each process will require a set of three register values, this scheme gets a little complicated.
- A scheme that simplifies the management of many different memory regions (code, data and stack for Process A, code, data and stack for Process B, etc.) is the memory segmentation scheme.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Segmentation of Memory

Segmentation

- Segmentation is the notion of dividing memory into separate pieces, called segments.
- Each segment has a unique name.
- Individual bytes of memory are addressed as a pair <segment name, offset>
- The O/S maintains a table mapping the logical addresses to the physical addresses.
- This scheme has the same effect as an unbounded number of base/bounds register pairs

A form of information hiding:

- The OS can move any segment to any location.
- A segment can be removed from main memory.
- Every address reference passes through the OS.
  - A process which does not have a segment name in its table is denied access to that segment.
  - Handled by combination of hardware and software.

Benefits of segmentation:

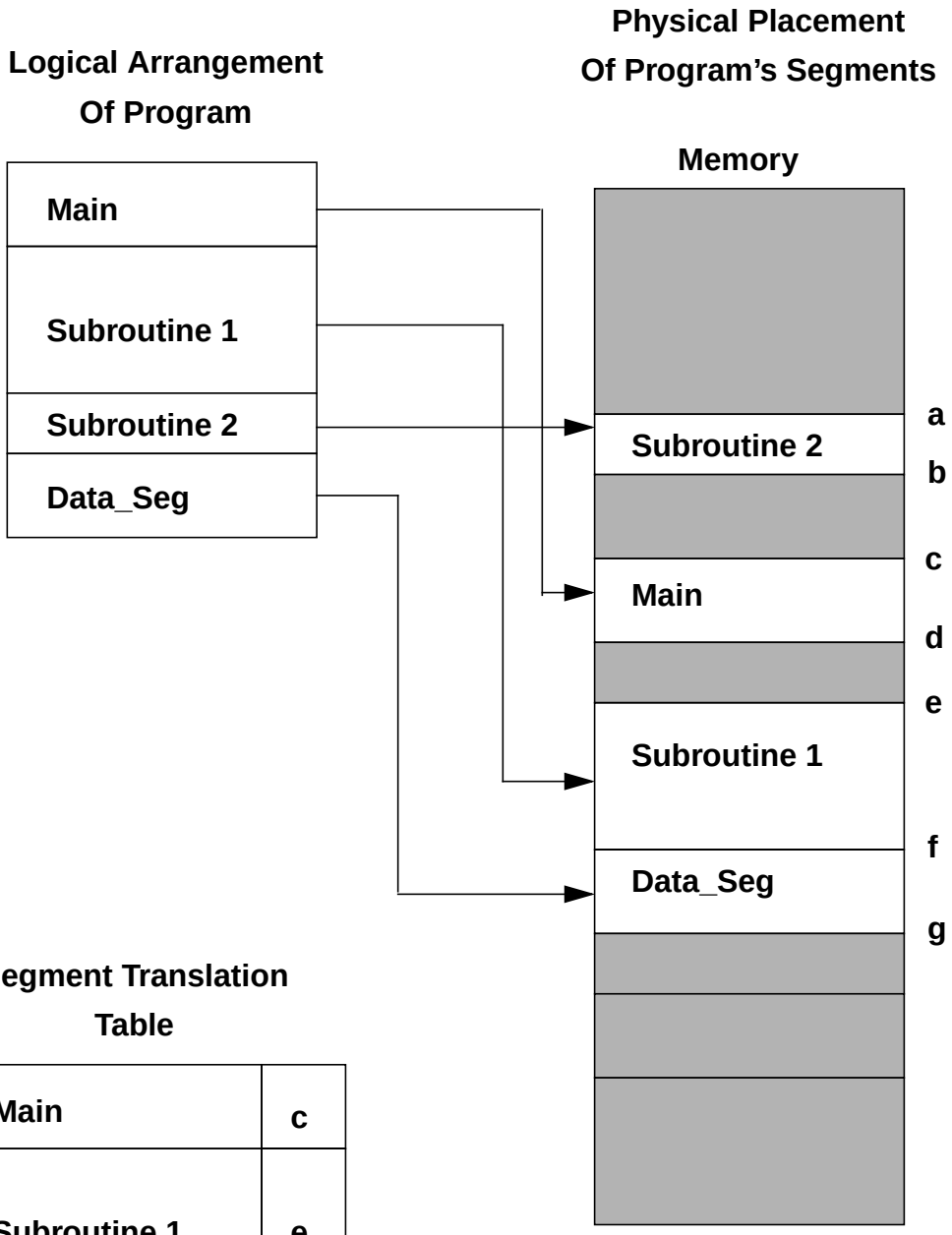
- Each segment can be assigned a different level of protection (e.g., access class label or value).
- Each request for a segment can be checked for appropriate access (perfect for Reference Monitor / Security Kernel implementations).
- Two or more users can share access to a segment, but with different access rights.
- It is impossible for a user to generate an address or gain access to an unpermitted segment.

Inherent problems:

- Segment names are inconvenient to encode.
- Segmentation leads to fragmentation of main memory.
- If swapping is used then additional memory management techniques must be employed (e.g. LRU).

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Protection of Memory and Addressing



Segment Translation Table

Main	c
Subroutine 1	e
Subroutine 2	a
Data_Seg	f

Logical and Physical Representation of Segments

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Segmentation Continued

Use of segmentation:

- Each process has a table of segments that it can access (Intel uses the terminology "Descriptor Table").
- This table specifies the address space of the process.

Paged Segmentation

- Paged memory schemes are convenient since they automatically manage the task of swapping in and swapping out pages of memory as needed by programs.
- Break each segment into equal sized pages.

Advantages of hardware that supports the segmentation of memory:

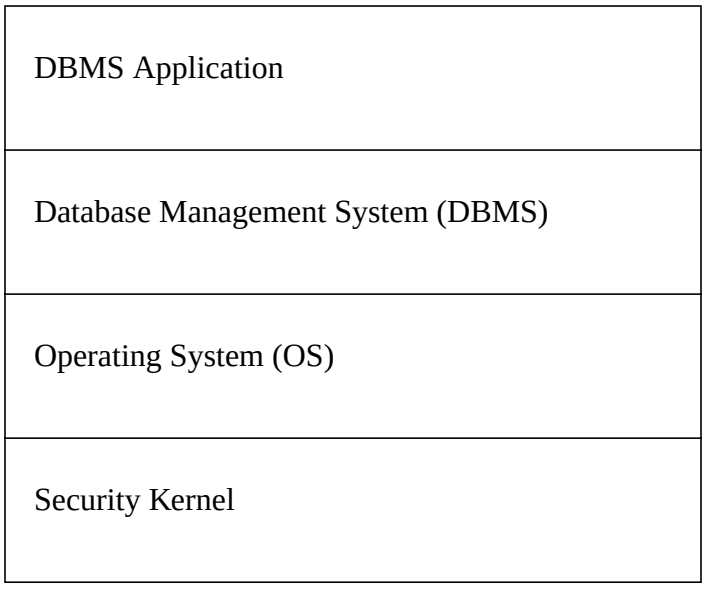
- Supports the isolation of processes by providing a simple means for specifying process address spaces.
- Supports labelling of objects (perfect for Reference Monitor / Security Kernel implementations).
- The next few slides will show how segmentation can support the implementation of execution domains.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Execution Domains

Motivation:

- The Process / Subject discussion a few pages back assumed the existence of two execution domains, the OS domain and the User Program domain.
- In general more than two execution domains are desired to support the principle of least privilege.
- The figure below shows an architecture that uses four distinct domains of execution for each process:
- When a process is running in the "DBMS Application" domain it cannot directly affect any of the code or data in the lower more privileged domains.
  - It can indirectly affect data in the lower domains by making calls to that domain.
  - For example, the Application can ask the DBMS to modify a database table, but it cannot directly affect the table without using the DBMS.



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Execution Domains Continued

Implementation Details of Execution Domains:

Execution Domains can either be implemented in software or hardware.

- Intel x86 (8086, 80286, 80386, 80486 and Pentium) chips support four hardware execution domains.
- Intel uses the terminology "Hardware Privilege Level".
- Multics hardware supported 32 domains.
- Execution domains are sometime referred to as *rings*.
- The term *hardware rings* implies hardware enforced domains and *software rings* implies software enforced domains.
- An important aspect of a ring implementation is the mechanism that allows outer (less privileged) subjects to call into inner (more privileged) subjects.
- This mechanism is called a *call gate*.

Call Gate Issues:

- Call gates (or gates) limit the way the less privileged subjects invoke the services (procedures and functions) of the more privileged subjects.
- Gates specify exactly which entry points of a domain may be called from above. For example, the Security Kernel ring in the previous figure may contain 323 functions and procedures, but the Security Kernel gate only allows the upper subjects to call 27 of them.
- Gates prevent upper subjects from jumping into the middle of procedures and functions in the lower domain.
- Gates can also be used to limit which subjects may call into a domain. For example, in the previous figure, the Security Kernel gate may only allow calls from the OS subject. That is, the DBMS subject cannot call directly into the Security Kernel.
- Gates are also used to validate pointers that are passed into a procedure of a more privileged ring. The gate ensures that all pointer parameters don’t point to any addresses that are part of the more privileged domain.
- Intel x86 chips provide robust gate mechanisms for supporting the four privilege levels.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Why Care about Hardware Privilege Levels?

Hardware Versus Software Implementation

- Hardware is far more efficient than software.
- High Assurance Products Use Hardware Mechanisms.

Hardware Platforms of High-Assurance Trusted Products

Trusted Product	Target Rating	Base Architecture
Boeing MLS LAN	A1	80x86 Multiprocessor
Gemini Trusted Network Processor	A1	80x86 PC or Multiprocessor
Wang XTS 300	B3	80x86
Verdix VSLAN	B2	80x86 Custom Board
TIS Trusted Xenix	B2	80x86 PC

Table from: Sibert et al, *The Intel 80x86 Processor Architecture: Pitfalls for Secure Systems*

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

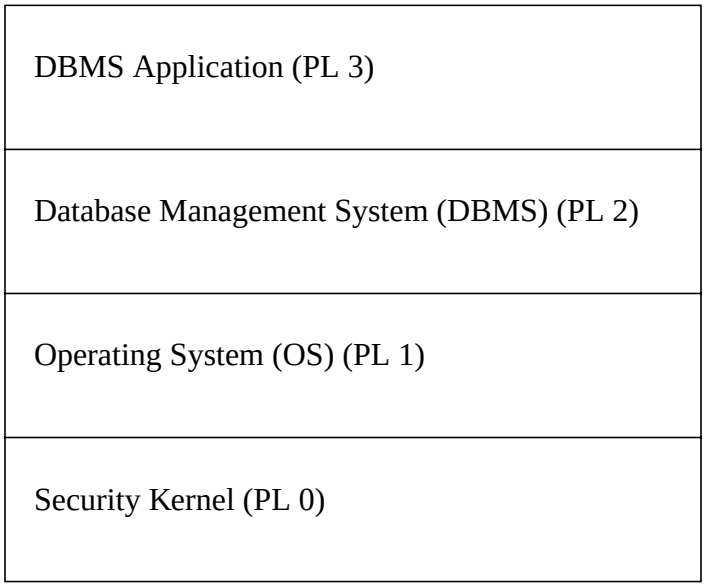
Ring Issues

Software Rings

- With software rings, the Security Kernel creates the ring abstraction, (i.e., it enforces the ring policy).

General Ring Issues

- A process may run in any one of several rings at any one time, moving from ring to ring during execution.
- A process running in a given ring is protected from other processes running in the same ring (process isolation).
  - Recall, a process running in a given ring is called a subject.
- Intel Privilege Levels are denoted as PL 0 through PL 3.
  - See the figure below.
- Ring mechanisms enforce a ring policy.
- A common ring policy is where a subject running in ring i can access all data and functions in ring j, if $i \leq j$.
  - This is the policy enforced by the Intel hardware Privilege Level mechanisms.



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Ring Brackets

Ring Bracket Motivation

- Ring brackets are often part of ring enforcement mechanisms.
- The use of ring brackets allow for a ring policy that is more robust than the policy enforced by Intel hardware Privilege Levels.
  - See the example below.
- A set of ring bracket values are associated with each segment of memory.
- The following example demonstrates a ring bracket policy and encoding.

Example

- Three ring bracket values R1, R2 and R3 are associated with each segment.

  0-R_1 is the write bracket

  0-R_2 is the read bracket

  0-R_3 is the execute bracket

Ring Bracket values:	R_1	R_2	R_3

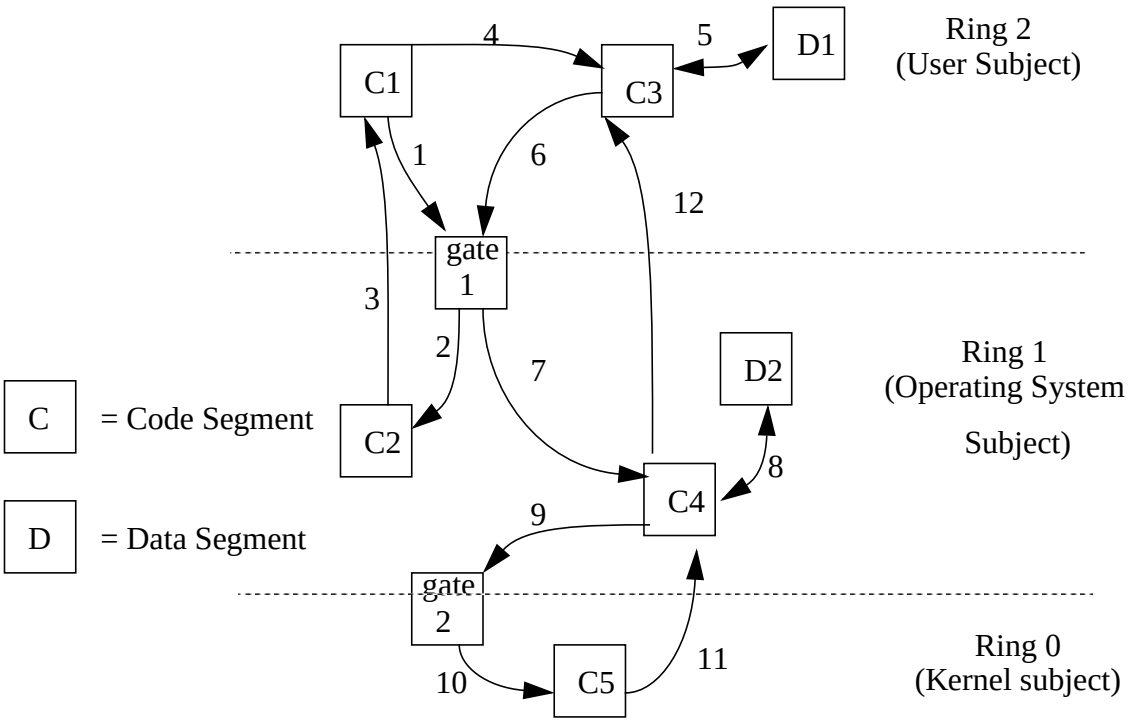
A segment with ring brackets of (4,5,7) is writable from rings 0 through 4, readable from rings 0 through 5, and executable from rings 0 through 7. (it is assumed that $R_1 \leq R_2 \leq R_3$)

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Example of Subjects, Processes and Gates

Single Process, Multi-subject Scenario

- In the figure below, execution starts in the upper left-hand code segment, C1.
- Subsequent processing goes as follows:
- C1 calls a procedure (via arrows 1 and 2) in C2. It needs to go through gate 1.
- The procedure in C2 finishes and execution returns (via arrow 3) to the calling point in C1.
- C1 (via arrow 4) calls a procedure in C3.
- C3 reads and writes data (via arrow 5) from/to D1.
- C3 (via arrows 6 and 7) calls through gate 1 to a procedure in C4.
- C4 reads and writes data (via arrow 8) from/to D2.
- C4 (via arrows 9 and 10) calls through gate 2 to a procedure in C5.
- The procedure in C5 finishes and execution returns (via arrow 11) to the calling point in C4.
- Similarly the procedure in C4 returns to C3 (via arrow 12).



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Design of Secure Operating Systems Overview

Qualities of Secure Systems

1. Security policy - well defined and enforced by system.
2. Identification - every subject must be uniquely identified.
3. Marking - objects labeled for comparison when access requested.
4. Accountability - must maintain complete and secure records.
5. Assurance - must contain mechanisms which enforce security and must be able to measure their effectiveness.
6. Continuous protection - mechanisms must be protected themselves.

Basic Considerations

- Security must be considered in every aspect of the design of operating systems.
- It is difficult to add on security features.

Principles of Design

1. Least privilege - fewest possible privileges for user.
2. Economy of mechanism - protection system should be small, simple and straight forward.
3. Open design - mechanism should be open to scrutiny.
4. Complete mediation - check every access.
5. Permission based - default permission should be denial of access.
6. Separation of privilege - one permission should not give away the entire system.
7. Least common mechanism - avoid shared objects.
8. Easy to use.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

This page is intentionally blank.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Section 5

Malicious Software and Intrusion Detection

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Malicious Software

Greedy programs

- Background task assumes greater priority.
- May not be malicious in nature.
- Infinite loops (are you guilty?)
  - Most systems use time-outs.
  - I/O time not usually checked.

Trapdoor

A secret, undocumented entry point into a module.

- Inserted sometime during code development.
  - Most often debugging hooks.
  - May permit direct change of variables.
  - Produce unwanted side effects.
- Poor error checking.
  - Unacceptable input not caught.
- Most instances are not malicious in nature.
  - Even if not malicious others may utilize it.

Trojan Horse

Performs a hidden function in addition to its stated function.

- Generally distributed as object code along with documentation of overt use.
- Micro users particularity susceptible.
- Can be introduced by binary manipulation (DEBUG).
- Instructions may be scattered with jumps.
- Instructions may be encrypted.

New Threats

- Malicious remote executables.
  - Downloaded Java
  - Agentware
- Really variations on known problems.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Malicious Software

Viruses

S*elf replicating and infectious program.*

- May be relatively harmless or disastrous.
- More prevalent on PCs
  - Do not compile their own source code like mainframes.
  - Swapping of programs.
  - Easier to get infected (opportunity is there).
  - Ignorance.
- Viruses can be categorized by:
  - How do they infect others?
    Overwriting virus?
    Non-overwriting virus?
  - Where do they live?
    Boot infector?
    System infector?
    Application infector?(specific or generic)
- Viral hiding techniques:
  - Self encrypting to avoid detection of the signature.
  - Polymorphic to present a different signature every time.
  - File compressing to be able to hide in files without increasing the size of the host file.

Worm

A program that can run independently and can propagate a fully working version of itself to other machines!

- Does not require the host program to be run to activate it (as is the case for a virus).
- Not all are malicious
  - file compression routines
  - automatic back-up routines

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Viruses

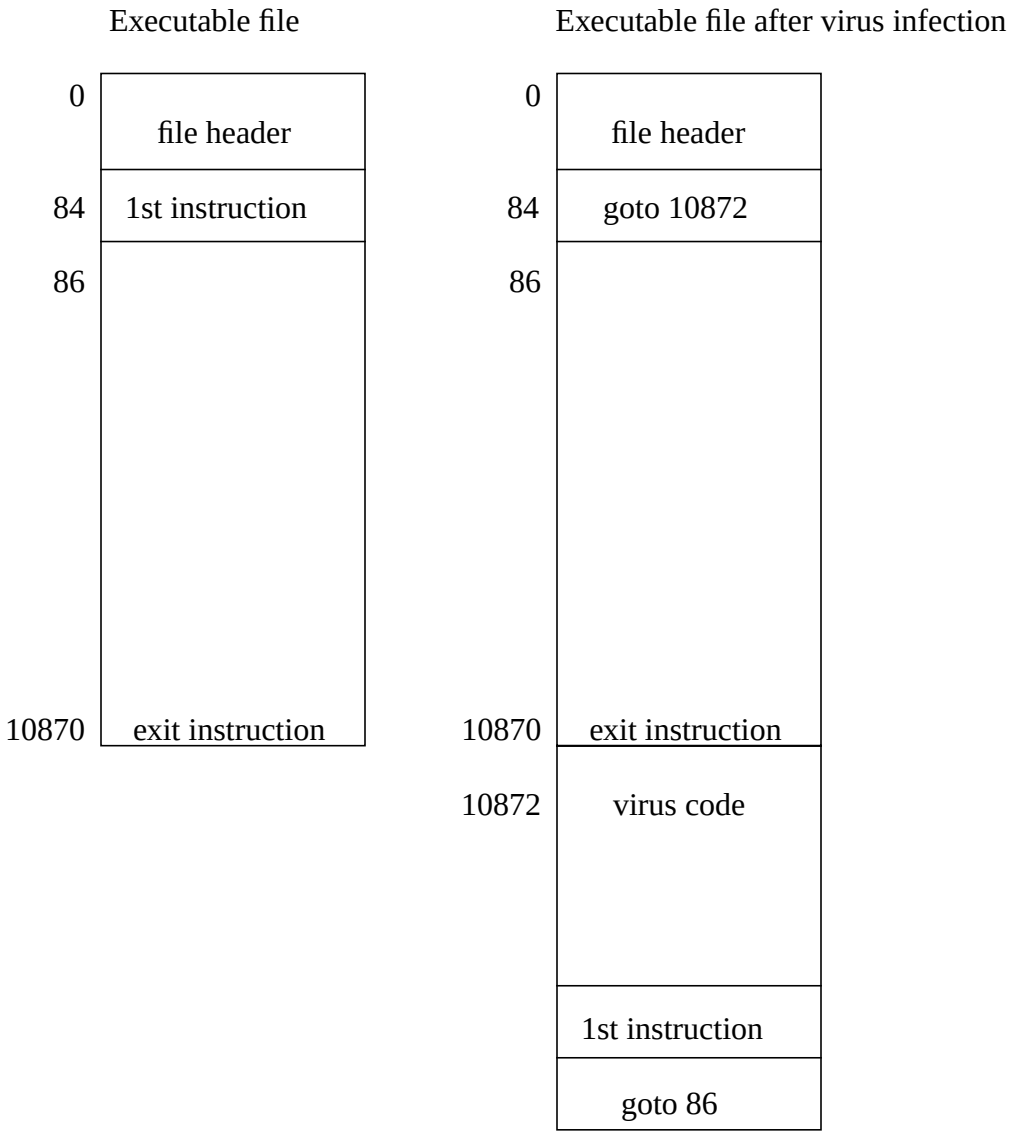
Historical overview

- 1949 - John von Neumann publishes “Theory and Organization of Complicated Automata” a report which dealt with the subject of self reproducing code.
- 1960’s - Bell Lab programmers H. Douglas McIlroy, Victor Vysottsky, and Robert Morris, create an after hours recreation game called “Core Wars” which experiments with code designed to reproduce and gain control of the computer’s core memory.
- “Core Wars” concept becomes a popular pass time at several other industrial and academic research centers but remains a closely guarded secret.
- 1970’s -Several futuristic novels, notably: “Shockwave Rider” by Thomas Brunner and “The adolescence of P-1” by Thomas J. Ryan feature worms and intelligent, information-seeking viruses.
- 1984 - Disclosure of recipe for “Core Wars” virus revealed to general public.
- 1986 - University of Delaware comes under attack by both the Brain virus and the Scores virus.
- 1988 - Princeton University requests assistance to combat *nVir* virus which was attacking their systems. Three hours later Stanford University reported a similar incident. One week later another *nVir* virus attack took place at Oulu University in Finland.
- March 1992 - Michelangelo virus is broadly publicized and overall the general public becomes more aware of the destructive potential of computer viruses.
- MS Word macro virus
  - Really executed
  - CERT advisory

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Viruses

General Virus Infection Scheme



How could this threat be limited or prevented?

- By isolating incoming code so that it cannot modify system code or important applications.
- By using file access control mechanisms (DAC or MAC).
- Antiviral software.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Viral Infection Rates

Viral Infection Models vs. Reality

- A number of researchers have developed models, based upon biological models, in an attempt to describe the infection rate of viral infections.
- These models, which have been around since the late 1980's, suggest that every computer system in the world should be infected with some form of virus today. We know this is not true, so these models fail to account for other factors which reduce the overall effect of viral infections.
- There are several reasons why these models do not accurately model the real world. Two obvious reasons stand out from all the rest:
  1. The models do not accurately model the effect of anti-viral precautions.
  2. The models do not take into account the effect of wide scale virus detection/removal upon discovery of new viruses.

The Best Strategy in the Anti-Viral War
Education of the user community and rigid anti-viral efforts!

Viral Infection Sources (Source: Dataquest, 1992)

- 43% Disk from home
- 25% Don’t know / refuse to say
- 7% Electronic Bulletin Board (EBB)
- 6% Sales demo disk
- 6% Repair or service disk
- 3% Shrink-wrapped application
- 2% Download other than EBB
- 2% Intercompany disk
- 1% Came with PC

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

The Virus Threat

Preventing a Virus Infection:

- Use only commercial software acquired from reliable, well-established vendors.
- Test all new software on an isolated system.
- Make backup bootable installation and recovery disks.
  - Keep disks write-protected during reboot.
- Use virus detectors.
  - Can be configured to run periodically or when new files are imported.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Content Checking Systems

These systems may be associated with firewalls.

The objective of content checking systems is to scan executable content being brought into the system in order to flag executables that potentially contain malicious code.

Can this really work for all cases?

- Can we solve the halting problem? No. We can’t solve this problem for arbitrary code in arbitrary formats either. One would have to make an omniscient system that was able to analyze all possible runs of the executable content.
- The best these systems can do is look for signatures that are characteristic of malicious code. The signatures can be very precise, in which case we are looking for an exact match with a piece of known malicious code (anti-virus software comes to mind here). This will mean that we will permit downloading of malicious executables about which we have little knowledge. There will be a non-negative False Accept Rate.
- We can look for "characteristics" of malicious code. In this case, code that isn’t malicious may be inadvertently flagged resulting in a non-negative False Reject Rate.
- Some agencies really interested in security (A Navy CAPT told us this) permit no attachments containing executable content whatsoever.
  - no Java
  - no MS Word document
  - no etc.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Auditing: Cornerstone of Intrusion Detection

Policy Review

In any enterprise, the control of authorization to disclose or modify information is at the heart of the security policy. To effectively enforce its principle policies, an enterprise may impose two additional **supporting policies**: one for identification and authentication. and the other for auditing. Identification and authentication supports the system in two ways: it insures that only authorized individuals are able to perform work within the system (here used in the most general sense) and it can provide a binding between the individual and the actions that may take place within the system. That binding is crucial for the effective use of the auditing mechanism. Through auditing, we have a trail of evidence that provides accountability for user actions.

The importance of accountability cannot be overemphasized. In corporate settings, accountability insures that individuals (or corporate entities) are held responsible for their actions. A few examples are:

- Terri-the-Teller moves $1000 from your account to his account
- Danni-the-Doctor instructs the radiology system to deliver a lethal dose of x-rays
- Eddi-the-Ensign modifies the targeting coordinates to strike allies instead of enemies

Enterprises will want audit trails that are sufficiently credible to be used in litigation, government inquiries, etc.

- Auditing systems can reveal malicious activity by both insiders and intruders.
  - Attempts to gain unauthorized access or unauthorized privilege
  - Attempts to exercise covert channels
  - Execution of malicious software, i.e. unusual activity associated with resource access or usage

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Where is auditing performed?

Operating system.
Auditing can be inserted into system calls so that their use can be monitored.

Libraries.
Libraries are not in the operating system, but may be used by a variety of applications. The assurance of capture of all relevant activity is not as strong as in the case of the operating system audit trail.

Applications
Applications can be written so that audit records pertinent to the application are collected and stored. For example, a database will collect and maintain an audit trail.

Selective auditing

Auditing of all activity on a system could result in an unmanageable quantity of audit information. Often auditing mechanisms are designed so that the system security administrator can configure the audit mechanism. Configuration parameters might include

- specific functions (e.g. write might be audited, but not read)
- selected individuals (audit everything that Oscar does, but don't bother to audit Alice or Bob)
- selected objects (always audit changes to the password file)
- time of day (run the audit mechanism between noon and 1 pm and on weekends)

How does an auditing mechanism work?
Principle of least common mechanism asks us to consolidate and modularize auditing. If we had audit functionality sprinkled around in the system code, then the mechanism might be incomplete or functionally inconsistent.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

An Audit Service

Several main databases (more granularity might be achieved in a real system)

- audit configuration database
  - stores configuration for selective audit
  - stores whether auditing is on/off
  - managed by security administrator
- audit collection database
  - stores audit records in raw form
- audit reduction database
  - stores configuration information for audit reduction
  - stores reduced audit trail

Writing code for audit

If auditing is "on" other functions call the audit service.

```
begin read( fd, nbytes, buffer)
   variable definitions
        ...
   if AUDIT_ON then
       call audit(read, fd, ...)
   ... rest of code ....
end read
```

Because identification and authentication have created a binding between the person on the outside of the system and the active entity executing code in the system and because the call to audit will run on behalf of that entity, we have a connection through the process identifier back to the actual person making the calls.

How is AUDIT_ON determined?

- Global variable - not very attractive
- Initialization parameter for functional module
- Function call to audit

If we are interested only in saving audit information to record a chain of events to be used for accountability purposes, then the above description of audit is sufficient. What if we want to use audit more dynamically--to catch Oscar while he is conducting unauthorized activity?

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

What should be audited?

The Guide to Understanding Audit in Trusted Systems offers helpful suggestions

Auditable Events

Actions taken within the framework of the system that are security relevant.

Any event that can be selected for inclusion in the audit trail. These events should include, in addition to security-relevant events, events taken to recover the system after failure and any events that might prove to be security-relevant at a later time.

For discretionary only system

- Use of identification and authentication mechanisms
- Introduction of objects into a user's address space
- Deletion of objects from a user's address space
- Actions taken by computer operators and system administrators and/or system security administrators
- All security-relevant events (as defined in Section 5 of this guideline)
- Production of printed output

When system handles multiple security classifications

- Any override of human readable output markings (including overwrite of sensitivity label markings and the turning off of labelling capabilities) on paged, hard-copy output devices
- Change of designation (single-level to/from multi-level) of any communication channel or I/O device
- Change of sensitivity level(s) associated with a single-level communication channel or I/O device
- Change of range designation of any multi-level communication channel or I/O device

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Audit Specifics

Audit information

Specific system information that permits records of auditable events to be recorded.

For discretionary only system

- Timestamp (date/time)
- The unique identifier on whose behalf the subject generating the event was operating
- Action or type of event
- Success or failure of the event
- Origin of the request (e.g., terminal ID) for identification/authentication events
- Name of object introduced, accessed, or deleted from a user's address space
- Description of modifications made by the system administrator to the user/system security databases

When system handles multiple security classifications

- Security level of the object
- Subject sensitivity level

Format of an audit record

Audit trails are just another form of a database.

Table 1: Audit record

Date/Time	Subject ID (sensitivity level)	Type	Success/ Failure	Origin	Object targeted (level)	security database modifications

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Can Audit Go Awry?

Audit mechanisms are not risk free.

They can fail for a number of reasons

Corrupted Audit Administrator

- Trusted administrator logs on as someone else and commits crimes under an innocent person's name.
- Trusted administrator could turn audit off or falsify audit records.

Solutions

- Careful selection of administrator
- Separation of duty: have separate accounts for administrator and audit log administrator
- audit configuration administrator, i.e. what is audited.
- Separation of audit roles

Loss of Audit Data

- like all other system data, the audit trail could be lost if it is stored on media that could be damaged due to power failures, natural disasters, etc.

Solutions

- use reliable technology for storage of audit
- use backups
- store backups in a safe place
- insure that backups are readable
- use redundancy

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Intrusion Detection

Rationale and Objectives

- Provides a deterrent to insider attack - users know that intrusion detection is running
- Provides security staff with a mechanism that informs them of the effectiveness of other security methods in the system.
  - Suppose that the firewall is breached. If you have a really good intrusion detection system, then it might be possible to know that the firewall is weak.
- Permits system administrators to gather information so that they can plan a response to the intrusion.
  - An **incident response capability** should be in place so that administrators have a plan for reacting to an intrusion
  - quick detection can stop an intruder
  - administrators can learn about intrusion techniques

Definition:

- The ability to detect security lapses, ideally while they occur.

Techniques:

- Wait until someone complains - not proactive
- Periodic review of audit logs - may be too late to prevent major damage
  - Audit must be enabled
  - Administrators must review the logs regularly
  - intruders might disable auditing or erase the logs. This is a problem with any mechanism that is not adequately protected.
- Integrity monitoring tools - Tripwire is an example: performs checksums
  - ongoing checks required
  - skilled administrators required
- Intrusion detection software builds usage patterns of the normal system and triggers an alarm any time the usage is abnormal.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Statistical Anomaly Detection

Based upon an attempt to define "normal" user actions, these techniques break down into two varieties:

Threshold detection

"Do that one more time and ....."

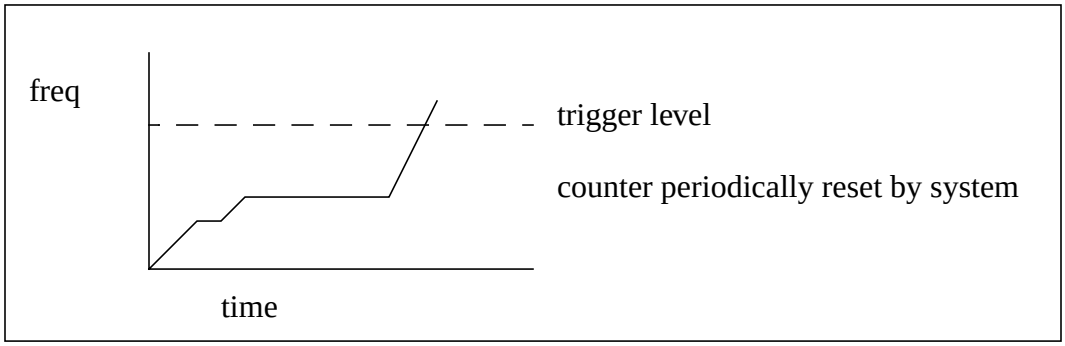
Count the occurrences of some event over a given interval and, if the number is greater than some threshold, send an alarm.

User (group) profiles

Have a description of characteristic activities for a particular user. If the user's activity falls out of range, send an alarm.

Various metrics are applied to audit records:

- counters recording the frequency of specific events



- gauges - rise and fall
- interval timing
- resource consumption

What if the user gradually modifies his/her behavior? Will abusive behavior be detected?

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Determining Average Behavior

Mean and standard deviation
Look for outliers

Multivariate analysis
Look for correlations between variables

Markov model
Transition probabilities between user states

Time series
Is it an automated attack? things happening too quickly?

Operational model
What is normal, what is abnormal (Sam is using the *ls* and *cd* commands too much.)

.

PROBLEM

The major disadvantage to all intrusion detection systems is that they will only find the anomalous behavior that they are programmed to find. Thus if your adversary has an attack that he knows will work, but has been saving for a time of crisis, the defenders will be unaware of this new form of penetration attack

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Rule-Based Intrusion Detection

Anomaly Detection

This technique is a form of pattern analysis. Historical records are used to describe "normal" usage patterns. New audit records are collected and compared with the "normal" pattern. If activity is anomalous, send an alarm.

A large number of rules may be required to characterize "normal" activity.

Penetration Detection

Uses expert system techniques. Known penetrations or "suspicious" activities are encoded into rules. As user behavior is audited, it is compared to the penetration attacks.

UC Santa Barbara system, USTAT, uses a state machine approach. Abstract events are defined and system events are mapped to abstract events on a many to one basis. A series of state transitions characterizing an intrusion can be developed. User activity is compared to these state transition sequences. (It requires far fewer rules than a more general penetration detection system.)

Want to know more? See Masters Thesis of Phil Porras, UCSB.

Disadvantages to the expert system approach

- must design system for a particular platform
- only known attacks can be encoded
- what if several attackers collaborate to penetrate the same system?

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Distributed Intrusion Detection

Challenges

- Different audit trail formats -- audit trail information must be normalized
- Analysis point must receive information from nodes
- Audit information must be protected during transmission for confidentiality and integrity

Centralized or distributed analysis centers can be used.

May use a hierarchy of collection points: host, LAN, central analyzer

- UC Davis developed an elaborate distributed intrusion detection system.
- Emerald team headed by Phil Porras at SRI is conducting research on a hierarchical intrusion detection system

Approaches to Data Analysis

To understand sophisticated attacks it may be necessary to collect vast amounts of information. (NPS had a visitor in the Summer of 1998 who told us that to help a customer deal with an ongoing attack, his computer forensics included collecting all audit and network traffic for analysis. He was filling up hugh GByte disk drives in less than 24 hours.) Once data is collected it must be studied.

Network forensics experts and security administrators are suffering from information overload. If the intrusion information is viewed as a large database, then several emerging techniques may help in intrusion detection and analysis

- database mining
- visualization techniques
  - For example at Battelle NW, visualization experts can display network packet information grouping it according to various components.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Network Management and Security

A number of vendors provide systems for network performance monitoring and management. Now both commercial and government interests are pursuing techniques to provide network performance management combined with security management.

Consider an attack scenario.

Suppose that a network node has several subnets and that an intrusion detection monitoring hub detects an attack on several of these subnets. An appropriate response to the attack might include:

- increasing the monitoring level of the intrusion detection system from routine monitoring to an alert monitoring status. This might entail the collection of additional information, local analysis, revision of triggers, etc.
- reconfiguration of one or more firewalls to prevent traffic from particular addresses
- termination of processes on selected hosts
- additional access control mechanism put into place at servers
- increased auditing at database server
- etc.

How are these responses to be orchestrated?

Security Service Desk Concept

- Network security monitoring hub run by experts.
- In the military context, might have command and control input regarding network security posture.

Components

- Sensory information monitoring mechanisms at selected points throughout network
- Network status displays
- Statement of policy supported by policy server
- Policy changes may result in administrative control information to elements in network

Problem

Protected and guaranteed bandwidth for sensory and control information

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

This page is intentionally blank.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

This page is intentionally blank.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

This page is intentionally blank.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Section 6

Accreditation, Certification and Disaster Planning

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Accreditation

DoD Information Technology Security Certification and Accreditation Process (DITSCAP)

DITSCAP Objective:

To establish a DoD standard infrastructure-centric approach that protects and secures the entities comprising the Defense Information Infrastructure (DII)

Definitions:

Accreditation

Formal declaration by the Designated Approving Authority (DAA) that an IT system is approved to operate in a particular mode using a prescribed set of safeguards at an acceptable level of risk.

Certification

Comprehensive evaluation of the technical and non-technical security features of an IT system and other safeguards, made in support of the accreditation process, to establish the extent that a particular design and implementation meets a set of specified security requirements.

Designating Authority (DAA)

Official with authority to formally assume the responsibility for operation a system or network at an acceptable level of risk.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Operating Modes

- Dedicated Security Mode
  - The mode of operation in which the system is specifically and exclusively dedicated to and controlled for the processing of one particular type of classification of information, either for full-time operation or for a specified period of time.

- System High Mode
  - The mode of operation in which system hardware/software is only trusted to provide need-to-know protection between users. In this mode, the entire system, to include all components electrically and/or physically connected, must operate with security measures commensurate with the highest classification and sensitivity of the information being processed or stored. All system users in this environment must possess clearances and authorizations for all information contained in the system, and all system output must be clearly marked with the highest classification and all system caveats, until information has been reviewed manually by an authorized individual to ensure appropriate classification and caveats have been affixed.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Operating Modes continued

- Multilevel Security Mode
  - The mode of operation which allows two or more classification levels of information to be processed simultaneously within the same system when some users are not cleared for all levels of information present.

- Controlled Mode
  - The mode of operation that is a type of multilevel security mode in which a more limited amount of trust is placed in the hardware/software base of the system, with resultant restrictions on the classification levels and clearance levels that may be supported.

- Compartmented Security Mode
  - The mode of operation which allows the system to process two or more types of compartmented information (information requiring a special authorization) or any one type of compartmented information with other than compartmented information. In this mode, system access is secured to at least the Top Secret (TS) level, but all system users need not necessarily by formally authorized access to all types of compartmented information being processed and/or stored in the system.

- Service policies may define other modes of operation.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

DITSCAP Overview

DITSCAP Establishes:

- a process
- set of activities
- general task descriptions
- management structure

DITSCAP Foundation:

The key to DITSCAP is the System Security Authorization Agreement (SSAA) between

- the DAA
- the CA
- the IT system program manager
- the user representative.

The SSAA is an evolving document. It is refined and augmented throughout the entire DITSCAP process

SSAA Objectives:

- Document the conditions of the C & A for the system
- To guide actions, document decisions, document level of effort, identify possible solutions

The SSAA will reduce redundancy by consolidating the security relevant documentation into one document

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

DITSCAP Phases

DITSCAP is composed of four phases:

- Definition
- Verification
- Validation
- Post Accreditation

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

DITSCAP Phases continued

Phase 1

Definition:
Focuses on understanding the mission, environment and architecture to determine the security requirements and level of effort necessary to achieve accreditation.

Objective:
To agree on the intended system mission, environment, architecture, security requirements, certification schedule, level of effort and resources required.

Phase 2

Verification:
Verifies the evolving or modified system's compliance with the information agreed on in the SSAA.

Objective:
To produce a fully integrated system ready for certification testing.

Phase 3

Validation:
Validates compliance of the fully integrated system with the information stated in the SSAA.

Objective:
To produce the required evidence to support the DAA making an informed decision to grant approval to operate the system (e.g., accreditation).

Phase 4

Post Accreditation:
Includes those activities necessary for the continuing operation of the accredited IT system in its computing environment and to address the changing threats a system faces through its life-cycle.

Objective:
To ensure secure system management, operation, and maintenance to preserve an acceptable level of residual risk.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

DITSCAP Phase Activities

Each DITSCAP phase consists of identified activities:

Phase 1 contains three activities:

- Documenting the mission need, which includes:
  - system mission, functions, and system interfaces
  - operational organization
  - information category and classification
  - expected life-cycle
  - system user characteristics
  - operational environment

- Registration, which includes:
  - informing the DAA, CA and user representative that the system will require C&A support
  - prepare mission description and system identification
  - prepare environment and threat description
  - prepare system architecture description and C&A boundary
  - determine the system security requirements
  - identify organizations that will be involved in the C&A
  - tailor the DITSCAP tasks, determine the level of effort, and prepare a DITSCAP plan
  - develop the draft SSAA

- Negotiation: (Forging an agreement between all involve parties on an implementation strategy to be used to satisfy the security requirements), which includes:
  - review initial SSA
  - conduct certification requirements review
  - approve SSAA

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

DITSCAP Phase Activities continued

Phase 2 contains three activities:

- Refining the SSAA, which includes:
  - at a minimum all participants re-read the SSAA
  - include more specific details regarding the certification effort
  - update with any new system modifications

- System development activity, which includes:
  - develop or integrate the system

- Certification analysis, which includes:
  - system architecture analysis
  - software design analysis
  - network connection rule compliance analysis
  - integrity analysis of integrated products
  - life-cycle management analysis
  - configuration identification procedures
  - configuration control procedures
  - configuration audit procedures
  - trusted distribution plans
  - contingency, continuity of operations, and back-up plans
  - vulnerability assessment

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

DITSCAP Phase Activities continued

Phase 3 contains four activities:

- Refine the SSAA, which includes:
  - ensuring that its requirements and agreements still apply
  - refine with additional details

- Certification evaluation of the integrated system, which includes:
  - Security test and evaluation
  - penetration testing
  - TEMPEST verification
  - validation of COMSEC compliance
  - system management analysis
  - site accreditation survey
  - contingency plan evaluation
  - risk-based management review

- Develop recommendation to the DAA

- Accreditation decision

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

DITSCAP Phase Activities continued

Phase 4 contains four activities:

- Maintenance of the SSAA, which includes:
  - review the SSAA
  - obtain approval of changes
  - document changes

- System operation, which includes:
  - system maintenance
  - system security management
  - contingency planning

- Change management, which includes
  - support system configuration management
  - risk-based management review

- Compliance validation tasks, which includes:
  - physical security analysis
  - review the SSAA
  - risk-based management review
  - procedural analysis
  - compliance reverification

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Vulnerability Assessment / Risk Analysis

Types of risk analysis:

Types based on an Annual Loss Expectancy (ALE)

Quantitative Example:

- Method I of DON AIS Guidelines

Qualitative Example:

- Method II of DON AIS Guidelines

Types based on check-off lists and pre-defined acceptable levels of risk

Examples:

- Naval Computer and Telecommunications Command (NCTS) Trusted Risk Analysis Method
- NAVSO PUB 5239-16 method

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

ALE-Based Risk Analyses

Overview:

Calculate the Annual Loss Expectancy for a system.

Estimate the expected loss to all assets (physical equipment, data, down time, corporate image, etc) during a typical year.

Advantages of ALE-based risk analyses:

- they yield a dollar figure final result (management likes dollar figures)
- they can be used in conjunction with a Return Of Investment (ROI) analysis of safeguards to judiciously indicate where to allocate money earmarked for security safeguards

Disadvantages of ALE-based risk analyses:

- they involve a lot of time and effort
- they are based on many assumptions

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

ALE Calculation

ALE Calculation

The ALE is defined as the total expected loss to all assets in a year.

It can be computed in either one of two ways (both ways should yield the same answer)

One way is to:

- consider each asset A_i
- calculate the annual expected loss for A_i as a result of all threats
- sum all the annual loss expectancies for all assets A_i

The other way is:

- consider a threat T_i
- calculate the annual expected loss to all assets as a result of T_i
- sum all the annual loss expectancies for all threats T_i

Threat/Asset Matrix

	$Asset_1$	$Asset_2$	. . .	$Asset_n$	
$Threat_1$	$(V_1 \times O_1) +$	$(V_2 \times O_1) +$	. . .	$+ (V_n \times O_1)$	ALE_{t1}
$Threat_2$	$(V_1 \times O_2) +$	$(V_2 \times O_2) +$	. . .	$+ (V_n \times O_2)$	ALE_{t2}
. . .	. . .	. . .	. . .	. . .	. . .
$Threat_m$	$(V_1 \times O_m) +$	$(V_2 \times O_m) +$	. . .	$+ (V_n \times O_m)$	ALE_{tm}
	ALE_{a1}	ALE_{a2}	. . .	ALE_{an}	ALE

ALE_t = Annual Loss Expectancy for $Threat_t$
ALE_a = Annual Loss Expectancy for $Asset_a$
V_a = Value of $Asset_a$ (0 to n assets)
O_t = Estimated number of occurrences of $Threat_t$ (0 to m threats)

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

ALE Calculation continued

To calculate an ALE accurately you need:

- to know all the assets of an organization
- to know the value of all assets
- to know all possible threats to all assets
- to know the likelihood of each threat occurring
- how the threats affect the assets
  - (e.g., the occurrence of a 6.0 earthquake may only destroy 10% an organization's assets)

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Asset Identification

- Hardware
- Software
- Data
- People
- Documentation
- Supplies

- The following table represents a partial listing of possible assets.

Data Assets	Communications Assets
Classified	***Communications equipment***
Operations	Communications lines
Tactical	Communications procedures
Planning	Multiplexors
Financial	Switching devices
Statistical	Telephones
Personal	Modems
Logistic	Cables
Other	Local area networks

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Asset Identification continued

Hardware Assets

Central Machine

- CPU
- Main Memory
- I/O Channels
- Operator's Console

Storage Medium

- Magnetic Media
  - Disk Packs
  - Magnetic tapes
  - Diskettes
  - Cassettes
  - Drums
- Non-Magnetic media
  - Punched cards
  - Paper tape
  - Paper printout

Special Interface Equipment

- Network front ends
- Database machines
- Intelligent controllers

I/O Devices

- User directed I/O devices
  - Printer
  - Card Reader
  - Terminals - local and remote
- Storage I/O Devices
  - Disk drives
  - Tape drives

Microcomputer Equipment

- CPU
- Monitor
- Keyboard

Software Assets

Operating systems

Programs

- Applications
- Standard applications
- Test programs
- Communications
- Microcomputer

Personnel Assets

Computer Personnel

- Supervisory personnel
- Systems analyst
- Programmers
  - Applications programmers
  - Systems Programmers
- Operators
- Librarian
- Security officer
- Maintenance personnel
- Temporary employees
- Consultants
- System evaluator/Auditors
- Clerical Personnel

Building Personnel

- Janitors
- Guards
- Facility engineers
- Functional users

Installation Management

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Asset Identification continued

Administrative Assets	Physical Assets
Documentation	***Environmental Systems***
Software	Air-conditioning
Hardware	Power
File	Water
Program	Lighting
JCL	***Building***
System	***Computer Facility***
Operations	Computer room
Schedules	Data reception
Operating guidelines	Tape and disk library
Audit documents	Customer engineer room
Procedures	I/O area
Emergency plans	Data preparation area
Security procedures	Physical plant room
I/O procedures	***Backup equipment***
Integrity controls	Auxiliary power
Inventory records	Auxiliary environmental controls
Operational Procedures	Auxiliary supplies
Vital records	***Supplies***
Priority-run schedule	Magnetic media
Production procedures	Paper
	Ribbons
	Office Spaces

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Asset Valuation

- Hardware
  - What is the replacement cost at current price?
  - How long will it take to replace the system/component?
  - If the work can be done manually, how many more people are required to do the job? How much overtime?
  - If customers contract for services, what are the lost revenues?
- Software
  - How long will it take for a programmer to find the problem?
  - How long will it take to reload and test the program?
  - If it is proprietary software, how long will it take to rewrite the software?
  - If the source code for proprietary software has been disclosed then, what is the probable associated cost?
- Data
  - Can it be replaced?
  - How much will it cost to reconstruct it?
  - Are criminal penalties involved? (police records, tax info, medical info, “Privacy Act” related info)
  - Is the information classified or company confidential? (sales, financial info, product data, weapons research, military operations)
  - Is there a possible loss of life or injury? (life support systems)
- Personnel
  - How many people will have to work overtime?
  - How much will training for the new person cost?

- Difficult to measure
  - Psychological effect (value of customer)
  - Effect of proprietary release (projected sales loses)

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Determining Threats

- Experience, research and imagination all provide help.
  - What are the effects of natural disasters?
  - What are the effects of outsiders?
  - What are the effects of malicious insiders?
  - What are the effects of unintentional errors?

Threats	Threats
Natural	***Intentional Acts***
Earthquake	Bomb threats
Flooding	Disclosure
Hurricane	Employee sabotage
Landslide	Enemy overrun
Lightning	Fraud
Sandstorm	Riot/Civil disorder
Snow/Ice storm	Strike
Tornado	Theft
Tsunami	Unauthorized use
Volcanic eruption	Vandalism
Accidents	
Disclosure	
Electrical disturbance	
Electrical interruption	
Emanation	
Environmental failure	
Fire	
Hardware failure	
Liquid leakage	
Operator/User error	
Software error	
Telecommunications interruption	

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Estimate Likelihood of Exploitation

- Data from general population.
- Observed data for specific system.
- Estimate number of occurrences in a given time.
- Estimate likelihood from table.
- Delphi Approach
  - several raters compare independent estimates
  - revise until consensus
- Factors affecting threat occurrence:
  - geographic location
  - facility environment
  - proximity to population centers
  - data sensitivity
  - protection/detection features
  - visibility
  - proficiency level
  - security awareness
  - emergency training
  - morale
  - local economic conditions
  - redundancies
  - written procedures
  - compliance level
  - past prosecutions

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Determining the Likelihood of Threats

- The following data is based upon nation statistics and is normalized to annual occurrences.

Threat	Occurrence Rate Range	Threat	Occurrence Rate Range
Natural		**Intentional Acts**	
Earthquake	.005 - .2	Alteration of data	.083 - .462
Flooding	.01 - .5	Alteration of software	.00225 - .0125
Hurricane	.05 - .5	Bomb threat	.01 - 100
Landslide	0 - .1	Disclosure	.2 - 5
Lightning	.07 - 50	Employee sabotage	.1 - 5
Sandstorm	.01- .5	Enemy overrun	?
Snow/Ice storm	0 - 10	Terrorist activity	009 - .10
Tornado	.00001 - 2	Fraud	.09 - .5
Tsunami	0 - .125	Riot/Civil disorder	0 - .29
Volcanic eruption	0 - .01	Theft	.015 - 1
Windstorm	.01 - 10	Unauthorized use	.009 - 5
Accidents		Vandalism	.008 - 1.0
Disclosure	.2 - 5		
Electrical interruption	.1 - 30		
Emanation	.1 - 10		
Environmental failure	.1 - 10		
Fire	.001 - .9		
Hardware failure	10 -200		
Liquid leakage	.02 - 3		
Operator/User error	10 - 200		
Software error	1 - 200		
Telecommun. failure	.5 - 126		

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Normalizing the Likelihood of Threats

- Frequency of occurrence is normalized based on annual occurrence:

Frequency		Value
Never		0.0
Once in 300 yrs.	1/300	.00333
Once in 200 yrs	1/200	.005
Once in 100 yrs	1/100	.01
Once in 50 yrs	1/50	.02
Once in 25 yrs	1/25	.04
Once in 5 yrs	1/5	.2
Once in 2 yrs	1/2	.5
Yearly	1/1	1.0
Twice a year	2/1	2.0
Once a month	12/1	12.0
Once a week	52/1	52.0
Once a day	365/1	365

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

ALE Example (per asset)

Example

$Asset_1$ = Data Center

Threat = Electrical Power Surge

Cost of incident = $100,000

Event frequency is three (3) times per year

$$ALE_{1,1} = \$100,000 \times 3 = \$300,000$$

$Asset_1$ = Data Center

Threat = Earthquake

Cost of incident = $1,500,000

Event frequency is once every two years

$$ALE_{1,2} = \$1,500,000 \times .5 = \$750,000$$

$Asset_1$ = Data Center

Threat = Flood

Cost of incident = $3,000,000

Event frequency is once every 10 years

$$ALE_{1,3} = \$3,000,000 \times .10 = \$300,000$$

ALE for Asset #1

$$ALE = ALE_{1,1} + ALE_{1,2} + ALE_{1,3}$$

$$ALE = \$300,000 + \$750,000 + \$300,000 = \$1,350,000$$

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

ALE Example (per threat)

Example

Asset = Server

Threat$_1$ = Electrical Power Surge

Cost of incident = $10,000

Event frequency is three (3) times per year

$$ALE_{1,1} = \$10,000 \times 3 = \$30,000$$

Asset = Disk Drive

Threat$_1$ = Electrical Power Surge

Cost of incident = $1,000

Event frequency is three (3) times per year

$$ALE_{2,1} = \$1,000 \times 3 = \$3,000$$

Asset = Data Center

Threat$_1$ = Electrical Power Surge

Cost of incident = $100,000

Event frequency is three (3) times per year

$$ALE_{3,1} = \$100,000 \times 3 = \$300,000$$

ALE for Threat #1

$$ALE = ALE_{1,1} + ALE_{2,1} + ALE_{3,1}$$

$$ALE = \$30,000 + \$3,000 + \$300,000 = \$333,000$$

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Return Of Investment (ROI)

- For each safeguard identified:
  1. Identify those vulnerabilities which may be reduced by implementation of the control.
  2. Assign an effectiveness rating for each event/control pair.
  3. Estimate the annual cost of implementing the control.
  4. Calculate the Return on Investment (ROI).

$$ROI = \frac{r_k \times ALE_t}{C_k}$$

C_k = Annual cost for $Control_k$
r_k = Effectiveness rating of $Control_k$
ALE_t = ALE of $Threat_t$

Basis for Selection of Addition Safeguards
• Greatest ROI
• Minimized ALE

Calculation of ROI

$$ALE = ALE_{1,1} + ALE_{2,1} + ALE_{3,1}$$

$$ALE = \$30{,}000 + \$3{,}000 + \$300{,}000 = \$333{,}000$$

Threat = Electrical Power Surge

Control = Surge Suppressors (100)

C_k = \$50 x 100 = \$5,500

r_k = .70

$$ROI = \frac{(0.70 \times 333,000)}{(5,500)} = \frac{233,100}{5,500} = 42 : 1$$

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Quantitative ALE Schemes in General

- The method we have just described is the general quantitative approach.

- Method I in the DON AIS Guidelines is a quantitative ALE scheme

- Fundamental problems with the quantitative method:
  - Difficult to find good numbers for threat frequencies.
  - Difficult to estimate the intangible value of an asset, in particular the “availability” of the information the system was designed to provide.
  - Methodology is essentially incapable of discriminating between low-frequency high-impact threat events (fires) and high-frequency low impact threat events (operator error).
  - Inherent subjectivity of the numbers involved.
  - Labor intensive, time consuming and therefore costly

The Plain Fact Is:
A *truly* quantitative method has not yet been developed!

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Qualitative ALE Schemes

- Rather than using pseudo-exact numbers, the qualitative approach uses even fuzzier metrics for asset values, threat frequencies, and control effectiveness:
  - High, Medium, Low
  - One, Two, Three (1,2,3)
  - Vital, Critical, Important, Convenient and Informational

- Advantages
  - Less labor intensive
  - Less Time consuming

- Disadvantages
  - Hard to get support for something with an associated term like “very important” to management
  - Numbers are even more subjective

- Method II in the DON AIS Guidelines uses a qualitative risk analysis approach

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Qualitative Values

Example Value tables used in Qualitative Analysis

Financial Loss Table

Financial Loss	Score
Less than $2,000	1
Between $2,000 and $15,000	2
Between $15,000 and $40,000	3
Between $40,000 and $100,000	4
Between $100,000 and $300,000	5
Between $300,000 and $1,000,000	6
Between $1,000,000 and $3,000,000	7
Between $3,000,000 and $10,000,000	8
Between $10,000,000 and $30,000,000	9
Over	10

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Other Risk Analysis Techniques

Techniques base on check-off lists and predefined acceptable levels of risk

Overview

- When doing an ALE-based risk analysis, much of the work (lists of assets, list of threats, likelihood of threats, etc) will be similar between organizations or sites.
- Methods based on check-off lists attempt to eliminate redundant work by providing a fairly extensive list of assets and vulnerabilities that are common to most enterprises.
- These lists need to be tailored to meet the precise needs of any particular site, but the amount of effort required to tailor them is relatively small.
- In addition to just providing canned lists, these techniques also identify pre-defined criteria for acceptable levels of risk.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

TRAM

Naval Computer and Telecommunications Command (NCTS) Trusted Risk Analysis Method (TRAM)

TRAM philosophy

Make all systems TCSEC (Orange book) C2 compliant.

Compliance can be met by combinations of any of the following four controls:

1. operationally, by having C2 evaluated equipment
2. operationally, by using certified add-on software or hardware to satisfy C2 requirements
3. environmentally, by using environmental controls to satisfy C2 requirements
4. procedurally, by using procedural controls

An example of 2 above:
A system that does not enforce a user Identification and Authentication (I&A) policy may satisfy the I&A requirement if a certified add-on I&A package is installed.

An example of 3 above:
A system that does not enforce a user Identification and Authentication (I&A) policy may satisfy the I&A requirement if access to the system is restricted to only one authorized user by environmental controls (e.g., locked office door).

The TRAM includes checklists that cover all aspects of a system that are required for the system to meet the TCSEC C2 criteria. It readily identifies safeguards that are required to bring all systems up to C2

The TRAM must be custom tailored to work on systems that need more than C2 requirements

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

NAVSO PUB 5239-16

NAVSO PUB 5239 contains INFOSEC Program Guidelines.

Module 16 of 5239 is the Risk Assessment Guidebook.

This document provides:

- procedures for performing cost-effective risk assessment on stand-alone systems, Local Area Networks (LANs), Wide Area Networks (WANs).

The method is applicable to dedicated, system high and multilevel modes of operation and unclassified, classified and sensitive but unclassified (SBU) information.

The check-off lists can be used to perform any of the following four levels of risk assessment

- Survey
- Basic
- Intermediate
- Full

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

NAVSO PUB 5239-16

The check-off lists are very extensive.

Example entries in the check-off list for Network Auditing Services

Network Auditing Services	Assessment	Comments
Network monitoring services		
Network monitoring features forward abnormal "indicators" to the network Monitor Center for review and disposition		
"Failed" log-in attempts		
Unauthorized access attempts (e.g., ungranted privileges(s))		
Computational resource threshold(s) reached		
Actual or suspected (malicious or not) penetrations attempts		

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Evaluation Issues

Important Evaluation Criteria

These criteria do not specify or address how to implement the required security features.

NCSC Criteria

- Trusted Computer System Evaluation Criteria (TCSEC)
- a.k.a. Orange Book
- a.k.a. Criteria
- Ties assurance with features.
- More on following slides.

Common Criteria

- Signed by participating countries in September 1998.
- Uses protection profiles.
- Attempt to harmonize criteria of EC, US, and Canada.
- Highly flexibility is an objective.
- Threat perspective.
- Considered by some to be dangerous because of arbitrary mix of features and assurance.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

TCSEC Issues

The TCSEC defines four basic divisions; A, B, C and D.

- Class **A** designates the highest level of assurance of policy enforcement.
- Within a division, numbers are used to designate a finer distinction of levels, (i.e., **B1**, **B2**, **B3**).
- A greater number indicates higher assurance.
- Classes **C** through Class **B1** might be add on measures to existing operating system
  - Division **D** is failure
- At Class **B2** and above security must be included in system design.
- Class **A1** systems subjected to formal methods.
- **TCSEC class requirements are cumulative.**

The TCSEC analysis of systems is divided into four requirements areas:

- Policy
- Accountability
- Assurance
- Documentation

Each requirements area is divided into a number of finer requirements.

- Lower assurance systems (e.g., C2) must satisfy a specific set of these requirements.
- Higher assurance systems (e.g., B2) must satisfy a larger specific set of these requirements.
- The Highest assurance systems (A1) must satisfy all of these requirements.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

TCSEC Requirements Chart

Security Policy
Discretionary Access Control
Object Reuse
Labels
Label Integrity
Exportation of Labeled Information
Labeling Human Readable Output
Mandatory Access Control
Subject Sensitivity Labels
Device Labels
Accountability
Identification and Authentication
Audit
Trusted Path
Assurance
System Architecture
System Integrity
Security Testing
Design Specification and Verification
Covert Channel Analysis
Trusted Facility Management
Configuration Management
Trusted Recovery
Trusted Distribution
Documentation
Security Features User's Guide
Trusted Facility Manual
Test Documentation
Design Documentation

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

TCSEC Requirements

Security Policy

Discretionary Access Control

Object reuse

- When a storage object (page frame, disk sector, magnetic tape, etc.) is initially assigned, allocated or reallocated to a subject, the TCB will ensure that the object contains no residual data.

Labels

- This is a requirement for labels (sensitivity or integrity) associated with each system resource (e.g., subject, object).
  - Label Integrity - Exported labels shall accurately reflect internal labels.
  - Exportation of Labeled Information - I/O devices will be labeled either single-level or multilevel.
- Labeling of Human Readable Output - The TCB will mark human readable output.

Mandatory Access Control

Subject Sensitive Labels

- The TCB will notify each terminal user of each change in the security level associated with the user.

Device Labels

- Minimum and maximum security levels will be assigned to all attached devices.

Accountability

Identification and Authentication

Audit

Trusted Path

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

TCSEC Requirements

Assurance

System Architecture

- The TCB shall maintain a domain for its own execution that is protected from tampering. It shall be internally structured in well-defined largely independent modules. It shall make effective use of available hardware to separate those elements that are protection-critical from those that are not.

System Integrity

- There shall be features that can be used to periodically validate the correct operation of the hardware and firmware elements of the TCB.

Security Testing

- The security mechanisms shall be tested.

Design Specification and Verification

- Formal or informal models shall be used to verify system correctness.

Covert Channel Analysis

- The developer shall conduct a through search for covert channels and make a determination of the maximum bandwidth of each identified channel.

Trusted Facility Management

- The TCB shall support separate operator and administrator functions.

Configuration Management

- A configuration management system shall be used.

Trusted Recovery

- Procedures and/or mechanisms shall be provided that can recover a system without a comprise of protection.

Trusted Distribution

- Trusted distribution facilities shall be used.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

TCSEC Requirements

Documentation

Security Features User's Guide

- A summary of protection mechanisms.

Trusted Facility Manual

- An administrator manual about running a secure facility.

Test Documentation

- Documentation of the test plan and test results.

Design Documentation

- A description of the design.

The following symbols are used in the chart on the next page.

No requirement	■
New or enhanced requirement	⊗
No additional requirement	⇒

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

TCSEC Requirements Chart

Criteria	D	C1	C2	B1	B2	B3	A1
Security Policy							
Discretionary Access Control		⊗	⊗	⇒	⇒	⊗	⇒
Object Reuse			⊗	⇒	⇒	⇒	⇒
Labels				⊗	⊗	⇒	⇒
Label Integrity				⊗	⇒	⇒	⇒
Exportation of Labeled Information				⊗	⇒	⇒	⇒
Labeling Human Readable Output				⊗	⇒	⇒	⇒
Mandatory Access Control				⊗	⊗	⇒	⇒
Subject Sensitivity Labels					⊗	⇒	⇒
Device Labels					⊗	⇒	⇒
Accountability							
Identification and Authentication		⊗	⊗	⊗	⇒	⇒	⇒
Audit			⊗	⊗	⊗	⊗	⇒
Trusted Path					⊗	⊗	⇒
Assurance							
System Architecture		⊗	⊗	⊗	⊗	⊗	⇒
System Integrity		⊗	⇒	⇒	⇒	⇒	⇒
Security Testing		⊗	⊗	⊗	⊗	⊗	⊗
Design Specification and Verification				⊗	⊗	⊗	⊗
Covert Channel Analysis					⊗	⊗	⊗
Trusted Facility Management					⊗	⊗	⇒
Configuration Management					⊗	⇒	⊗
Trusted Recovery						⊗	⇒
Trusted Distribution							⊗
Documentation							
Security Features User's Guide		⊗	⇒	⇒	⇒	⇒	⇒
Trusted Facility Manual		⊗	⊗	⊗	⊗	⊗	⇒
Test Documentation		⊗	⇒	⇒	⊗	⇒	⊗
Design Documentation		⊗	⇒	⊗	⊗	⊗	⊗

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Class D and C1

Class D Systems: Minimal Security

- There are no evaluated systems in this class.

Class C1 Systems: Discretionary Security Protection

- C1 systems provide rather limited security features.
- C1 systems are an environment of "cooperating users processing data at the same level of security."
- Two main features:
  - I and A, e.g., passwords
  - DAC
  - Does not require a distinction between read, write and execute.
  - Allows wildcards. E.g., M* = all users whose names begin with M.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Class C2

Class C2 Systems: Controlled Access Protection

- Accountability through password controls and audit.
- More detailed discretionary controls.
- Object reuse requirement.

DON CAP Program (C2 by 92)

Controlled Access Protection (CAP) Guidebook

(NAVSO P-5239-15)

- Functional interpretation of “Class C2” requirements.
- Describes minimum set of automated controls for a system.
- All DoN systems must be assessed for CAP compliance.
- All DoN systems are considered to process “sensitive unclassified” data as a minimum and therefore must adhere to “Class C2” requirements due to data aggregation and connectivity.
- Waivers may be granted but must be reviewed annually.

CAP Assessments

Basic

- Used to determine if a set of CAP features exist in a product (documentation review)

Detailed

- Used to determine whether CAP features function as described or claimed

Recognized-Authority

- Compliance assessments by:
  - NCSC (National Computer Security Center)
  - NESSEC (Naval Electronic Sys. Security Eng. Center)
  - NRL (Naval Research Laboratory)
  - AFCSC (Air Force Cryptologic Support Center)

C2 Advantages - Abuse of authority, Direct probing.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Class B1

Class B1 Systems: Labeled Security Protection

- An informal or formal model of the Security Policy is required.
- All "major" objects are required labeled and these labels are used to enforce a MAC policy.
- B1 is often call "C2 with labels", since B1 systems do not require much more assurance than C2 systems.
- The labels must be implemented in a way such that a system has the potential to support different Human Readable Labels.
  - The internal labels are probably just numbers and a Human Readable Label Manager maps the numbers to the Human Readable Labels.
  - DoD might use Top Secret, Secret, etc.
  - Non-DoD might use Sensitive, Non-sensitive, etc.
- Requires a Security Officer.
- Requires Security Officer documentation.
  - Administration of labels
  - Securely manage user clearances.
- Requires Human Readable Labels on output.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Class B1 Summary

Advantages

Prevention and Detection

- Abuse of Authority
- Direct Probing

Some Protection Against Probing with Malicious Software

Risks

Direct Penetration

Subversion of Mechanism

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Class B2

High Assurance versus Low Assurance

- Systems rated at class B1 and below are commonly referred to as "low assurance" systems.
- Systems rated at B2 and above are commonly referred to as "high assurance" systems.

Class B2 Systems: Structured Protection

- Not much additional user-visible security features.
- Instead, B2 has extended features and additional assurance that the features were designed to work properly.
- The system is relatively resistant to penetration.

Requirements:

- Formal Security Policy Model.
- MAC for all subjects and objects.
- Greater isolation for Security Kernel.
- Methodical configuration management.
  - Protects against illicit modifications.
- Greater use of modularity and use of hardware features.
- Trusted path.
- Covert channel analysis.
- Identification and isolation of non-security relevant code.
- Penetration testing will augment interface testing.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Class B2 Summary

Advantages

Prevention and Detection

- Abuse of Authority
- Direct Probing

Some Protection Against Probing with Malicious Software

Some Protection Against Direct Penetration

Risks

Direct Penetration

Subversion of Mechanism

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Class B3

Class B3 Systems: Security Domains

- There are no new user-visible features.
- Must satisfy Reference Monitor implementation requirements.
  - Simple
  - Tamper-proof
  - Impossible to bypass
- Exclude code from Security Kernel that is not security relevant.
- Highly resistant to penetration.
- Trusted Facility Management.
  - Assignment of specific individual as security officer.
- Requires Trusted Recovery.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Class B3 Summary

Advantages

Prevention and Detection

- Abuse of Authority
- Protection against Direct Probing

Some Protection Against Probing with Malicious Software

Significant Protection Against Direct Penetration

Some Protection Against Subversion of Mechanism

Risks

Subversion of Mechanism

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Class A1

Class A1 Systems: Verified Protection

- Pretty much functionally equivalent to B3 systems.
- Trusted Distribution is the only new feature.

Additional assurance provided by:

- Formal analysis and mathematical proof that the system design matches the system's security policy and its design specifications.
- Trusted Distribution
  - This decreases the possibility of subversion during distribution, i.e., replacement of TCB parts.
- Life-cycle configuration management.
  - Hardware
  - Software
  - Specifications
  - Development tools
- Life-cycle covers:
  - design
  - development
  - production
  - distribution
- Formal Top Level Specification can be analyzed by computer tools to find all covert storage channels.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Class A1 Summary

Advantages

Prevention and Detection

- Abuse of Authority
- Protection against Direct Probing

Protection Against Probing with Malicious Software

Increased Assurance Against Direct Penetration

Increased Assurance Against Subversion of Mechanism

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

TPEP Program

NCSC Trusted Product Evaluation Program (TPEP)

- Resulted from DoD Directive 5215.1 in 1982.
- The NSA is responsible for evaluating commercial products through an independent evaluation based on TCSEC requirements by a qualified team of experts.
- TPEP phases:
  - Proposal phase
  - Vendor assistance phase
  - Design analysis phase
  - Evaluation phase
  - Rating Maintenance Phase (RAMP)

TPEP Guidelines and Interpretations

- *a.k.a. The Rainbow Series*
- Guidelines
  - A Guide to Understanding Discretionary Access Control in Trusted Systems
  - A Guide to Understanding Trusted Distribution in Trusted Systems
  - A Guide to Understanding Configuration Management in Trusted Systems
  - A Guide to Procurement of Trusted Systems
  - Guidance for Applying the DoD TCSEC in specific Environments

...

- Interpretations:
  - Trusted Network Interpretation (TNI)
  - Trusted Database Interpretation (TDI)

Evaluated Products List (EPL)

- List of products that have completed evaluations and those that are in evaluation.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Common Criteria

Basic Issues:

Considers *Functionality Security Requirements* and *Assurance Security Requirements* separately

Evaluations based on an analysis of *Security Objectives* which drive *Security Requirements* (both Functional and Assurance)

Based on 3 types of requirement constructs

- Packages
  - intermediate combination of components that meets an identifiable subset of security objectives
  - the predefined Evaluation Assurance Levels (EALs) are examples of packages
  - Intended to be reusable

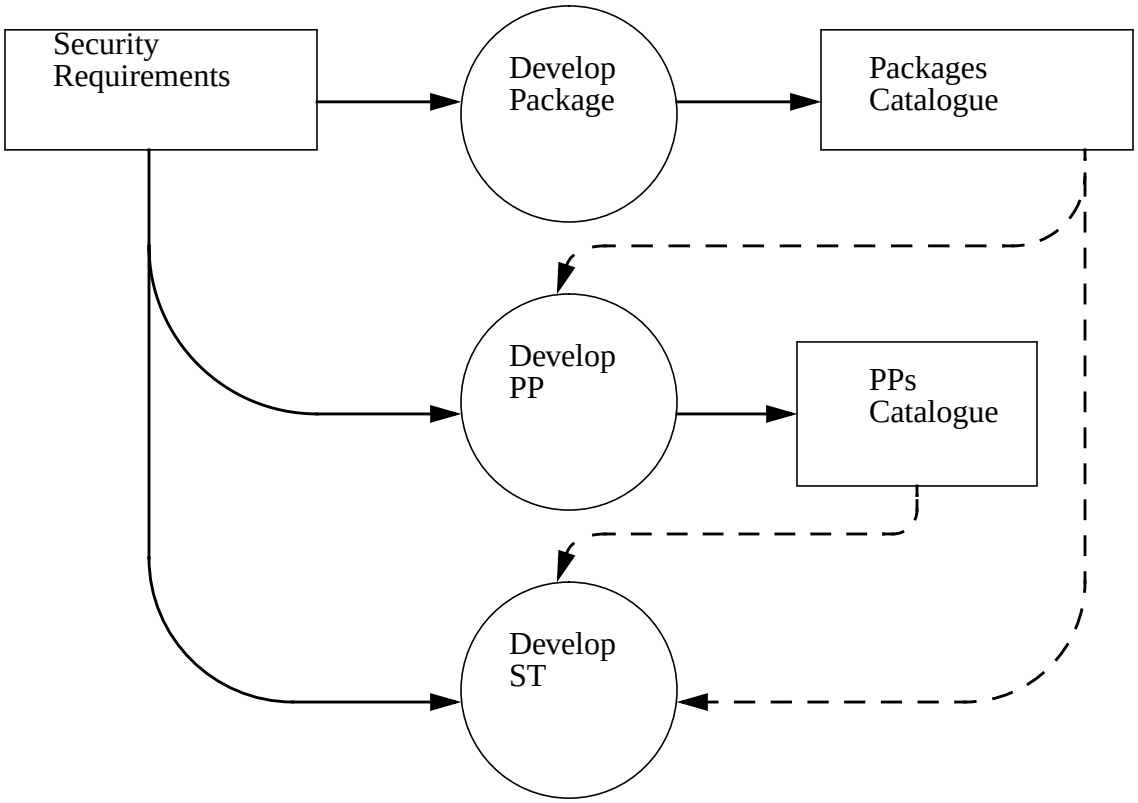
- Protection Profiles (PP)
  - A meaningful set of both security requirements (both functional and assurance) and rationale for the security objectives and requirements.
  - Intended to be reusable
  - (E.g., could write a PP that is equivalent to the TCSEC C2 class)

- Security Targets (ST)
  - The set of security requirements for a target evaluation system.
  - Target evaluation systems are called Targets Of Evaluation (TOE)

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Common Criteria

Use of Security Requirements



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Common Criteria

Types of Evaluation

- PP Evaluation
  - Demonstrate that the PP is complete, consistent, technically sound and suitable for use as a statement of requirements for an evaluatable TOE

- ST Evaluation
  - Demonstrate that the ST is complete, consistent, technically sound and suitable for use as a basis for the corresponding TOE evaluation
    or
  - Demonstrate that the ST properly meets the requirements of a PP (in cases where the ST claims conformance to a PP)

- TOE Evaluation
  - Demonstrate that the TOE meets the security requirements contained in the ST.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Common Criteria

Protection Profile (PP) Contents

Protection Profile

- PP introduction
  - PP identification
  - PP overview
- TOE description (usually a hypothetical TOE)
- TOE Security Environment
  - Assumptions
  - Threats
  - Organizational security policies
- Security Objectives
  - Security objectives for the TOE
  - Security objectives for the environment
- IT Security Requirements
  - Functional Security Requirements
  - Assurance Requirements
- Rationale
  - Security Objectives rationale
  - Security Requirements rationale

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Common Criteria

Security Target (ST) Contents

Security Target

- ST introduction
  - ST identification
  - ST overview
- TOE description
- TOE Security Environment
  - Assumptions
  - Threats
  - Organizational security policies
- Security Objectives
  - Security objectives for the TOE
  - Security objectives for the environment
- IT Security Requirements
  - Functional Security Requirements
  - Assurance Requirements
- TOE Summary Specifications
  - TOE security functions
  - Assurance requirements
- PP claims
- Rationale
  - Security Objectives rationale
  - Security Requirements rationale

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Common Criteria

Functional Classes and Number of Corresponding Family Members

Functional Class Names	Number of Family Members
Communication	2
I & A	6
Privacy	4
Protection of TOE Security Functions	16
Resource Allocation	3
Security Audit	6
TOE Access	6
Trusted Path	2
User Data Protection	13
Cryptographic Support	2
Security Management	6

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Common Criteria

Family Members for the I & A Class and the number of components for the corresponding members

Family members of the I & A Class	Number of components
Authentication of failures	1
User attribute definition	1
Specification of secrets	2
User authentication	7
User identification	2
User-subject binding	1

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Common Criteria

Components of the User Authentication family

Components of the User Authentication Family
Timing of authentication
User authentication before any action
Unforgeable authentication
Single-use authentication mechanisms
Multiple authentication mechanisms
Re-authenticating
Protected authentication feedback

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Common Criteria

Evaluation Assurance Level (EAL) Summary

Assurance Class	Assurance Family	EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Configuration management	ACM_AUT				**1**	1	**2**	2
	ACM_CAP	**1**	**2**	**3**	**4**	4	**5**	5
	ACM_SCP			**1**	**2**	**3**	3	3
Delivery and operation	ADO_DEL		**1**	1	**2**	2	2	**3**
	ADO_IGS	**1**	1	1	1	1	1	1
Development	ADV_FSP	**1**	1	1	**2**	**3**	3	4
	ADV_HLD		**1**	**2**	2	**3**	**4**	**5**
	ADV_IMP				**1**	**2**	**3**	3
	ADV_INT					**1**	**2**	**3**
	ADV_LLD				**1**	1	**2**	2
	ADV_RCR	**1**	1	1	1	**2**	2	**3**
	ADV_SPM				**1**	**3**	3	3
Guidance documents	AGD_ADM	**1**	1	1	1	1	1	1
	AGD_USR	**1**	1	1	1	1	1	1
Life cycle support	ALC_DVS			**1**	1	1	**2**	2
	ALC_FLR							
	ALC_LCD				**1**	**2**	2	**3**
	ALC_TAT				**1**	**2**	**3**	3
Tests	ATE_COV		**1**	**2**	2	2	**3**	3
	ATE_DPT			**1**	1	**2**	2	*3*
	ATE_FUN		**1**	1	1	1	**2**	2
	ATE_IND	**1**	**2**	2	2	2	2	**3**

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Assurance Class	Assurance Family							
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Vulnerability assessment	AVA_CCA					**1**	**2**	2
	AVA_MSU			**1**	**2**	2	**3**	3
	AVA_SOF		**1**	1	1	1	1	1
	AVA_VLA		**1**	1	**2**	**3**	**4**	4

The numbers identify a specific assurance component.

Bold type is used to highlight differences between requirements.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Overview of CM Capabilities (ACM_CAP)

- ACM_CAP.1
  - Version numbers
- ACM_CAP.2
  - Configuration items
- ACM_CAP.3
  - Authorization controls
- ACM_CAP.4
  - Generation support and acceptance procedures
- ACM_CAP.5
  - Advanced support; (E.g., controls to insure unauthorized modifications, master copy identification, acceptance procedures for modifications, etc.)

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Balanced Assurance

Technically Sound

- Security Architectures Use TCB Subsets
- Require Greatest Assurance for Most Critical Policies
- Enforcement Mechanism for Most Critical Policies is Most Privileged

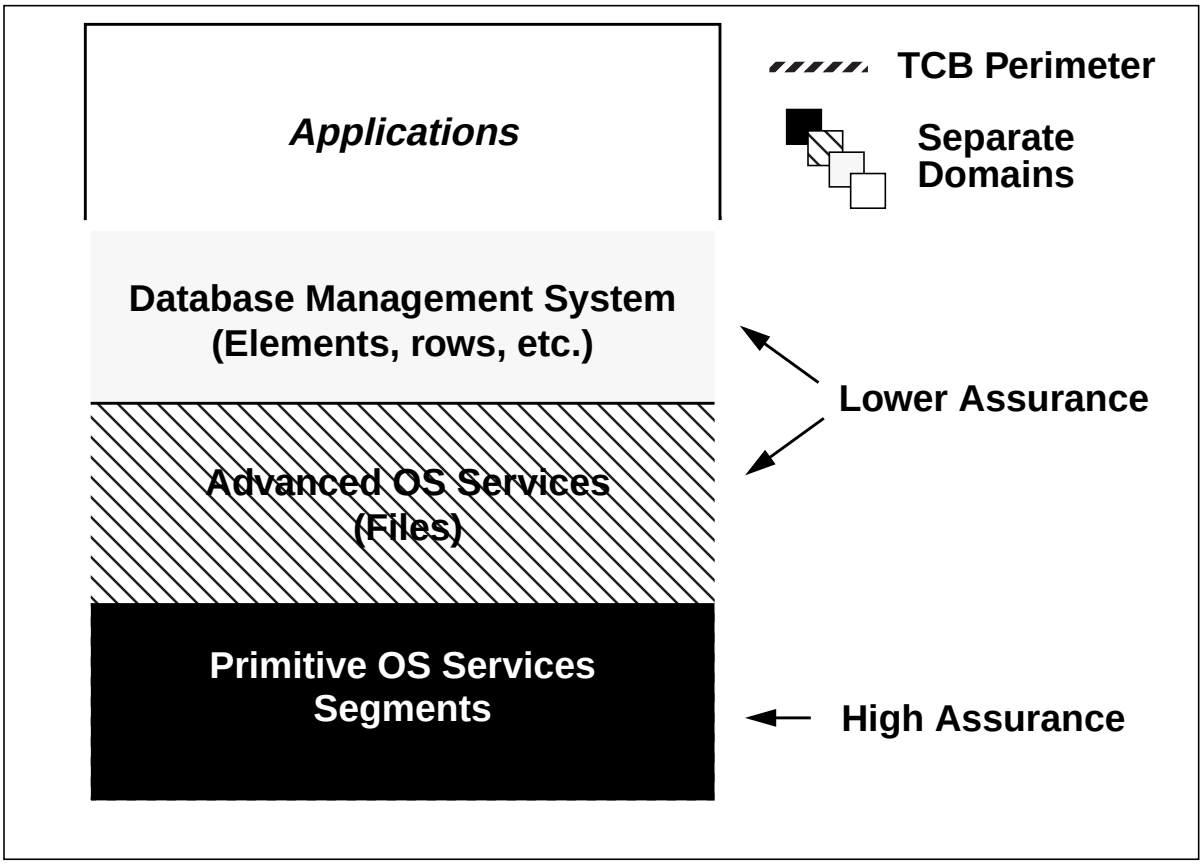
Commercially Attractive

- Can Build System Using Incrementally Evaluated Components

Permits High Assurance Where Critical

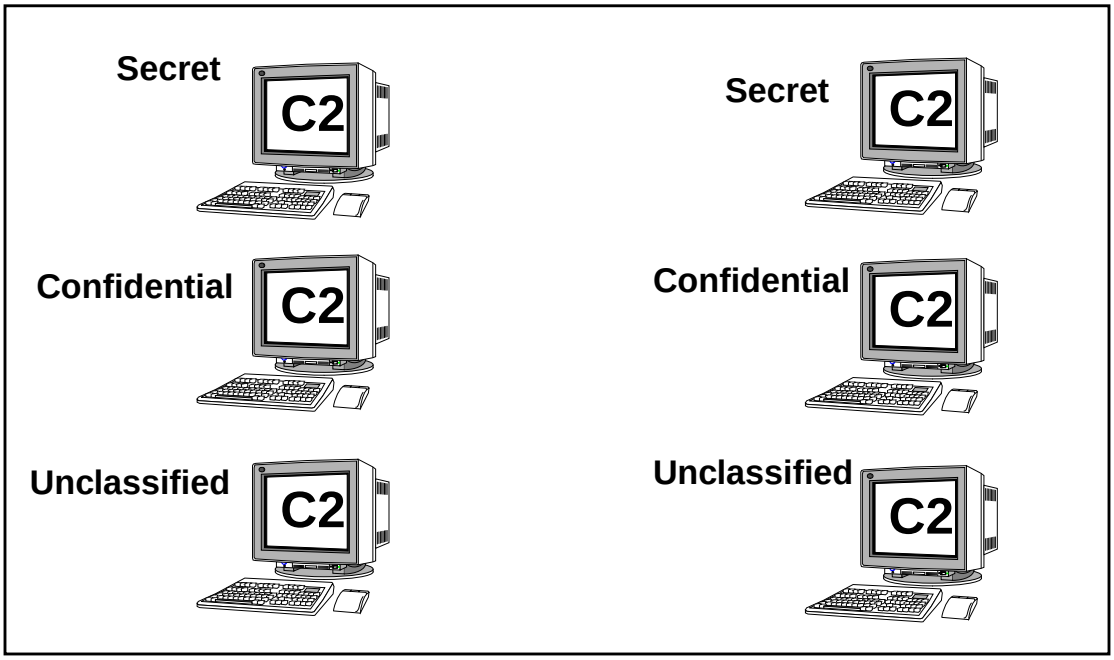
Does not Impose High Assurance Requirements Unnecessarily

Hierarchical Balanced Assurance Architecture



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

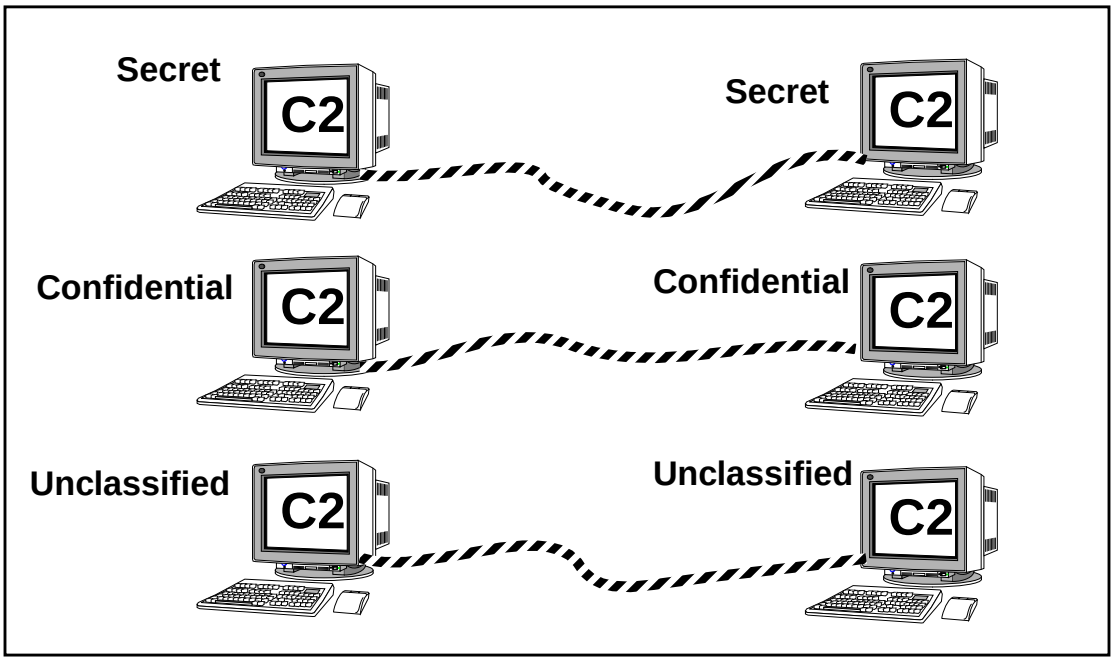
Architecture Before Networking



Separate systems

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

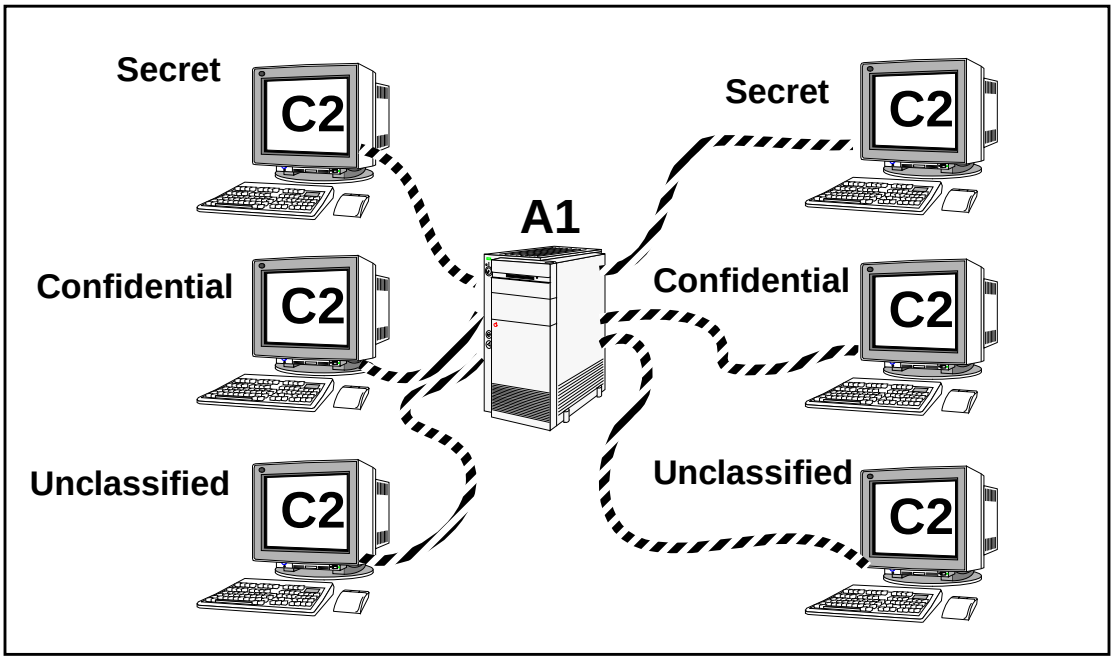
Single-Level Connection Architecture



Systems Connected at a Single Security Level

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Networked Balanced Assurance Architecture



Systems Connected Through A High Assurance Guard

With one minor caveat, one could argue convincingly that the resultant network above could satisfy the Orange Book A1 requirements, since the mandatory enforcement is being done by an A1 rated component. The minor caveat is that A1 DAC requires the ability to deny access down to the granularity of a single user (i.e., Alice is denied read access to file X, even though file X is readable by everyone else in the world). The C2 systems shown above are not required to have this functionality.

The phrase “C2+” is often used to describe systems that satisfy C2 functionality and assurance requirements and, in addition, implement a DAC policy that has the ability to deny access down to the granularity of a single user. Thus, if C2+ systems are used in the configuration shown above, a convincing argument could be made that the entire network should warrant an A1 rating.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Yellow Book Provides Standard Guidance

Security Matrix for Open Security Environments								
	Maximum Data Sensitivity							
Minimum Clearance or Authorization of System Users		U	N	C	S	TS	1C	MC
	U	C1	B1	B2	B3	*	*	*
	N	C1	C2	B2	B2	A1	*	*
	C	C1	C2	C2	B1	B3	A1	*
	S	C1	C2	C2	C2	B2	B3	A1
	TS(BI)	C1	C2	C2	C2	C2	B2	B3
	TS(SBI)	C1	C2	C2	C2	C2	B1	B2
	1C	C1	C2	C2	C2	C2	C2†	B1‡
	MC	C1	C2	C2	C2	C2	C2†	C2†

Systems for Classes C1 or C2 are assumed system high

C2† -- If users not authorized for all categories, then Class B1 or higher

B1‡ -- If 2 categories, Need Class B2

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

System Composition Dangers

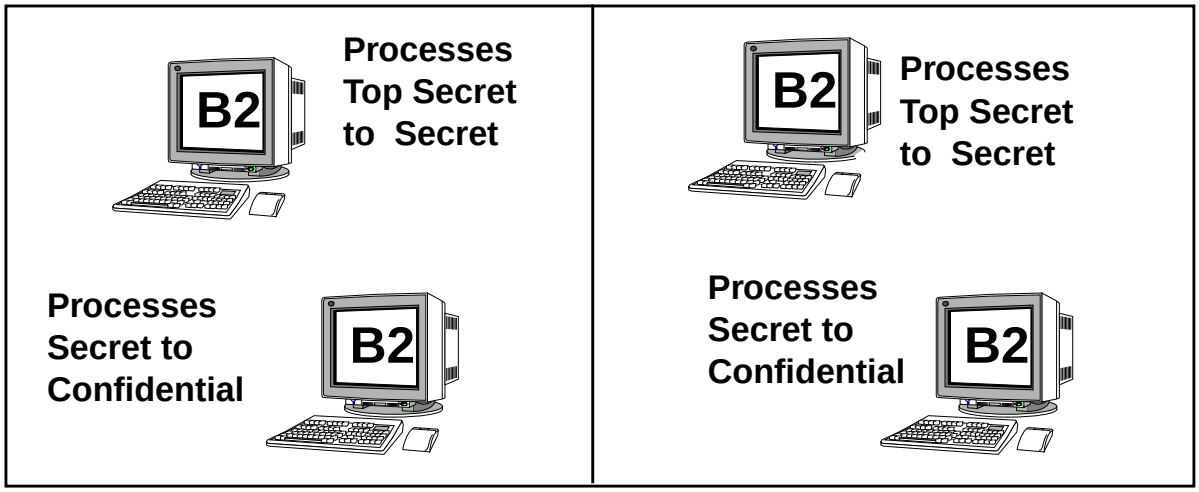
Need to Combine Systems or Components

- composition can be challenging
  - connection of separately secure systems may be insecure
  - need more research on theory of composition

Cannot Assume Adequate Assurance Although Individual Components are Sufficient for Isolated systems

Cascade Problem Example

Separate Systems have Adequate Assurance



Systems with Secret-to-Secret Connection are Inadequate

- Information can “cascade” from TS to S to C

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Criteria Dangers

Mix and Match Approach

- Arbitrary Protection Profiles to Counter Specific “Threats”
- Separation of *Functional* Requirements from *Assurance* Requirements

Post Evaluation TCB Extensions

Relying on Vendor “Pedigrees”

- Most Popular University Operating System is Notably Insecure
- Example: Building Nice WYSIWYG Interfaces Does not Imply Security Competence
- Vendors May Cut Corners for Greater Profit
- Changes in Personnel or Business Strategy Unknown to Evaluation Authority
- Customer Will Not Know Assurance Lacking Until Too Late

Assurance Summary

Codify What is Demonstrated--Worked Examples

Unify with COMSEC Practice

- Utilize synergy
- Identify uses for Cryptographic Techniques -- integrity of labels
- Identify Trusted Processing -- keys

TCB Subsetting Techniques -- TNI gave a start

Trusted Subject Methods

- Covert Storage Channel Analysis
- Sufficient Design Constraints
- Definitive Tie to Cryptography

Explicitly Address Balanced Assurance

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Color	Title and Summary of Contents
Orange Book	*Department of Defense (DoD) Trusted Computer System Evaluation Criteria (TCSEC)* Contains basic requirements in four categories for trusted operating systems: security policy, accountability, assurance, and documentation.
Green Book	*Department of Defense (DoD) Password Management Guideline* Contains a set of good practices for the design, implementation, and use of password systems used for authentication. Many trusted systems comply explicitly with this guideline.
Light Yellow Book	*Computer Security Requirements - Guidance for Applying the Department of Defense (DoD) Trusted Computer System Evaluation Criteria (TSEC) in Specific Environments* Contains information on different modes of security (closed security environment, open security environment, dedicated security mode, controlled security mode, and multi-level security mode) and the “risk index” associated with each environment.
Yellow Book	*Technical Rationale Behind CSC-STD-003-85: Computer Security Requirements - Guidance for Applying the Department of Defense (DoD) Trusted Computer System Evaluation Criteria (TSEC) in Specific Environments* Companion to the Light Yellow Book. Contains background information on determining the class of trusted system required for different risk indexes.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Color	Title and Summary of Contents
Dark Blue Book	*Department of Defense (DoD) Magnetic Remanence Security Guideline (FOUO)* Contains recommendations for using products that purge magnetic media via various types of data sanitization and magnetic remanence techniques.
Tan Book	*A Guide to Understanding Audit in Trusted Systems* Contains an interpretation of the auditing requirements included in the Orange Book. Auditing keeps track of sensitive activities in a system and provides a way of determining who performed these activities.
Aqua Book	*Trusted Product Evaluations: A Guide for Vendors* Contains procedures to follow when submitting a trusted system (or a network product, a database product, or a subsystem) to the NCSC for evaluation.
Salmon Book	*A Guide to Understanding Discretionary Access Control (DAC) in Trusted Systems* Contains an interpretation of the discretionary access control requirement included in the Orange Book. DAC protects files and other objects in a system at the discretion of the owner.
Dark Green Book	Glossary of Computer Security Terms Contains definitions for common terms used in government computer security publications.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Color	Title and Summary of Contents
Red Book	*Trusted Network Interpretation (TNI) of the Trusted Computer System Evaluation Criteria (TCSEC)* Contains an interpretation of the Orange Book requirements for networks, and a summary of specific network services: communications integrity, denial of service, and compromise protection.
Coral Book	*A Guide to Understanding Configuration Management in Trusted Systems* Contains an interpretation of the configuration management requirements included in the Orange Book. These requirements manage changes to the Trusted Computing Base and to the system documentation.
Burgundy Book	*A Guide to Understanding Design Documentation in Trusted Systems* Contains an interpretation of the design documentation requirements included in the Orange Book, including the suggested scope and level of effort for this documentation.
Lavender Book	*A Guide to Understanding Trusted Distribution in Trusted Systems* Contains an interpretation of the trusted distribution requirements included in the Orange Book. These requirements ensure that all elements of the TCB distributed to a customer arrive exactly as intended by the vendor. They include recommendations for packaging, security locks, courier service, etc.
Venice Blue Book	Contains an interpretation of the Orange Book requirements for computer security add-on products and subsystems. Subsystems typically provide features in one or more or the following categories: discretionary access control, object reuse, identification and authentication, and audit.

Color	Title and Summary of Contents
Dark Red Book	*Trusted Network Interpretation Environments Guideline* Companion to the Red Book. Contains information helpful when integrating, operating, and maintaining trusted computer networks, including the minimum security required in different network environments.
Pink Book	*Rating Maintenance Phase (RAMP) Program Document* Contains procedures for keeping an Orange Book rating up to date via the RAMP program. Participation in RAMP is required for C1, C2 and B1 systems.
Purple Book	*Guidelines for Formal Verification Systems* Contains procedures to follow when submitting a formal design and verification tool to the NCSC for evaluation.
Brown Book	*A Guide to Understanding Trusted Facility Management* Contains an interpretation of the trusted facility management requirements included in the Orange Book. These requirements mandate certain types of system and security administration - for example, the separation of operator, security administrator, and account administrator functions.
Light Blue Book	*Trusted Product Evaluation Questionnaire* Contains an extensive list of questions aimed at vendors of trusted systems. Examples are “What are the subjects in your system?” and “How can an operator distinguish the TCB-generated banner pages from user output?” The goal of the list is to help vendors understand what technical information is required for the system to be evaluated successfully.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Color	Title and Summary of Contents
Gray Book	*Trusted UNIX Working Group (TRUSIX) Rationale for Selecting Access Control List Features for the UNIX System* Contains a description of access control lists (ACLs), their use in enforcing the discretionary access control (DAC) feature included in the Orange Book, and the reasons for selecting this mechanism as a standard for trusted UNIX systems
Lavender 2	*Trusted Database Management System Interpretation* Contains an interpretation of the Orange Book requirements for database management systems.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

This page is intentionally blank.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Section 7

Basics of Cryptography

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Services Provided by Cryptosystems

Secrecy

- Secrecy requires that an intruder should not be able to determine the plaintext corresponding to given ciphertext, and should not be able to reconstruct the key by examining ciphertext for known plaintext.

Authenticity

- Authenticity requires that the sender can validate the source of a message; i.e., that it was transmitted by a properly identified sender and is not a replay of a previously transmitted message.

Integrity

- Integrity requires the ability to assurance that a message was not modified accidentally or deliberately in transit, by replacement, insertion or deletion.

Nonrepudiation

- Protection against a sender of a message later denying transmission.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Introductory Concepts

Definitions

- Encryption - encode.
- Decryption - decode.
- Cryptology - study of encryption and decryption.
- Cryptography - using encryption to conceal text.
- Cryptanalysis - the breaking of secret writing.

- Plaintext - the original message P
  - Sometimes called cleartext.

  $P = [p_1, p_2,..., p_n]$

- Ciphertext - the encrypted message C

  $C = [c_1, c_2,..., c_n]$

Encryption Algorithms

- C = E(P) (E is the encryption algorithm)
- P = D(C) (D is the decryption algorithm)
  - The term encipher is sometimes used for encryption.
  - The term decipher is sometimes used for decryption.

Clearly, we must have:

- P = D(E(P))

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Substitution Ciphers

The Caesar Cipher

- Each character of the plaintext is replaced with the character three to the right, modulo 26. I.e., A is replaced with D, B is replaced with E, ..., Z is replaced with C.
- x modulo y (or simply x mod y) is the remainder obtained when x is divided by y. I.e., $28 \bmod 26 \equiv 2$.
- Modulo is used to handle “wrap around” situations.
- The table below shows how plaintext is encrypted into ciphertext.

p_i	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
c_i	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Example																										
p_i	P	R	O	F	E	S	S	I	O	N	A	L		C	O	U	R	T	E	S	Y
c_i	S	U	R	I	H	V	V	L	R	Q	D	O		F	R	X	U	W	H	V	B

Variation of Caesar Cipher

p_i	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
c_i	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O

- With any cipher that is a variation of the Caesar Cipher, the message receiver only needs to know what the character A maps to in order to be able to decrypt the whole message. I.e., once you know what character A maps to, you can figure out what all the other characters map to.
- Thus:
  - The *key* of the Caesar Cipher is D.
  - The *key* of the Caesar Cipher variation is P.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Substitution Ciphers

Weaknesses of Caesar Ciphers

- A **brute force** attack is one where all possible keys are tried in an attempt to break a cipher.
- Caesar Ciphers only have 25 effective keys, and is therefore subject to a quick brute force attack.
- Encrypting a message multiple times with multiple keys does not add any security.

  Example:

  Encrypting the word “help” twice, using the keys of C and K, produces the following:

  ciphertext = tqxb

  Encrypting the word “help” once, using the key of M produces:

  ciphertext = tqxb

- For every encryption key, there exists a second key that if used to encrypt the output a second time, will actually decrypt it.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Substitution Ciphers

A General Substitution Cipher

- A more general substitution cipher is produced by using a mapping of characters that is not so simplistic as the previous two examples. Consider the mapping below:

p_i	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
c_i	X	D	G	J	M	P	S	V	Y	B	E	H	K	N	Q	T	W	Z	C	F	I	L	O	R	U	A

- In order for the message receiver to decrypt a message using this cipher, they need to know what every character in the alphabet maps to.

- Hence the key needs to be something like:
  - (X,D,G,J,M,P,S,V,Y,B,E,H,K,N,Q,T,W,Z,C,F,I,L,O,R,U,A).

Cryptanalysis Attack of These Codes:

- There are 26! (26 factorial = 26 x 25 x 24 x ... x 1) keys for the general substitution cipher example. $26! \cong 4 \times 10^{26}$

- This number of keys is too great to attempt a brute force attack.

- In spite of this, this type of cipher is easy to crack.

- With the exception of the brute force attack, the General Substitution Cipher has the same weaknesses as the Caesar Cipher.

- Letter frequency analysis is commonly used to break substitution ciphers because the letters change, but their properties do not.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Frequency Distribution Analyses

Frequency Distribution of Characters in English:

- The following tables list the relative frequency of characters in the English language.

Character	E	T	R	N	I	O	A	S	D	L	H	C	F
percent	12.75	9.25	8.50	7.75	7.75	7.50	7.25	6.00	4.25	3.75	3.50	3.50	3.00

Character	U	M	P	Y	G	W	V	B	K	X	Q	J	Z
percent	3.00	2.75	2.75	2.25	2.00	1.50	1.50	1.25	0.50	0.50	0.50	0.25	0.25

Consider the following ciphertext:

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

- The distribution of characters in this message is:

Character	P	Z	S	U	O	M	H	D	E	V	X	F	W
percent	13.13	11.67	8.33	8.33	7.50	6.67	5.83	5.00	5.00	4.17	4.17	3.33	3.33

Character	Q	T	A	B	G	Y	I	J	C	K	L	N	R
percent	2.50	2.50	1.67	1.67	1.67	1.67	0.83	0.83	0.00	0.00	0.00	0.00	0.00

- It seems likely that cipher letters P an Z are the equivalents of plaintext letters E and T, but it is not certain which is which.
- The letters S, U, O, M and H are all of high frequency and probably correspond to plaintext letters from the set {R, N, I, O, A, S}.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Frequency Distribution Analyses

Additional Strategies:

- The frequency of two-letter combinations (known as digraphs) can also provide clues.
- For example, the digraph ZW appears three times.
- The most common digraph is “TH”.
- Also ZWP appears in the ciphertext and we conjectured that P might stand for E in plaintext.
- Furthermore, ZWSZ appears in the first line.
- It is possible that S stands for A.
- Given these assumptions we have the following structure:

U	Z	Q	S	O	V	U	O	H	X	M	O	P	V	G	P	O	Z	P	E	V	S	G	Z	W	S
	t		a									e			e		t	e			a		t	h	a

Z	O	P	F	P	E	S	X	U	D	B	M	E	T	S	X	A	I	Z	V	U	E	P	H	Z	H
t		e		e		a								a				t				e		t	

M	D	Z	S	H	Z	O	W	S	F	P	A	P	P	D	T	S	V	P	Q	U	Z	W	Y	M	X
		t	a		t		h	a		e		e	e			a		e			t	h				

U	Z	U	H	S	X	E	P	Y	E	P	O	P	D	Z	S	Z	U	F	P	O	M	B	Z	W	P
	t			a			e			e		e		t	a	t			e				t	h	e

F	U	P	Z	H	M	D	J	U	D	T	M	O	H	M	Q
		e	t												

- At this point, trial and error should yield the plaintext.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

More On Substitution Ciphers

Monoalphabetic Substitution Ciphers:

- All ciphers discussed so far are examples of monoalphabetic ciphers.
- Throughout the whole message, each character of plaintext is always replaced by the same character of ciphertext.
  - For example, when using the Caesar Cipher, the plaintext letter I is always replaced with the ciphertext letter L.
- Any cipher that has this property is called a monoalphabetic cipher.
- If the message is long enough, the distribution of letters in the ciphertext will be similar to the distribution of letters in English.
  - If the letter E occurs 13% of the time in the plaintext, then the letter that E encrypts to will occur 13% of the time in the ciphertext.
- Thus, monoalphabetic ciphers lend themselves to character frequency analyses and are **relatively easy to break**.

Advantages of substitution ciphers:

- Can be performed by direct lookup.
- Time to encrypt a message of n characters is proportional to n.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Polyalphabetic Ciphers

Polyalphabetic Cipher Issues:

- Polyalphabetic Ciphers are an improvement over the simple monoalphabetic technique.
- Polyalphabetic Ciphers use two or more monoalphabetic ciphers when encrypting a message.

A Simple Polyalphabetic Cipher

- Consider an example which uses a variation of the Caesar Cipher with key D on even letters and key M on odd letters.
- In this example the letter E in the plaintext sometimes encrypts to the letter H (when E is in an even position in the plaintext) and sometimes encrypts to letter Q (when E is in an odd position in the plaintext).
- The resulting ciphertext will not then exhibit the same frequency distribution of characters as the plaintext.
- If E occurs 13% of the time in the plaintext, there may be no character in the ciphertext that occurs 13% of the time, because sometimes E is mapped to H and sometimes it is mapped to Q.
- This example cipher is still relatively easy to break.
- After failing to break a message using a straight forward frequency analysis, the cryptanalyst might assume that the cipher is a polyalphabetic cipher and might start looking at frequency distributions of every other letter or every third letter or every fourth letter and so on.
- In the case of this example, if the message is long enough or if enough messages have been intercepted, a frequency analysis of every other letter would break the code.

Advantages of polyalphabetic ciphers:

- Flattens letter frequencies.
- Double letter pairs not so obvious.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Vigenère Cipher

Vigenère Cipher Details:

- The Vigenère Cipher is one of the best known polyalphabetic ciphers.
- Consider the following example:

key	m	o	n	t	e	r	e	y	m	o	n	t	e	r	e	y	m	o	n	t	e	r	e	y	m	o	n
plaintext	w	e	a	r	e	d	i	s	c	o	v	e	r	e	d	s	a	v	e	y	o	u	r	s	e	l	f
ciphertext	I	S	N	K	I	U	M	Q	O	C	I	X	V	V	H	Q	M	J	R	R	S	L	V	Q	Q	Z	S

- In this example the key used is the word “monterey”.
  - Note how the word “monterey” is written repeatedly for the whole length of the message in the top row of the table.
- The key specifies which variation of the Caesar Cipher is used for each letter of the message.
  - For example, the first letter of the message will be encrypted with a Caesar Cipher variation of key M.
  - The second letter will be encrypted with a Caesar Cipher variation of key O.
  - An so on.

Attacking this type of cipher.

- After failing to break a message using a straight forward frequency analysis, the cryptanalyst might assume that the cipher is a polyalphabetic cipher and might start looking at frequency distributions of every other letter or every third letter or every fourth letter and so on.

- In this example, if the message is long enough or if enough messages have been intercepted, a frequency analysis of every eighth letter would break the code.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

One Time Pad Scheme

How To Make An Unbreakable Cipher

- If in the previous example, the key word was random and as long as the message, how would a cryptanalyst attack this cipher?
- The cryptanalyst would need to intercept many messages to develop a statistical relationship between the ciphertext and the plaintext.
    - I.e., the first letter of each message would always be encrypted using the same variation of the Caesar cipher, as would the second letters and so on.
- If enough messages are intercepted, the code could be broken.
- If each random key is only used once (no two messages use the same random key), how would a cryptanalyst attack this cipher?
- No successful attack is possible because the ciphertext has no statistical relationship to the plaintext.
- This type of cipher is called a One Time Pad and it is unbreakable.
- Note however, that this scheme requires the secure distribution of many long (as long as the messages) keys.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Binary Substitution Ciphers

The Vernam cipher:

- The Vernam cipher is a version of the One Time Pad cipher that is implemented using binary keys, plaintext and ciphertext.
- Consider the example below:

key	1	1	0	0	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0
plaintext	1	0	0	0	1	0	0	0	0	1	0	1	1	0	0	1	0	1	0	1	0	1	1	1	0	1	0
ciphertext	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	0	1	1	1	1	1	0	1	0	0	1	0

- The ciphertext is obtain by XORing key bits with plaintext bits.
- I.e., $C_i = P_i \oplus K_i$

- If the key is not as long as the entire message and is therefore repeated, a statistical relationship will exist between the plaintext and the ciphertext and the cipher may be broken.

- If the key is as long as the message, but the key is used for several messages, there will again be a statistical relationship between the plaintext and the ciphertext so that the cipher may be broken.

- If the key is as long as the message and only one message is encrypted with any one key, the code is functionally equivalent to a One Time Pad and is unbreakable.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Transposition Techniques

Transposition Ciphers:

- All examples so far involved the substitution of a ciphertext symbol for a plaintext symbol.
- A very different kind of mapping is achieved by performing a permutation of the plaintext letters.
- Pure transposition ciphers (and monoalphabetic ciphers) are easily recognized because they have the same letter frequencies as the original plaintext.

Example 1:

Writing the message backwards - not very hard to analyze.

- The plaintext "thetimehascomethewalrussaid"
- The ciphertext "diassurlawehtemocsahemiteht"

Example 2:

Transposing adjacent letters - not very hard to analyze.

- The plaintext "thetimehascomethewalrussaid"
- The ciphertext "httemihesaocemhtwelaurssiad"

Example 3: Columnar Transposition

Write the message down in columns and reading off the rows becomes the ciphertext (or vice versa).

- The plaintext "thetimehascomethewalrussaid"

t	e	i	e	a	c	m	t	e	a	r	s	a	d
h	t	m	h	s	o	e	h	w	l	u	s	i	

- The ciphertext "teieacmtearsadhtmhsoehwlusi".
- In this example, the key is the number of rows used.
- Cryptanalysis is fairly straightforward and involves laying out the ciphertext in matrices of various shapes and sizes.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

More Transposition Techniques

Example 4:

- Again the plaintext is written down by column and it is read off by rows, but this time the rows are read off in a permuted order.
  - The key column specifies the order in which the rows are read off.
- The plaintext "thetimehascomethewalrussaid"

key						
3	t	m	c	h	r	i
1	h	e	o	e	u	d
4	e	h	m	w	s	**z**
5	t	a	e	a	s	**z**
2	i	s	t	l	a	**z**

- The ciphertext "heoeudistlaztmchriehmwsztaeasz"
- The 'z's added at the end of some rows are called **nulls** or **padding**. They are required so the message can be decrypted properly. In a real application, these would probably be random letters.
- The cipher is still fairly easy to break, by playing around with different permutations of rows and columns.
- Digraph (common two-letter combinations) and trigraph (common three-letter combinations) frequency tables can be useful to help determine the key.

Multiple Stage Ciphers:

- Transposition ciphers can be made significantly more secure by performing more than one stage of transposition.
- The result is a more complex permutation that is not easily reconstructed.
- Ciphers consisting of multiple stages of transpositions and multiple stages of substitutions can be very secure.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Encryption Issues

Data Compression:

- Compressing the message before encrypting it can enhance a cipher's ability to resist being broken.
- Many of the crypanalytic techniques discussed so far involved making guesses and then looking for English words.
- If the message is compressed before encryption, it does not look like English when it is correctly decrypted (it must be uncompressed to recover the English text).

Types of attacks on ciphers:

- Often today it is assumed that the adversary knows what encryption algorithm is being used.
- When this is true, the adversary is only attempting to determine the key used during the encryption process.
- Cryptographers try to determine the strength of ciphers given that a cryptanalyst may possess different types of information.
- Possible scenarios include:
- Ciphertext only
  - The cryptanalyst knows the algorithm and the ciphertext.
- Known plaintext
  - The cryptanalyst knows the encryption algorithm and a plaintext-ciphertext pair (the plaintext that corresponds to a ciphertext).
  - Somehow the cryptanalyst has obtained the plaintext corresponding to a ciphertext.
- Chosen plaintext
  - The cryptanalyst knows the encryption algorithm and a plaintext-ciphertext pair, such that the plaintext was chosen by the cryptanalyst.
  - Somehow the cryptanalyst has tricked someone into sending a message that might reveal information or structure about the key being used.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

More Encryption Issues

Computational Security:

- An encryption scheme is said to by computationally secure if
  - the cost of breaking the cipher exceeds the value of the encrypted information and
  - the time required to break the cipher exceeds the useful life-time of the information.

Abstract measures of a cipher's effectiveness:

- Confusion:
  - Confusion obscures the relationship between the plaintext and the ciphertext.
  - The easiest way to do this is through substitution.
- Diffusion:
  - Diffusion dissipates the redundancy of the plaintext by spreading it out over the ciphertext.
  - The simplest way to cause diffusion is through transposition.
- *Confusion and diffusion are the cornerstone of good block cipher design.*
- Bit-sensitivity
  - Bit-sensitivity looks at the impact on the ciphertext (how many bits change) as a result of changing either one bit of the plaintext or of the key. We want every bit of the ciphertext to depend on every bit of the plaintext and on every bit of the key.

Codes based on hard problems:

- Just because a cipher is based on a "hard problem" it does not mean that the cryptanalyst needs to solve that problem to break the code.
- Recall that the General Substitution Cipher has 26! keys, which is far too many to try exhaustively.
- But this cipher is easily broken using a frequency distribution analysis of the ciphertext.
- Brute force attacks are usually impractical.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

The Data Encryption Standard (DES)

History of DES

1972 -NBS issued a call for proposals:

- Must provide high level of security.
- Must be completely specified and easy to understand.
- The security of the algorithm must reside in the key; the security should not depend on the secrecy of the algorithm.
- Must be available to all users.
- Must be adaptable for use in diverse applications.
- Must be economical to implement in electronic devices.
- Must be efficient.
- Must be able to be validated.
- Must be exportable.

1974 - IBM responded with "Lucifer" (renamed - DEA).

- Note that Lucifer algorithm used a 128-bit key and DES uses a 56-bit key.
- IBM consulted NSA on design issues.
  - NSA suggested changes to some of the S-boxes.

1976 - DES officially adopted.

Overview of DES

- Combination of:
  - Substitution technique (for confusion).
  - Transposition technique (for diffusion).
- These two techniques are repeated for 16 cycles one on top of the other.
- Plaintext is encrypted in blocks of 64 bits.
- Keys are 64 bits long (only 56 are really needed).
- Uses only standard arithmetic and logical operations on up to 64 bit numbers.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

More on DES

DES Has Four Modes of Operation

- ECB -Electronic Code Book
- CBC - Cipher Block Chaining
- OFB - Output Feedback
- CFB - Cipher Feedback
- More will be said about the differences between these modes of operation later.

The following description of the DES algorithm will assume the ECB mode of operation.

When used for encryption:

- Data is input in a block, which consists of 64 bits.
- A 64 bit key is input.
  - Only 56 bits of the key are used.
  - Every 8th bit is discarded.
  - The extra bits can be used as parity-check bits to ensure the key is error free.
- A 64 bit block of ciphertext is output.

When used for decryption:

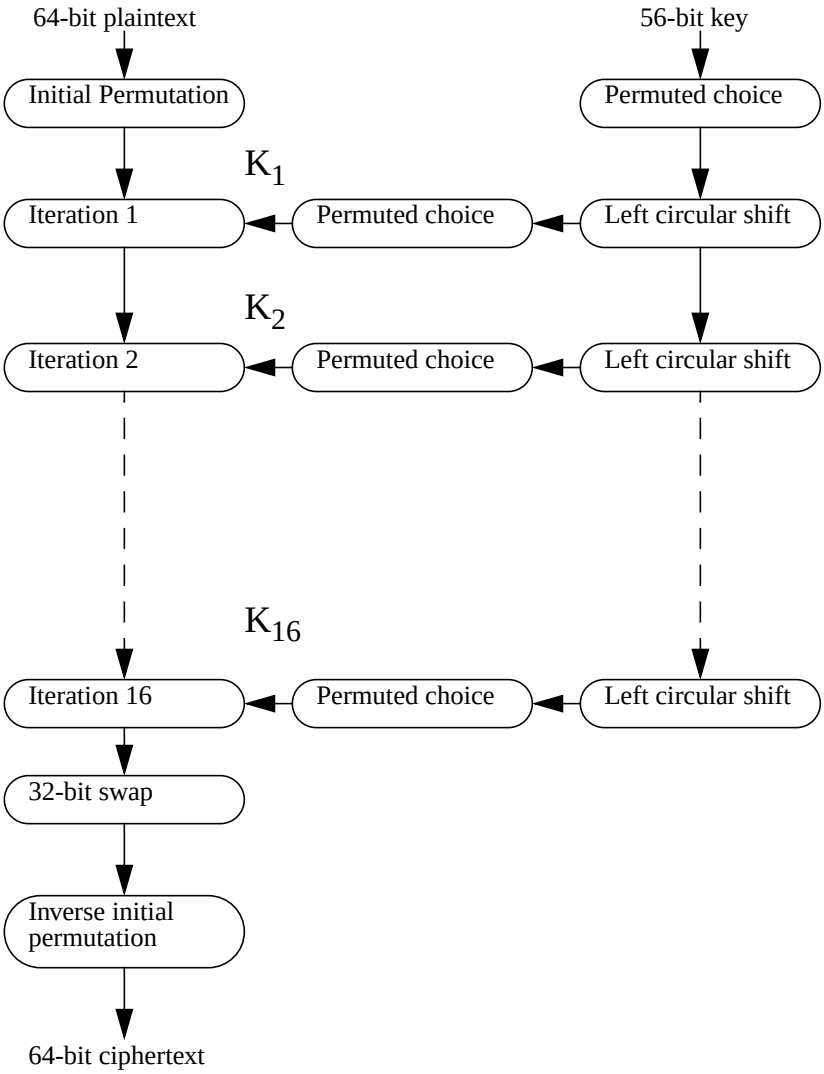
- A 64 bit block of ciphertext is input
- The same key used during encryption is input.
- A 64 bit block of plaintext is output.

Basic algorithm structure

- The figure on the following page reveals the basic algorithm structure.
- The algorithm has 16 iterations.
- The Key goes through 16 transformations.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Overall DES Scheme



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Internal DES Details

DES Structure

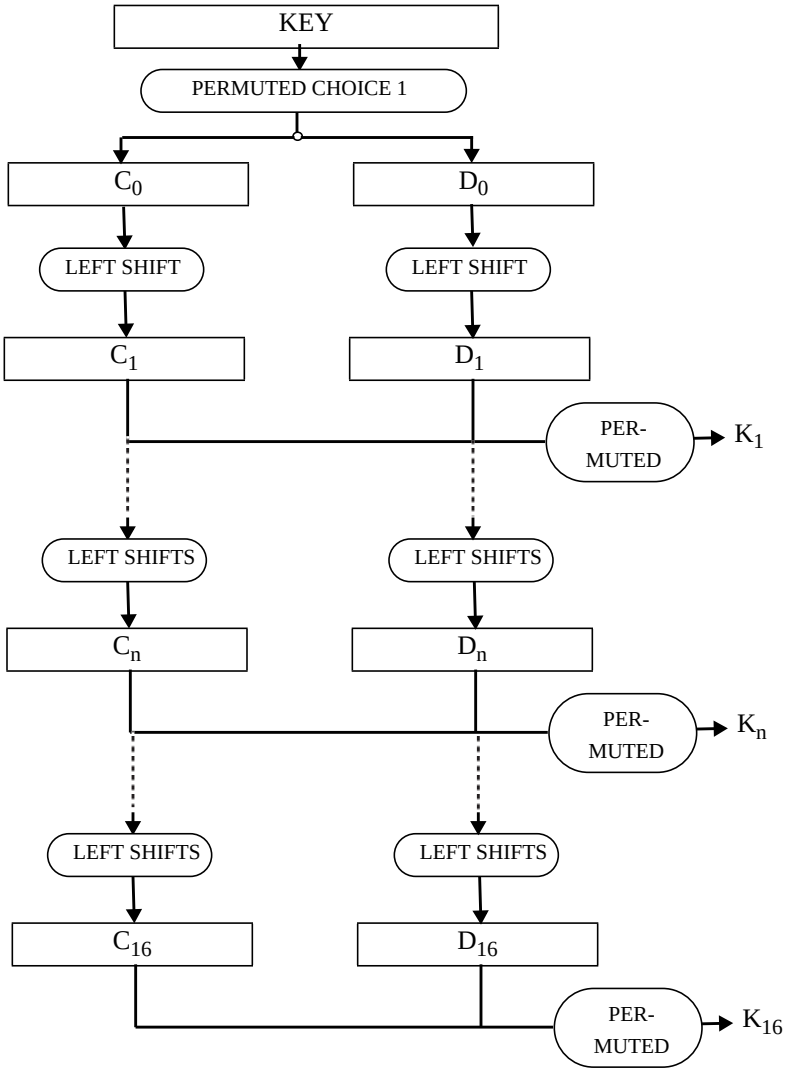
- Notice the two separate lines of processing in the figure on the previous page.
  - An encrypting algorithm on the left.
  - A key transforming algorithm on the right.

- The following two pages discuss the inner structure of the key transforming algorithm.
  - This processing produces the various key values (K_1 through K_{16}) that are used by the encrypting algorithm.

- Following the discussion of the key transforming algorithm is a discussion of the plaintext encrypting portion of the algorithm.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

DES Key Transforming Algorithm



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

DES Key Transforming Algorithm

The Key Scheduler

- Contains a set of bit-shifts and permutations totally independent of the encrypting algorithm.
- The key schedule is usually computed before encrypting takes place.

Step 1

1. The key is subjected to an initial Permuted Choice P-Box.
2. The result is divided into two 28-bit halves labeled C_0 and D_0.

Step 2

1. Both C and D are given a left circular shift according to the shift table.

Step 3.

1. C and D are concatenated to produce CD_1.

Step 4

1. CD_1 is then subjected to a Permuted Choice in which the key is permuted.
2. Bits 9, 18, 22, 25, 35, 38, 43 and 54 are removed to produce a 48-bit key K_1.
3. K_1 is used in cycle 1 of the cryption algorithm.

Steps 2 through 4

1. Repeated a total of 16 times.

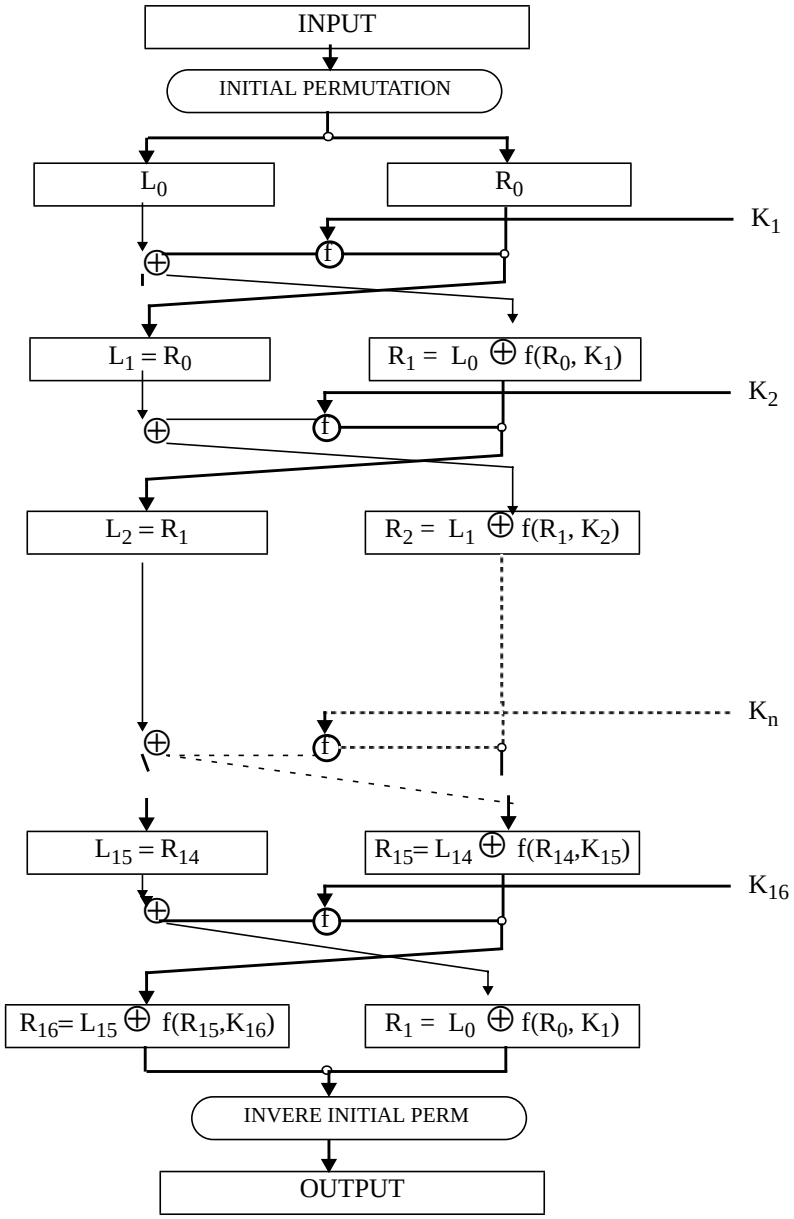
Note:

The only difference in each cycle is the number of bits shifted in the circular shift.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

DES Encrypting Algorithm

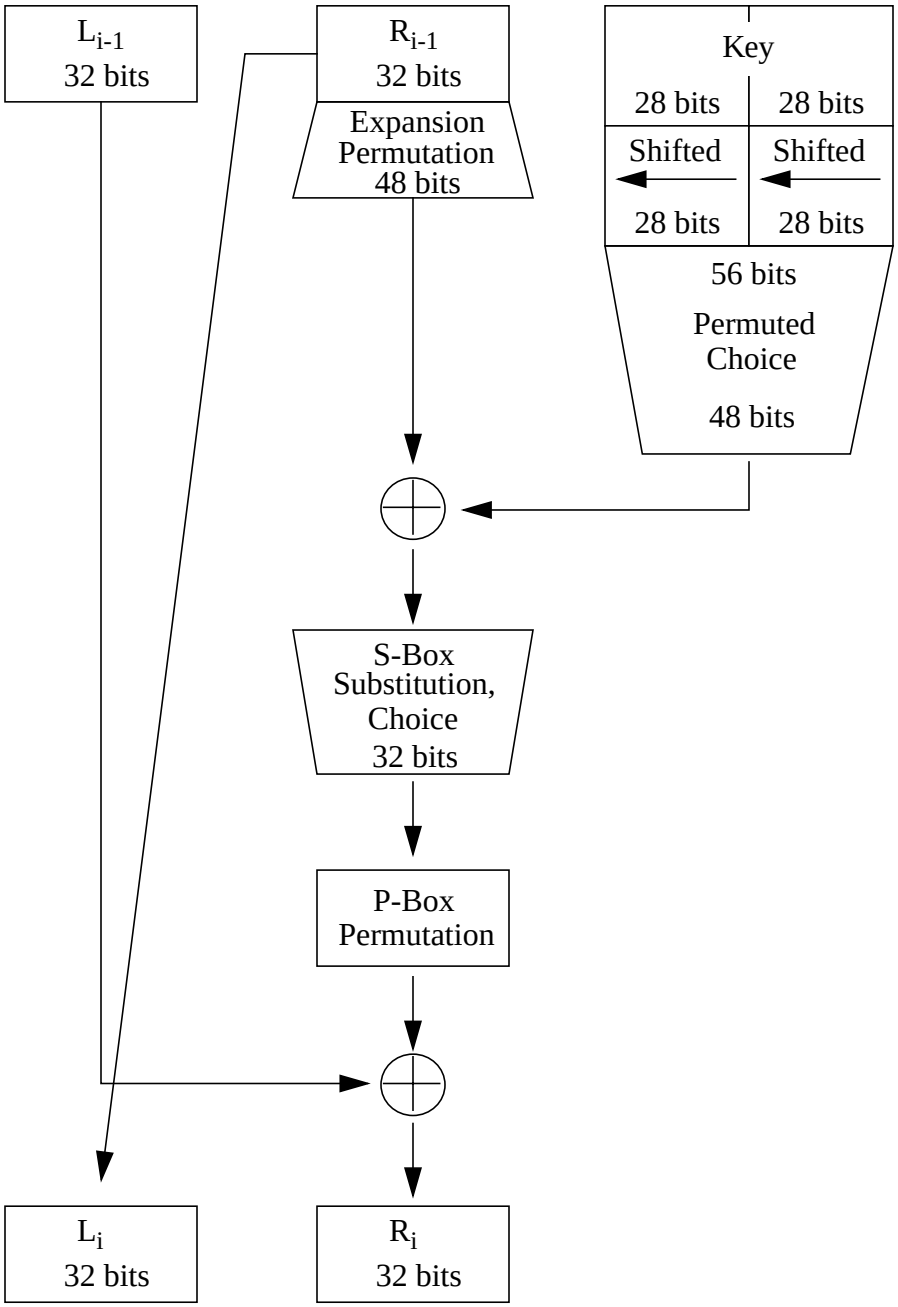
Below is a block diagram of the plaintext encryption algorithm.



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

DES Encrypting Algorithm

Below is a closer look at the actual details of each encryption round, showing the S- and P-boxes.



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

DES Encrypting Algorithm

A Single Cycle of the DES

Step 1.

1. Input 64 bits of plaintext.

Step 2.

1. Rearrange by a P-Box known as the Initial Permutation (IP).

Step 3.

1. Split block into two 32 bit segments called the Left (L) and Right (R) halves.
2. Save a copy of the Right half and label R_0.

Step 4.

1. Subject R to a special permutation box called a Permutation Expansion (P_E) which takes 32 input bits and produces 48 output bits.

Step 5.

1. Take the expanded R and XOR it against a 48-bit segment of the key.

   (Note: This is the only place in each cycle which involves the key)

Step 6.

1. Output from XOR is called Pre-S block.
2. 48-bit Pre-S block is broken into eight 6-bit segments.
3. Each segment processed by a different S-Box in parallel.
4. Each S-Box produces four bits.
5. A total of 32 bits output called Post-S.

Step 7.

1. Post-S is fed to a final P-box.
2. Takes a 32-bit input and returns a 32-bit output.
3. The output is called the result.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

DES Encrypting Algorithm

Note:

Steps 4 through 7 are often grouped into one function in DES diagrams **f**(R_{n-1},K_n)

Step 8.

1. The Left (L) is now XORed against the output of F(R,K_n) to produce the “New R”
2. R_0 now becomes the “New L”

Steps 3 to 7

1. repeated 16 times for each 64-bit block to be encrypted

- Finally, the PREOUTPUT is subjected to a reverse of the initial permutation (IP)
- This is required for the algorithm's invertibility.
- Decryption uses the exact same algorithm except that the order in which the keys are used is reversed.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Substitution Boxes (S-Boxes)

Substitution Box (S-Box)

- Introduces confusion and non-linearity to DES
- Interpret bits as numbers
- One number replaced by another from a table
  - table has values ranging from 0 (0000) to 15 (1111)
  - duplications among the elements
- Takes 6-bit input and returns 4-bit output
  1. First and last bits choose row into S-box substitution table.
  2. The middle four bits chooses the column
  3. The table returns a four bit number
- They are the heart and soul of the algorithm's secrecy

S-Box #1 of 8

	Column Number															
Row No	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Example S-box Input/Output	
INPUT	binary 101011 = decimal 43
First and Last bits	binary 11 = decimal 3
Middle four bits	binary 0101 = decimal 5
OUTPUT	binary 1001 = decimal 9

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

ECB Mode

Electronic Codebook Mode (ECB)

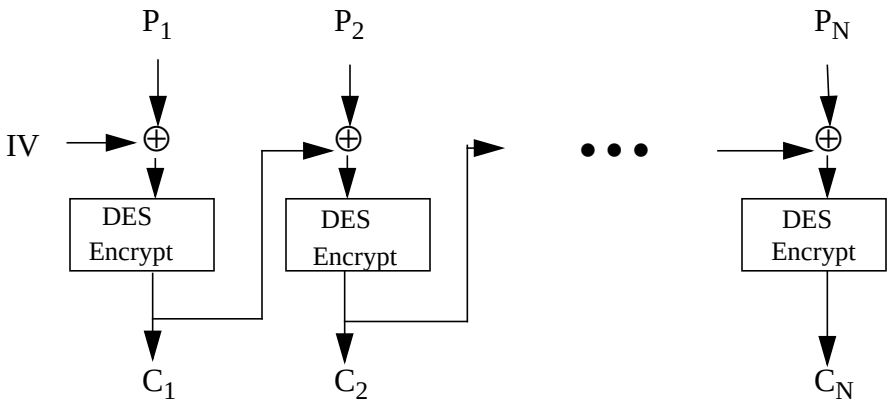
- Each 64-bit plaintext block is encrypted independently of all other plaintext blocks.
- The term codebook is used because, for a given key, there is a unique ciphertext for every 64-bit block of plaintext.
- Abstractly, one could imagine a gigantic codebook with an entry for every 64-bit plaintext block and the corresponding 64-bit ciphertext block.
- If a message is highly structured, it may contain blocks of plaintext that are identical. And since this mode encrypts them to identical ciphertext blocks, some structure of the message maybe revealed.
- Hence, this mode is not considered too secure for long messages.
- An advantage of this mode is that due to the independence of the block encryptions, an error that occurs during transmission in one block will only affect the decryption of that block.
  - I.e., errors do not propagate.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

CBC Mode

Cipher Block Chaining Mode (CBC)

- An enhanced version of the ECB that chains together blocks of ciphertext.
- The CBC mode encrypts each block using the plaintext, the key and the output of the previous block (except the first in the cycle which uses an Initializing Vector (IV)).
- The CBC mode has an advantage over the ECB mode in that repeating blocks are hidden.
  - See the diagram below.
- The CBC is frequently employed in generating Message Authentication Codes (MACs), frequently referred to as Message Digests, which are a type of cryptographic checksum used to ensure message integrity.
  - The MAC consists of the last block of ciphertext and is generally sent along with a plaintext version of the message.
  - The rest of the ciphertext message is discarded when all that is desired is a checksum (MAC).



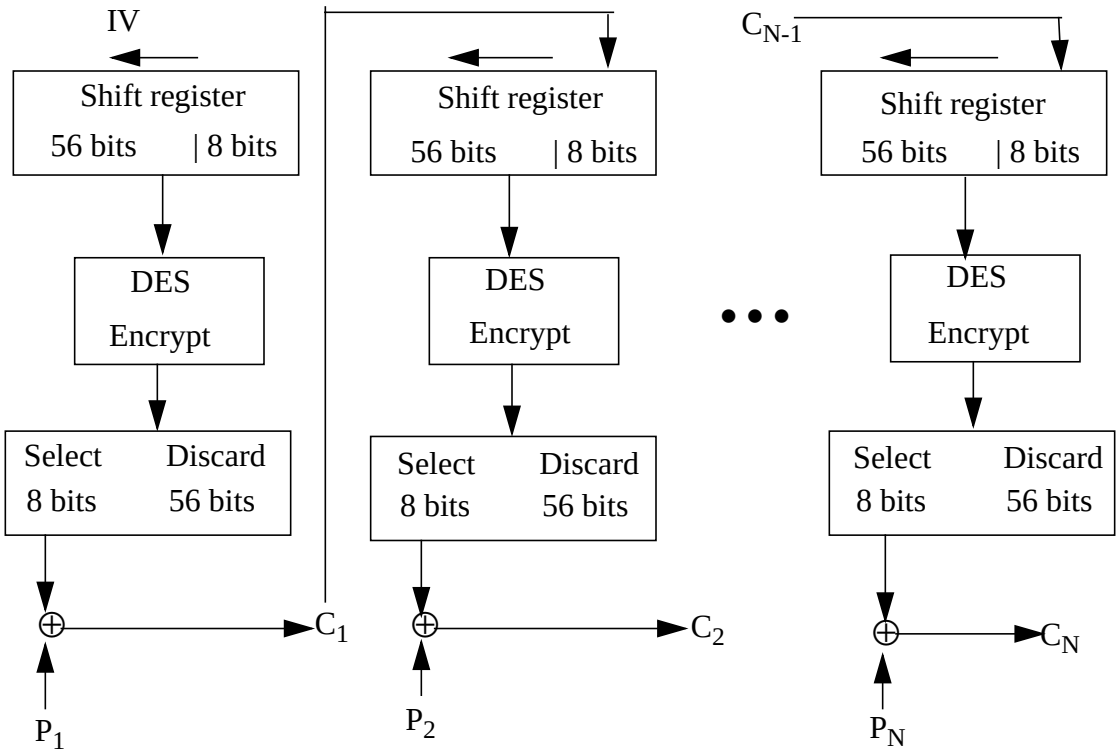
- Because of the chaining affect on blocks, an error in transmission will cause decryption errors in subsequent blocks.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

CFB Mode

Cipher Feedback Mode (CFB)

- This mode uses the block nature of DES in a way that produces a stream cipher.
- Stream ciphers act on small chunks of data, usually 8-bit chunks.
- It eliminates the need to pad messages into 64-bit blocks.
- It can operate in real-time. That is, each character can be encrypted and transmitted immediately.
- The scheme requires an Initial Vector (IV) to start the process.
- In the diagram below, P_i is an 8-bit piece of plaintext and C_i is the corresponding 8-bit piece of ciphertext.
- Notice that the plaintext nevers gets directly processed by the DES Encryption algorithm. Instead, it is XORed with the output of the DES Encryption algorithm.

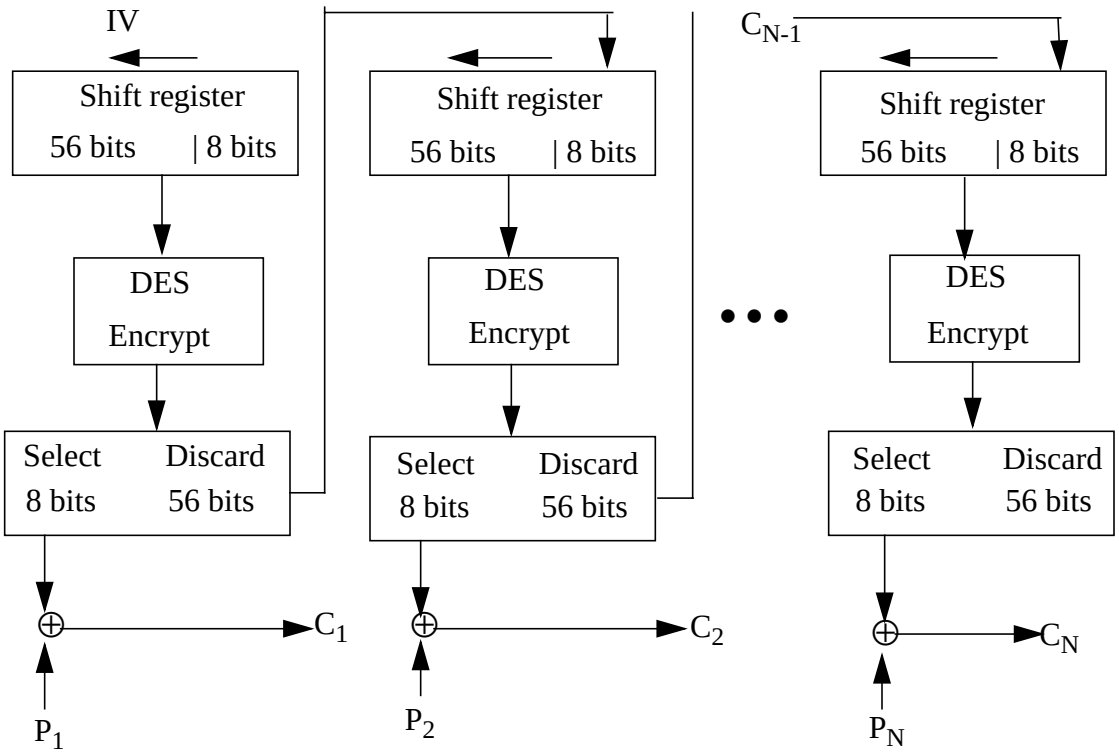


Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

OFB Mode

Output Feedback Mode (OFB)

- This mode is similar in operation to the CFB mode.
- The difference is that in CFB mode the previous 8-bit chunk of ciphertext is shifted into the shift register and used as input to DES and in OFB mode the selected 8-bits of DES output are shifted into the shift register.
- One advantage of OFB mode is that errors in transmission do not propagate.



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

DES Decryption

The process of decryption with DES is essentially the same as the encryption process.

The same algorithm and key are used for encryption and decryption, except that during decryption the internal keys (K_i) are used in reverse order.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

DES Issues

Criticisms of the DES

- Number of iterations - is 16 enough?
- Key length
  - 2^{56} possible keys to try.
  - Massively parallel system could try all keys in 1 day (although it would be a very expensive proposition).
  - *According to an article in the October 1, 1996 San Jose Mercury News, a government agency can break a DES encrypted message in 12 seconds.*
  - Triple encryption may be the answer. See below.
- NSA involvement - Do they hold a 'trapdoor'?

Weaknesses of the DES

- Weak keys (e.g. all zeros or all ones).
- Semi-Weak keys (2 separate keys can decrypt the same message).
- The same DES algorithm is used!
- Key length

Triple DES

- Provides an effective key length of 112 bit key (i.e. independent 56 bit keys); thereby making a brute force attack infeasible.
- Most common variant is EDE mode (encrypt-decrypt-encrypt for encryption and decrypt-encrypt-decrypt for decryption).
  - Encrypt plaintext with DES using key #1.
  - Decrypt resulting cipher text with key #2.
  - Encrypt resulting Ciphertext with key #1.
- About half as fast as standard DES
- The keys are used in reverse order.

Bit Sensitivity

- If only one bit of either the input plaintext or key is changed, each bit of the ciphertext is affected.

A New Standard

- DES will soon be replaced as the national encryption standard.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

International Data Encryption Algorithm (IDEA)

Overview

- Operates on 64-bit plaintext block.
- Uses 128 bit key.
- Same algorithm is used for encryption and decryption (like DES).
- Considered by some to be superior to DES
- It is a symmetric algorithm.

General Description

- 64 bit input block is divided into four 16 bit blocks: X1, X2, X3, and X4 which become the input blocks to the first round of the algorithm.
- In each of the eight total rounds, the four sub-blocks are XORed, added, and multiplied with one another and with six 16 bit sub-blocks of key material.
- Between each round the second and third sub-blocks are swapped.

Speed of IDEA

- Software implementation speeds are comparable with those for DES.
- Hardware implementations are just slightly faster.

Want to know more?

If you are interested in learning more about IDEA and other cryptographic techniques then you might want to read the following book:

Applied Cryptography

Protocols, Algorithms, and Source Code in C,

Second Edition
by

Bruce Schneier

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Skipjack Algorithm

Overview

- Developed by NSA.
  - started design in 1985 and finished evaluation in 1990
- Developed for use by Clipper and Capstone.
- Actual algorithm is classified SECRET.
  - To prevent the construction of devices that will interoperate with Skipjack devices, but which don't support the "Law Enforcement Field" mechanisms.

General Description

- It is a symmetric algorithm.
- It has an 80 bit key and encrypts 64-bit blocks of plaintext.
- It can be used in either ECB, CFB, OFB, or CBC modes.
- There are 32 rounds of processing per single encrypt or decrypt operation.
- The strength of Skipjack does not merely depend upon the secrecy of the algorithm (like any good cipher).

Speed of Skipjack

- The algorithm was designed to achieve high data throughput for use in real-time communications system.

Skipjack Issues:

- It is intended to only be implemented in a tamper-proof chip.
- It is also intended that the implementation will provide a Law Enforcement Field (LEAF) that will enable law enforcement agencies to decrypt encrypted messages.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Symmetric Versus Asymmetric Algorithms

Two Categories of Encrypting Algorithms

- Encryption algorithms can be divided into two categories:
  - Symmetric key algorithms or ciphers
  - Asymmetric key algorithms or ciphers

Symmetric Key Ciphers:

- In symmetric key ciphers, both the encryption algorithm and the decryption algorithm use the same key.
- All ciphers discussed so far, including DES, are symmetric key algorithms.
- Other names for symmetric ciphers:
  - Private key.
  - Secret key.
  - Single key.
  - Shared key.
  - Conventional encryption.
- Symmetric key schemes require both the sender and receiver to possess the same key.
  - I.e., the key must be securely distributed.
- The amount of information a cryptanalyst can gain about a key is directly proportional to the number and length of messages encrypted with the key.
- For security reasons, keys should be changed periodically, which means that keys need to be securely distributed fairly often.

Asymmetric Key Ciphers:

- In asymmetric key ciphers the key used for encryption is different from the key that is used for decryption.
- Other names for asymmetric ciphers:
  - Public key.
- Asymmetric key ciphers do not require the secure distribution of keys.
  - They, however, have other key distribution problems.
  - These problems are discussed on future slides.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Public-Key Cryptography

Asymmetric Key Cryptography

- Public key cryptographic systems use two keys, one private and one public key, to make the necessary transformations.

Summary of Public Key Protocols

- Each user generates two keys - a public key and a private key.
- Each user keeps the private key in a secure manner.
- Each user gives the public key to everyone else.

Example for sending a secret.

- Alice wants to send a message to Bob, such that if the message is intercepted it cannot be read.
- Alice has Bob’s public key. (Only Bob has Bob’s private key!)
- Alice encrypts the message with Bob’s public key.
- Alice sends the encrypted message to Bob.
- Bob uses his private key to decrypt the message.
- If the message is intercepted in transit, it can only be decrypted by someone who has Bob’s private key and no one but Bob has Bob’s private key.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

RSA Encryption

Rivest-Shamir-Adelman (RSA) Encryption

- Two key system (public and private) based on the difficulty of factoring very large numbers.
  - Encryption $C \equiv P^e \bmod n$
  - Decryption $P \equiv C^d \bmod n$
- Key_e and Key_d are carefully chosen such that:

$$P \equiv (P^e)^d \bmod n \equiv (P^d)^e \bmod n \text{ (i.e. E(D(M)) = D(E(M)) = M)}$$

Choosing Keys for RSA Method

- Underlying problem is based on factoring very large numbers!

  Encryption $C \equiv P^e \bmod n$

  Decryption $P \equiv C^d \bmod n$

  where

  Encryption key = (e, n)

  Decryption key = (d, n)

- The first task is to select n
  - n is normally very large (approx 200 digits)
  - n is a product of two large primes p and q (typically 100 digits each)
- Next a large integer e is chosen such that
  - e is relatively prime to $(p\text{-}1) * (q\text{-}1)$
  - I.e., e and $(p\text{-}1) * (q\text{-}1)$ have no factors in common.
  - e is usually picked as a prime larger than both $(p\text{-}1)$ and $(q\text{-}1)$

- Next select d such that: $e * d \equiv 1 \bmod (p\text{-}1) * (q\text{-}1)$
- Then if we have selected our numbers correctly,

A MINOR MIRACLE OCCURS

.

.

.

$(P^e)^d \equiv P \bmod n$

Note: Minor miracle provided courtesy of Euler and Fermat.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

RSA Encryption

Summary of RSA Encryption

Public Key

n => product of two primes, p and q
(p and q remain secret)
e => relatively prime to $(p\text{-}1) * (q\text{-}1)$

Private key

$d \equiv e^{-1} \pmod{(p\text{-}1) * (q\text{-}1)}$

Encrypting

$c \equiv m^e \pmod{n}$

Decrypting

$m \equiv c^d \pmod{n}$

Example: RSA Encryption

Let $p = 11$ and $q = 13$ {both primes}

Then $n = p * q = 143$ and
$(p\text{-}1) * (q\text{-}1) = 10 * 12 = 120$

Next choose e such that it is relatively prime to $(p\text{-}1) * (q\text{-}1)$. We will choose 11!

Recall that: $d \equiv e^{-1} \pmod{(p\text{-}1) * (q\text{-}1)}$
$d * e \equiv 1 \pmod{(p - 1) * (q - 1)}$

In other words: $11 * 11^{-1} \pmod{120} \equiv 1 \pmod{120}$
$121 \bmod 120 \equiv 1 \bmod 120$

In this case both e and d are the same (11)

Let the plain message P be the letter 'H'
(7 in our 0-25 schema)

E('H') = E(7) => $7^{11} \bmod 143 \equiv 106$
D(106) = $106^{11} \bmod 143 \equiv 7$ => 'H'

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Introduction to Hash Functions

Hash Functions and Message Digests.

- A hash function H accepts a variable-size message M as input and outputs a fixed-size representation H(M) of M, sometimes called a message digest. In general H(M) will be much smaller than M; e.g., H(M) might be 64 or 128 bits, whereas M might be a megabyte or more.
- A hash function can serve to detect modification of a message. That is, it can serve as a cryptographic checksum (also known as an MDC = manipulation detection code or MAC = message authentication code).
- It is theoretically possible that two distinct messages could be compressed into the same message digest (a collision). The security of hash functions thus requires collision avoidance. Collisions cannot be avoided entirely, since in general the number of possible messages will exceed the number of possible outputs of the hash function. However, the probability of collisions must be low.

Properties of Hash Functions

- To serve the authentication process properly a hash function F must have the following properties:
  - **a.** F can be applied to an argument of any size.
  - **b.** F produces a fixed-size output.
  - **c.** F(x) is relatively easy to compute for any given x.
  - **d.** For any given y it is computationally infeasible to find x with F(x) = y.

    Property (d) guarantees that an alternative message hashing to the same value as a given message cannot be found. This prevents forgery and also permits F to function as a cryptographic checksum for integrity.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Example Hash Function

Ultra simple example:

- Divide the message up into 8-bit chucks.
- Pad the beginning with zeros if necessary.
- Interpret each 8-bit block as a number.
- The values will be between 0 and 255.
- The message M is M_1,M_2,M_3, ..., M_N
- The hash of message M (hash(M)) $\equiv (M_1^2 + M_2^2 + M_3^2 + \cdots + M_N^2) \bmod 256$
- M = 11001010,00101010,01011100
- hash(M) $\equiv ((202)^2 + (42)^2 + (92)^2) \bmod 256$
- hash(M) $\equiv (40804 + 1764 + 8464) \bmod 256$
- hash(M) $\equiv (51032) \bmod 256$
- hash(M) = 88
- How hard is it to find another message that also hashes to 88?

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

MD4/5 and MD2

MD4/MD5 and MD2

- MD4 is a one-way hash function designed by Ron Rivist. MD stands for Message Digest, and the algorithm produces a 128-bit hash, or message digest, of the input message.
- Rivest's goals for the design of MD4 algorithm were:
  - Security - It should be computationally infeasible to find two messages that hash to the same value. No attack should be more efficient than brute force.
  - Direct Security - Not to be based on any fundamental assumption like the difficulty of factoring.
  - Speed - Should be suitable for high speed 32 bit software implementations.
  - Simplicity - Should be as simple as possible without large data structures.
  - Architecture - favor microprocessor architectures (specifically Intel microprocessors).
- After a successful attack was made on the first three rounds of the algorithm was achieved in 1990, Rivest strengthened the algorithm which is now known as MD5 (MD5, to date is considered secure).
- MD2 is another one-way hash function also created by Rivest and is used as the basis for Privacy Enhanced Mail (PEM).

General Description of MD5

- After some initial processing, MD5 processes the input text in 512-bit blocks, divided into sixteen 32-bit sub-blocks. The output of the algorithm is a set of four 32-bit blocks, which concatenate to form a single 128-bit hash value.
- The main loop, which continues for as many 512-bit blocks as there are in the message, consists of four rounds of sixteen operations each. Each operation performs a nonlinear function on three of four 32 bit variables. The result is then added to the fourth variable.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Secure Hash Algorithm (SHA)

Secure Hash Algorithm (SHA)

- NIST, with assistance from NSA, designed the Secure Hash Algorithm (SHA) for use with the Digital Signature Algorithm (DSA). The standard is known as the Secure Hash Standard (SHS).
- When a message of any length is input, The SHA produces a 160-bit message digest.
- SHA is very similar in operation to MD4. It differs in that it adds an additional expansion operation, an extra round and the whole transformation was designed to accommodate the DSS block size for efficiency.
- Most cryptographers feel that the SHA is more secure than MD5 because of its fundamental design as well as its resistance to brute force attack on the 160-bit message digest versus the 128-bit digest produced by MD5.

Want to know more?

If you are interested in learning more about hash functions and other cryptographic techniques then you might want to read Chapter 14 of the following book:

Applied Cryptography

Protocols, Algorithms, and Source Code in C

Second Edition
by

Bruce Schneier

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

This page is intentionally blank.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

This page is intentionally blank.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

This page is intentionally blank.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Section 8

Cryptographic Protocols

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Protocols

The Purpose of Protocols

- In daily life, there are informal protocols for almost everything that we do; ordering over the telephone, playing poker, exchanging cash for products or services, banking, voting in an election, to mention just a few. We don't give much thought to these protocols since they have evolved over time and everyone knows how to use them and how they work.
- Increasingly, people are communicating and doing business over computer networks rather than face-to-face as they have in the past. Many face-to-face protocols rely on people's presence to ensure fairness and security; however, in our new cyber world we do not have that luxury. It is therefore necessary to develop formal protocols for computing that can ensure that business is conducted fairly, honestly and securely.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Protocols

Protocol

An orderly sequence of steps taken by two or more parties to accomplish some task.

- Characteristics
  1. Established in advance
  2. Mutually subscribed
  3. Unambiguous
  4. Complete

Arbitrator Protocol

- Arbiter
  - A Trustworthy, disinterested third party.
  - Directly involved in transaction.
  - A person, program or machine.
- Disadvantages
  - Suspicion
  - Cost
  - Delay
  - Bottleneck
  - Secrecy

Adjudicated Protocols

- Adjudicator
  - A third party who judges whether a transaction was conducted fairly.
  - A notary public
- Disadvantages
  - Detects failure to cooperate after the fact.

Self Enforcing Protocols

- Advantages
  - Guarantees fairness
  - No Outsider needed

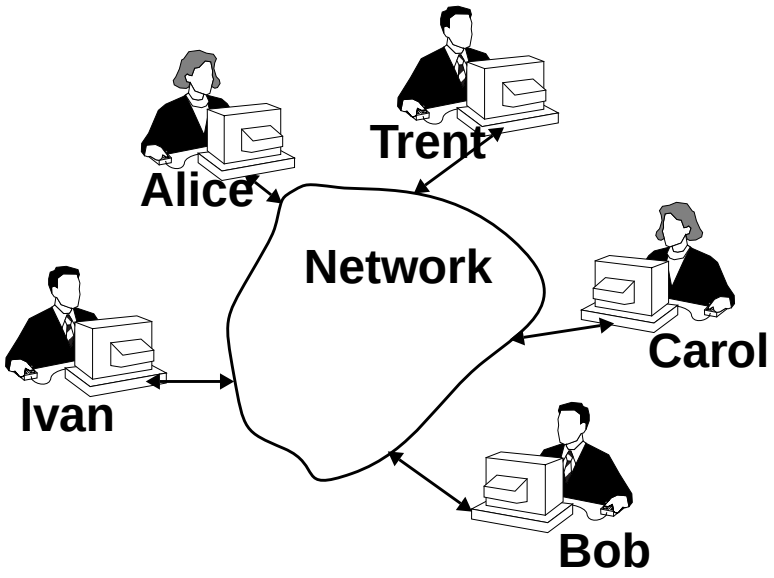
Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Using DES To Support Secrecy

Alice wishes to send a secret message M to Bob.

Both Alice and Bob possess $\mathbf{K}_{AB}$

1. Alice encrypts M with the DES key $\mathbf{K}_{AB}$, producing M'
2. Alice transmits M' to Bob
3. Bob decrypts M' with the DES key $\mathbf{K}_{AB}$, producing M



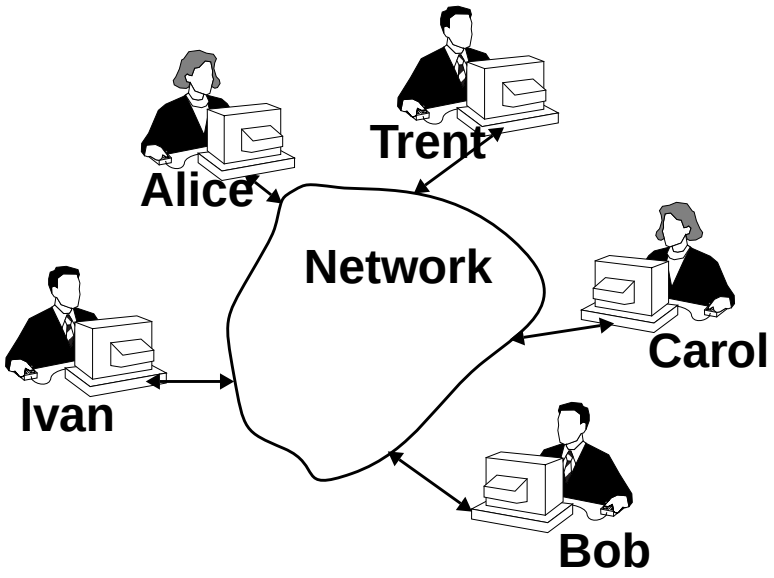
- Since both only Alice and Bob share the key $\mathbf{K}_{AB}$, nobody else can read Alice's secret message M, unless they have managed to obtain a copy of the private key $\mathbf{K}_{AB}$.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Using DES To Support Authenticity

Alice wishes to send a message M to Bob such that Bob is assured that the message could only have originated from Alice.

1. Alice encrypts M with the DES key $\mathbf{K}_{AB}$, producing M'
2. Alice transmits M' to Bob
3. Bob decrypts M' with the DES key $\mathbf{K}_{AB}$, producing M



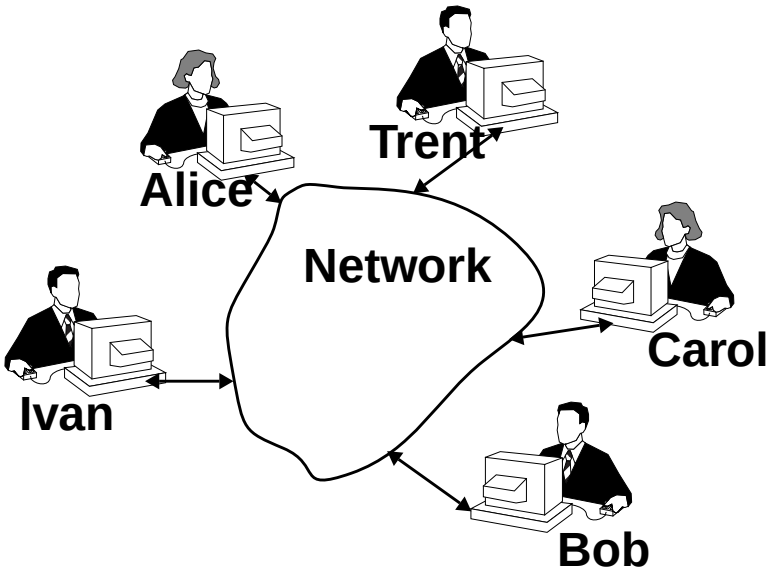
- Since both only Alice and Bob share the key $\mathbf{K}_{AB}$, Bob knows that Alice is the only person who could have sent M. Nobody else could have sent it, unless they have managed to obtain a copy of the private key $\mathbf{K}_{AB}$.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Using DES To Support Integrity

Alice wishes to send a message M to Bob such that Bob is assured that the message was not modified during its transit from Alice to Bob.

1. Alice encrypts M with the DES key $\mathbf{K}_{AB}$, producing M'
2. Alice encrypts M with the DES key $\mathbf{K}_{AB}$, using CBC mode and produces a 64 bit MAC
3. Alice transmits both M' and the 64 bit MAC to Bob
4. Bob decrypts M' with the DES key $\mathbf{K}_{AB}$, producing M
5. Bob encrypts M with the DES key $\mathbf{K}_{AB}$, using CBC mode and produces a 64 bit MAC'
6. Bob compares MAC and MAC' to see if they are the same.



- Since both MAC and MAC' are the same Bob knows that the message M has not been altered.

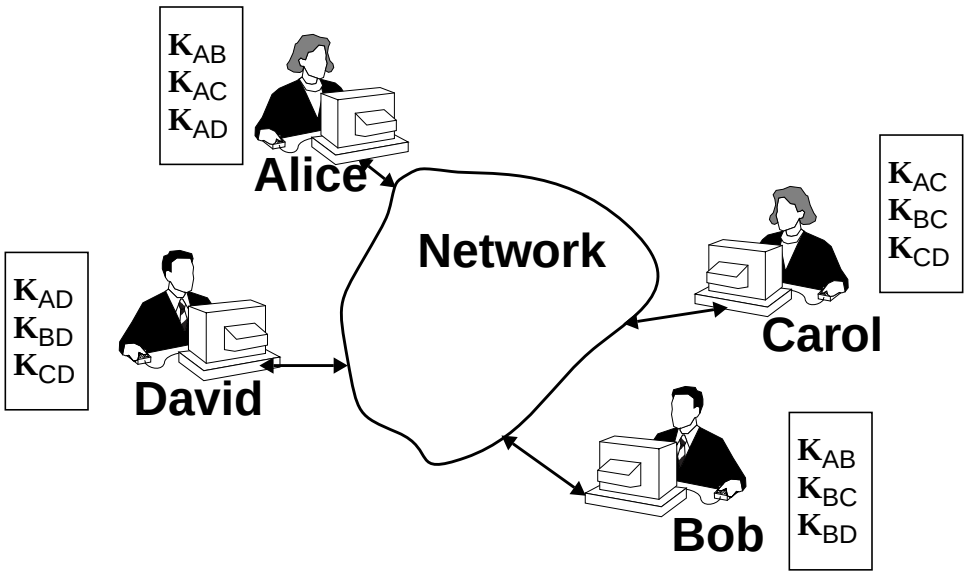
Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Disadvantages of Conventional Key Systems

- With a conventional key system a separate key is needed for every pair of users.

  $n * (n\text{-}1)/2$ keys are required for n users.

Example
3 users requires three keys k_{AB}, k_{AC} and k_{BC} 4 users requires six keys k_{AB}, k_{AC}, k_{BC}, k_{AD}, k_{BD} and k_{CD} In general, we are choosing from *n* items *k* at a time or: $n * (n\text{-}1)/2$ keys are required



KEY MANAGEMENT IS A NIGHTMARE!!

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Disadvantages of Conventional Key Systems

Distribution of Shared Keys:

- Keys must be distributed in a secure manner.
  - Bonded courier.
  - Registered mail.

- As previously mentioned, the amount of information a cryptanalyst can gain about a key is proportional to the number and length of messages encrypted with the key.

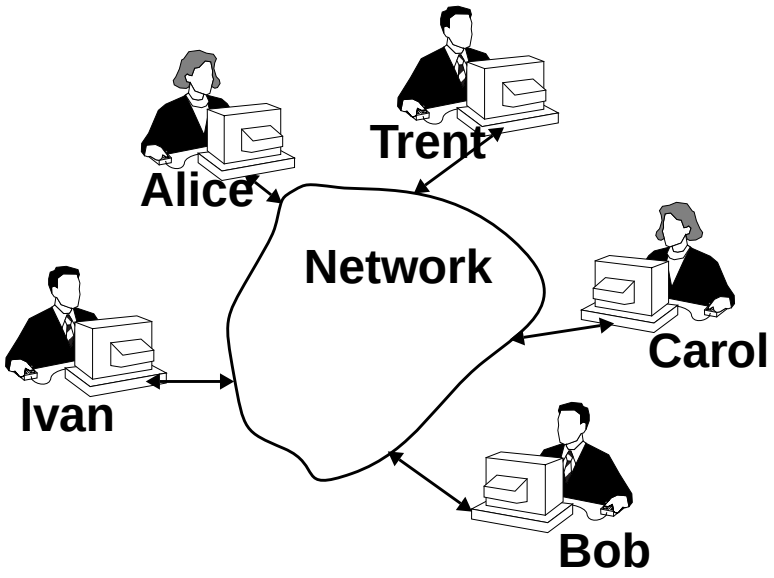
- Thus, for security reasons, keys should be changed periodically, which means that keys need to be securely distributed fairly often.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Using RSA To Support Secrecy

Alice wishes to send a secret message M to Bob.

1. Alice encrypts M with Bob's public key $\mathbf{K}_{BOB\text{-}PUB}$, producing M'
2. Alice transmits M' to Bob
3. Bob decrypts M' with his private key $\mathbf{K}_{BOB\text{-}PRI}$, producing M



Problem!!

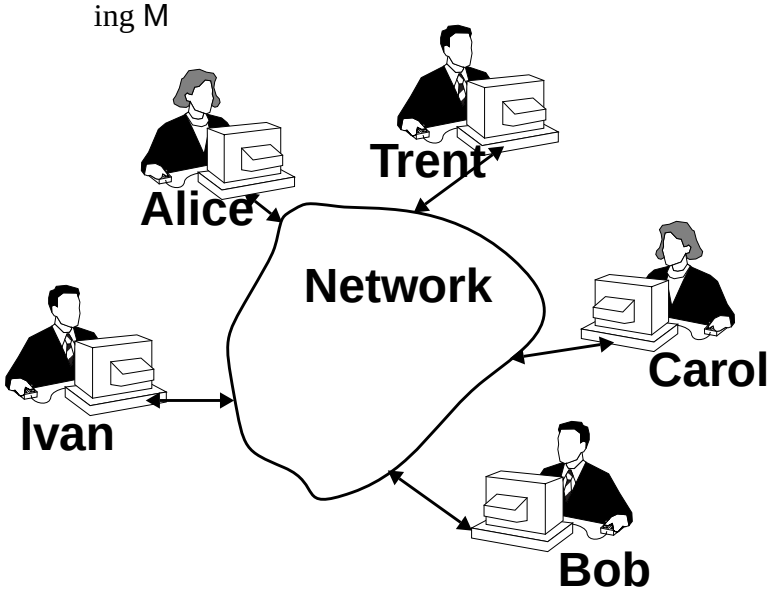
- Nobody else can read M since Bob is the only one who possesses $\mathbf{K}_{BOB\text{-}PRI}$; however, Bob has no assurance that M came from Alice since anyone could have used his public key $\mathbf{K}_{BOB\text{-}PUB}$.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Using RSA To Support Authenticity

Alice wishes to send a message M to Bob such that Bob is assured that the message could only have originated from Alice.

1. Alice encrypts M with Alice's private key $\mathbf{K}_{\text{ALICE-PRI}}$, producing M'
2. Alice transmits M' to Bob
3. Bob decrypts M' with Alice's public key $\mathbf{K}_{\text{ALICE-PUB}}$, producing M



Problem!

- This protocol ensures authenticity, but secrecy is non-existent since anyone can obtain Alice's public key $\mathbf{K}_{\text{ALICE-PUB}}$. Thus M is accessible to an eavesdropper like Ivan.

Problem!

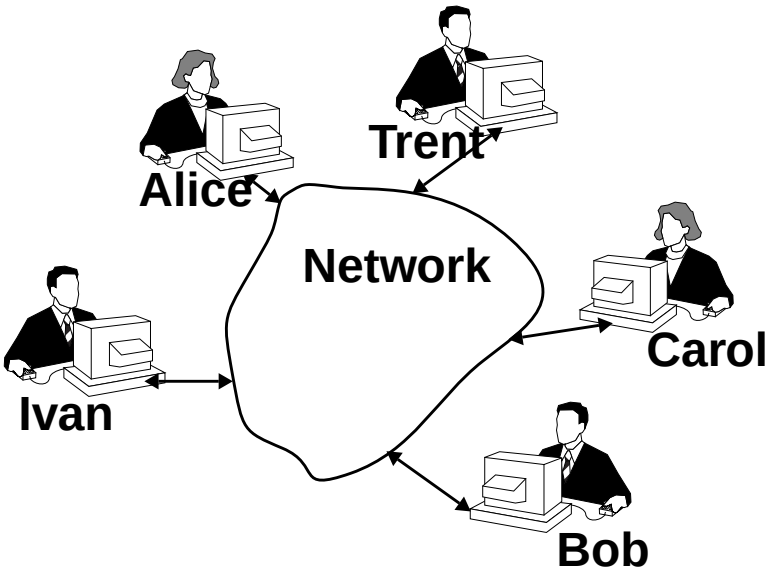
- Bob is still not absolutely certain that M' was not altered in transit.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Using RSA To Support Secrecy, Authenticity and Integrity

Alice wishes to send a secret message M to Bob such that Bob is assured that the message could only have originated from Alice and that the message was not modified during its transit from Alice to Bob.

1. Alice uses a mutually available hash function H to produce a hash $H(M)_{ALICE}$ of the original message M.
2. Alice signs $H(M)_{ALICE}$ by encrypting it with her private key $\mathbf{K}_{ALICE\text{-}PRI}$ producing $H(M)'_{ALICE}$
3. Alice encrypts M with Bob's public key $\mathbf{K}_{BOB\text{-}PUB}$, producing M'.
4. Alice transmits both M' and $H(M)'_{ALICE}$ to Bob
5. Bob decrypts M' with his private key $\mathbf{K}_{BOB\text{-}PRI}$, producing M
6. Bob uses the same hash function H to produce a hash $H(M)_{BOB}$ of the message M he just decrypted in step 5.
7. Bob decrypts $H(M)'_{ALICE}$ using $\mathbf{K}_{ALICE\text{-}PUB.}$
8. Bob compares $H(M)_{ALICE}$ and $H(M)_{BOB}$ to see if they are the same.



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Public-Key Systems Issues

Distribution of Keys:

To some degree, public key systems solve the key distribution problem that private key systems suffer from.

- The number of keys that a large community requires can be greatly reduced if the members of the community use a public key scheme instead of a private key scheme.
- If a community of *n* members uses a public key scheme they will only require *n* private keys and *n* public keys.
  - This is a total of 2*n* keys.
- There is no need for maintaining secrecy when distributing public keys.

Note that there is no need for secrecy, but there is a big need for authenticity.

- Consider the following scenario:
- Alice computes a private key / public key pair and sends the public key to Bob in a message such that Bob thinks that the public key is from Carol.
  - I.e., Bob thinks that this key is Carol's public key.
- Now if Bob wants to send a secret to Carol, he would encrypt the secret with Carol's public key and send it off to Carol.
- Alice intercepts the message an decrypts the message with the private key and reads the secret.
- This example illustrates the need for authenticity when receiving public keys.
  - This problem leads to public key management and certificate authority schemes.
- More on this topic later.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Efficiency Considerations

Public Key Versus Private Key Algorithms:

- Private key algorithms (such as DES and IDEA) are much faster than public key algorithms (such as RSA).

Question:

- How do we take advantage of public key cryptography for key distribution and private key cryptography for bulk encryption?

Answer:

- Use a hybrid scheme (such as PGP).

Hybrid Schemes:

- In a hybrid scheme, a public key algorithm is used to encrypt (and decrypt) a shared key (such as a DES key).
- The message is encrypted with the shared key.
- The encrypted message and the encrypted shared key are transmitted.
- The receiver, decrypts the shared key (using a public key algorithm) and then uses the shared key to decrypt the message.

Analysis of the hybrid scheme:

- Only keys (which are relatively short) are encrypted using the slow public key algorithm.
- The message (which may be very long) is encrypted with the fast shared key algorithm.
- The shared key may only be used for one message.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Digital Signatures

Digital signatures.

- A digital signature is the electronic analogue of a handwritten signature. A common feature is that they must provide the following:
  - A receiver must be able to validate the sender's signature.
  - A signature must not be forgeable.
- There are two major variants of implementation:
  - True signatures.
  - Arbitrated signatures.
- In a true signature system, signed messages are forwarded directly from signer to recipient. In an arbitrated system, a witness (human or automated) validates a signature and transmits the message on behalf of the sender. The use of an arbitrator may be helpful in event of key compromise as noted below.
- Digital signatures provide authentication, nonrepudiation and integrity checks. In some settings authentication is a major consideration; In some cases it is desirable even when secrecy is not a consideration.
- When using a public key scheme, encrypting a message (or the hash of a message) with a private key is effectively "signing" the message, since only one person in the world has the private key.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Key Management in Shared Key Systems

Conventional system key management.

- In a conventional (one-key) system, two users who wish to communicate securely must first securely establish a common key. One possibility is to employ a third party such as a courier. In practice, it may be necessary to establish a new key from time to time for security reasons. This may make use of a courier or similar scheme costly and inefficient.

- An alternative is for the two users to obtain a common key from a central issuing authority with whom each can communicate securely. Security is then a major consideration: a central authority having access to keys is vulnerable to penetration. Due to the concentration of trust, a single security breach would compromise the entire system. In particular, a central authority could engage in passive eavesdropping for a long period of time before the practice was discovered; even then it might be difficult to prove.

- In large networks it might become a bottleneck, since each pair of users needing a key must access a central node at least once. Additionally, failure of the central authority could disrupt the key distribution system. A hierarchical (tree-structured) system, with users at the leaves and key distribution centers at intermediate nodes may be one way to alleviate this problem. However, this creates a new security problem, since a multiplicity of entry points for intruders is created. Furthermore, it might be inefficient unless pairs of users communicating frequently were associated to a common subtree; otherwise the root of the tree would, once again, become a bottleneck.

- Some of these disadvantages can also be mitigated by a public-key approach to key distribution.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Key Management in Public Key Systems

Public-Key System Key Management.

- Prior to using a public-key cryptosystem for exchanging conventional secret keys, users Alice and Bob must exchange their public keys. It is a simpler problem than exchanging secret keys, since public keys do not require secrecy in storage or transit. Public keys can be managed by an on-line or off-line directory service; they can also be exchanged directly by users. However, authenticity is an issue. If Alice thinks that $\mathbf{K}_{\text{IVAN-PUB}}$ is really $\mathbf{K}_{\text{BOB-PUB}}$ then Alice might encrypt using $\mathbf{K}_{\text{IVAN-PUB}}$ and inadvertently allow Ivan to decrypt using $\mathbf{K}_{\text{IVAN-PRI}}$. A second problem is integrity: any error in transmission of a public key will render it useless. Hence some form of error detection is desirable. Regardless of the scheme chosen for public key distribution, at some point a central authority is likely to be involved. However, exchange of public keys between users need not involve the central authority, since the primary concern is with authenticity. Therefore, the implications of compromise of the central authority are not as severe as it would be in a conventional key system.
- Validity is an additional consideration: a user's public key may be invalidated because of compromise of the corresponding private key, or for some other reason such as expiration. This creates a stale-data problem in the event that public keys are stored or accessed through a directory.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Nonces and Timestamps

Replay attacks

Some protocols are vulnerable to a replay attack, even though the protocol may use cryptography.

Replay attacks are characterized by an attacker replaying one or more packets that the attacker has captured at an earlier time. In many instances the attack may be successful even though the attacker cannot read or modify the packets.

Nonces or timestamps are used to prevent this class of attacks.

Definition

- A *nonce* is a random number that is often used in protocols to prevent the "replay" problem.

Example use:

- Alice generates a new nonce and sends a message containing the nonce. If she then receives a return message containing the same nonce value, she knows that the return message is not a replay of a previous message that had been sent to her.

Example replay attack:

Every day Bob sends Alice troop information. They use a symmetric key algorithm to prevent disclosure of this information.

Ever day Alice generates a symmetric key K_{AB} and distributes the key to Bob by sending him the message:

$((K_{AB})Alice_{PRIVATE})Bob_{PUBLIC}$

No one can read this message, other than Bob, and Bob is able to verify that it came from Alice.

After Bob receives this message, he decrypts it and retrieves the key K_{AB}. After obtaining this key, Bob uses this key to send troop movement information to Alice.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Nonces and Timestamps continued

One day an attacker eavesdrops on the microwave communication link and records this message and all the subsequent encrypted troop movement traffic that Bob sends to Alice.

After 5 months work the attacker eventually determines K_{AB} by using brute force on the troop movement information that Bob sent Alice that day. Note that the attacker **was not able** to break the public key scheme.

Now when the attacker wants to eavesdrop on troop movement information, the attacker sends Bob a copy of the message that Alice sent to Bob 5 months ago, namely

$((K_{AB})Alice_{PRIVATE})Bob_{PUBLIC}$

When Bob receives this message he decrypts it and verifies that it came from Alice. He starts sending the troop movement data off to Alice using K_{AB}, which is the same key that he used 5 months ago (but he does not know it).

Since the attacker has K_{AB} the attacker can read the information in real time.

To prevent this kind of attack, Alice must include a timestamp or nonce in the key distribution message that she sends to Bob. If the protocol uses a nonce, then Bob must record ever nonce that he has ever received. Every time he receives a new message he checks to see if the nonce in the message is the same as the nonce in some previous message. If it is, he assumes the message is a replay attack and discards the message.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Key Management Via Certificates

Use of Certificates in Public Key Systems.

- A technique to obtain a solution to both authenticity and integrity in distribution of public keys is the use of certificates. A certificate-based system assumes a central issuing authority CA as in the secret-key case. Again it must be assumed that each user can communicate securely with the CA. This is relatively simple since it merely requires that each user possess $\mathbf{K}_{\text{CA-PUB}}$, the public transformation of the CA. Then Alice may register $\mathbf{K}_{\text{ALICE-PUB}}$ with the CA. Since $\mathbf{K}_{\text{ALICE-PUB}}$ is public, this might be done via the postal service, an insecure electronic channel, a combination of these, etc.
- Normally Alice will follow some form of authentication procedure in registering with the CA. Alternatively, registration can be handled by a tree-structured system: the CA issues certificates to local representatives (e.g., of employing organizations), who then act as intermediaries in the process of registering users at lower levels of the hierarchy.
- In any case, in return Alice receives a certificate signed by the CA and containing $\mathbf{K}_{\text{ALICE-PUB}}$. That is, the CA constructs a message M containing $\mathbf{K}_{\text{ALICE-PUB}}$, identification information for A, a validity period, etc. Then the CA computes $\text{CERT}_{\text{ALICE}} = \mathbf{K}_{\text{CA-PRI}}(\text{M})$ which becomes Alice's certificate. $\text{CERT}_{\text{ALICE}}$ is then a public document which contains both $\mathbf{K}_{\text{ALICE-PUB}}$ and authentication, since the certificate is signed by the CA. Certificates can be distributed by the CA, by users, or used in a hierarchical system. The inclusion of the validity period is a generalization of timestamping. The importance of timestamping is in guarding against the use of compromised keys.
- However, the problem of stale data is not wholly solved by timestamping: a certificate may be invalidated before its expiration date, because of compromise or administrative reasons. Hence if certificates are cached by users (as opposed to being redistributed by the CA each time they are used), the CA must periodically issue lists of invalidated certificates.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Key Management Via Certificates

A phone-book approach to certificates.

- Some of the features of the previous schemes could be combined in a phone-book approach, using an electronic equivalent such as a floppy disk containing certificates. This would optimize ease of use since a user could communicate securely with another by accessing the latter's certificate very rapidly. However, again the central authority would have to issue “hot lists”. Periodic distribution of the compilations of certificates would be a separate management process. Additionally, the security of such a scheme is clearly in question since phone-book entries might contain errors, or entries could be altered.

Certificate

Name:	Alice
Address:	alice@host.domain
Date Issued:	950901
Date Expires:	960831
Public Key:	clpg55kzxplvwqlfdrg tuoksfidolbkfgcsdarp 0qxsjhrxfjsdf2yun0ql dxklrtortwdgrr6ee8l4

Signed: Mr. Trustworthy Trent

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Key Management Via Certificates

Decentralized Management.

- Users may be responsible for managing their own certificates. In this event the protocol is much simpler. When A wishes to initiate communication (e.g., exchange of a secret key) with B, he sends a message to B containing A's certificate, A's identification, and other information such as a date, random number etc. as described in the protocol in the previous section. This message also requests B's certificate. Upon completion of the certificate exchange, employing some protocol such as the handshake above, A and B will possess each other's authenticated certificates. A can validate the certificate C_B by decrypting the certificate with $\mathbf{K}_{CA\text{-}PUB}$. Then $\mathbf{K}_{B\text{-}PUB}$ may be retrieved. The certificate must contain information properly identifying B to A, so that an intruder cannot masquerade as B. The certificate must also have a validity period. In turn B may proceed similarly.
- The central authority must periodically issue lists of certificates which have become invalid before their expiration dates due to key compromise or administrative reasons. It is likely that in a large system this would be done, e.g., on a monthly basis. Hence a user receiving a certificate from another user would not have complete assurance of its validity, even though it is signed by the CA. Thus this system trades higher efficiency for some security loss, compared to the previous scheme.
- For greater assurance of validity, a user could access a centrally-managed list of invalid certificates; presumably these would be very current. This would require on-line access to a central facility, which would create the same type of bottleneck we have noted previously. However, the accesses would be very quick, since presumably only a certificate sequence number would be transmitted.
- Another on-line possibility would be for the central authority to enforce coherence by a global search of cached certificates each time a certificate is invalidated. Again there is a trade-off of validity assurance and efficiency.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Certificate Example

Sample PGP certificate and signed message hash

- -----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.6.i

mQCNAi+btRkAAAEEAKxQ9HwqfsQc9apOIQmFTo2wqbCL6Q1xlvN6CjxkBbtviaLq
EgmVPnb/FGD5wwxDMjCCJDwBFfLLRwASQAyyy5RjukkZx1Gn8qHzmoyIOVTFOIJI
TFDWyVjMSSvUKACDqXv/xVFunsPlPc7d6f4MwxD1kw2BBpoV7k64di/cua4BAAUR
tCRTc2ggZGlzdHJpYnV0aW9uIGtleSA8eWxvQGNzLmh1dC5maT6JAJUCBRAvm7Vv
qRnF8ZYfSjUBAW7pBACQ7G2pYStkBM5aOK2udb/m/YAAZ/NlY2emSgEJfYrAysSY
0yfbhKGt0K59fGSotmSRcMOpq0tgTMm7lQjsUr5ez1Ra/0Dv7e3xoGQYJ8764X9w
popC+u9JuxLeGTtgWYwPUZIHFcQanZslUmCDr36kvesx/2wXBf8+StghMbA3vw==
=aGik
- -----END PGP PUBLIC KEY BLOCK-----

-----BEGIN PGP SIGNATURE-----
Version: 2.6.i

iQCVAgUBMAPhQqkZxfGWH0o1AQHgngP/dbcRUFqJF549VvVOWgDtAxu/UoO6hnei
26/OpczgH6j8+6fZh8TV81yVAh95K6EhHsKo85j5hXTmKSG3xLn6fw26q1DPGHpQ
Sa4xQ4oL20qcvgOeaEi3gZxxTD5etzdl8eBNbe8vSIkk91yrsAiZL7h8St7UHGsA
N5WqXSMI8pg=
=tXr9
-----END PGP SIGNATURE-----

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

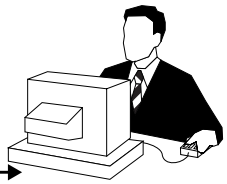
Key Distribution Example

Exchange of DES key using a one-way authentication protocol with certificates:

1. Alice chooses some random string R_{ALICE}

2. Alice constructs $M = K_{ALICE-PRI}(T_{ALICE}, R_{ALICE}, ID_{BOB}, K_{BOB-PUB}(K_{AB}))$

3. Alice sends Bob ($CERT_{ALICE}$, M)

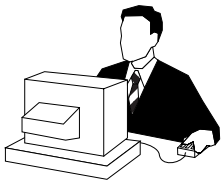


4. Bob decrypts $CERT_{ALICE}$ with K_{CA-PUB} to obtain $K_{ALICE-PUB}$

5. Bob decrypts M with $K_{ALICE-PUB}$ to obtain (T_{ALICE}, R_{ALICE}, ID_{BOB})

6. Bob verifies T_{ALICE}, R_{ALICE}, ID_{BOB}

7. Bob checks R_{ALICE} to make sure that M is not a replay.



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

CLIPPER

The CLIPPER Chip

- The CLIPPER chip is one implementation of the SKIPJACK algorithm. The Clipper chip designed for the AT&T commercial secure voice products has the following characteristics.
  - Functions specified by NSA; logic designed by MYKOTRONX; chip fabricated by VLSI, Inc.
  - Resistant to reverse engineering against the most sophisticated techniques which may be used by a well funded adversary.
  - 15-20 MB/S encryption/decryption rate once synchronization is established.
  - The chip programming equipment writes (once) the following information into a special memory on the chip.
  - serial number (unique) unit key (unique) family key specialized control software
    1. Upon generation (or entry) of a session key in the chip, the chip performs the following actions.
    2. Encrypts the 80-bit session key under the unit key producing an 80-bit intermediate result.
    3. Concatenates the 80-bit result with the 25-bit serial number and a 23-bit authentication pattern (a total of 128 bits).
    4. Enciphers this 128 bits with the family key to produce a 128-bit cipher block chain called the Law Enforcement Access Field (LEAF).
    5. Transmits the LEAF at least once to the intended receiving Clipper chip.
    6. If law enforcement agencies want to decrypt the session, they intercept the LEAF and messages of this session. They use the LEAF to obtain the session key from the Escrow authorities.
    7. The two chips use this field together with a random initialization vector (IV) to establish cryptographic synchronization.
    8. Once synchronized, the CLIPPER chip uses the session key to encrypt/decrypt data in both directions.
- The chips can be programmed to not enter secure mode if the LEAF has been tampered with.
- CLIPPER chip prices currently cost $16 unprogrammed and $26 programmed.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

CLIPPER

The Great Debate - Key Escrowing

- The CLIPPER chip was intended to protect private communications while at the same time permitting government agents to obtain the keys upon presentation of legal authorization. The the two halves of the unique unit key are to be held by two separate escrow agents and would allow the government to access the encrypted private communications. The use of CLIPPER, so far, is voluntary; however, government agencies are being strongly encouraged to adopt its use.
- The National Institute of Standards and Technology (NIST) and the Department of the treasury were designated on February 4, 1994 as the two escrow agents that will hold the keys in escrow.
- The topic of key escrowing has been the subject of a great deal of heated debate over the past two years. On one side are those that feel that individual privacies are at risk and those that argue that law enforcement officials must be given the technological ability to do their jobs. The most notable proponent for key escrowing is Dorothy Denning who believes that CLIPPER is necessary to stop crime. The Electronic Freedom Foundation (and notably John Perry Barlow) adamantly opposes the key escrowing concept because the temptation for government to usurp the rights of private individual's rights is too tempting.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Anonymous Key Distribution

Anonymous Key Distribution

- Consider the problem of key distribution. If we assume that people do not have the ability to generate their own keys then they must use the services of a Key Distribution Center (KDC). The problem is that the keys must be distributed in such a fashion that no one may determine who gets what keys.

1. Alice generates a public/private key pair (for this protocol, she keeps both keys secret).
2. The KDC generates a continuous stream of keys.
3. The KDC encrypts the keys, one-by-one, with its own public key.
4. The KDC transmits the encrypted keys, one-by-one, onto the network.
5. Alice chooses a key at random.
6. Alice encrypts the chosen key with her public key.
7. Alice waits for a while and sends the double encrypted key back to the KDC.
8. The KDC decrypts the double encrypted key with its private key, leaving a key encrypted with Alice's public key.
9. The KDC sends the encrypted key back to Alice.
10. Alice decrypts the key with her private key.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Nonrepudiation

Nonrepudiation.

- Nonrepudiation is contingent upon users keeping their private keys secret. If Alice's private key $\mathbf{K}_{\text{ALICE-PRI}}$ should be compromised, then Alice might be able to repudiate messages sent even before the compromise.
- The use of a central authority is suggested for this purpose. In this scheme, the receiver of a message sends a copy to the central authority. The latter can attest to the instantaneous validity of the sender's signature (i.e., that it has not been reported that the sender's private key has been compromised at the time of sending).
- In a public-key system augmented by a hash function H, Alice might send a message M to Bob as follows (ignore secrecy considerations): Alice sends M and a signed hash $H(M)'_{\text{ALICE}}$ to Bob. Bob uses Alice's public key $\mathbf{K}_{\text{ALICE-PUB}}$ to retrieve H(M), then computes $H(M)_{\text{BOB}}$ and compares the two values for authentication. For nonrepudiation, Bob retains M, $\mathbf{K}_{\text{ALICE-PUB}}$ and $H(M)'_{\text{ALICE}}$. If Alice attempts to repudiate M, a judge can use the three items to resolve the dispute by completing the same steps Bob did and attesting to the validity of Alice's private key at the time of transmission of M.
- The preceding schemes satisfy another desirable property: in the adjudication process, they do not compromise security by exposing private keys to a judge.
- Another solution involves timestamps. This again may involve a network of automated arbitrators, but is very simple in nature. Receivers obtain timestamps along with the received message. If a receiver needs to be sure of the validity of a signature, he may check the validity of the sender's private key by checking with a central authority. As long as the received message is timestamped before the validity check, the receiver is assured of nonrepudiation. If users are permitted to change their keys, a central authority should retain past keys for use in disputes which may arise later.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Secret Sharing Algorithms

Secret Sharing Algorithms

- Let's assume that you're setting up a launch program for a nuclear missile. You want to make sure that no single individual can initiate a launch. You have five officers whom you feel can be trusted with the individual secret codes to initiate a launch; however, you additionally desire that only three officers need be present to launch a missile.
- This problem can be solved by a secret sharing scheme, called a threshold scheme. In its simplest form any message (a launch code in this case) can be divided into *n* pieces, called shadows, such that any *m* of them can be used to reconstruct the entire message.
- Lets say you desired to create a (3,5) threshold scheme to satisfy our launch code scheme, then we could use the following technique developed by Shamir.

1. Generate a quadratic polynomial $ax^2 + bx + \mathrm{M}$ *(*mod *p)* in which *p* is a random number. The coefficients, *a* and *b*, are chosen at random and are kept secret and discarded after the shadows are handed out. M is our secret launch code. The prime *p* is made public.
2. The shadows k_i are obtained by evaluating the polynomial at five points $k_i = F(x_i)$.
3. Since the quadratic polynomial has three unknown coefficients, *a*, *b*, *m*, any three shadows can be used to create three equations since the other two equations are redundant.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Secret Sharing Algorithms

Secret Sharing Algorithms

Example

Let:

M = 11 (our secret launch code)
a = 7 b = 8 (our chosen randoms)

Which generates the quadratic:

$$F(x) = 7x^2 + 8x + 11 \pmod{13}$$

We create the following shadows:

$$k_1 = F(1) = 7 + 8 + 11 = 0 \pmod{13}$$
$$k_2 = F(2) = 28 + 16 + 11 = 3 \pmod{13}$$
$$k_3 = F(3) = 63 + 24 + 11 = 7 \pmod{13}$$
$$k_4 = F(4) = 112 + 32 + 11 = 12 \pmod{13}$$
$$k_5 = F(5) = 175 + 40 + 11 = 5 \pmod{13}$$

To reconstruct M from three of the shadows, let's say, k_2, k_3 and k_5 we solve the following set of linear equations to find M:

$$3 = a * 2^2 + b * 2 + M \pmod{13}$$
$$7 = a * 3^2 + b * 3 + M \pmod{13}$$
$$5 = a * 5^2 + b * 5 + M \pmod{13}$$

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Blind signatures

Blind signatures

- Usually we desire people to be aware of the contents of a document before signing it; however, there are times when we wish to have people sign a document without their seeing the contents. This has an obvious application in the real world, specifically to the notarization process. Less obvious perhaps is that we can use blind signatures in voting protocols.
- Let's assume that Bob is a notary public. Alice wants him to sign a document but does not want him to have any idea what he is signing. Bob doesn't care because he is just certifying that he notarized the document at a given time. The following simple protocol can be used.

    1. Alice takes the document and multiplies it by a random value, called a blinding factor.
    2. Alice sends the blinded document to Bob.
    3. Bob signs the document and returns it to Alice.
    4. Alice divides out the blinding factor, leaving the original document signed by Bob.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Secure Elections

Secure Elections

- Although on the surface it may seem like a very simple protocol to develop, secure voting can involve rather detailed and complicated protocols. We will look at only at a few of the more simplistic protocols which involve a Central Tabulating Facility.

- Ideally, the protocol we desire has the following properties:

    1. Only authorized voters can vote.
    2. No one can vote more than once.
    3. No one can determine for whom anyone voted.
    4. No one can change anyone else's vote without being discovered.
    5. All voters can make sure that their vote has been taken into account in the final tabulation.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Simplistic Secure Voting Protocols

Simplistic Secure Voting Protocols

- Let's look at a very simplistic protocol that does not work:

    1. All voters encrypt their vote with the public key of the CTF.
    2. All voters send their vote to the CTF.
    3. The CTF decrypts the votes, tabulates them and makes the result public.

O-o-o-o-ps!
This protocol does not work because it violates every one of the characteristics we desire in a voting protocol.

- Let's try an improvement over the first protocol that satisfies some of our properties but still falls short of the mark:

    1. All voters sign their vote with their private key.
    2. All voters encrypt their signed vote with the CTF's public key.
    3. All voters send their vote to the CTF.
    4. The CTF decrypts the vote, checks the signatures, tabulates the results, and makes the results public.

O-o-o-o-ps!
This protocol is only a slight improvement since it only satisfies properties (1) and (2) of our ideal protocol.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Voting with Blind Signatures

Voting with Blind Signatures

- The problem that we now face is how to dissociate the vote from the voter, while still maintaining authentication:

1. All voters generate a pair of votes (“yes’ and “no”) to which a very large randomly generated serial number is attached.
2. All voters blind their pairs of votes and send them to the CTF.
3. The CTF checks its database to make sure voters have not submitted blinded votes for signature previously. Then it signs the votes and sends them back to the voters and stores the names of the voters in its database.
4. The voters unblind the messages, leaving a a “yes” and “no” vote, each of which has been signed by the CTF.
5. The voters choose one of the votes and encrypts it with the CTF’s public key.
6. The voters send in their votes.
7. The CTF decrypts the votes, checks the signature, checks its database for a duplicate serial number, saves the serial number, and tabulates the votes. It publishes the results of the election, along with every serial number and its associated vote.

A-a-a-a-h!

- The protocol above works even better if all votes are collected in an electronic ballot box prior to giving it to the CTF, since this would make it impossible for the CTF to keep track of who sent in what vote.
- The problem with this protocol is that it does involve a CTF which must be trusted. In order to eliminate the CTF we must resort to rather complex and cumbersome protocols which appear to be extremely impractical for large scale elections.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Digital Cash

Digital Cash

- Once again an everyday protocol which we frequently take for granted, and which also appears to have a relatively simple solution in cyber-space, requires a complex and cumbersome solution. There is no single solution which fits our concept of an ideal digital cash world. The following characteristics have been identified as six properties of an ideal digital cash system:

    1. Independence - The security of the digital cash is not dependent on any physical location. The cash can be transferred through computer networks.
    2. Security - The digital cash cannot be copied or reused.
    3. Untraceability - No one can trace the relationship between users and their purchases.
    4. Off-Line Payment - The point of purchase does not need to be linked to a host to process the user's payment.
    5. Transferability - The digital cash can be transferred to other users.
    6. Divisibility - A piece of digital cash in a given amount can be subdivided into smaller pieces of cash in smaller amounts.

- There are a number of digital cash systems which have been proposed and several implementations currently exist; however, no practical solution has been implemented which satisfies all of the constraints above.
- Okamoto and Ohta have developed a schema which satisfies all the constraints. The total data transfer for a payment is about 20 kilobytes, and the protocol can be completed in several seconds.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

S/Key - One-Time Password System

S/Key - One-Time Password System

- Developed and implemented at Bellcore.
- One-time password system, to counter eavesdropping on network connections to obtain user/account information and passwords.
- The user's secret password never crosses the network during login.
- User's secret password is never stored anywhere, including on the host being protected.
- Based on publicly available hash function algorithm (MD4/MD5).
- Takes 8 bytes of input to MD4 then by folding the output byte pairs in the 16-byte MD4 to produce an 8-byte output (yielding a 64-bit password).

How one-time passwords are generated:

1. The very first one-way password is created by running the secret password *s* through the hash function *f* some specified number of times, *N*.

$$p_0 = f^N(s)$$

2. The next one-way password is generated by running the secret password through the hash function *N-1* times;

$$p_1 = f^{N-1}(s)$$

3. In general, each subsequent one-time password p_i is generated by:

$$p_i = f^{N-i}(s)$$

An eavesdropper will not be able to generate the next one-time password in the sequence because doing so would require inverting the hash function.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

S/Key - One-Time Password System

How one-time passwords are generated:

1. The host is initially given p_0.
2. When a client attempts to be authenticated, the seed and the current value of *i* are passed to the client.
3. The client returns the next one-time password by taking a the seed value, which is concatenated to the password, and running the modified secret password through the hash function *i-1* times.
4. The host temporarily saves the clients result, then applies the hash function to it.
5. If the result compares to its previously stored value then the temporary copy replaces the previously stored value, and the client is given access.

- After the user has used *N-1* passwords then it is necessary to reinitialize the system through the use of *keyinit*, which is a special version of the UNIX *passwd* command.
- S/Key is available for anonymous *ftp* from: *thumper.bellcore.com* in the *pub/nmh* subdirectory

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

This page is intentionally blank.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

This page is intentionally blank.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

This page is intentionally blank.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Section 9

Network Security I

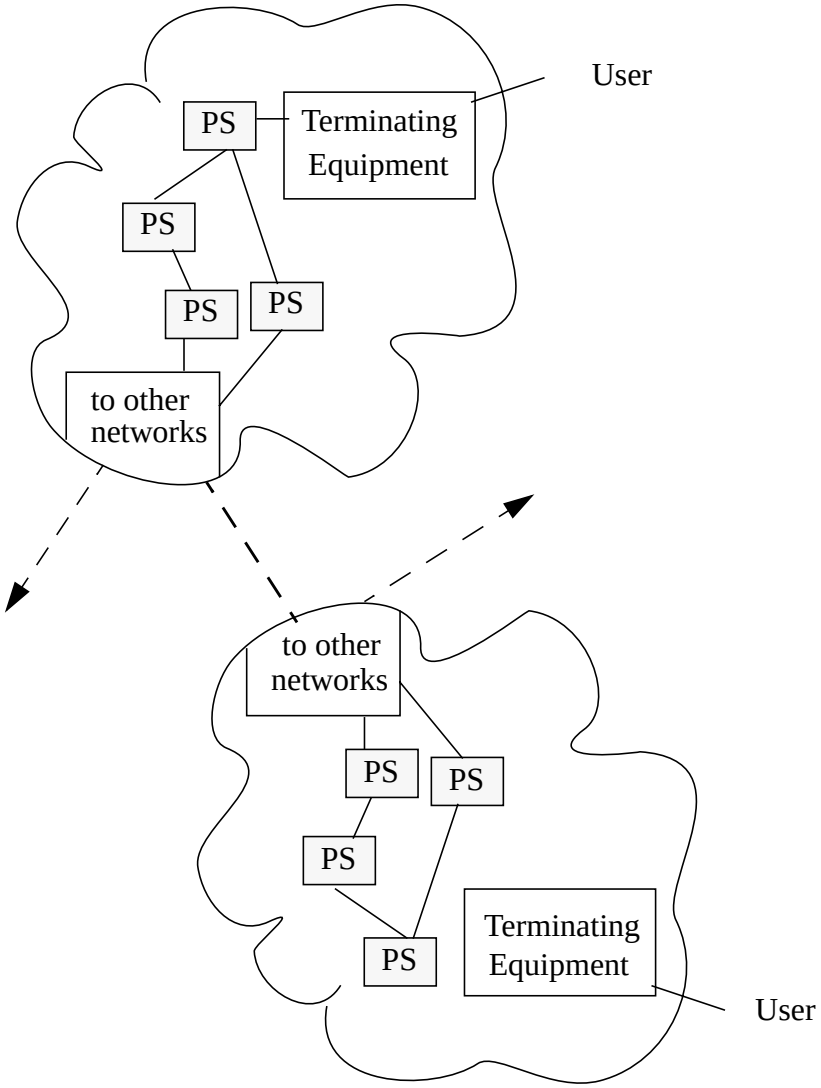
Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Introduction to Networks

Network

A collection of interconnected functional units providing data communications services among components attached to it. These components are comprised of both hardware and software



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Threats to Network Security

Their distributed nature and the possibility of increased interconnectivity render networks more vulnerable than monolithic systems.

Traffic Flow Analysis

Inference of information through the examination of message attributes rather than message contents

- Traffic frequency
- Source addresses
- Destination addresses
- Dominoes Pizza Traffic

Denial of Service

The prevention of authorized access to physical resources through theft or disruption. Delaying for time-critical access is a form of denial of service

- flooding of network with traffic
- blocking of transmissions based on addresses
- message replay

Spoofing (Impersonation)

Use of services under a false identity. May obtain unauthorized access to information or may maliciously modify information

- replay of passwords
- modification of source addresses
- compromise of passwords
- message replay

Eavesdropping

Illicit capture of information while it is enroute between communicating parties

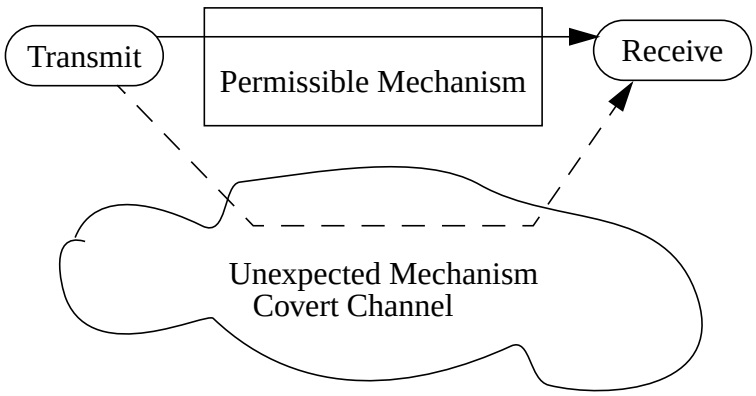
- Radio link interceptions
- Wiretapping
- Emanations from communications equipment
- Grabbing unprotected ethernet packets in a LAN

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Threats to Network Security

Covert Channels

The use of system mechanisms in an unexpected manner which causes the leakage of information in violation of the system security policy.



- Encodings using message length
- Encodings using message addresses

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Network Security Services

There is overlap between these areas. Note that ISO lists the following: Non-repudiation, access control, authentication, data confidentiality, and data integrity

Access control

Enforcement of security policy when requests for access are made

Information Confidentiality

Protection of information from unauthorized disclosure

Information Integrity

Protection of information from unauthorized modification

Authentication and Non Repudiation

Insure the proper authentication of active system entities. Prevents impersonation or masquerading

Prevent the repudiation of prior events

- Proof of origin
- Proof of receipt

Availability

Insure that the network services are both available and of appropriate quality

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Confidentiality

Objective:

- Restrict access to data in transit or in storage to TCB components

Method:

- transform data to render contents unreadable -- encryption
- store data in system-only domain -- secured computing systems or trusted processor states
- Use of label-based controls, e.g. derived from Bell and LaPadula Model
  - Prohibit flow of information down in levels of confidentiality
  - Permit entities at higher confidentiality levels to observe information at lower confidentiality levels

Considerations for location of confidentiality controls

- Which protocol layer?
- Multiple protocol layers?
- End system provides?
- Intermediate system provides?

Problems

Key Management

- Key distribution centers for private keys
- Certificate servers for public keys

Separation of Encrypted from Unencrypted

- Where is the TCB?

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Integrity

Objective:

- Prevent unauthorized modification of data. Includes
  - integrity of information in transmission packets
  - ordering of transmission packets
  - insuring transmission of complete message to intended destination

Method:

- use message authentication codes (MAC), also known as integrity locks
- Use of label-based controls, e.g. derived from Biba Integrity Model
  - Prohibit flow of information up in levels of integrity
  - Permit entities at lower integrity levels to observe information of higher integrity

Considerations for location of integrity controls

- Which protocol layer?
- Multiple protocol layers?
- End system provides?
- Intermediate system provides?

Problems

Management

- Protection of integrity enforcement mechanisms
- Determination of specific integrity attributes, e.g. labels or MACs

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Authentication and Non-repudiation Services

Objective:

- Establish message origin (author)
- Obtain proof that message was sent.

Method:

- use signatures, time-stamps

Considerations for location of authentication and non-repudiation services

- Which protocol layer?
- Multiple protocol layers?
- End system provides?
- Intermediate system provides?

Problems

Protocol designs

- The protocols can be complicated.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Availability Services

Although desirable, cannot be specified in the same precise, global, and persistent manner as confidentiality and integrity policies

- Subjective
  - What is progress?
  - What is termination?
  - What is success?
  - what is sufficient?
  - Cannot look at individual components, entire system must be examined
- Authorized users may compete for system resources
- To reduce scope of problem, external users may need to be excluded
  - totally
  - provided with circumscribed services
- How are availability attributes defined?
- How are availability attributes allocated and revoked?

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Network Security Mechanisms

Overview of Network Security Mechanisms

	Encryption	Trusted Network Base	Physical Protection
Authentication	Yes. Validate authenticity of requestor of service. Sometime "certificates are produced for future requests	Yes. Mandatory integrity fields, but information must be under TNB control	Yes, should use error detection techniques to support physical protection
Access Control	No, but can help enforce TCB access control decisions	Yes, Label-based access control decisions	Restrict use of end system only to authorized users
Confidentiality	Yes, Semantics of data altered so that they are unintelligible	Yes, but data must remain under continuous control of the TNB, otherwise supplemental mechanisms, such as encryption, are needed	Separate users who should have access to information from those who should not
Integrity	Yes, Encrypted data may have internal check sum Cleartext may have cryptographically computed checksum	No in the case of communications Enforcement of Biba integrity model within system. Encryption for communications integrity	Malicious modification can be avoided by combined physical and personnel controls

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Cryptography for Secure Networking

Objective:

transform information so that it is unreadable without use of a key to transform ciphertext back to plaintext

Permits secure communication over an insecure channel

Two kinds of Cryptography

Private Key -- Symmetric Key

- examples include Skipjack, DES
- Same Key and algorithm are used to encrypt and decrypt message

$$D_k(E_k(P)) = P$$

- Supports multiple receivers
- Key must be agreed upon by sender and receiver
  - trusted protocol for key distribution
  - synchronization of rekeying
  - deliver keys by trusted courier
- encipherment is in fixed block sizes
- high bandwidths supported
  - hardware implementations
  - stream mode -- 10^8 bps
  - change keys with each protocol data unit -- 10^6 to 10^7 bps

Public Key -- Asymmetric Key

Examples include RSA

Sender and receiver have different keys for encryption and decryption

$$D_{priv}(E_{pub}(P)) = P$$

The encryption key is public and may be stored in a “directory”

The decryption key must be protected

Bandwidth lower ~10^6 bps

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Benefits of Cryptography

Cryptography contributes to communications security (COMSEC)

Confidentiality

- Once a sensitive message has been encrypted using a strong encryption algorithm, it can only be decrypted using the key
- Encrypted data is consider to be unclassified

Integrity

- Digital checksums
- Message authentication codes
- Integrity Lock, also known as “Spray Paint”
- Message Digests
  - likelihood of two messages producing the same message digest is low
  - no keys are required
  - MD5 is a popular choice

Authentication

- Use key to validate identity
- Private Key Authentication
  - Three-way protocol usual
  - Trusted key server required
  - Kerberos is an example
- Public Key Authentication -- digital signatures
  - NIST digital signature standard (DSS) - no privacy
  - Basic mechanism: sender signs message through encryption with private key and receiver authenticates by decrypting with public key
  - Replay is prevented by using time stamps
  - Digests can improve efficiency
  - Problem: public keys must be distributed

Access Control -- NO!

Cryptography does not provide access control. If a TCB selects pair-wise keys based on security labels, then cryptography can support TCB access control decisions.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Challenges to Network Cryptography

Key Management

Classification of key is the least upper bound of the classifications of data encrypted with the key

Who should receive individual keys?

- each message
- each security level
- individual users
- groups of users
- hosts
- each connection

Key Generation

- need to have good random number generators
- need to avoid weak keys
  - all 1s
  - all 0’s
  - etc.

Need Key Distribution and Revocation Plan

- may differ for symmetric and public keys
- need key revocation lists
- may choose to use mixed schemes
- public key cryptography to distribute symmetric (shared) keys
- symmetric key cryptography to encrypt messages
- Secure Data Network System (SDNS) supports key distribution relying on public key methods
- Blacker and Kerberos are systems where a particular network node is allocated key distribution responsibility

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

ISO Reference Model

Open Systems Interconnection (OSI)

- Describes computer network communications.
- Model describes peer-to-peer correspondence, relationship between corresponding layers of sender and receiver.
- Each layer represents a different activity performed in the actual transmission of a message.
- Each layer serves a separate function
- Equivalent layers perform similar functions for sender and receiver

.

Layer	Responsibility	Actions
Layer 7 Application	User Program	Initiates message; optional encryption
Layer 6 Presentation	System Utilities	Breaks message into blocks, text compression; optional encryption
Layer 5 Session	Operating System	Establishes user-to-user session, header added to show sender, receiver and sequencing information, recovery
Layer 4 Transport	Transport Manager	Flow control, priority service, information added concerning the logical connection
Layer 3 Network	Network Manager	Routing, message blocking into packets, routing information added to blocks
Layer 2 Data Link	Hardware	Transmission error recovery, message separation into frames; optional encryption, header and trailer added for correct sequencing and error detection
Layer 1 Physical	Hardware	Physical signal transmission, by individual bits

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

OSI Security Service Matrix

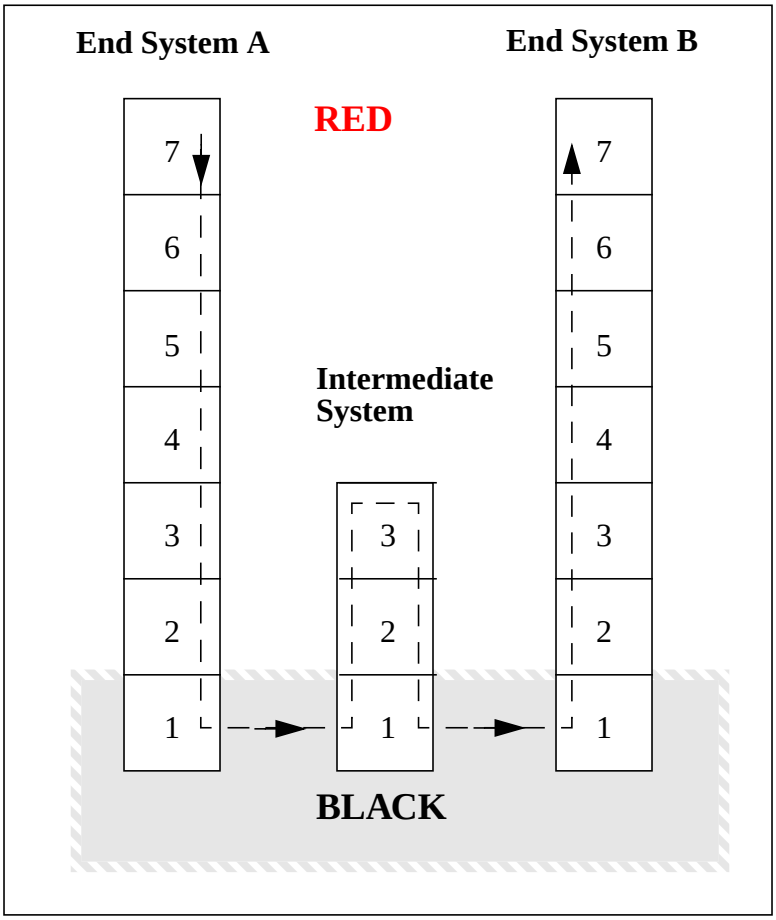
Services	1	2	3	4	5	6	7
peer entity authentication			X	X			X
data origin authentication			X	X			X
access control			X	X			X
connection confidentiality	X	X	X	X		X	X
connectionless confidentiality		X	X	X		X	X
selective field confidentiality							X
traffic flow security	X		X				X
connection integrity			X	X			X
connectionless integrity			X	X			X
nonrepudiation							X
	Physical	DataLink	Network	Transport	Session	Presentation	Application

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Cryptography Placement Issues

- Data in Red is in the clear.
- Data in the Black is encrypted.
- Encryption takes place as data passes from the Red region to the Black region.
- Decryption takes place as data passes from the Black region to the Red region.

- Link Encryption -- Encryption is a link layer
- All Intermediate systems must be trusted
- Protects against compromise during transmission

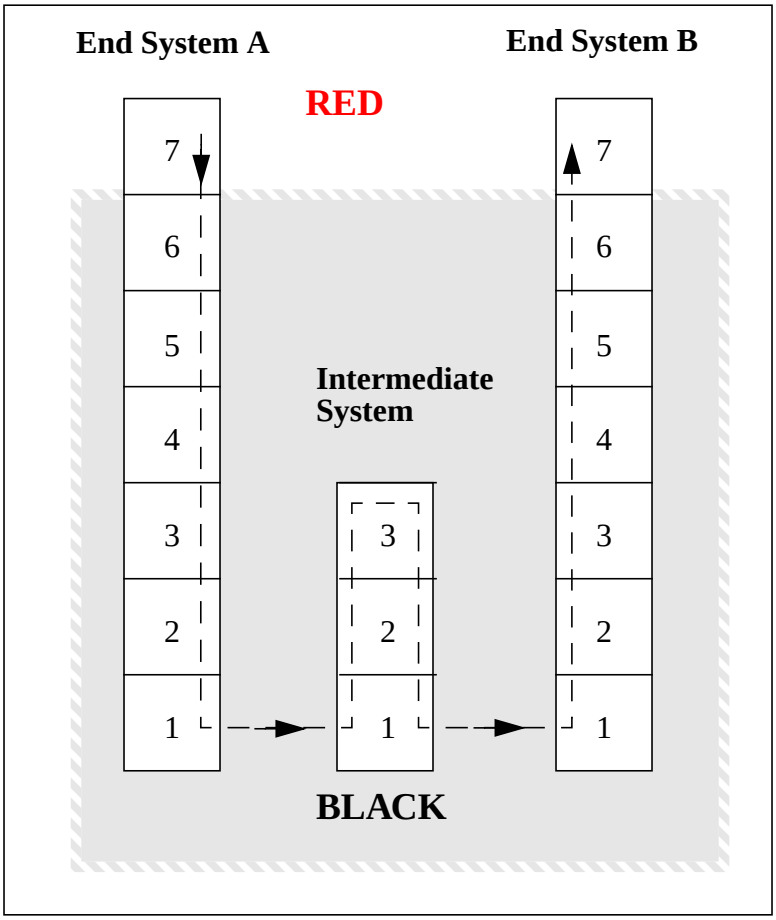


Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Cryptography Placement Issues

End-to-End Encryption

- Encryption is at application layer
- Application information is protected
- Potential for many attacks at lower protocol levels



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

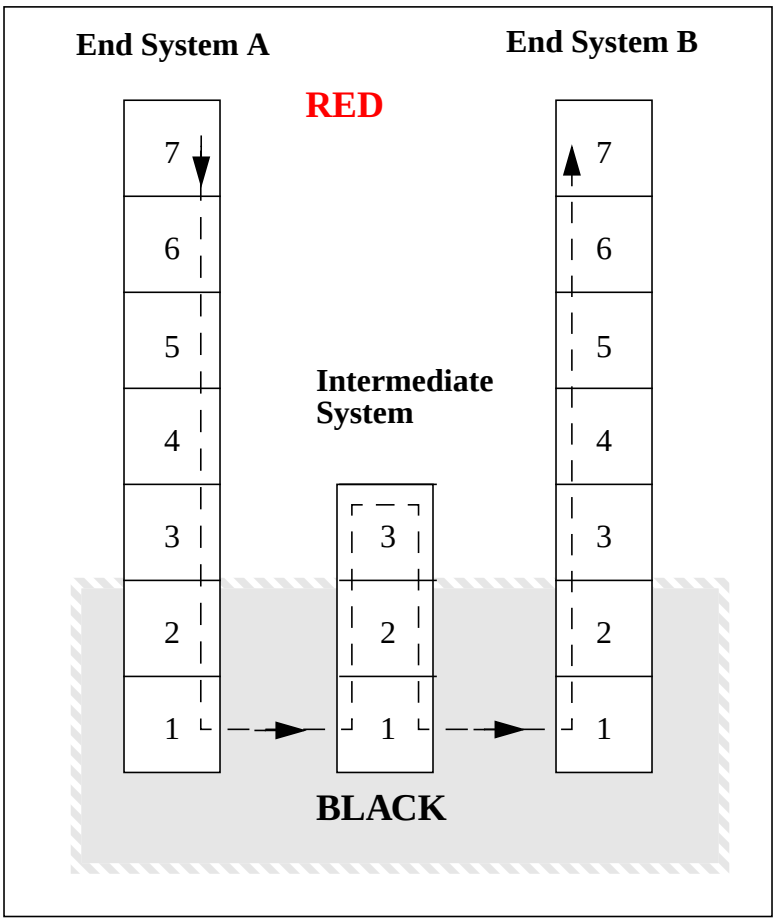
Cryptography Placement -- SILS

Standard for Interoperable LAN Security

IEEE 802.10 LAN Security Working Group

- vendors, government, and users

Security Services at layer 2 of OSI framework



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Benefits and Disadvantages of SILS

SILS Permits Routing

Recall Layer 3 provides routing services

Advantages

- routing may be optimized
- Many protocols exist at Layer 3, e.g. Novell, TCP/IP, etc.
  - With encryption at Layer 2, all are supported

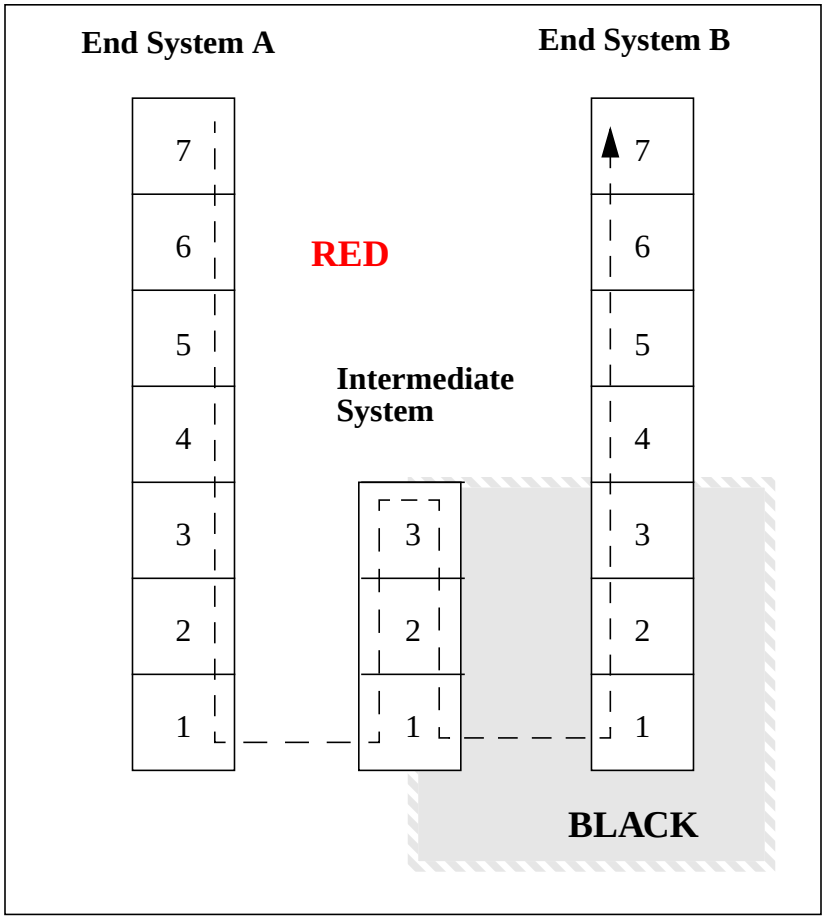
Disadvantages

- with encryption below the routing layer, then clear text must be available in intermediate systems
- the addresses provided by the source will not necessarily be used because the router may choose “better” addresses
- all routing paths used must be able to encrypt/decrypt so that destination can decrypt and reconstruct the complete message
- Performance at high bandwidth may require multiplexing of multiple cipher streams

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Mixed Systems

Here some systems use encryption while others do not



Challenges

- maintain DAC of End Systems
- separate messages according to security level
- keep encrypted and unencrypted information separate

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Mixed Systems - continued

Red systems may be LANS

- inexpensive
- local protection
- use secure facilities
- do not need encryption, so performance is better
- must be trusted
- separate information by security level
- enforce DAC policy, which may include a policy regarding how messages will be routed when they are transmitted from the LAN

Intermediate systems must be trusted

- must separate Red and Black
- must protect keys and cryptographic methods
- may need to select cryptography based upon message security levels
- routing policy may need to be enforced

Black systems require no mandatory trust

- all messaged are treated as having same level
- DAC routing policy may need to be enforced
- may choose to protect communications lines against
  - traffic analysis
  - denial of service

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Where are Encryption Mechanisms Located?

Within the host system

- If you want to use encryption at the application level
- May use different encryption for different applications
  - files
  - e-mail
  - etc.

need trusted system

  - separate Red/Black
  - manage keys and cryptography
  - information is read from Red and written to Black

As a front end device

- helps to localize encryption
- could be device on I/O port
- could be card on system bus (part of hardware, but not “in” the host system)

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Network Evaluations

Introduction

The Trusted Network Interpretation (TNI) assumes that it is possible to evaluate a network under the TCSEC.

- implies that the network can be treated as if it were a monolithic computer system
- Strategy is to partition the TCB both logically and physically among components of the network.
- This results in a Network TCB (NTCB).

Evaluation procedure

- must have decomposition of overall network security policy into policies for individual components
- evaluate individual components
- use the network security architecture to support the assertion that the network is a sound composition of its components
- given a sound composition assert that since each supports its allocated policy correctly that overall network security policy is supported We will inspect the rationale for doing this.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Network Security Policies

Need to examine policy and how it will be allocated to Network components.

Security Policy

Security policy is broadly expressed in terms of people and information

- need a **single** uniform network security policy
- if there are multiple organizations involved, then the security policy needs to be defined during the **early** stages of the network development process

Mandatory Access Control Policy

- based on a comparison of labels associated with information with clearances of users.
- may need to merge systems of classes and clearances
- commercial organizations may have trivial MAC policies
- label-based, i.e., labels on data units and communications entities

Discretionary Access Control Policy

- these are much more diverse
  - modes of access
  - composition of groups
  - kinds of named objects for access control
  - mechanisms to limit or propagate permission to access information
- expect intensive generalization of policy among organizations
- overall policy -- depends upon the underlying capabilities ascribed to the network
- based upon the identity of the entity requesting service
  - Host
  - Users

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Network Supporting Policies

Supporting Policy Issues:

- additional capabilities relating to accountability of individuals for security-relevant activities
- provide environment for enforcement and monitoring of MAC and DAC policies
- two major sub-categories

Identification and Authentication

- supports MAC and DAC
- authenticates ID and clearance
- basis for determining group membership for DAC

Audit

- security relevant events are uniquely associated with a user
- hold users accountable for actions

for network, must formulate mutually acceptable set of overall supporting policies

- likely to be harder than DAC

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Network Security Policy Issues

Formal Security Policy Model

- starting point for a chain of arguments leading to higher assurance
- form of model may be influenced by technical characteristics of system to be built
  - want intuitive resemblance to subjects, objects and access characteristics of intended implementation

For each component in the network a Reference Monitor is needed.

- each Reference Monitor should have a Formal Security Policy Model (at Class B2 and above)
- do not need a Formal Security Policy Model for the entire network system
  - instead must argue that each model represents the overall security policy

Security Policy Summary

Must have a priori policy statements.

For Class B2 and above, must have Formal Security Policy Models.

Formal Security Policy Models are not required for supporting policy components.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Network TCB -- Introduction

Objectives

1. a subject is confined to a single network component for its lifetime
2. a subject may directly access only objects within its component
3. every component contains a component Reference Monitor which mediates all access which are made locally.
4. communications channels which link components do not compromise information

If a network succeeds in achieving the points described above, then its collection of Reference Monitors constitute a comprehensive network

Reference Monitor

- all network accesses are mediated
- there are no non-local accesses
- the network Reference Monitor cannot be tampered with
- no component reference monitor can be tampered with.

We need to design the network so that the above axioms can be validated.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Method to Achieve Objectives

Confine Subjects

Subjects must be confined to a single component.

notion of a <process, domain> pair for a subject

limit objects to the same component

- must assure that no domain encompasses objects from another component

- remote processes result in the creation of a new subject in the remote component.

Objects in Local Component

Subjects can directly access objects only within the component with which the subject is associated.

What about information being transmitted between components?

- information ``in motion'' is not treated as an object
- if it is not an object then it cannot be access until it ``comes to rest''

Components Contain Component Reference Monitor

- in some components a degenerate component reference monitor may suffice
  - this means that if the component is single level the reference monitor is degenerate
  - no accesses need be checked because
    1. all objects have the same label
    2. all subjects have the same class
- recall that each component reference monitor needs to enforce only the policy pertinent to the particular component's local accesses

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Special Concerns for Distributed TCBs

Fragmented TCB Domain

Trusted Paths Between Components

Trusted Protocols

Fault tolerance

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Fragmented TCB Domain

Monolithic system

- all aspects of security state are local
- all aspects of security state are immediately available
- state transitions are well defined

Distributed system

- many devices
- maintaining integrity of TCB more difficult

- A single device may not comprise totality of all system integrity constraints
  - Delays possible
  - No guarantees of stability

- Concurrent transitions at various locations may not permit total ordering of state transitions

- Security state may be replicated, consistency must be maintained

- Labels must be consistent

- Methods for comparing labels must be consistent

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Trusted Paths Between Components

Must provide assurance that TCB data is passed in a trusted path

Trusted Path must provide

- message received from trusted path originated from a trusted source
- message received from trusted path was not modified
- labels of messages sent on the trusted path have not been altered
- message ordering is preserved on pair-wise trusted paths
  - prevents replays
  - this may be optional

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Trusted Protocols

Protocol interpreters at different OSI levels may need to be trusted

Implementation of Trusted Path

- Insert into protocol interpreters at Transport level or below
  - delivery assurances may be part of protocol
  - expensive to verify
- Or, use cryptographic authentication with end-to-end transport level protocol
  - need not guarantee delivery
  - not too expensive to verify

Implement System-level atomic State Transitions

- May need application level protocols for making global state transitions.
- Very difficult to verify

System-level Concurrency Control

- Use to achieve atomic state transitions

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Fault tolerance

Everything may not function all of the time

Must have fail-secure properties

even though something is not working, the system is still secure

just show that failure states are secure

- do not worry about the “usual” safety issues
  - progress
  - termination
  - delivery of service

Denial of Service

- People do worry about this
- System should be designed to provide partial policy enforcement in the face of failures
  - example: DAC could still work even though only one security level might be serviced within a LAN on a distributed system
- Traditional fault-tolerant techniques can be applied to TCB relevant data

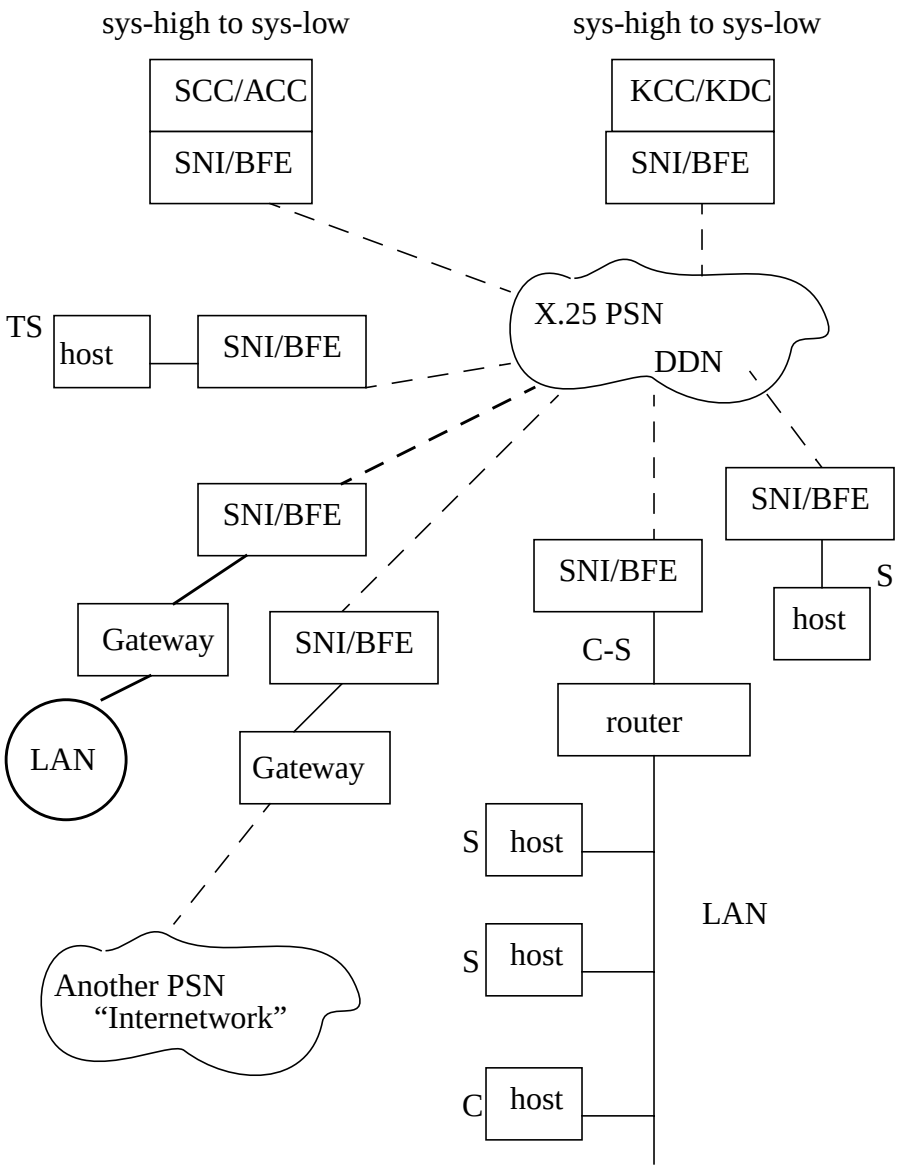
Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

BLACKER

Multilevel secure, with compartments

Class A1 (but pre-TNI)

Complete development but future use problematic



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Blacker Overview

Secure System Applique to Defense Data Network (DDN)

Major Components

Blacker Front End

Smart encryption box between host and Packet Switch

Blacker Initialization Parameters Carrier

hand-held device for initializing BFEs

Access Control Center (ACC)

- maintains authorization tables
- controls permissions for hosts to exchange messages
- security officer activity
- maintains audit trails
- triggers auditing alarms

Key Distribution Center (KDC)

distributes encryption keys to BFEs under direction of ACC

Blacker "Domain"

- 1000 BFE's
  - These correspond to hosts
- Single
  - ACC
  - KDC
  - (Note that for reliability ACCs and KDCs were replicated)

Red/Black Separation achieved through COMPUSEC

- Security Kernel used to separate Red and Black
- Traditional has crypto device "sandwiched" between Red and Black
  - Double the hardware cost
  - Must synchronize Red/Black -- difficult
- Kernel
  - Separates Red/Black
  - Manages Crypto as MLS device

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Secure Network Interface

SNI, AKA Blacker Front End (BFE)

Transparent:

Network interface presented to host

Host interface presented to network

Separates

Host-to-host shared keys

Security Levels

Enforce a connectivity property

A host may send or receive messages over a crypto connection only if it has current access to that crypto connection

Hosts Supported

Periods Processing

Multilevel

Automatic Crypto Key Support

obtains keys on demand

rekeys connections

Obtains permissions from SCC/ACC and through them enforces access control

When SCC/ACC and KCC/KDC are unavailable, emergency mode for secure communications

Connection Database

new connections added only if received by trusted path from SCC via KCC

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Access Control Center

Security Control Center, a.k.a Access Control Center (ACC)

Mediates

- request for connection between source and destination
- request for mediation comes from SNI/BFE
- Message security level must be within ranges of source and destination
- Source and destination must be on each others DAC lists
- If successful will request that KCC/KDC generate keys for the connection

Security Administration

- Access Control Database
- Security Administration Interface
- Generates data to initialize SNIs
- Maintains configuration database of host/SNI sites

Within a Domain

- Replicated SCCs
- two phase commit used for consistency

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Key Distribution Center

KCC, a.k.a. Key Distribution Center

All keys are encrypted

- Must communicate with BFEs at hosts' security level
  - therefore MLS
  - Use crypto seals for all imports/exports associated with KDC
  - Cryptoseals makes KDC similar to a security Guard

Centralized Key Management

Distribution of Keys controlled by Access Control Center

Separate administration of COMSEC and Access Control concerns

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

This page is intentionally blank.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Section 10

Network Security II

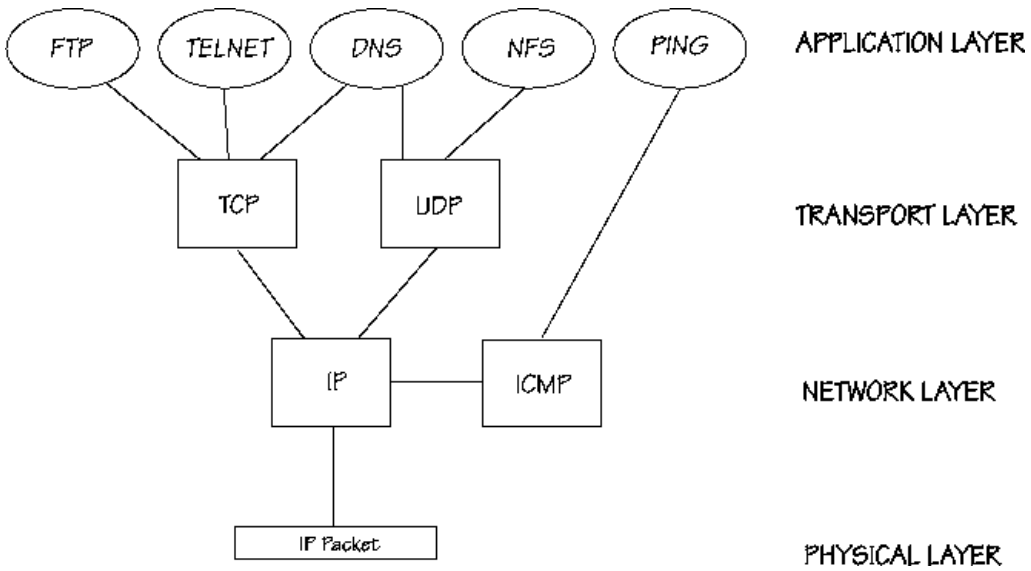
Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Overview of TCP/IP Internals

The Protocols

- TCP/IP is a suite of protocols including TCP and IP, UDP (User Datagram Protocol), ICMP (Internet Control Message Protocol), and several others.
- TCP/IP protocol suite does not conform exactly to the Open Systems Interconnection's seven layer model, but rather is pictured as shown:



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Overview of TCP/IP Internals

IP - Internet Protocol

- Low level protocol
- IP addresses
- 0-255.0-255.0-255.0-255
- E.g., 1.2.3.4
- NPS is 131.120.*.*
- Specify source and destination of IP packets

- The IP layer receives packets delivered by lower-level layers and passes the packets ``up" to the higher-layer TCP or UDP layers. and also transmits packets that have been received from the TCP or UDP layers to the lower-level layer.
- IP packets are unreliable datagrams because IP does nothing to ensure that IP packets are delivered in sequential order or are not damaged by errors.
- The IP packets contain the source address of the host from which the packet was sent, and the destination address of the host that is to receive the packet.
- TCP and UDP services generally assume that the source address in a packet is valid when accepting a packet.
- IP address forms the basis of authentication for many services; the services trust that the packet has been sent from a valid host and that host is indeed who it says it is.
- IP contains an option known as IP Source Routing, which can be used to specify a direct route to a destination and return path back to the origination. A source routed IP packet, to some TCP and UDP services, appears to come from the last system in the route as opposed to coming from the true origination.
- IP source address is problematic and can lead to break-ins and intruder activity; furthermore it can be used to trick systems into permitting connections from systems that otherwise would not be permitted to connect.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Overview of TCP/IP Internals

TCP

- If the IP packets contain encapsulated TCP packets, the IP software will pass them ``up'' to the TCP software layer. TCP sequentially orders the packets and performs error correction, and implements virtual circuits, or connections between hosts. The TCP packets contain sequence numbers and acknowledgments of received packets so that packets received out of order can be reordered and damaged packets can be retransmitted.
- Connection oriented services, such as TELNET, FTP, rlogin, X Windows, and SMTP, require a high degree of reliability and therefore use TCP. DNS uses TCP in some cases (for transmitting and receiving domain name service databases), but uses UDP for transmitting information about individual hosts.

UDP

- UDP interacts with application programs at the same relative layer as TCP. However, there is no error correction or retransmission of misordered or lost packets. UDP is therefore not used for connection-oriented services that need a virtual circuit. It is used for services that are query-response oriented, such as NFS.
- It is easier to spoof UDP packets than TCP packets, since there is no initial connection setup (handshake) involved.

ICMP

- ICMP (Internet Control Message Protocol) is at the same relative layer as IP; its purpose is to transmit information needed to control IP traffic. ICMP redirect messages inform hosts about more accurate routes to other systems, whereas ICMP unreachable messages indicate problems with a route. Additionally, ICMP can cause TCP connections to terminate ``gracefully'' if the route becomes unavailable (*ping* is a commonly-used ICMP-based service).

- ICMP redirect messages can be used to trick routers and hosts acting as routers into using ``false'' routes; these false routes would aid in directing traffic to an attacker's system instead of a legitimate trusted system. This could in turn lead to an attacker gaining access to systems that normally would not permit connections to the attacker's system or network.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Overview of TCP/IP Internals

- ***TCP - Transmission Control Protocol***
- ***UDP - User Datagram Protocol***
  - Higher level protocol
  - Use destination port numbers to identify specific TCP or UDP service
  - Use source port numbers to distinguish between multiple sessions
  - Standard destination ports
  - FTP data - port 20
  - FTP control - port 21
  - Telnet - port 23
  - X11 (X-Windows) - port 6000
  - SMTP (Simple Mail Transfer Protocol) - port 25
  - Source ports are random above 1023

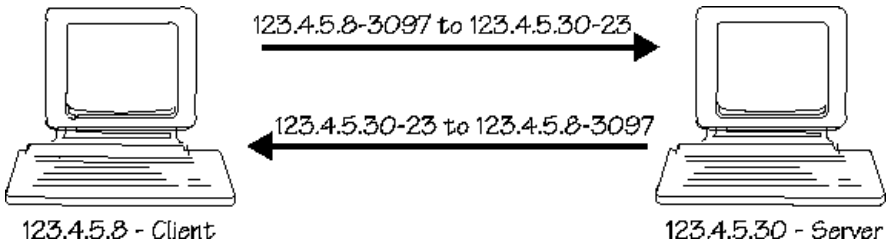
TCP and UDP Port Structure

- TCP and UDP services generally have a client-server relationship. (example - TELNET)
- A TCP or UDP connection is uniquely identified by
  - source IP address - the address of the system that sent the packet
  - destination IP address - the address of the system that receives the packet
  - source port - the connection's port at the source system
  - destination port - the connection's port at the destination system.
  - [There is a somewhat-uniform rule that only privileged server processes, i.e., those processes that operate with UNIX superuser privileges, can use port numbers less than 1024 (referred to as privileged ports).]

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Example - how ports are used for sending and receiving messages

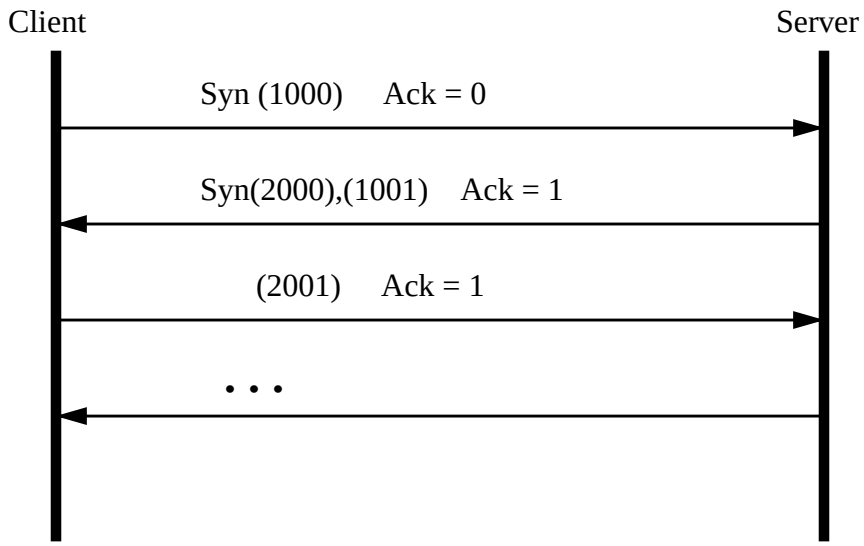
The TELNET server listens for incoming messages on port 23, and sends outgoing messages to port 23. A TELNET client, on the same or different system, would first request an unused port number from the operating system, and then use this port when sending and receiving messages. It would place this port number, say 3097, in packets destined for the TELNET server so that the server, when responding to the client, could place the client's port number in its TCP packets. The client's host, upon receiving a message, would examine the port and know which TELNET client should receive the message.



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

TCP Connections

- ***TCP is a connection oriented protocol (UDP is not)***
  - TCP packets are sequenced
  - TCP packets are acknowledged
  - TCP packets are retransmitted, if necessary
- ***Example TCP connection handshake***



- Intended to provide some measure of host authenticity.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Internet security problems

Contributing Factors:

- Internet was not designed to be very secure.
- Phenomenal success of the Internet.
- Introduction of different types of users.
- Vulnerable TCP/IP services - a number of the TCP/IP services are not secure and can be compromised by knowledgeable intruders.
- Services used in the local area networking environment for improving network management are especially vulnerable.
- Ease of spying and spoofing - the majority of Internet traffic is unencrypted; e-mail, passwords, and file transfers can be monitored and captured using readily-available software, intruders can then reuse passwords to break into systems.
- The role and importance of system management is often short-changed in job descriptions, resulting in many administrators being, at best, part-time and poorly prepared
- Lack of policy - many sites are configured unintentionally for wide-open Internet access without regard for the potential for abuse from the Internet; many sites permit more TCP/IP services than they require for their operations and do not attempt to limit access to information about their computers that could prove valuable to intruders.
- Complexity of configuration - host security access controls are often complex to configure and monitor; controls that are accidentally misconfigured often result in unauthorized access.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Internet security problems

Use of weak, static passwords.

- Passwords can be ``cracked'' a number of different ways, however the two most common methods are by cracking the encrypted form of the password and by monitoring communications channels for password packets.
- The UNIX operating system usually stores an encrypted form of passwords in a file that can be read by normal users. The password file can be obtained by simply copying it or via a number of other intruder methods. Once the file is obtained, an intruder can run readily-available password cracking programs against the passwords to obtain passwords that can be used to gain access into the system.

Host Authentication

- Some TCP or UDP services are able to authenticate only to the granularity of host addresses and not to specific users.
  - For example, an NFS (UDP) server cannot grant access to a specific user on a host, it must grant access to the entire host. The administrator of a server may trust a specific user on a host and wish to grant access to that user, but the administrator has no control over other users on that host and is thus forced to grant access to all users (or grant no access at all).

Ease of Spying/Monitoring

- When a user connects to his/her account on a remote host using TELNET or FTP, the user's password travels across in plaintext making the passwords susceptible to direct packet monitoring or network sniffers.
- Electronic mail - Most users do not encrypt e-mail, yet many assume that e-mail is secure and thus safe for transmitting sensitive information.
- The X Window System permits multiple windows to be opened at a workstation, along with display of graphics and multi-media applications (for example, the WWW browser Mosaic). Intruders can sometimes open windows on other systems and read keystrokes that could contain passwords or sensitive information.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Firewalls

Why a Firewall?

- The purpose of an Internet firewall is to provide a single point of defense with controlled and audited access to services, both from within and without an organizations private network.
  1. Protection from vulnerable services.
  2. Controlled access to site systems.
  3. Concentrated Security.
  4. Enhanced privacy.
  5. Logging and statistics on network use or misuse.
  6. Policy enforcement.

Issues and Problems with Firewalls

- Restricted access to desirable services, such as TELNET, FTP, X Windows, NFS.
- If unrestricted modem access is still permitted into a site protected by a firewall, attackers could effectively jump around the firewall.
- Firewalls generally do not provide protection from insider threats.
- MBONE - Multicast IP transmissions (MBONE) for video and voice are encapsulated in other packets; firewalls generally forward the packets without examining the packet contents.
- Viruses - Firewalls do not protect against users downloading virus-infected personal computer programs from Internet archives or transferring such programs in attachments to e-mail.
- Throughput - Firewalls represent a potential bottleneck, since all connections must pass through the firewall and, in some cases, be examined by the firewall.
- All eggs in single basket - A firewall system concentrates security in one spot as opposed to distributing it among systems. A compromise of the firewall could be disastrous to other less-protected systems on the subnet.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Internet security problems

Design Decisions

- There are two fundamental philosophies that determine the overall configuration of a firewall (The distinction in these two philosophies cannot be overemphasized):

"That which is not expressly permitted is prohibited."

- The firewall must be designed to block everything, and services must be enabled on a case-by-case basis.
- This approach is the easiest from an administrator's point of view since it provides a more "fail-safe" stance and does not demand that the administrator possess exception skills in the maintaining security of the system.

"That which is not expressly prohibited is permitted."

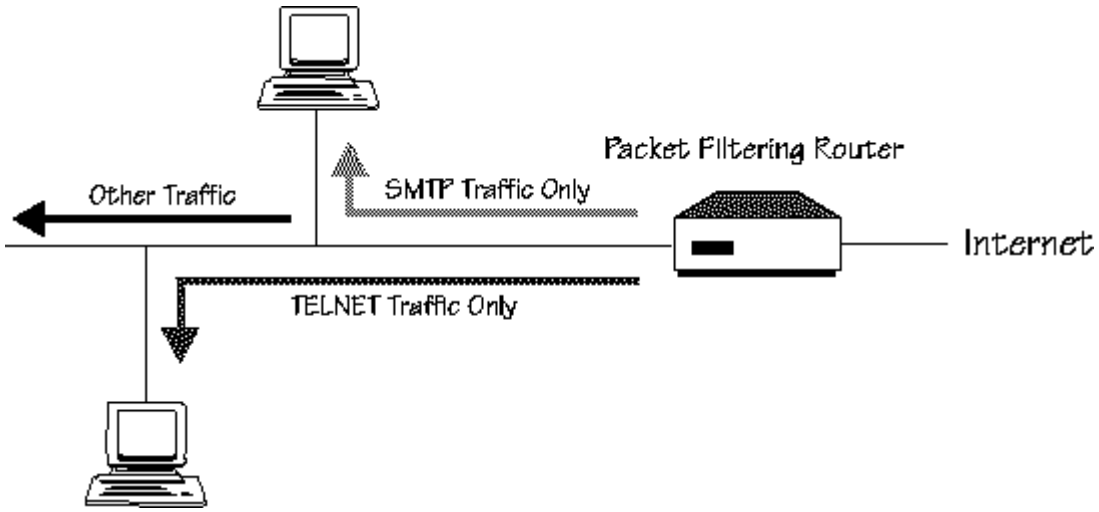
- The system administrator is placed in reactive mode to the actions of the user. Although this offers the user the greatest flexibility, it often pits the user against the administrator.
- This approach requires that the administrator anticipate what users will do and requires considerable skill in auditing and maintaining the overall security of the network.
- From a security standpoint, this approach is an administrator's nightmare.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Firewall Basics

Packet Filtering

- A packet filtering router usually can filter IP packets based on some or all of the following fields:
  - source IP address,
  - destination IP address,
  - TCP/UDP source port, and
  - TCP/UDP destination port.
  - ACK bit
- A site might wish to block connections from certain addresses, or block connections from all addresses external to the site (with certain exceptions, such as with SMTP for receiving e-mail).
- If a firewall can block TCP or UDP connections to or from specific ports, then one can implement policies that call for certain types of connections to be made to specific hosts, but not other hosts.



Representation of Packet Filtering on TELNET and SMTP.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

TCP/IP Threats

Threat 1 - Packet Sniffing

- ***Telnet and FTP passwords travel in the clear***

Threat 2 - Trusted Hosts and Source Routing

- ***Packets travel a predefined route to and from the destination.***
- ***Attacker routes traffic between two trusted hosts so that the traffic goes through the attackers system***
- ***Prevent by disallowing source routed packets at the firewall***

- Recall that the IP address of a host is presumed to be valid and is therefore trusted by TCP and UDP services. Using the IP source routing option, an attacker's host can masquerade as a trusted host or client. An example:

  1. the attacker constructs a source route from a trusted client to a server that passes through the attacker system. When packets use this route they will travel between the trusted client and the server (and back to the client) passing through the attacker system in each direction,
  2. the attacker sends packets using this route and the source routing option from the attacker to the server,
  3. the server accepts the packets as if they had origniated at the trusted client and sends acknowledgment packets back to the client,
  4. the attacker intercepts these packets (so the don’t continue on the trusted client) and continues the session as if they were the trusted client.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

TCP/IP Threats

Threat 3 - Trusted Hosts

- ***Attacker system masquerades as a trusted host by changing its IP address to that of a real trusted host***
- ***Packets are allowed in but they are not correctly returned, but who cares***
- ***Prevent with packet filtering firewall mechanism***
- ***Example packet filtering rules with NPS IP addresses***

Rule	Direction	Source IP address	Destination IP address	Action
A	Inbound	131.120.*.*	131.120.*.*	Deny
B	Any	Any	131.120.*.*	Allow

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

TCP/IP Threats

The following information demonstrates why packet filtering firewalls need to filter on the source port and the ACK bit

Threat 4 - X-Windows

- ***Attacker initiates a session with the X11 daemon running on port 6000.***
- ***Attacker monitors keystrokes***
- ***Easily overlooked when configuring firewall packet filtering rules***
- ***Example packet filtering rules that are intended to only allow SMTP (e-mail) in both directions***

Rule	Direction	Source IP address	Destination IP address	Destination port	Action
A	Inbound	Any	131.120.*.*	25	Allow
B	Outbound	131.120.*.*	Any	>1023	Allow
C	Outbound	131.120.*.*	Any	25	Allow
D	Inbound	Any	131.120.*.*	>1023	Allow
E	Any	Any	Any	Any	Deny

- ***These rules allow a connection between an external port 5150 and an internal port 6000 (X11)***
  - Rule D allows inbound packets
  - Rule B allows outbound packets

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

TCP/IP Threats

- ***Need Source Port Filtering to Counter This Attack***

Rule	Direction	Source IP address	Destination IP address	Source port	Destination port	Action
A	Inbound	Any	131.120.*.*	>1023	25	Allow
B	Outbound	131.120.*.*	Any	25	>1023	Allow
C	Outbound	131.120.*.*	Any	>1023	25	Allow
D	Inbound	Any	131.120.*.*	25	>1023	Allow
E	Any	Any	Any	Any	Any	Deny

- ***These rules allow a connection between an external port 25 and an internal port 6000 (X11)***
  - Rule D allows inbound packets
  - Rule C allows outbound packets
- ***Need Ack bit filtering to prevent this attack***

Rule	Direction	Source IP address	Destination IP address	Source port	Destination port	Ack bit set	Action
A	Inbound	Any	131.120.*.*	>1023	25	Any	Allow
B	Outbound	131.120.*.*	Any	25	>1023	Yes	Allow
C	Outbound	131.120.*.*	Any	>1023	25	Any	Allow
D	Inbound	Any	131.120.*.*	25	>1023	Yes	Allow
E	Any	Any	Any	Any	Any	Any	Deny

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Firewall Basics

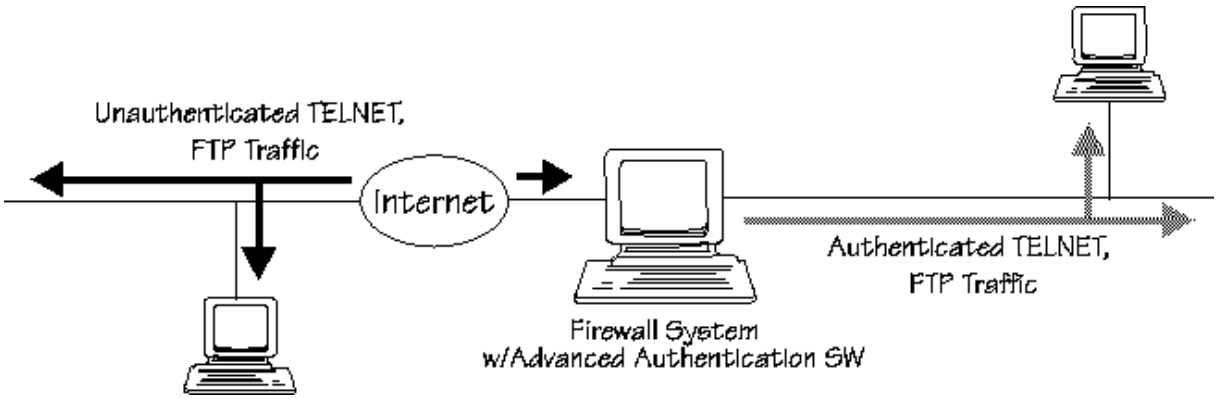
Application Gateways

- To counter some of the weaknesses associated with packet filtering routers, firewalls need to use software applications to forward and filter connections for services such as TELNET and FTP. Such an application is referred to as a proxy service, while the host running the proxy service is referred to as an application gateway.

An example of an application gateway:

- As an example, consider a site that blocks all incoming TELNET and FTP connections using a packet filtering router. The router allows TELNET and FTP packets to go to one host only, the TELNET/FTP application gateway. A user who wishes to connect inbound to a site system would have to connect first to the application gateway, and then to the destination host, as follows:

1. a user first telnets to the application gateway and enters the name of an internal host,
2. the gateway checks the user's source IP address and accepts or rejects it according to any access criteria in place,
3. the user may need to authenticate herself (possibly using a one-time password device),
4. the proxy service creates a TELNET connection between the gateway and the internal host,
5. the proxy service then passes bytes between the two connections, and
6. the application gateway logs the connection.



Virtual Connection Implemented by an Application Gateway and Proxy Services.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Benefits of Application Gateways

- Proxy services allow only those services through for which there is a proxy.
- The names of internal systems need not necessarily be made known via DNS to outside systems.
- Application traffic can be pre-authenticated before it reaches internal hosts and can be logged more effectively.
- Cost-effective because third-party software or hardware for authentication or logging need be located only at the application gateway.
- Rules at the packet filtering router will be less complex since the router need only allow application traffic destined for the application gateway and reject the rest.
- An e-mail application gateway serves to centralize e-mail collection and distribution to internal hosts and users. To outside users, all internal users would have e-mail addresses of the form:

  user@emailhost

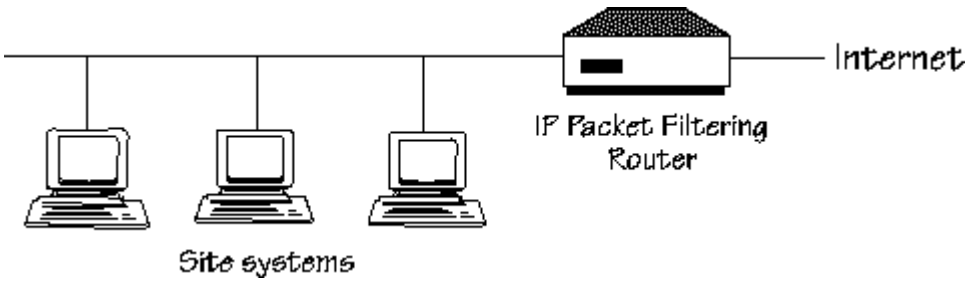
- The gateway would accept mail from outside users and then forward mail along to other internal systems as necessary. Users sending e-mail from internal systems could send it directly from their hosts, or in the case where internal system names are not known outside the protected subnet, the mail would be sent to the application gateway, which could then forward the mail to the destination host.
- Application gateways are used generally for TELNET, FTP and e-mail, as well as for X Windows and some other services. The application gateway can filter the FTP protocol and deny all puts to the anonymous FTP server; thus ensuring that nothing can be uploaded to the server.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Firewall Comparison

Packet Filtering advantages:

- The most common and easiest to employ for small, uncomplicated sites.
- The site systems usually have direct access to the Internet while all or most access to site systems from the Internet is blocked.
- Usually, inherently-dangerous services such as NIS, NFS, and X Windows are blocked.



Packet Filtering Firewall.

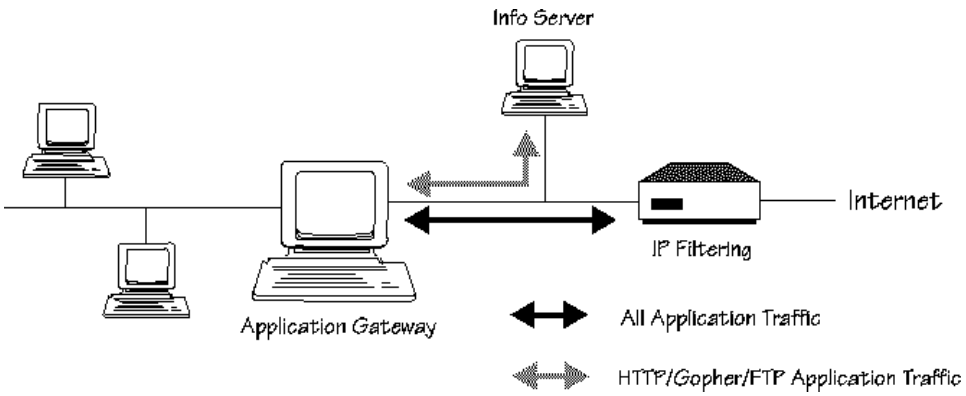
Disadvantages

- Little or no logging capability.
- Packet filtering rules are often difficult to test thoroughly.
- If complex filtering rules are required, the filtering rules may become unmanageable.
- Each host will require its own copy of advanced authentication measures.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Dual-homed Gateway Firewall

- A host system with two network interfaces with the host's IP forwarding capability disabled (i.e., the host can no longer route packets between the two connected networks).
- Services and access to site services is provided by proxy servers on the gateway.
- Simple firewall, yet very secure.
- The router can prevent direct Internet access to the firewall and force access to go through the firewall. If direct access is permitted to the server (which is the less secure alternative), then the server's name and IP address can be advertised by DNS. Locating the information server between the gateway and the router also adds to the security of the site, as any intruder penetration of the information server would still be prevented from reaching site systems by the dual-homed gateway.



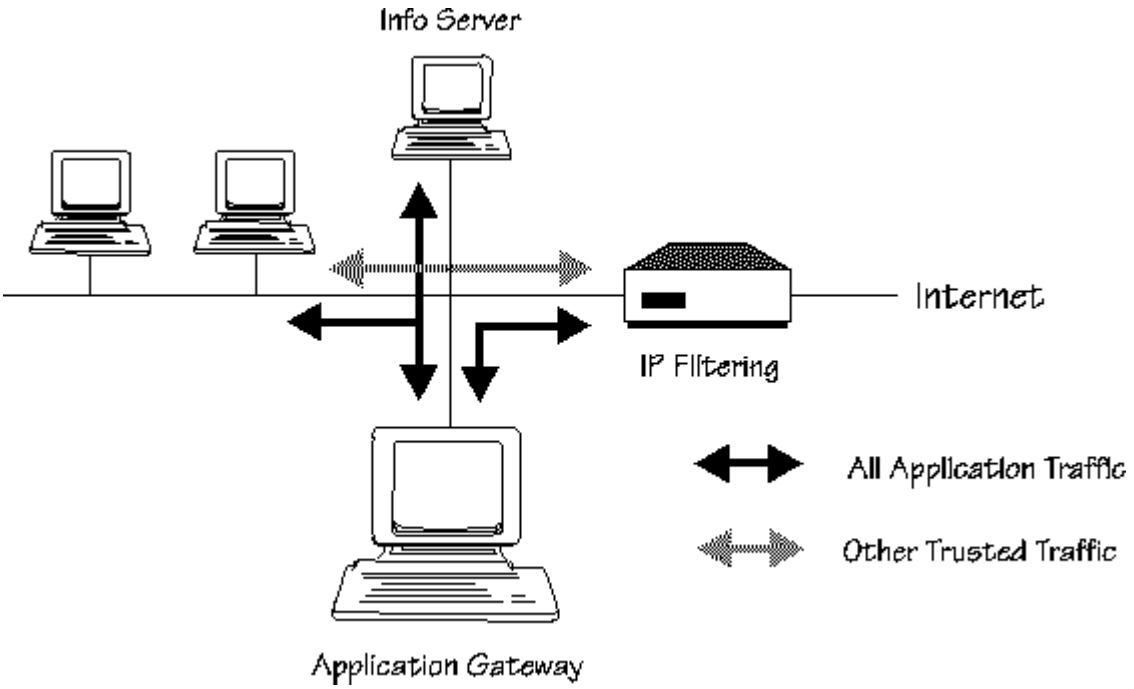
Dual-homed Gateway Firewall with Router.

- Design policy - deny all services unless they are specifically permitted, since no services pass except those for which proxies exist.
- Achieves a higher degree of privacy since the names and IP addresses of site systems are hidden from Internet systems, because the firewall does not pass DNS information.
- The firewall can house software to require users to use authentication tokens or other advanced authentication measures.
- The firewall can also log access and log attempts or probes to the system that might indicate intruder activity.
- The security of the host system used for the firewall must be very secure, as the use of any vulnerable services or techniques on the host could lead to break-ins.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Screened Host Firewall

- More flexible firewall than the dual-homed gateway firewall, but is less secure.
- Combines a packet-filtering router with an application gateway located on the protected subnet side of the router.
- The router filters inherently dangerous protocols
- Router rejects (or accepts) application traffic according to the following rules:
  - application traffic from Internet sites to the application gateway gets routed,
  - all other traffic from Internet sites gets rejected, and
  - the router rejects any application traffic originating from the inside unless it came from the application gateway.



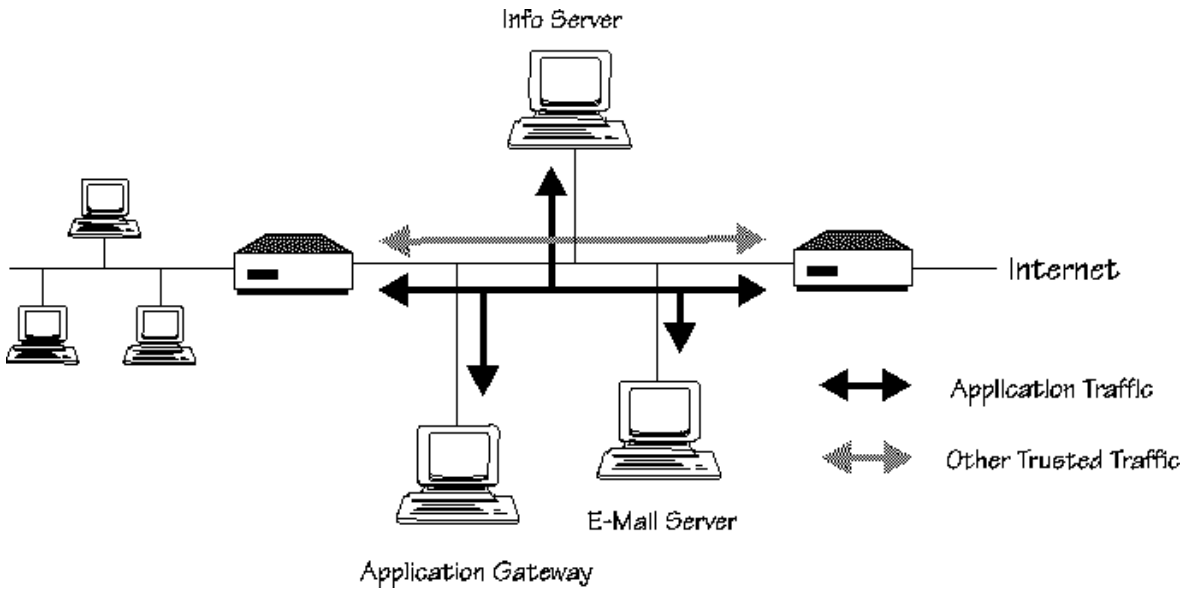
Screened Host Firewall.

- Needs only one network interface and does not require a separate subnet between the application gateway and the router.
- Permits the router to pass certain trusted services ``around'' the application gateway and directly to site systems.
- There are now two systems, the router and the application gateway, that need to be configured carefully.
- Opens up the possibility that the policy can be violated.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Screened Subnet Firewall

- Can be used to locate each component of the firewall on a separate system, thereby achieving greater throughput and flexibility.
- Two routers are used to create an inner, screened subnet



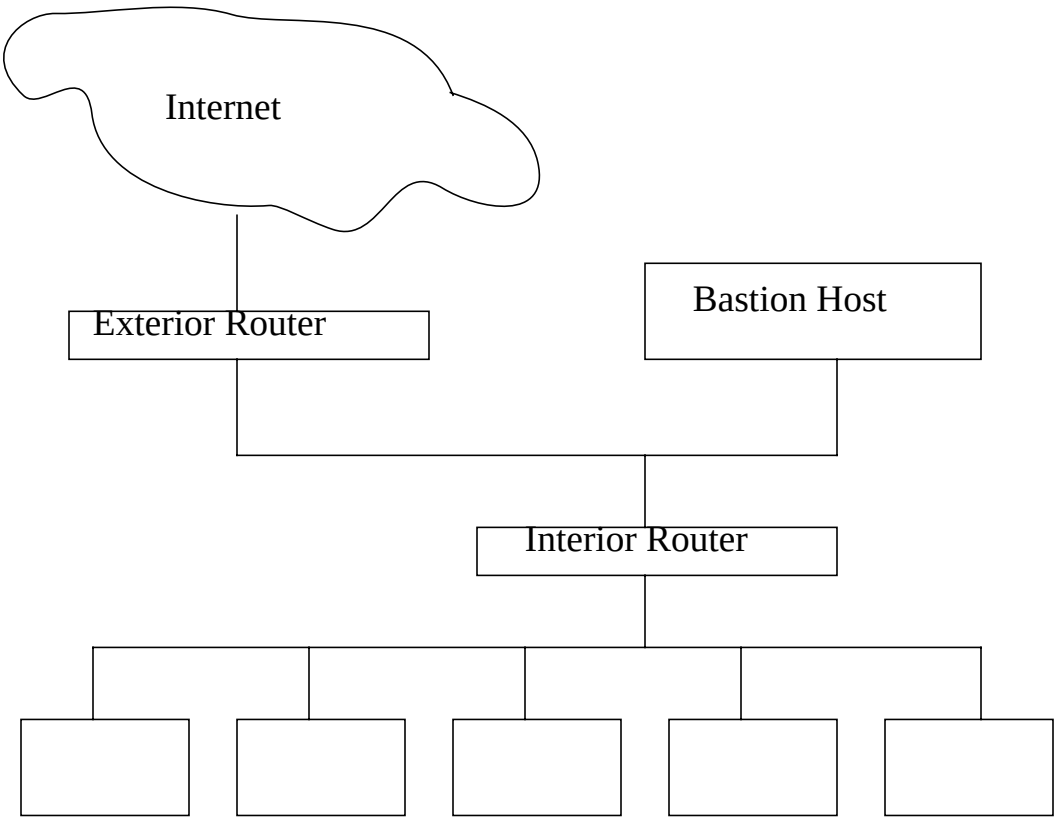
Screened Subnet Firewall with Additional Systems.

- No site system is directly reachable from the Internet and vice versa.
- Routers are used to direct traffic to specific systems, thereby eliminating the need for the application gateway to be dual-homed.
- Appropriate for sites with large amounts of traffic or sites that need very high-speed traffic.
- The two routers provide redundancy since an attacker would have to subvert both routers.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Example Filtering Rules

The filtering rules on the following 2 pages are example rules for an Interior router in a Screened Subnet firewall configuration, such as the one shown below.



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Example Filtering Rules

Rule	Direction	Source Address	Dest Address	Protocol	Source Port	Dest Port	ACK	Action
spoof	in	internal	any	any	any	any	any	deny
telnet1	out	internal	any	TCP	>1023	23	any	permit
telnet2	in	any	internal	TCP	23	>1023	yes	permit
ftp1	out	internal	any	TCP	>1023	21	any	permit
ftp2	in	any	internal	TCP	21	>1023	yes	permit
ftp3	out	internal	any	TCP	>1023	>1023	any	permit
fpt4	in	any	internal	TCP	>1023	>1023	any	permit
ftp5	out	internal	bastion	TCP	>1023	21	any	permit
ftp6	in	bastion	internal	TCP	21	>1023	any	permit
ftp7	in	bastion	internal	TCP	any	6000-6003	any	deny
ftp8	in	bastion	internal	TCP	>1023	>1023	any	permit
ftp9	out	internal	bastion	TCP	>1023	>1023	any	permit
smtp1	out	internal	bastion	TCP	>1023	25	any	permit
smtp2	in	bastion	internal	TCP	25	>1023	any	permit
smtp3	in	bastion	internal	TCP	>1023	25	any	permit
smtp4	out	internal	bastion	TCP	25	>1023	any	permit
nntp1	out	internal	any	TCP	>1023	119	any	permit
nntp2	in	any	internal	TCP	119	>1023	any	permit
nntp3	in	any	internal	TCP	>1023	119	any	permit
nntp4	out	internal	any	TCP	119	>1023	any	permit
http1	out	internal	bastion	TCP	>1023	80	any	permit
http2	in		internal	TCP	80	>1023	any	permit
dns1	out	internal	bastion	UDP	53	53	x	permit

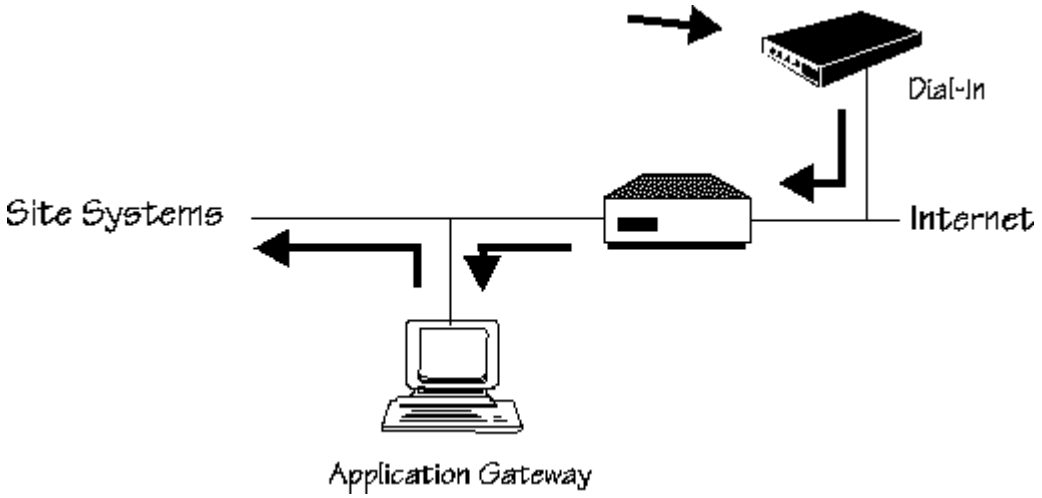
Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Rule	Direction	Source Address	Dest Address	Protocol	Source Port	Dest Port	ACK	Action
dns2	in		internal	UDP	53	53	x	permit
dns3	out	internal	bastion	TCP	>1023	53	any	permit
dns4	in	bastion	internal	TCP	53	>1023	any	permit
dns5	in	bastion	internal	TCP	>1023	53	any	permit
dns6	out	internal	bastion	TCO	53	>1023	any	permit
default1	out	any	any	ANY	any	any	any	deny
default2	in	any	any	ANY	any	any	any	deny

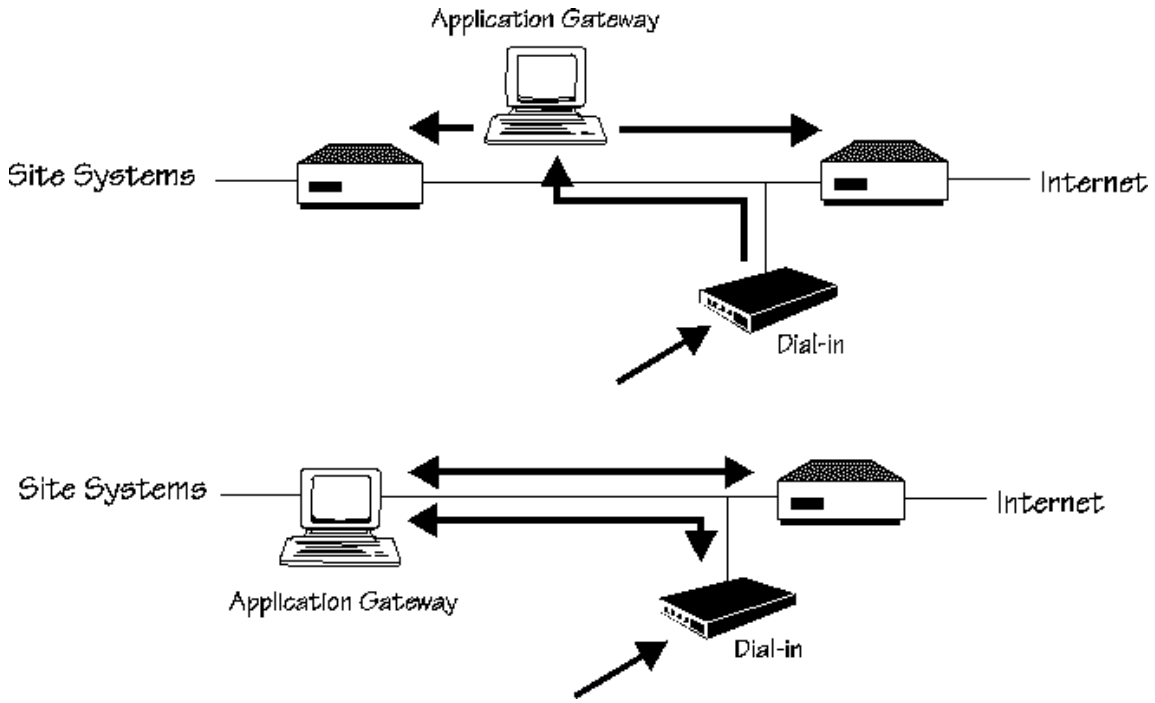
x -- UDP packets do not have and ACK bit

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Integrating Modem Pools with Firewalls



Modem Pool Placement with Screened Host Firewall.



Modem Pool Placement with Screened Subnet and Dual-Homed Firewalls.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

This page is intentionally blank.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Section 11

Network Security III
Public Key
Infrastructure

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Why is PKI Needed

Public Key Cryptography - a brief review

Unlike symmetric key cryptography, public key cryptography in its simplest form does not require a protected exchange of encryption keys.

- Alice and Bob, wish to communicate with each other.
- Each possesses a public key, Kpub, and a private key, Kpri.
- Both make their Kpub known to all potential correspondents.
- Both keep their Kpri secret.
- The encryption algorithm depends upon the use of both the public and the private key to transform plaintext to ciphertext and back to plaintext. Two services are described below. Other services are variants of these.

Service	**Transform Original Plaintext**	**Ciphertext Result**	**Transform Back to Plaintext**
Confidentiality	EKpub(P) where the public key used is that of the intended receiver	ciphertext that can only be decrypted by receiver	EKpri(C) where the private key used is that of the intended receiver
Integrity	EKpri(P) where the private key used is that of the sender	ciphertext that could only have been generated by the sender	EKpub(C) where the public key used is that of the sender

Problem

How do Alice and Bob distribute their public keys?

The implications of this problem and its generally accepted proposed solution have resulted in the need for a public key infrastructure.

Let’s look at how public keys might be exchanged

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Public Key Distribution

Implications of Poor Public Key Services

- binding of key to individual uncertain
- no way to verify public key
- in the case of compromise, no way to revoke old public key
- no way to replace keys after a specified "lifetime" has elapsed

NOTE!

Keep in mind that public keys may be rather large numbers. Key lengths of 512 and 1024 bits are common.
People are going to have to distribute big numbers.

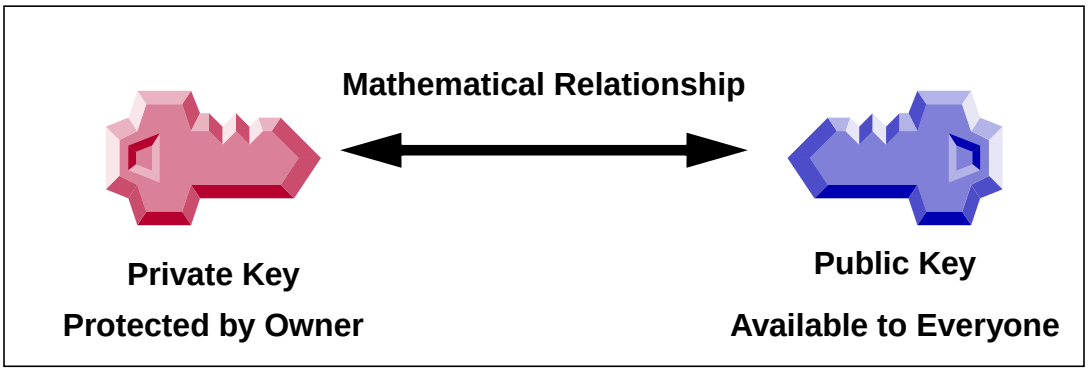
Methods for Public Key Distribution

- Hand carry or mail - floppy or written key
  - cumbersome
  - inefficient
  - error prone.
- Post public key on personal web page.
  - Whose web page?
  - Man in the middle attack
  - revocation
- Send public key in e-mail
  - Whose e-mail?
  - Man in the middle attack
- Have key held by trusted third party

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Public Key Certificates

Provide a reliable binding between the identity of the key's owner and the public key in a form that can be distributed openly. Certificates make public key cryptography feasible. For example, using certificates it is possible to verify digital signatures



Satisfy the following objectives

- Reliable binding between key owner and key
- Management of registration process off line
- Certificate validation process
- Automated services
- Dynamic administration

Issues that must be addressed

- Format of certificates
- Who issues certificates
- How are certificates revoked?
- Where are certificates stored?
- How do I find someone's certificate?
  - Are there privacy issues?
- How are certificates accessed?
- How do certificate holders interoperate?

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Uses of Public Key Certificates

The X.509-based certificate infrastructure is required for a number of protocols used for electronic commerce and secure communications. Among them are:

S/MIME

Secure/Multipurpose Internet Mail Extension allows the capability to sign and encrypt e-mail messages. S/MIME supports the following services:

- Enveloped data consisting of encrypted data with encrypted session key exchange
- Signed data formed using a digital signature of a message digest. The content and the signature are then encoded using base64 encoding, which maps 6-bit blocks of input into 8-bit blocks of output.
- Clear-signed data in which the payload is in the clear but the digital signature is base64 encoded. This permits receivers without S/MIME capability to view the contents of the message, although they cannot verify it.
- Signed and enveloped data consists of mix and match combinations of the above three services.

IP Security

IPSec provides mechanisms for the encryption and authentication of all formats of data at the IP level. This capability permits the protection of all application transmission across unknown networks rather than protection based upon application-level security services.

SSL

Secure Sockets Layer provides security for web traffic. Its services are interposed at the transport layer above TCP in the standard ISO protocol stack.

Secure Electronic Transactions

The SET protocol permits the use of credit cards over the Internet. Advocated by a number of credit card companies it provides for user privacy when making purchases, but insure that vendors are paid for their goods.

SET is an extremely complex protocol involving many communications stages. Although it is believed to provide better security than SSL only time will tell whether or not it will be adopted.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Basic PKI Entities

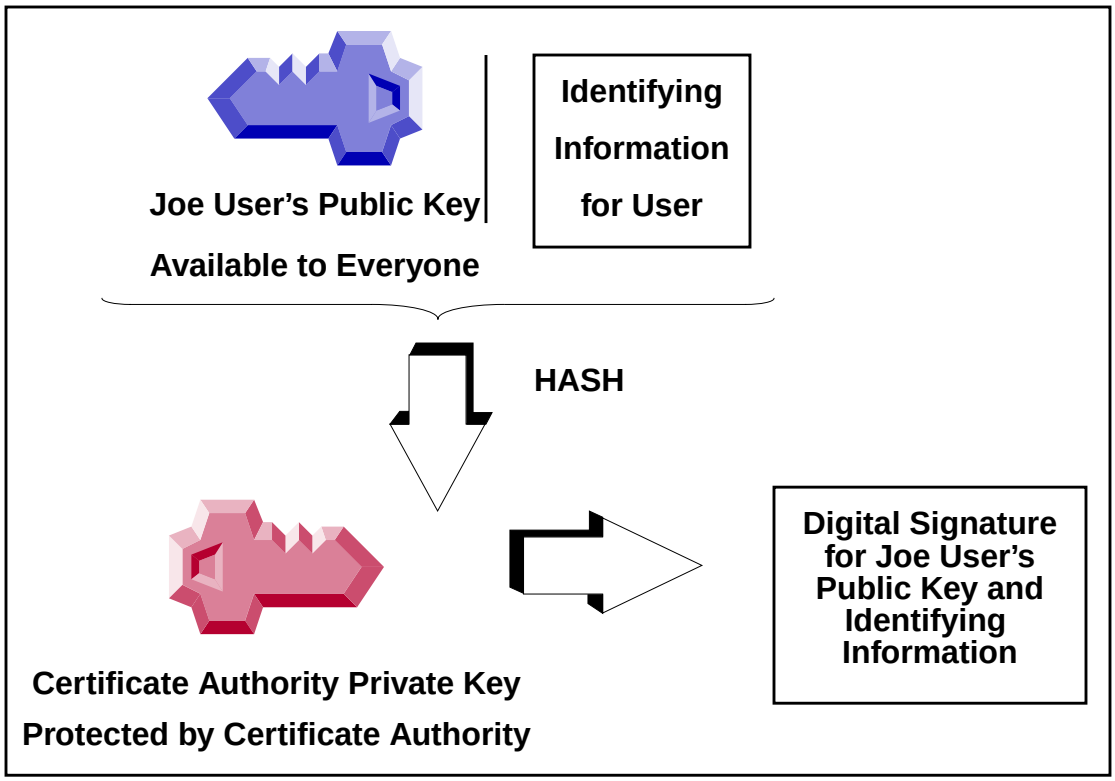
Registration Authority

Performs proof of identity checks insuring that the identity exists and that the registrant is associated with the identity. By bringing a birth certificate to the passport office a person proves their identity. At the registration authority one might be required to provide convincing evidence of identity. The registration authority is intended to be an outlet for the certificate authority that allows users to register in their locality with a certificate authority.

Use of a registration authority also separates the role of certificate issuance and signing from that of verification of identity and collection of registration information.

Certificate Authority

The principle task of the certificate authority is to bind the identity to the certificate and issue the certificate. A certificate authority may accomplish this by creating a digital signature from user public keys and identifying information.



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Basic PKI Entities

Certificate Authority (continued)

The certificate authority will issue the certificate which may be maintained in a certificate directory.

The certificate authority can be queried regarding the validity of a certificate.

The private key of the certificate authority must be carefully protected.

Root Certificate Authority

When certificate authorities are organized in a hierarchical

Certificate Directory

Provides a repository containing certificates for some group of entities.

X.500 directories were initially proposed as the standard, however there are now a variety of proprietary directories, so industry is attempting to standardize the directory access protocol using LDAP, Lightweight Directory Access Protocol.

Having been created by a protected certificate authority, certificates are unforgeable. Thus it is presumed that no additional protection need be provided for certificates. If all users subscribe to the same certificate authority, then they can send keys to each other and can verify them using the public key of their common certificate authority.

In large organizations or for inter-organization authentication there may be multiple certificate authorities. For example a group of users at university A may share a given certificate authority, but users at university B may use a different certificate authority.

To verify a copy of user Bob at university B, Alice must obtain a verified copy of the public key for the certificate authority serving university B.

This technique is known as ***chaining of certificates***.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Basic PKI Entities

Personal Security Environment

Within this protected environment, a user's private key or keys are stored. A number of techniques are available for private key protection. Some are not particularly secure.

passwords - keys are vulnerable to password cracking and must be used in memory for the algorithms.

memory cards - store keys. Problem keys must be exposed to be used in regular system memory.

smartcards - perform encryption and signature operations as well as store keys

Certificate Validation Protocols

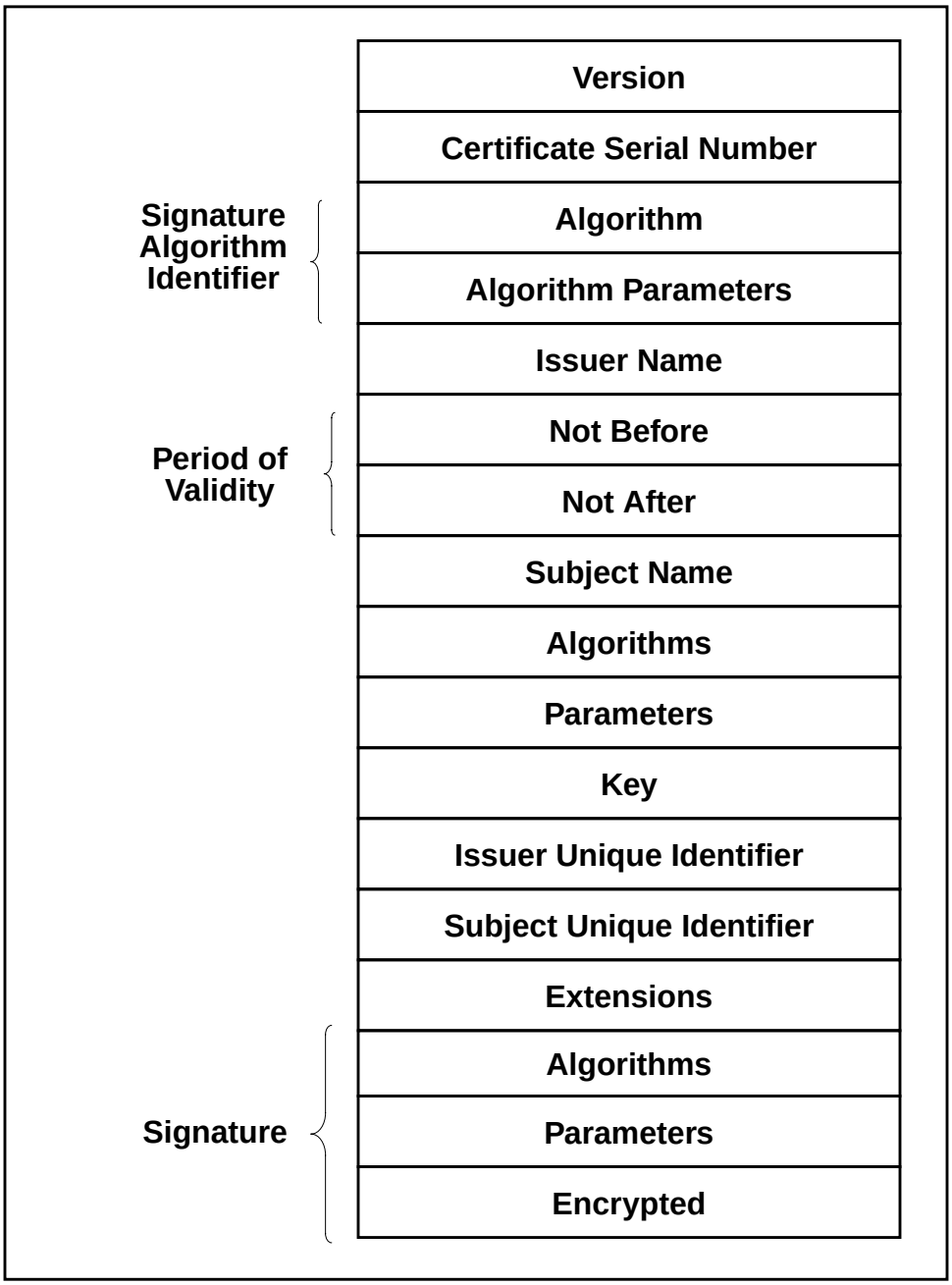
This may consist of certificate revocation lists or on-line verification protocols such as On-line Certificate Status Protocol.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Format of Certificates

Most certificates are based on the X.509 format.

Note, this is where the term PKIX derives. It stands for Public Key Infrastructure for X.509 and is a working group of the Internet Engineering Task Force.



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Certificate Contents

The various fields in the X.509 certificate are

Version

Our illustration is of an X.509 Version 3 certificate. The version number permits the differentiation of the various X.509 certificate versions. This field would be inspected and a switch statement in the code would permit appropriate processing.

Serial Number

A per-CA identifier for the certificate. This means that a certificate is uniquely identified by a combination of the identifier of the CA itself and the identified assigned to the certificate by the CA.

Signature Algorithm Identifier

Identified the algorithm used to sign the certificate. It is redundant.

Issuer Name

The X.500 name of the certificate authority that created and signed the certificate

Period of Validity

Places a time frame on the validity of the certificate

Subject Name

Name of the user whose identity is bound to the certificate.

Subject Public Key Information

The public key, the algorithm for which the public key is valid, and related parameters.

Issuer Unique Identifier

Optional unique identifier for the CA. This might be required if the X.500 name has been reused for different entities.

Subject Unique Identifier

Optional unique identifier for the CA. This might be required if the X.500 name has been reused for different entities.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Certificate Contents (continued)

Extensions

- Policy extensions contain additional information about the subject as well as issuer policy. Policy might identify the context in which the certificate is valid.
- Subject and issuer attribute extensions convey additional information about these entities
- Certificate path constraints may constrain the chaining of CAs and/or the types of certificates that may be issued by a CA.

Signature

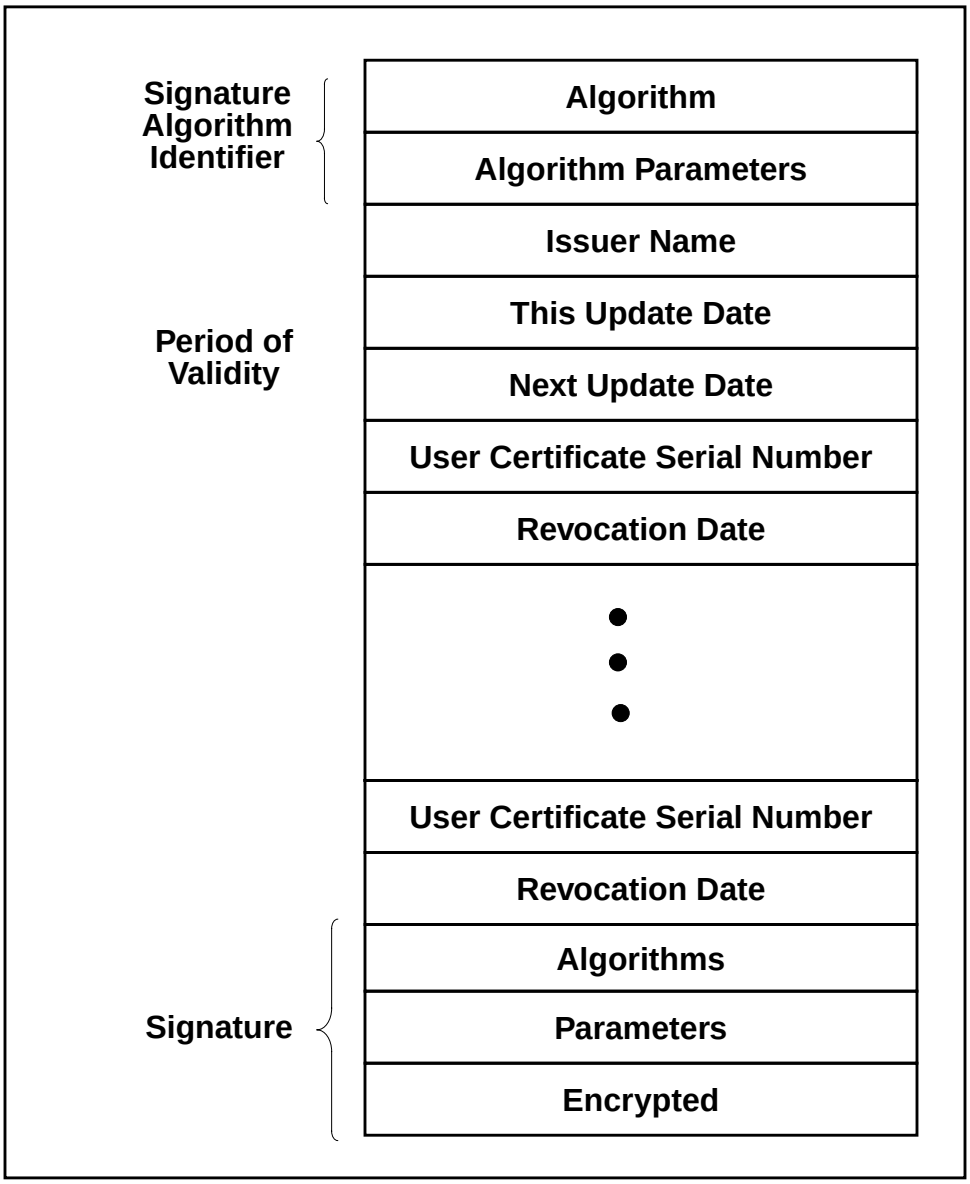
Hash of all other fields in the certificate encrypted with the CA's private key. The signature algorithm is identified.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Certificate Revocation List

Certificate revocation lists are analogous to the lists of cancelled credit card numbers that vendors used to be issued. When a customer presented his or her credit card, the cashier would look on the list to determine whether the credit card was still valid.

The certificate revocation list has one signature applied to the entire list. The update dates allow users to synchronize the use of CRLs.



Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

Certificate Validation

Certificate revocation lists may be partitioned based on the reason for the certificate's revocation. For example, a certificate revoked because an individual no longer works for a particular employer may require less immediate action than one revoked because the user has just been indicted for embezzlement.

Certificate revocation lists have the disadvantage that they are somewhat static. Unless the CRL is received, a user has no way of determining whether a particular certificate is still valid.

On-line Certificate Status Protocol (OCSP)

Just as the credit card industry has moved to on-line validation, so certificates can be validated on line. This provides a more timely check of the certificate's validity. New protocols include real time versions such as Real-time Certificate Status Protocol (RCSP).

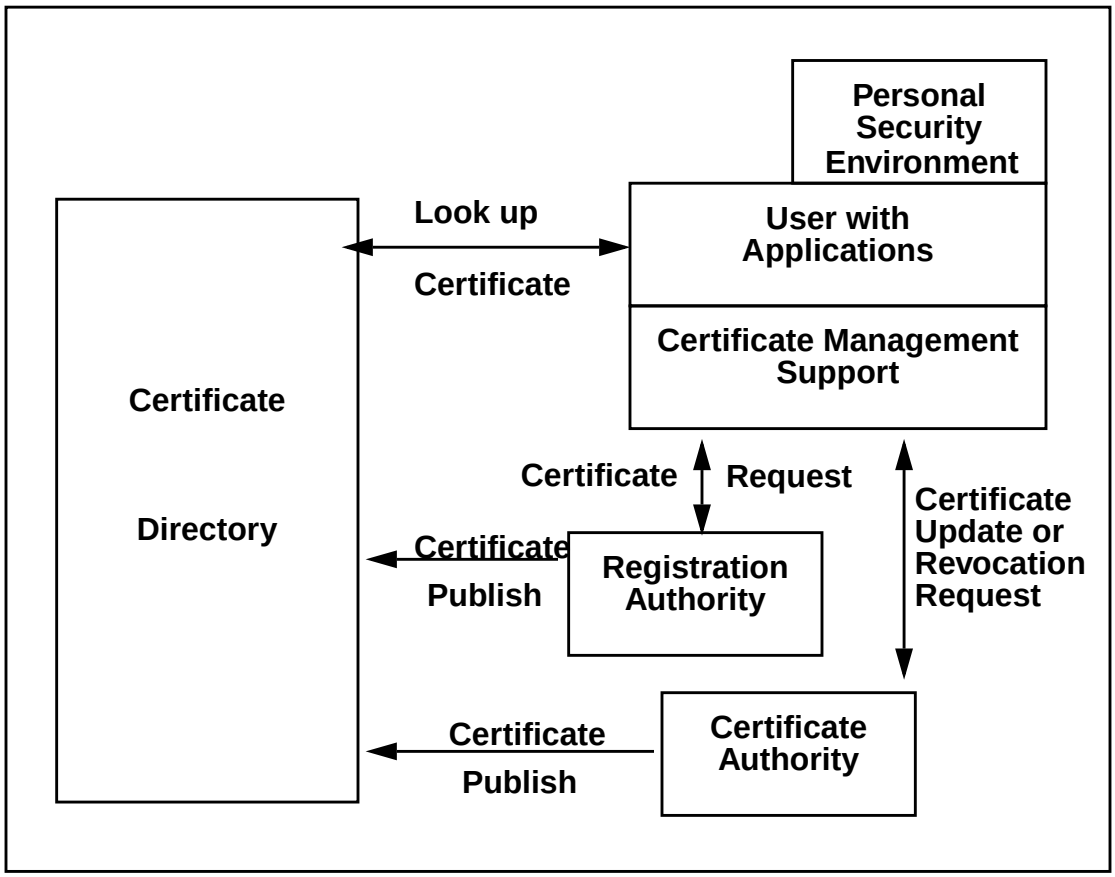
PROBLEM!

Most current applications (browsers) do not provide support for validation of certificates.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

PKI Usage

The figure below depicts how public keys might be used and managed.



The creation of certificates will be ruled by certain policies and procedures

- Verification procedures
- Certificate context - what certificates are good for
- Certificate content - is the certificate only for the key pair or for other organizational information
- Certificate lifetimes
- Assurance provided by CA
- Key Escrow and Recovery Services
- CA cross certification support

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

The Future of PKI

Today the PKI is in its formative stages.

Challenges include

Interoperability

an interesting attempt to deal with interoperabilty issues is the ACES project of GSA.

PSE protection issues

Certificate validation by applications and trust of certificate validation entities.

Key management issues.

For example what to do about lost keys

User Acceptance

Always the great challenge in security. If the user's won't use the security features, it really doesn't matter how much security they provide.

Liability issues

Who is to blame? The CA? The user? ...

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

PKI Etcetera

Orchestration of PKI takes place using two tiers of protocols and standardized formats

- Management Protocols
- PKIX CMP - certificate management protocol
- CMMF - certificate management message format
- Operational Protocols
  - certificate formats such as X.509
  - e-mail
  - HTTP - Hypertext Transfer Protocol
  - LDAP - Lightweight Directory Access Protocol
  - etc

PKI Glossary

PKI - Public Key Infrastructure

PKIX - Public Key Infrastructure for X.509

X.509 - certificate format

PKIX CMP - PKIX Certificate Management Protocol

CMMF - Certificate Management Message Format

CA - Certificate Authority

RA - Registration Authority

Public Key - Key available to everyone

PSE - Personal Security Environment

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

This page is intentionally blank.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

This page is intentionally blank.

Copyright (c) Naval Postgraduate School Center for INFOSEC Studies and Research

This page is intentionally blank.