Comp 5370 Midterm Allowed Note/Cheat Sheet

John Carroll – 902521946/jcc0044

**\*Confidentiality**

* Concept: data is not disclosed to unauthorized individuals, entities, or processes
* Assumption: reliance on correct data
* Via
  + encryption
  + access control mechanisms
  + hiding

**\*Integrity**

* Concept: trustworthiness of the data
  + data integrity
    - data has not been changed, destroyed, or lost in an unauthorized or accidental manner
  + origin integrity
    - data comes from a credible source
* Assumption: source of data and trust in source
* Via
  + prevention
    - block unauthorized attempts to change data
    - block changes that are not authorized
  + detection
    - identify loss of trust in data's integrity

**\*Availability**

* Concept: data is accessible and usable upon demand by an authorized system entity, according to performance specifications
* Assumption: criticality, reliability, survivability characteristics are known and valid
* Via
  + ?

Security Policies

* Policy partitions system states into:
  + Authorized (secure)
    - These are states the system can enter
  + Unauthorized (non-secure)
    - If the system enters any of these states, it is a security violation
* Secure system
  + Starts in authorized state
  + Never enters unauthorized state

Transformations

* Discretionary Access Control (DAC)
  + individual user sets access control mechanism to allow or deny access to an object
* Mandatory Access Control (MAC)
  + system mechanism controls access to object, and individuals cannot alter that access
* Originator Controlled Access Control (ORCON)
  + originator (creator) of information controls who can access information

Policies vs Mechanisms

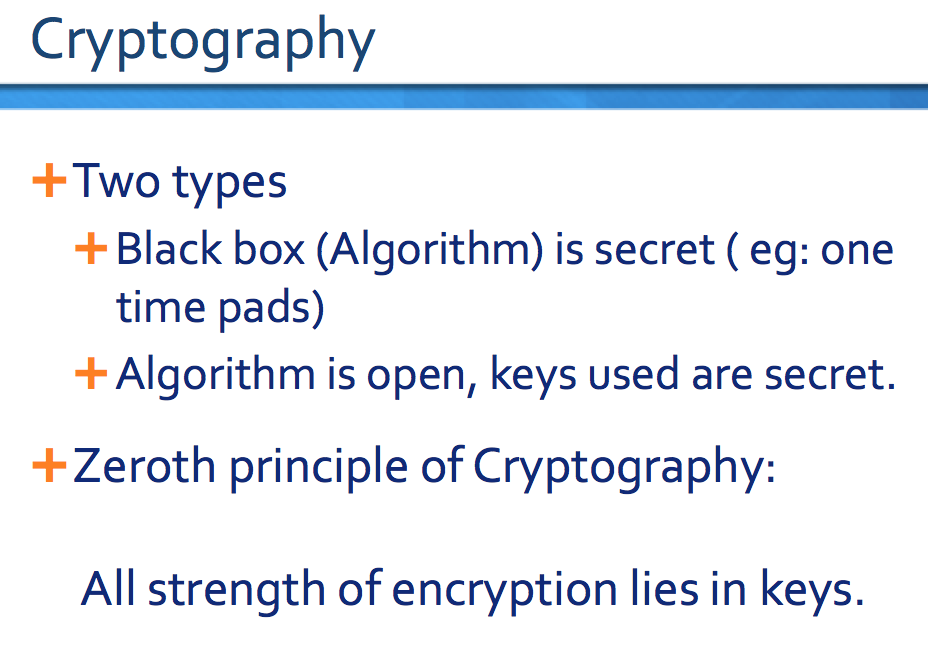
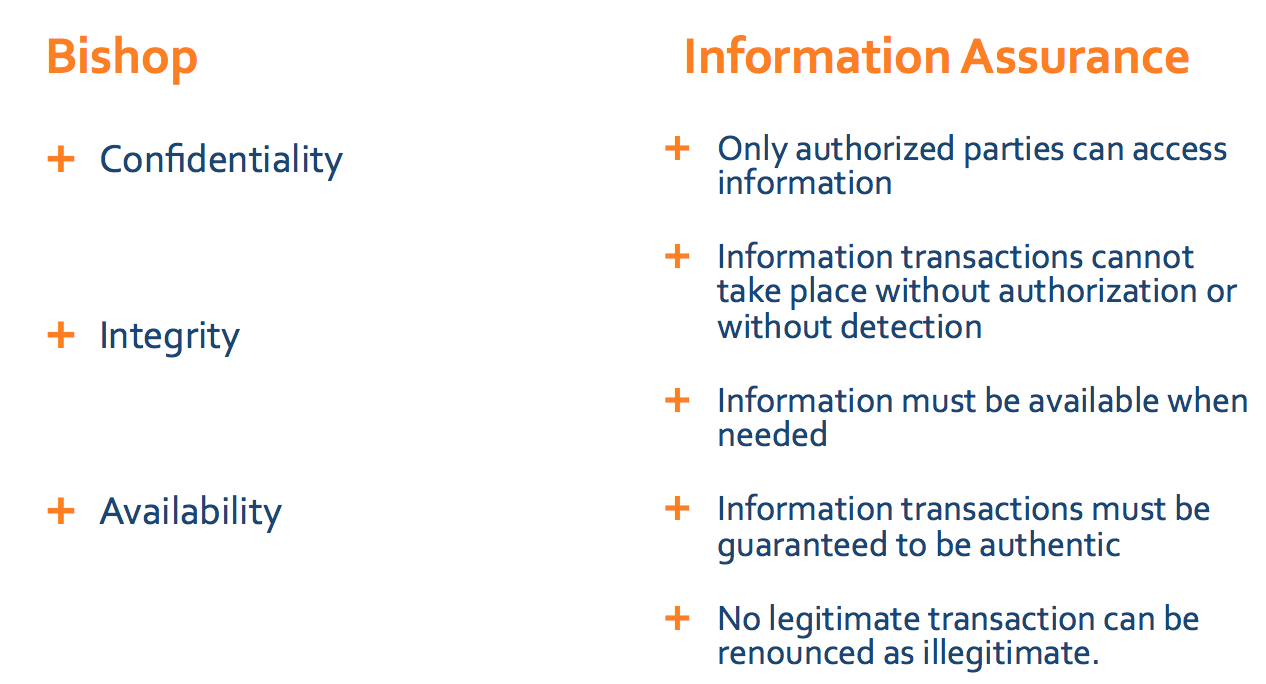
* Policies describe *what* is allowed
* Mechanisms control *how* policies are enforced
* *Trust* underlies everything

Confidentiality Policy

* Goal: prevent the unauthorized disclosure of information
  + Focus on information flow
  + Classical approach: Bell-LaPadula Model

Bell LaPadula Model

* Security levels arranged in linear ordering
  + Top Secret
  + Secret
  + Unclassified
* Objects have *security classification L*(*o*)
* Subjects have *security clearance L(s)*
* Expand notion of security level to include categories
* Security level is (*clearance*, *category set*)
* Examples
  + ( Top Secret, { NUC, EUR, ASI } )
  + ( Confidential, { EUR, ASI } )
  + ( Secret, { NUC, ASI } )



Socket programming

* Socket: telephone
* Bind: assign telephone number to a telephone
* Listen: turn on the ringer so that you can hear the phone call
* Connect: dial a phone number
* Accept: answer the phone
* Read/write: talking

Cryptosystems

* Quintuple (E, D, M, K, C)
  + M set of plaintexts
  + K set of keys
  + C set of ciphertexts
  + E set of encryption functions *e*: M × K → C
  + D set of decryption functions *d*: C × K → M

Attacks

* Opponent whose goal is to break cryptosystem is the *adversary*
  + Assume adversary knows algorithm used, but not key
* Three types of attacks:
  + *ciphertext only*: adversary has only ciphertext; goal is to find plaintext, possibly key
  + *known plaintext*: adversary has ciphertext, corresponding plaintext; goal is to find key
  + *chosen plaintext*: adversary may supply plaintexts and obtain corresponding ciphertext; goal is to find key
* Mathematical attacks
  + Based on analysis of underlying mathematics
* Statistical attacks
  + Make assumptions about the distribution of letters, pairs of letters (digrams), triplets of letters (trigrams), *etc.*
  + Examine ciphertext, correlate properties with the assumptions.
* Classic Cryptography
  + Cæsar cipher
  + Vigènere cipher
  + DES
* Public Key Cryptography
  + Diffie-Hellman
  + RSA
* Cryptographic Checksums
  + HMAC

Classical Cryptography

* Sender, receiver share common key
  + Keys may be the same, or trivial to derive from one another
  + Sometimes called *symmetric cryptography*
* Two basic types
  + Transposition ciphers
  + Substitution ciphers
  + Combinations are called *product ciphers*

Public Key Cryptography

* Two keys
  + *Private key* known only to individual
  + *Public key* available to anyone
    - Public key, private key inverses
* Idea
  + Confidentiality: encipher using public key, decipher using private key
  + Integrity/authentication: encipher using private key, decipher using public one
* Requirements
  + Computationally easy to encipher or decipher a message given the appropriate key
  + Computationally infeasible to derive the private key from the public key
  + Computationally infeasible to determine the private key from a chosen plaintext attack

RSA

* History
  + 1977 MIT
  + Ron Rivest, Adi Shamir and Leonard Adleman
* Exponentiation cipher
  + meaning, relies on the difficulty of determining the number of numbers relatively prime to a large integer *n*
* Asymmetric approach involving 3 parts
  + key generation
  + encryption
  + decryption

Security Services

* Confidentiality
  + Only the owner of the private key knows it, so text enciphered with public key cannot be read by anyone except the owner of the private key
* Authentication
  + Only the owner of the private key knows it, so text enciphered with private key must have been generated by the owner
* Integrity
  + Enciphered letters cannot be changed undetectably without knowing private key
* Non-Repudiation
  + Message enciphered with private key came from someone who knew it

Keyed vs Keyless

* Keyed cryptographic checksum: requires cryptographic key
  + DES in chaining mode: encipher message, use last *n* bits. Requires a key to encipher, so it is a keyed cryptographic checksum.
* Keyless cryptographic checksum: requires no cryptographic key
  + MD5 and SHA-1 are best known; others include MD4, HAVAL, and Snefru

Location of Encryption Device

* Link encryption:
  + A lot of encryption devices
  + High level of security
  + Decrypt each packet at every switch
* End-to-end encryption
  + The source encrypt and the receiver decrypts
  + Payload encrypted
  + Header in the clear
* High Security: Both link and end-to-end encryption are needed

Key points

* Two main types of cryptosystems: classical and public key
* Classical cryptosystems encipher and decipher using the same key
  + Or one key is easily derived from the other
* Public key cryptosystems encipher and decipher using different keys
  + Computationally infeasible to derive one from the other
* Cryptographic checksums provide a check on integrity

Cryptographic Key Infrastructure

* Goal: bind identity to key
* Classical: not possible as all keys are shared
  + Use protocols to agree on a shared key
* Public key: bind identity to public key
  + Crucial as people will use key to communicate with principal whose identity is bound to key
  + Erroneous binding means no secrecy between principals
  + Assume principal identified by an acceptable name

PKI vs WoT

* PKI
  + puts trust decision at single point
  + root certificate widely distributed
* WoT
  + leaves trust decisions in hands of users
  + certificates not widely distributed, but may be difficult to get

Key points

* Key management critical to effective use of cryptosystems
  + Different levels of keys (session *vs*. interchange)
* Keys need infrastructure to identify holders, allow revoking
  + Key escrowing complicates infrastructure
* Digital signatures provide integrity of origin and content

Much easier with public key cryptosystems than with classical cryptosystems

Security Architecture and Models – key points

* Basic functions of security
  + confidentiality, integrity, availability
* Terminology
  + threat, attack, policy, mechanism, assurance
* Protecting access
  + access control matrix
* Security Policies
  + Confidentiality Policies
    - Bell-LaPadula Model
  + Integrity Policies
    - Biba, Clark-Wilson
  + Hybrid Policies
    - Chinese Wall, CISS
  + Hybrid Models
  + TCSEC, TCB
  + IPSEC

Requirements of integrity policy

* Users must use existing production software
  + They cannot write their own software
* Programmers will develop and test programs on a non-production system
  + Explicit process defines how production data provided for dev
* Explicit process defines migration of dev to prod
  + Process is controlled and audited
* Managers and auditors have access to the system state and system logs

Models information

* Noninterference model - Covers ways to prevent subjects operating in one domain from affecting each other in violation of security policy
* State machine model - Abstract mathematical model consisting of state variables and transition functions
* Chinese Wall Model – provides a model for access rules in a consultancy business where analysts have to make sure that no conflicts of interest arise
* Lattice Model - The higher up in secrecy, the more constraints on the data; the lower in secrecy, the less constraints on the data

Definitions

* Access control - Prevention of unauthorized use or misuse of a system
* ACL - Access control list
* Access Mode - An operation on an object recognized by the security mechanisms - think read, write or execute actions on files
* Accountability- Actions can be correlated to an entity
* Accreditation - Approval to operate in a given capacity in a given environment
* Asynchronous attack - An attack exploiting the time lapse between an attack action and a system reaction
* Audit trail - Records that document actions on or against a system
* Bounds Checking - Within a program, the process of checking for references outside of declared limits. When bounds checking is not employed, attacks such as buffer overflows are possible
* Compartmentalization - Storing sensitive data in isolated blocks
* Configuration Control - management and control of changes to a system’s hardware, firmware, software, and documentation
* confinement - Ensuring data cannot be abused when a process is executing a borrowed program and has some access to that data
* Contamination – Corruption of data of varying classification levels
* Correctness Proof - Mathematical proof of consistency between a specification and implementation
* Countermeasure - anything that neutralizes vulnerability
* Covert Channel - A communication channel that allows cooperating processes to transfer information in a way that violates a system’s security policy
* covert storage channel involves memory shared by processes
* covert timing channel involves modulation of system resource usage (like CPU time)
* Criticality - Importance of system to mission
* Cycle - One cycle consists of writing a zero, then a 1 in every possible location
* Data Contamination - Deliberate or accidental change in the integrity of data
* Discretionary Access Control - An entity with access privileges can pass those privileges on to other entities
* Mandatory Access control - Requires that access control policy decisions are beyond the control of the individual owner of an object (think military security classification)
* DoD Trusted Computer System Evaluation Criteria (TCSEC) - orange book
* Firmware - software permanently stored in hardware device (ROM, read only memory)
* Formal Proof - Mathematical argument
* Hacker/Cracker – Individual who cause Damage
* Logic bomb - An unauthorized action triggered by a system state
* Malicious logic - Evil hardware, software, or firmware included by malcontents for malcontents
* Principle of Least Privilege - Every entity granted least privileges necessary to perform assigned tasks
* Memory bounds - The limits in a range of storage addresses for a protected memory region
* Piggy Back - Unauthorized system via another’s authorized access (shoulder surfing is similar)
* Privileged Instructions - Set of instructions generally executable only when system is operating in executive state
* Reference Monitor - A security control which controls subjects’ access to resources - an example is the security kernel for a given hardware base
* Resource - Anything used while a system is functioning (eg CPU time, memory, disk space)
* Resource encapsulation - Property which states resources cannot be directly accessed by subjects because subject access must be controlled by the reference monitor
* Security Kernel - Hardware/software/firmware elements of the Trusted Computing Base - security kernel implements the reference monitor concept
* Trusted Computing Base - From the TCSEC, the portion of a computer system which contains all elements of the system responsible for supporting the security policy and supporting the isolation of objects on which the protection is based -follows the reference monitor concept
* TCSEC - Trusted Computer Security Evaluation Criteria - Evaluation Guides other than the Orange Book
* ITSEC - Information Technology Security Evaluation Criteria (European)
* CTCPEC - Canadian Trusted Computer Product Evaluation Criteria
* CC - Common Criteria
* Trusted System
* follows from TCB
* A system that can be expected to meet users’ requirements for reliability, security, effectiveness due to having undergone testing and validation
* System Assurance
* the trust that can be placed in a system, and the trusted ways the system can be proven to have been developed, tested, maintained, etc.
* Virus - program that can infect other programs
* Worm - program that propagates but doesn’t necessarily modify other programs
* Bacteria or rabbit - programs that replicate themselves to overwhelm system resources
* Back Doors - trap doors - allow unauthorized access to systems
* Trojan horse - malicious program masquerading as a benign program

Key points Authentication

* Authentication is not cryptography
  + You have to consider system components
* Passwords are here to stay … at least for a while longer
  + They provide a basis for most forms of authentication
* Protocols are important
  + Selection leads to a more secure choice of authentication
  + Storing information properly contributes to security chain
* Discussing multiple factor authentication--- separate discussion!