Security of Information System

Operating Systems, Database and Program Security

Nandika Kasun

Department of Communication and Media Technologies
University of Colombo School of Computing
University of Colombo
Sri Lanka

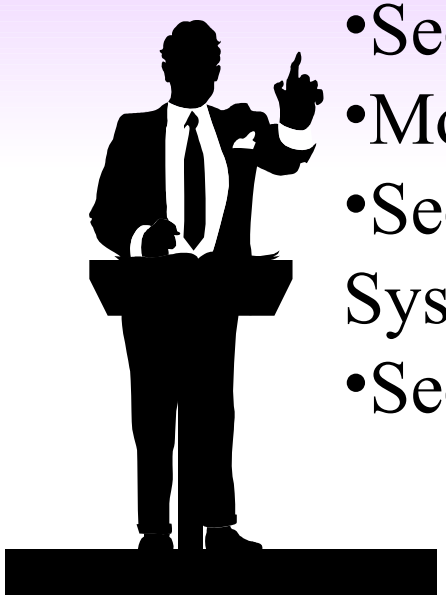
Objectives:

Operating systems, database and program security:

- Identify the security features of ordinary and trusted operating system
- Evaluate the operating system security
- Understand the security requirement of databases
- Describe the types of computer virus and protection methods

kasun@cmb.ac.lk

Operating Systems, Database and Program Security



4.1 Operating Systems Security

- Security Policies
- Models of Security
- Security Features of Ordinary Operating System
- Security Features of Trusted Operating System

Computer System Components

- **Hardware**
  - Provides basic computing resources (CPU, memory, I/O devices).
- **Operating system**
  - Controls and coordinates the use of the hardware among the various application programs.
- **Applications programs**
  - Define the ways in which the system resources are used to solve the computing problems of the users.
- **Users**
  - E.g., people, machines, other computers.

Operating System

Operating systems provide the fundamental mechanisms for securing computer processing. Since the 1960s, operating systems designers have explored how to build “secure” operating systems —operating systems whose mechanisms protect the system against a motivated adversary. Recently, the importance of ensuring such security has become a mainstream issue for all operating systems.

kasun@cmb.ac.lk

What Security Goals Does Operating System Provide?

- **Goal 1: enabling multiple users securely share a computer**
  - Separation and sharing of processes, memory, files, devices, etc.
- **How to achieve it?**
  - memory protection
  - processor modes
  - authentication
  - file access control

What Security Goals Does Operating System Provide?

- Goal 2: ensure secure operation in networked environment
- How to achieve it?
  - Authentication
  - Access Control
  - Logging & Auditing
  - Intrusion Detection
  - Recovery

kasun@cmb.ac.lk

Memory Protection: access control to memory

- Ensures that one user's process cannot access other's memory
  - fence
  - relocation
  - base/bounds register
  - segmentation
  - paging
  - …
- Operating system and user processes need to have different privileges

kasun@cmb.ac.lk

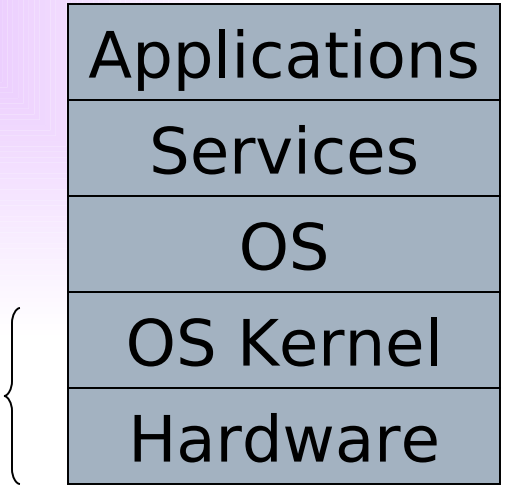
CPU Modes (a.k.a. processor modes or privilege

- **System mode (privileged mode, master mode, kernel mode)**
  - **can execute any instruction and access any memory locations, e.g., accessing hardware devices, enabling and disabling interrupts, changing privileged processor state, accessing memory management units, modifying registers for various descriptor tables .**
- **User mode**
  - **access to memory is limited, cannot execute some instructions**
- **Transition from user mode to system mode must be done through well defined call gates (system calls)**

Reading: http://en.wikipedia.org/wiki/CPU_modes

kasun@cmb.ac.lk

Placing Security in Lower Layers



Two good reasons to place security in lower layers:

1. **It may be possible to evaluate security to a higher level of assurance.**
2. **Putting security mechanisms into the core of the system reduces performance overheads caused by security.**

Three major tasks

- Operating systems must provide efficient *resource mechanisms*,
- Second, it is the operating system's responsibility to switch among the processes fairly
- Third, access to resources should be controlled, such that one process cannot inadvertently or maliciously impact the execution of another.

kasun@cmb.ac.lk

Operating System Security

- The reference monitor is an abstract concept, the security kernel is its implementation, and the trusted computing base contains the security kernel among other protection mechanisms.

Three rules:

1. Keep the security kernel of an operating system as simple as possible.
2. Users must not be able to modify the operating system.
3. Operating system has to prevent users from accidentally or intentionally accessing other users' data.

Access Control

An *access enforcement mechanism* authorizes requests from multiple *subjects* (e.g. users, processes, etc.) to perform *operations* (e.g., read, write, etc.) on objects (e.g., files, sockets, etc.).

An operating system provides an access enforcement mechanism.

Two fundamental concepts of access control:

- a *protection system* that defines the access control specification and
- a *reference monitor* that is the system's access enforcement mechanism that enforces this specification.

kasun@cmb.ac.lk

Protection system

A protection system consists of a protection state, which describes the operations that system subjects can perform on system objects, and a set of protection state operations, which enable modification of that state.

A protection system enables the definition and management of a protection state. A *protection state consists of the specific system subjects, the specific system objects, and the operations that those* subjects can perform on those objects.

The access matrix is used to define the *protection domain* of a process.

	File 1	File 2	File 3
Process 1	Read	Read, Write	Read, Write
Process 2	–	Read	Read, Write

Lanpson's access Matric

kasun@cmb.ac.lk

Problems with access matrix

Untrusted processes can tamper with the protection system.

A protection system that permits untrusted processes to modify the protection state is called a *discretionary access control* (DAC) system.

Mandatory protection system

A *mandatory protection system* is a protection system that can only be modified by trusted administrators via trusted software, consisting of the following state representations:

A *mandatory protection state* is a protection state where subjects and objects are represented by *labels* where the state describes the operations that subject labels may take upon object labels;

A *labelling state* for mapping processes and system resource objects to labels;

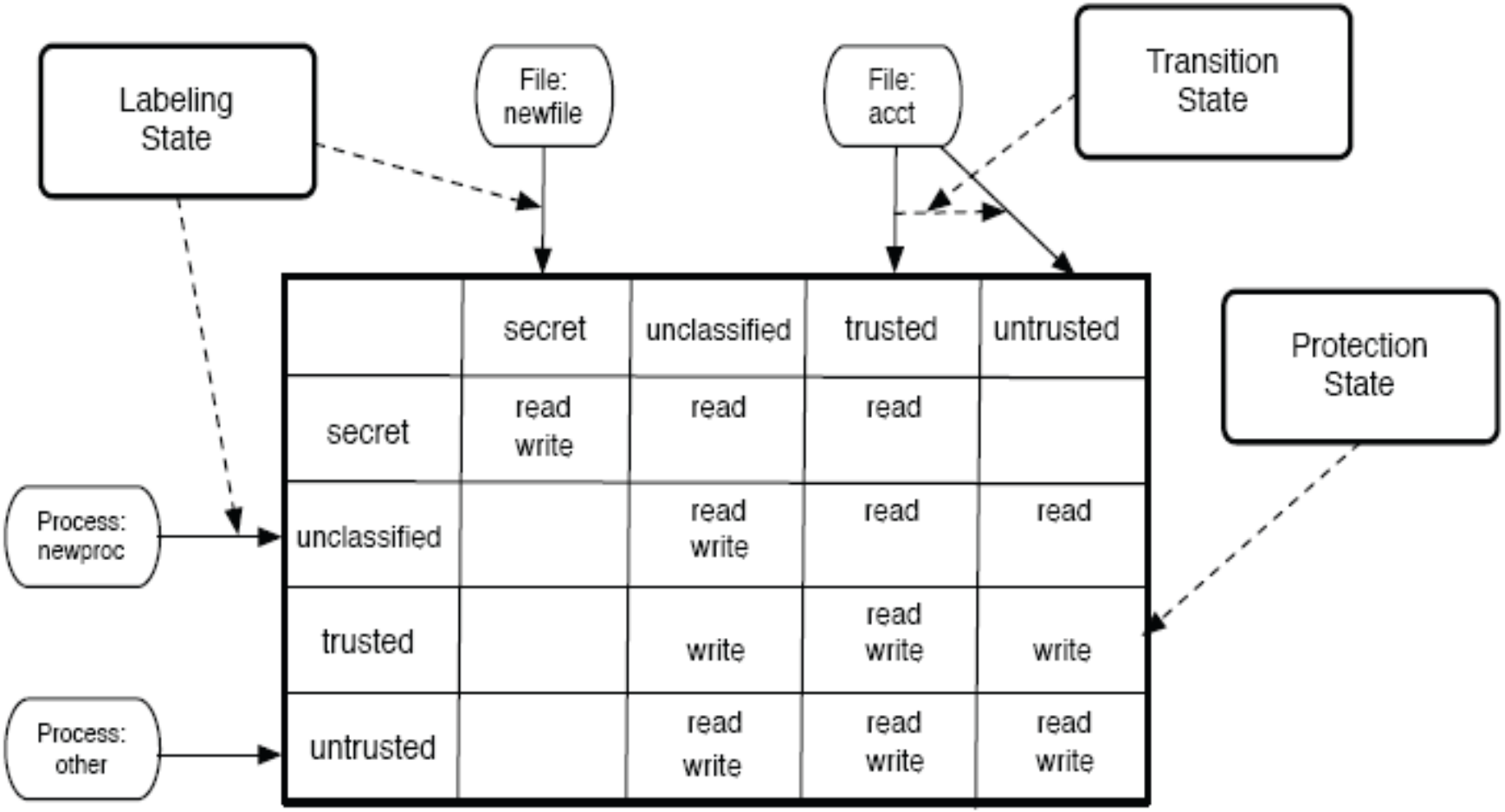
A *transition state* that describes the legal ways that processes and system resource objects may be relabeled.

Mandatory access control

A label is simply an abstract identifier—the assignment of permissions to a label defines its security semantics. Labels are tamperproof .

Trusted administrators define the access matrix's labels and set the operations that subjects of particular labels can perform on objects of particular labels. Such protection systems are *mandatory access control* (MAC) systems because the protection system is immutable to untrusted processes.

kasun@cmb.ac.lk



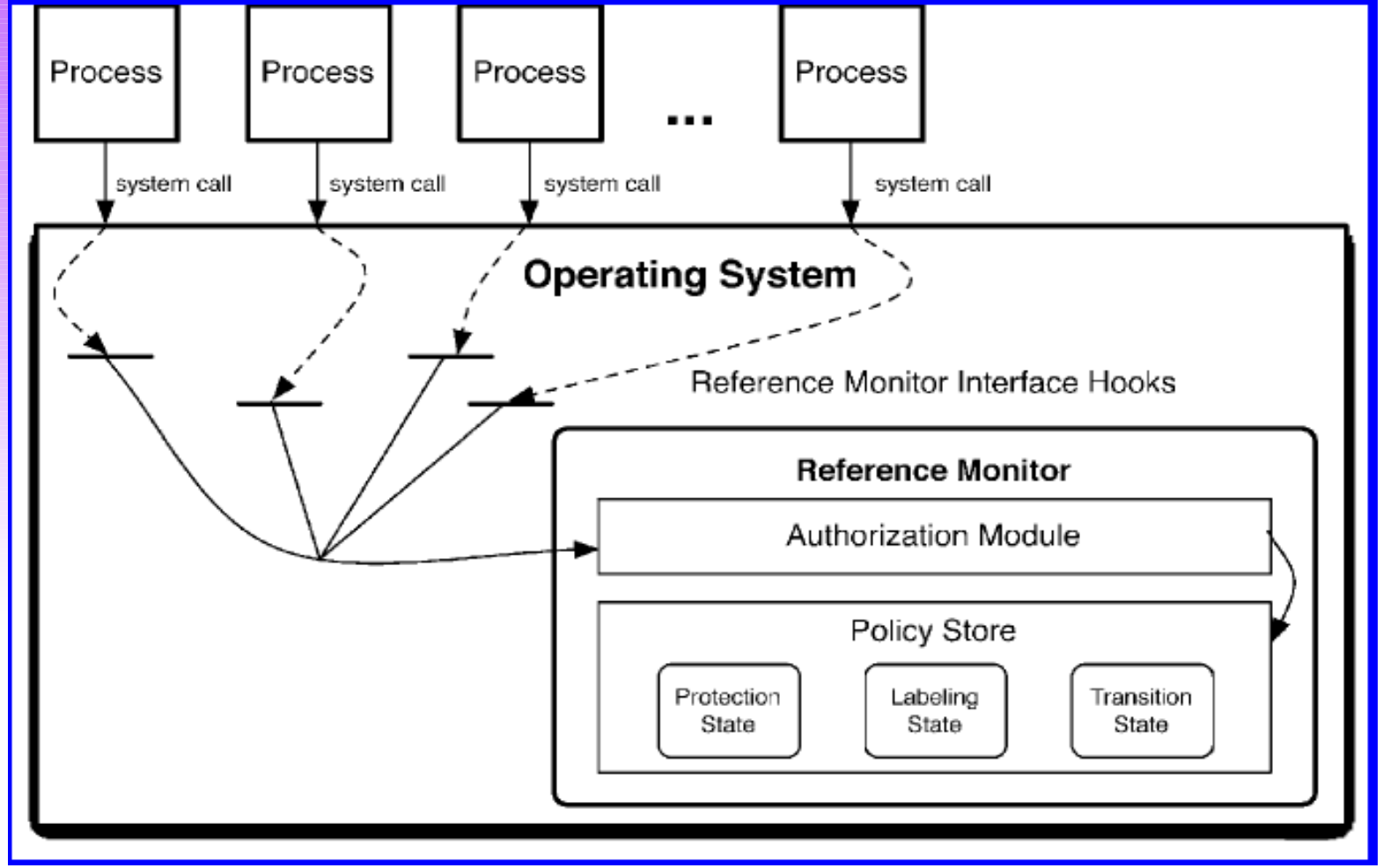
A Mandatory Protection System: The protection State is defined in terms of labels and is immutable. The immutable labeling state and transition state enable the definition and management of the labels for system subjects and objects.

A reference monitor is the classical access enforcement mechanism. It takes a request as input, and returns a binary response Indicating whether the request is authorized by the reference monitor's access control policy. We identify three distinct components of a reference monitor:

(1) its interface -The interface defines where the authorization module needs to be invoked to perform an authorization query to the protection state, a labeling query to the labeling state, or a transition query to the transition state.

(2) its authorization module-determines the exact queries that are to be made to the policy store. ;

(3) Its policy store-The policy store responds to authorization, labeling, and transition queries based on the protection system that it maintains.



A Reference monitor is a component that authorizes access requests at the reference monitor interface defined by individual hooks that invoke the reference monitor's authorization module to submit an authorization query to the policy store. The policy store answers authorization queries, labeling queries, and label transition queries using the corresponding states.

Microkernel vs. Monolithic System

- Monolithic kernel
  - one big kernel provides all services, e.g., file system, network services, device drivers, etc.
  - e.g., Linux 2.6 kernel has about 6 millions of code
- Microkernel
  - implement many services as processes
- Difference between kernel mode and processes running as root (or superuser, administrator)

Microkernel vs. Monolithic System

- Most operating systems are monolithic, that is the whole operating system is a single a.out file that runs in “kernel mode”.
- The alternative is a microkernel-based system, in which most of the OS runs as separate processes, mostly outside the kernel. They communicate by message passing.
- Reference: The Tanenbaum-Torvalds Debate available at http://www.oreilly.com/catalog/opensources/book/appa.htm

Security Evaluation

Users of secure systems need some kind of assurance that the products they use provide adequate security. They could:

1. **Rely on the word of the manufacturer/service provider?**
2. **Test the system themselves.**
3. **Rely on an impartial assessment by an independent body (evaluation).**

Evaluation Criteria

- The Trusted Computer Security Evaluation Criteria (TCSEC, Orange Book) were the first evaluation criteria to gain wide acceptance.
- A number of other criteria have since been developed to improve on the Orange Book and to unify different criteria which have arisen:
  - Information Technology Security Evaluation (ITSEC)
  - Canadian Trusted Computer Product Evaluation Criteria
  - Federal Criteria
  - Common Criteria

Target of the Evaluation

- **Evaluation criteria refer to either**
  - **Products**
    - We have to find an accepted set of generic requirements (Security classes of Orange Book and the protection profile of Federal and Common Criteria)
  - **Systems**
    - Requirements capture and analysis becomes part of each individual evaluation (ITSEC).
- **Where is the borderline between a security evalution and the task of security consultant?**

Method and Structure of the Evaluation

- Security Evaluation can be product oriented or process oriented.
- The concepts of Repeatability and Reproducibility
- Three aspects are addressed in an evaluation criteria:
  - Functionality: The security features of a system.
  - Effectiveness: Are the mechanisms used appropriate for the given security requirement?
  - Assurance: The thoroughness of the evaluation.

Orange Book

- Although the efforts were concentrated in the "national security in USA", the document also provides:
  - A yardstick for users to assess the degree of trust that can be placed in a computer security system.
  - Guidance for manufacturers of computer security systems
  - A basis for specifying security requirements when acquiring a computer security system.

Classification of OS Security

- D – Minimal Protection
- C1 – Discretionary Security Protection: intended for an environment where co-operating users process the data at the same level of integrity.
- C2 – Controlled Access Protection: make users individually accountable for their actions. Most reasonable class for commercial applications.
- B1 – Labelled Security Protection: intended to handle classified data and enforce mandatory policies. Include thorough security testing.
- B2 – Structured Protection: Increases assurance by adding requirements to the design. e.g. Covert channel analysis.
- B3 – Security Domains: Highly resistant to penetration.
- A1 – Verified Design: Adds formal model for security policy.

Common Criteria

- Starting in late 90's, the Common Criteria merges ideas from its various predecessors. The ultimate goal is an internationally set of criteria in the form of an ISO standard.
- It separates functional and security requirements from the intensity of required testing.
- Evaluation assurance levels from 1 to 7.
  - EAL1: Tester reads documentation and performs some tests to confirm documented functionality.
  - EAL7: Developer provides formal functional specification and high-level design, security functions must be simple enough for formal analysis.

Red Book

- Red Book attempts to address network security with the concepts and terminology of Orange Book.

Q: Are computer networks simply a specific example for computer systems?

A: We must distinguish two different types of networks:

1. Networks of independent components.
2. Centralised networks (only this one is considered in Red Book).

Trusted Computing (TC)

- The Trusted Computing Group is an alliance of Microsoft, Intel, IBM, HP and AMD which promotes a standard for a more secure PC.
  - Their definition of security is controversial, though.
- TC provides a computing platform on which you can not tamper with the application software, and where these applications can communicate securely with their authors and with each other.
  - The original motivation was digital rights management.

There is an excellent FAQ about "Trusted Computing" available at http://www.cl.cam.ac.uk/~rja14/tcpa-faq.html

Security Features

- Identification and Authentication
- Object Reuse Protection
  - Prevent leaks via reallocation
  - Clean before re-use

Complete Mediation

- Mediate all means of access
- File access plus direct memory access if possible
- Mediate on each access, not generally done for files
  -

More Security Features

- **Trusted Path**
  - **Give end user means to determine they are really talking with OS**
  - **Secure Attention Key (SAK): key sequence that cannot be intercepted by non-OS**
    - Ctl-Alt-Del in Windows
    - Rootkit…
  - **Or security relevant changes only made during system boot**

More Security Features

- **Audit**
  - Must be able to review and recreate security relevant changes
  - Must protect log
- **Log growth**
  - Originally assumed security officer would review directly
  - Can by used for backing evidence
- **Really want to detect anomalies**
  - Intrusion detection

Operating Systems, Database and Program Security

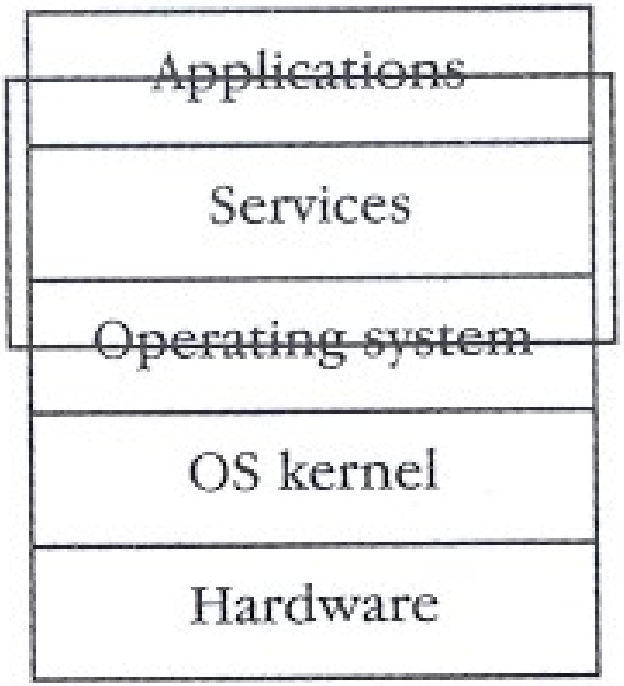
4.2 Database Security

- Security Requirements of Databases
- Reliability and Integrity
- Protection of Sensitive Data
- Inference Problem: Direct and Indirect Attacks
- Disaster Recovery

Database Security

- **Database system security is more than securing the database**
  - **Secure database**
  - **Secure DBMS**
  - **Secure applications**
  - **Secure operating system in relation to database system**
  - **Secure web server in relation to database system**
  - **Secure network environment in relation to database system**

Layered Database Security



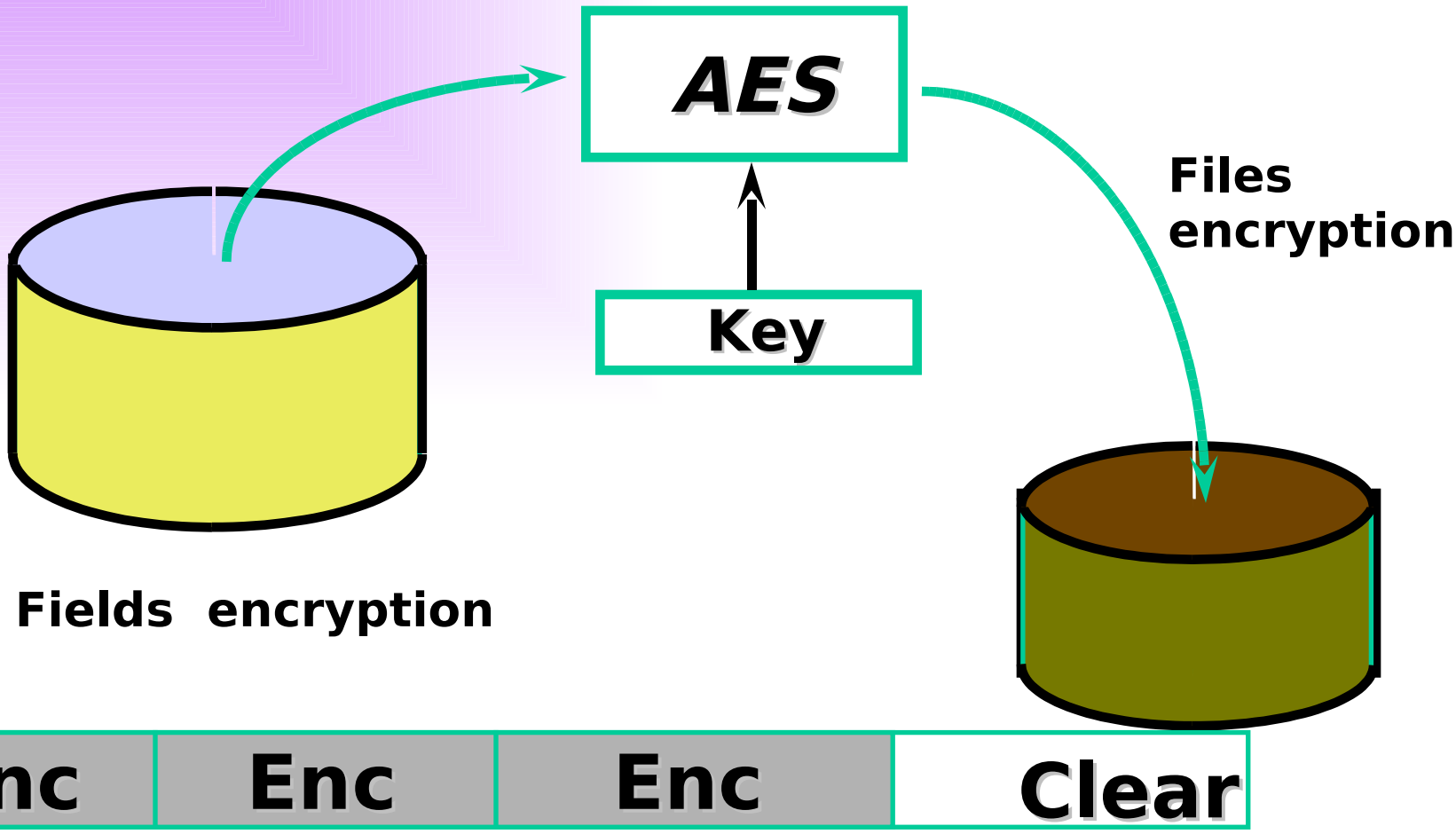
Location of database security

ucsc
kasun@cmb.ac.lk

Securing Database

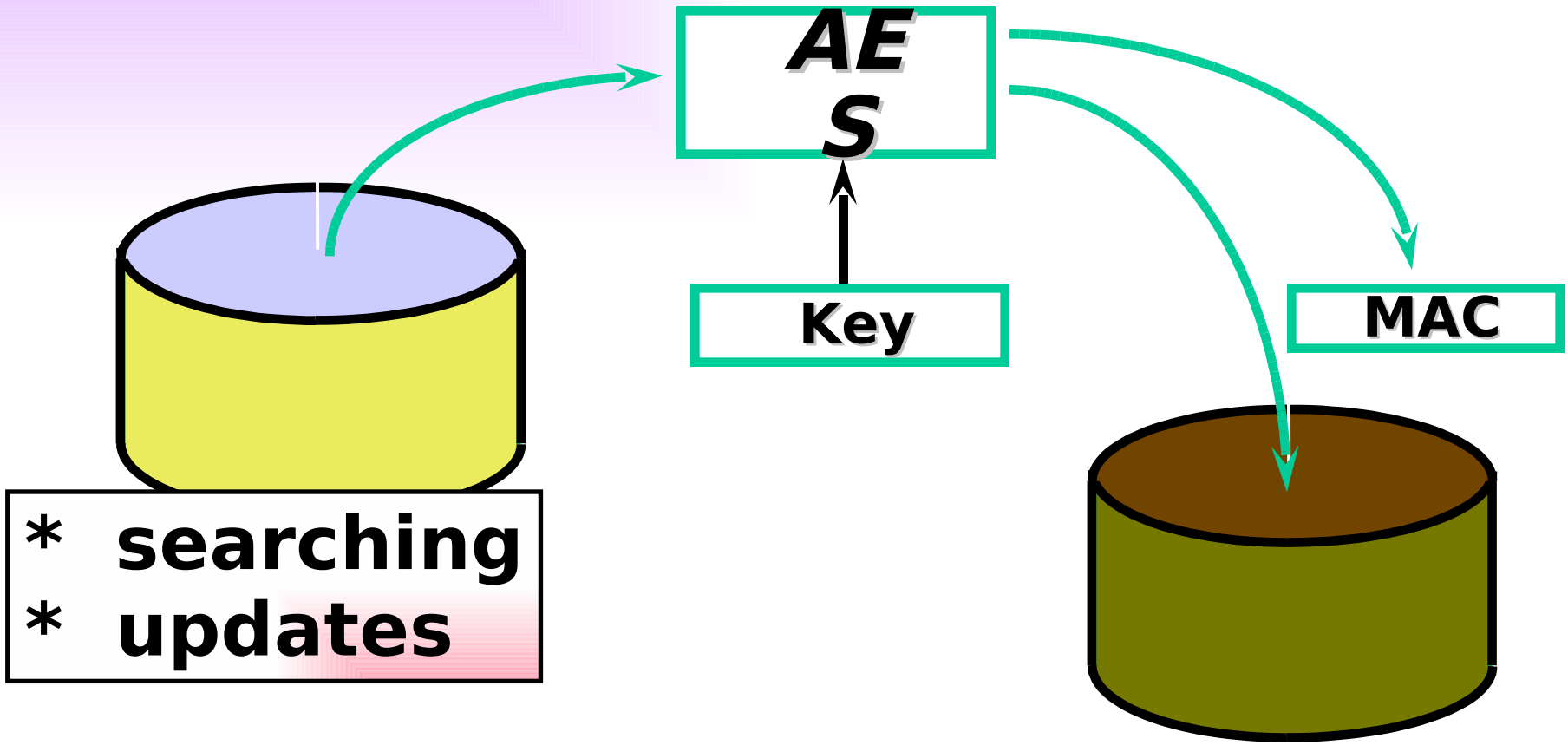
- **Users, Passwords**
  - Default users/passwords
    - sys, system accounts – privileged, change default passwords
    - scott account – well-known account and password, change it
    - **general password policies (length, domain, changing, protection)**
- **Privileges, Roles, Grant/Revoke**
  - Privileges
    - System - actions
    - Objects – data
  - Roles (pre-defined and user-defined role)
    - Collections of system privileges (example: DBA role)
  - Grant / Revoke
    - Giving (removing) privileges or roles to (from) users

Data Confidentiality



kasun@cmb.ac.lk

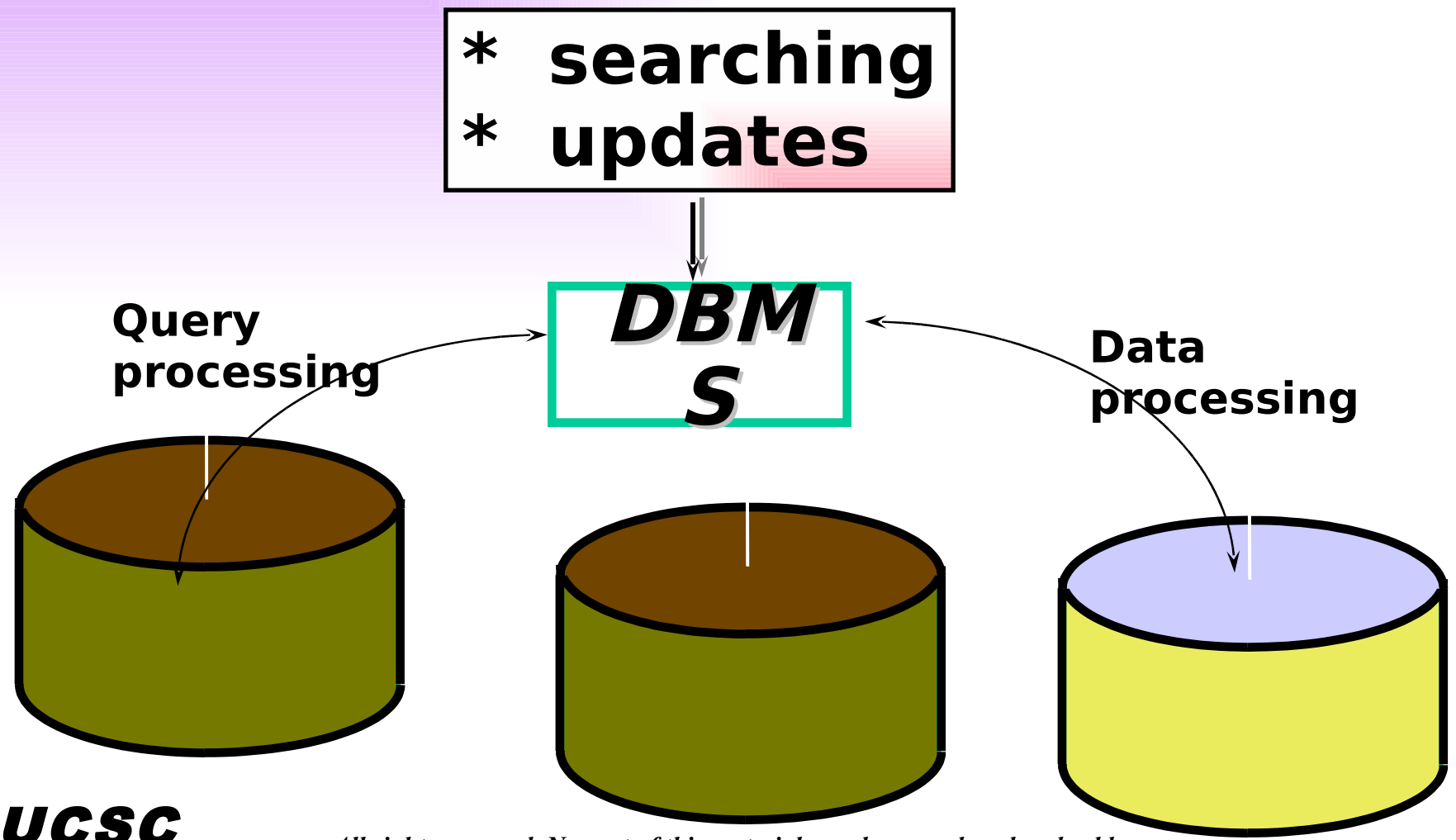
Confidentiality and Integrity Combined



ucsc
kasun@cmb.ac.lk

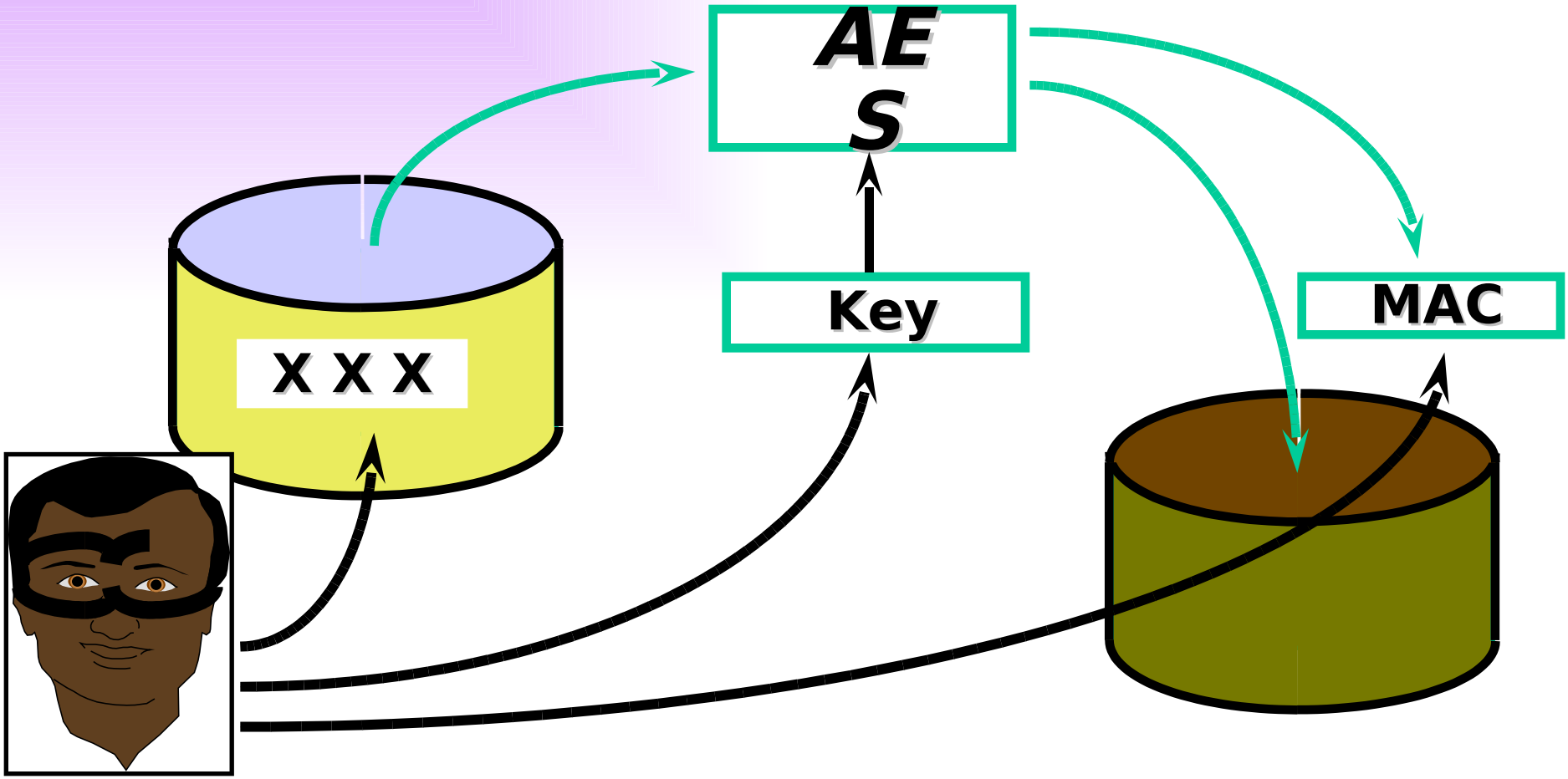
Processing of Encrypted Files in Database Management Systems (DBMS)

* searching
* updates



ucsc
kasun@cmb.ac.lk

Protection of Key and MAC

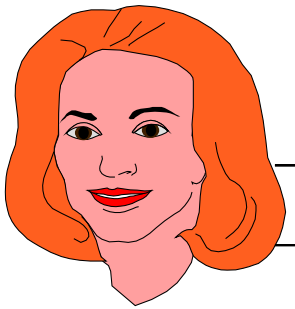


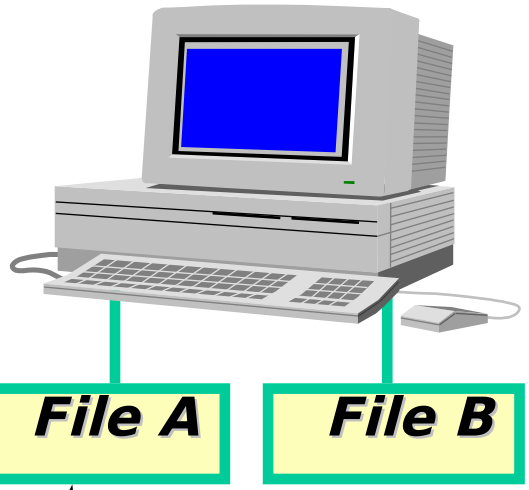
UCSC
kasun@cmb.ac.lk

Access Control – Principles

Access control

Who can do …
what …
with which **resource** ?





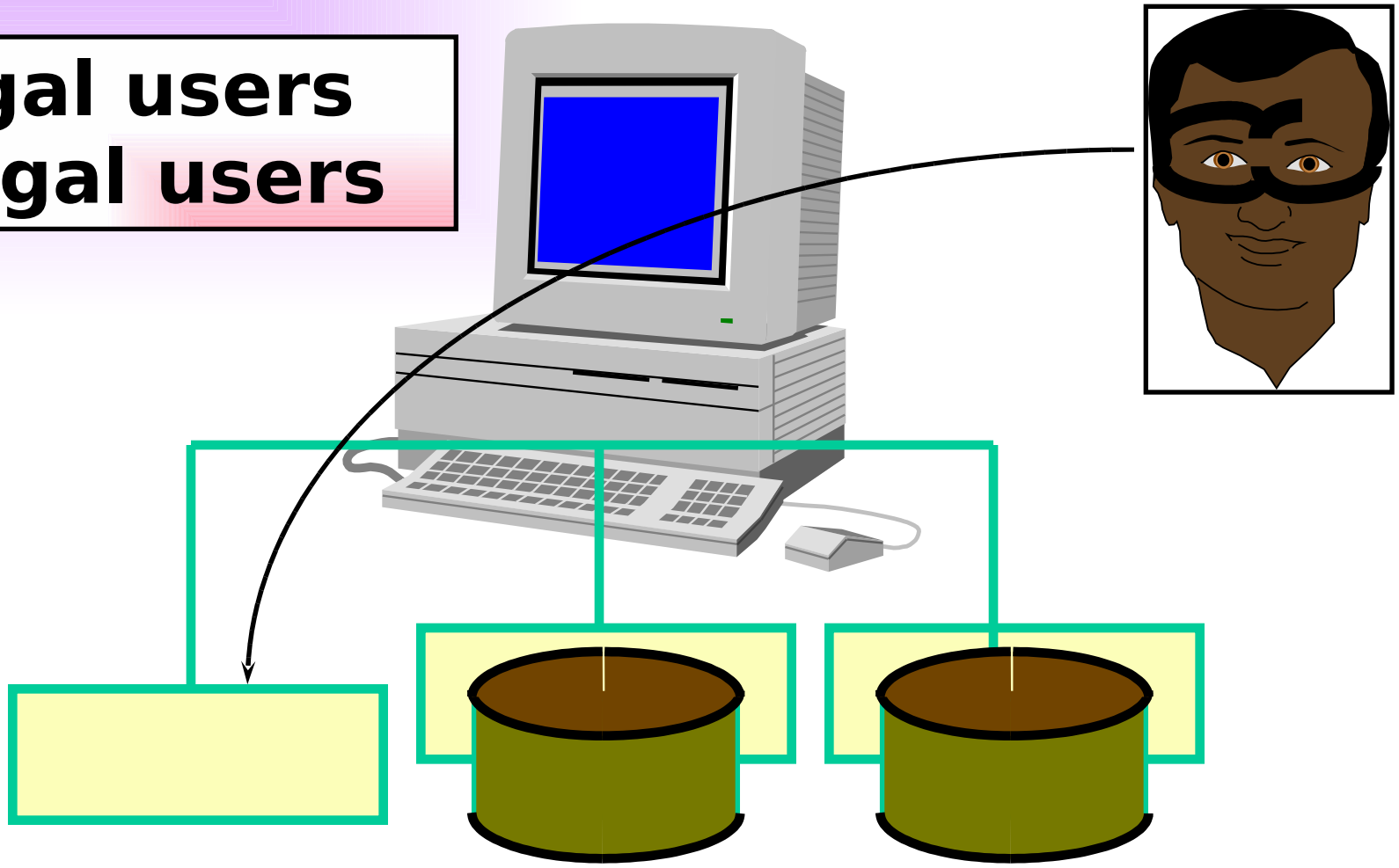
Read

Copy

kasun@cmb.ac.lk

Access Control – Files and Directories

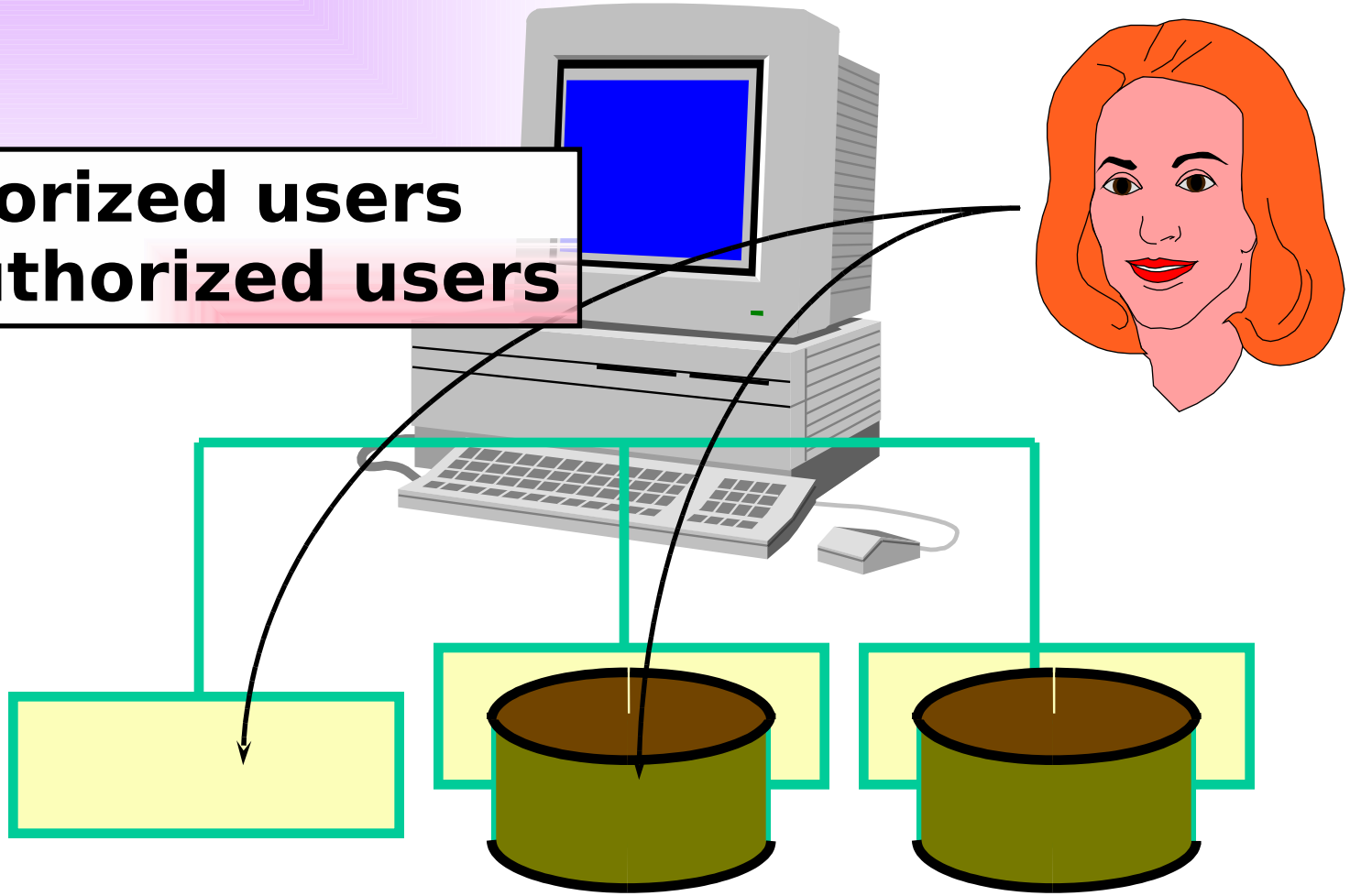
* Legal users
* Illegal users



kasun@cmb.ac.lk

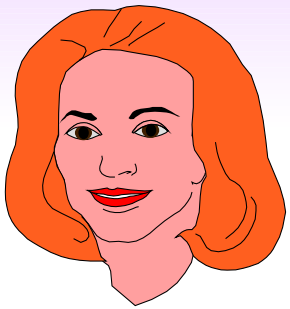
Access Control – Files and Directories

* Authorized users
* Unauthorized users

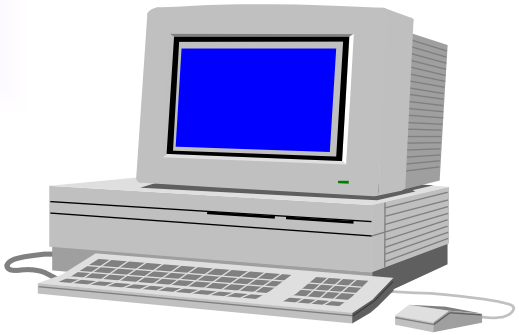


kasun@cmb.ac.lk

Fundamental Model of Access Control



Read



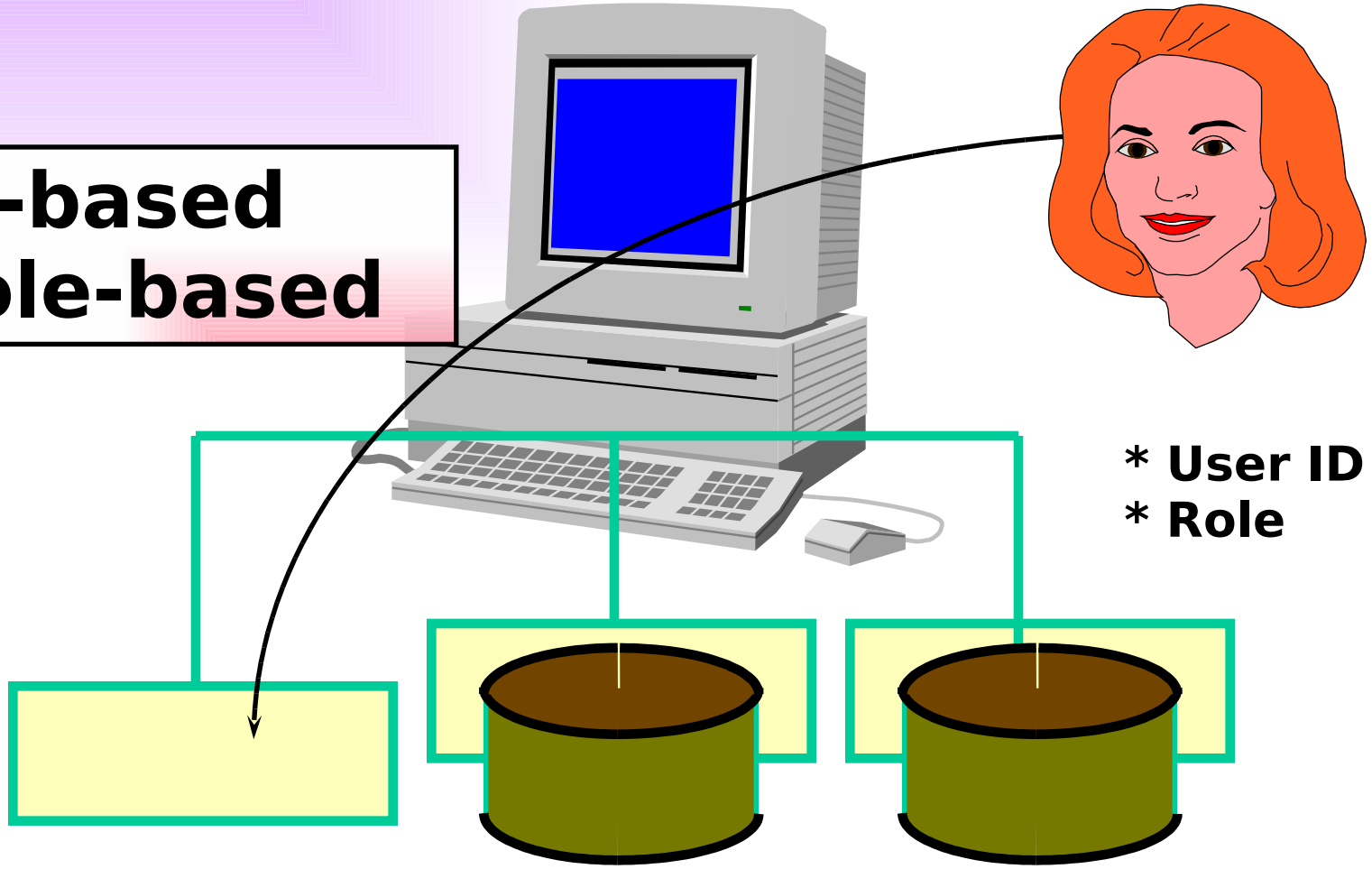
File A

Subjects → Access requests → Reference monitor → Objects

kasun@cmb.ac.lk

Authorization Schemes

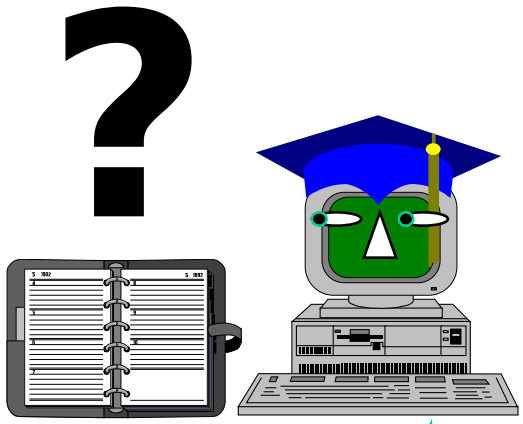
* ID-based
* Role-based



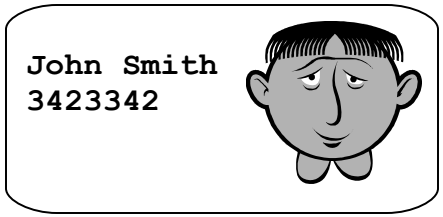
* User ID
* Role

kasun@cmb.ac.lk

Identity – based Access Control



Identity



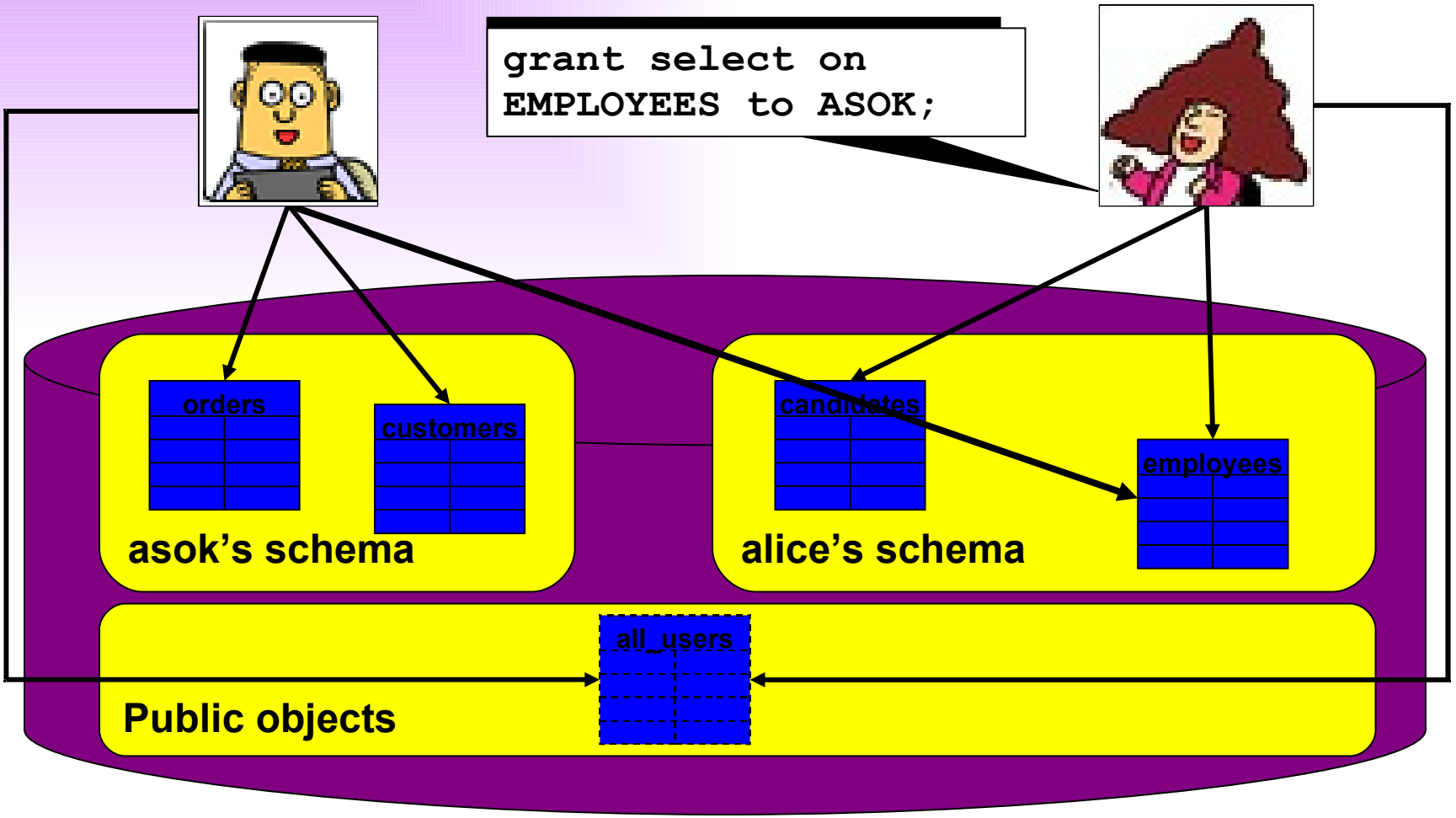
kasun@cmb.ac.lk

Identity – based Access Control

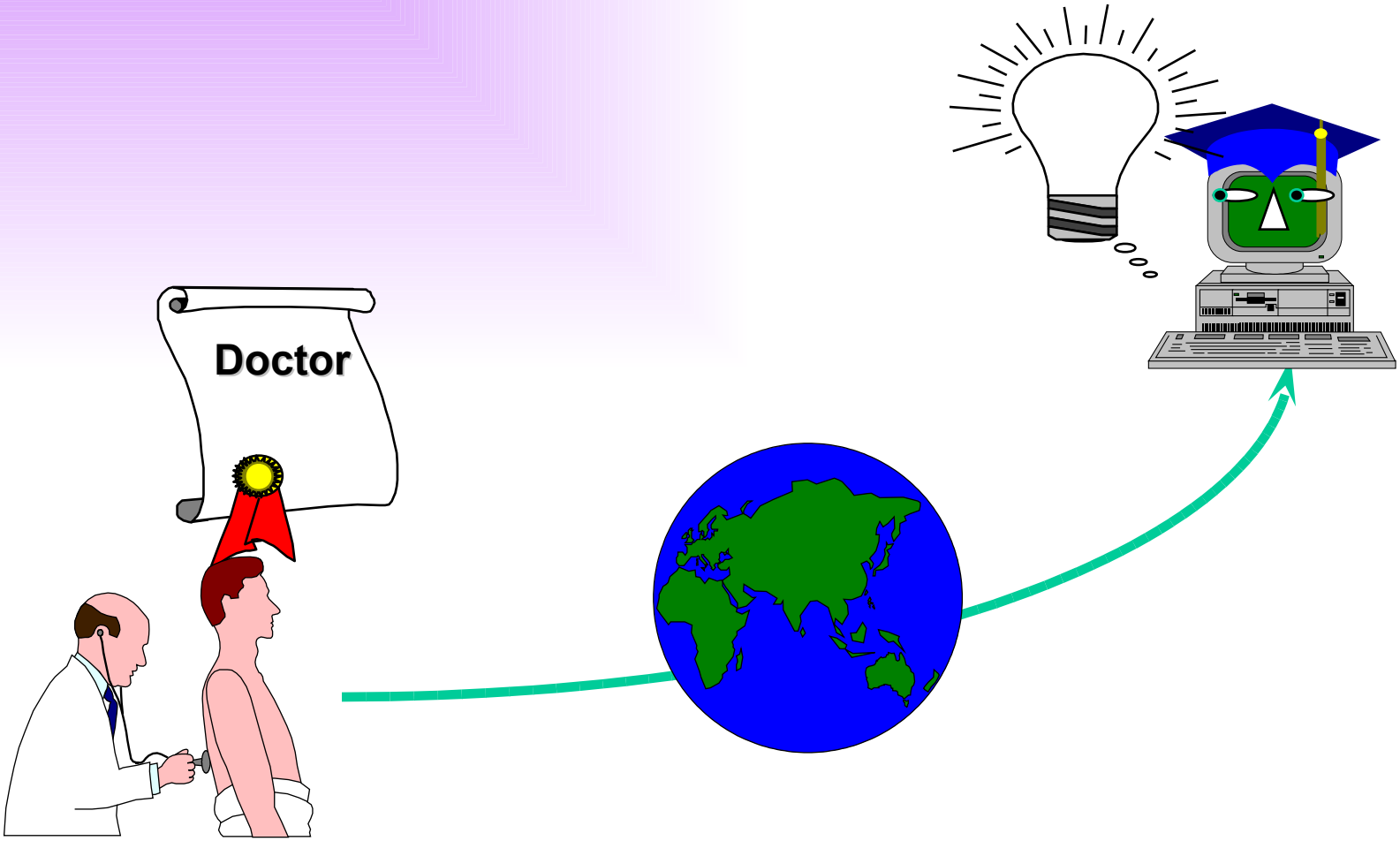
	O1	O2	O3	O4	O5	O6
S1		*r, w*				
S2			*x, d*			
S3						
S4						
S5					*l, c*	
S6						

ucsc
kasun@cmb.ac.lk

Identity – based Access Control

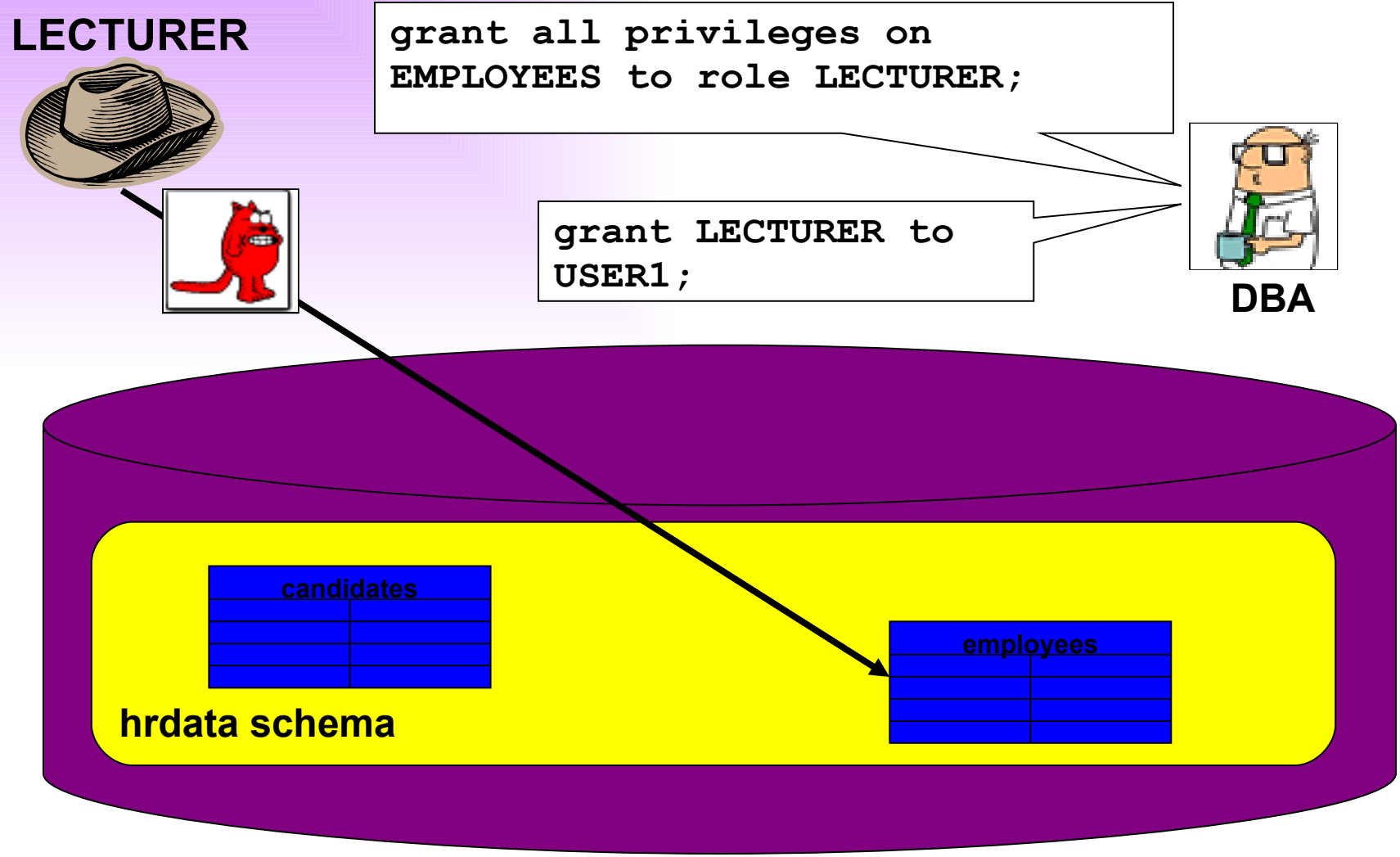


Role – based Access Control

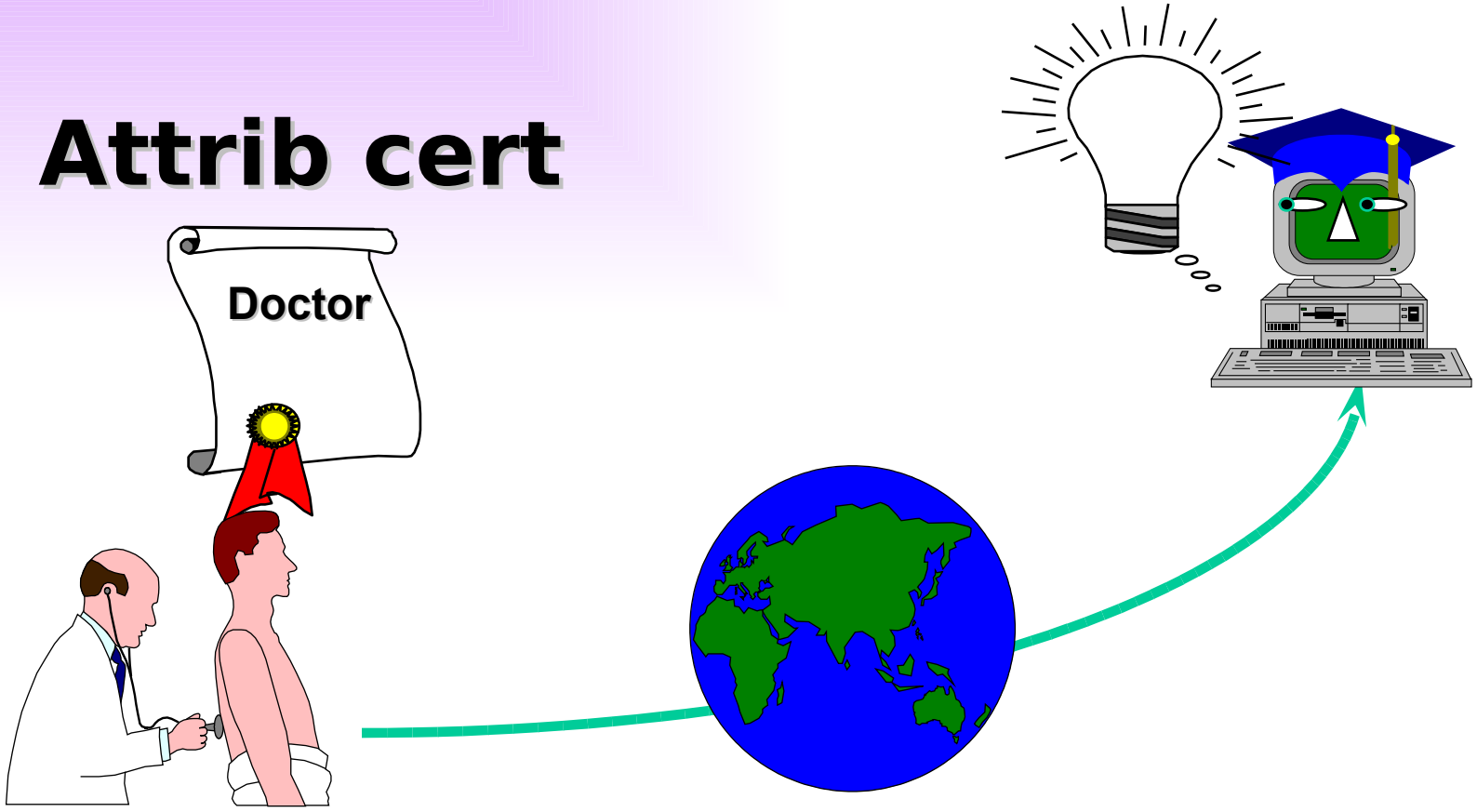


kasun@cmb.ac.lk

Role – based Access Control



Role – based Dynamic Access Control



kasun@cmb.ac.lk

Attribute Certificates (AC)

- **PKCs may include life/long lasting attributes.**
  - **if the attribute doesn't expire before the related PKC, then the attribute may be included in the PKC.**

- **ACs should be used for short lasting attributes:**
  - **if the attribute expires before the related PKC, then the attribute should placed in an AC.**

Difference between PKC and AC

PKC is passport and AC is visa

Public Key Certificate (PKC)	Attribute Certificate (AC)
Version	Version
Serial Number	Serial Number
Signature ID	Signature ID
Subject	Holder
Issuer	Issuer
Validity Period	Validity Period
Subject Public Key Info	Attributes
Extensions	Extensions
Signature	Signature

Public Key
PKC binds a subject and a public key

No Public Key
AC binds a holder and attributes

UCSC
kasun@cmb.ac.lk

Attribute Authority (AA)

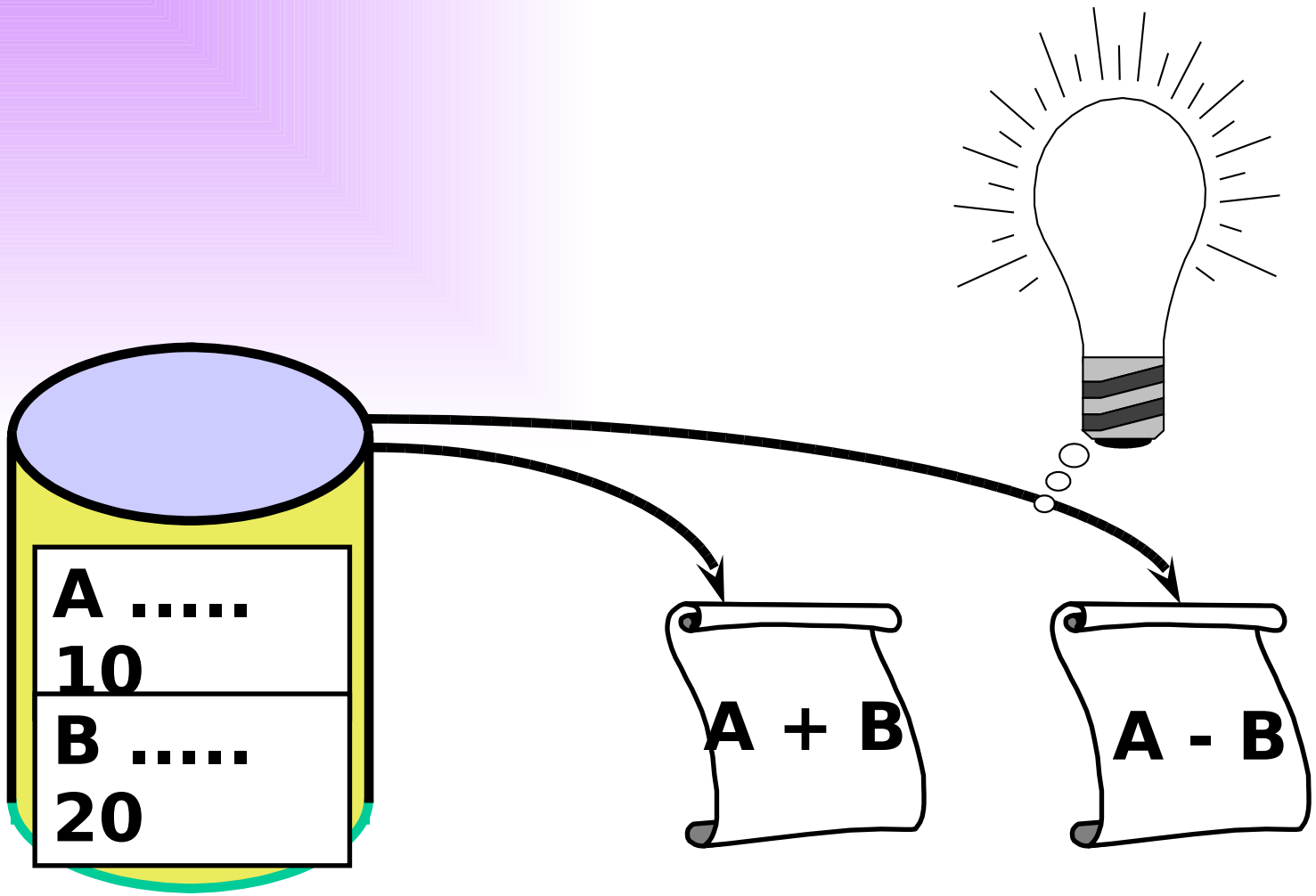
- **AA Attribute Authority (also called AC Issuer)**
  - **An authority trusted by one or more users to create and sign attribute certificate. It is important to note that the AA is responsible for the attribute certificates during their whole lifetime, not just for issuing them.**
- **AA can be any entity in the network having objects in its control.**

SQL Injection

Example: SQL Injection Attack through Web Application

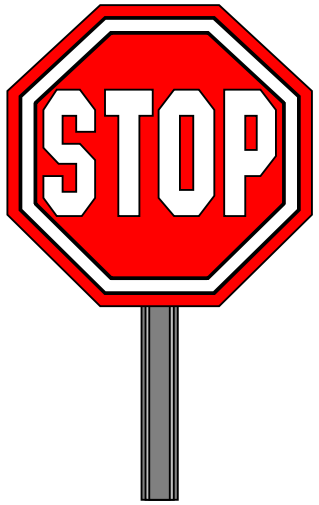
- **Application tracks own usernames and passwords in database**
- **Client accepts username and password, passes as parameters**
- **Application Java code contains SQL statement:**
  - String query = "SELECT * FROM users_table " +
  - " WHERE username = " + " ‘ " + username + " ‘ " +
  - " AND password = " + " ‘ " + password + " ‘ " ;
- **Expecting one row to be returned if success, no rows if failure**
- **Attacker enters any username, password of: Aa ‘ OR ‘ ‘ = ‘**
- **Query becomes: SELECT * FROM users_table WHERE username = ‘anyname‘ AND password = ‘Aa‘ OR ‘ ‘ = ‘ ‘;**
  // F or T => T
- **All user rows returned to application**
- **If application checking for 0 vs. more than 0 rows, attacker is in**

Data Inference

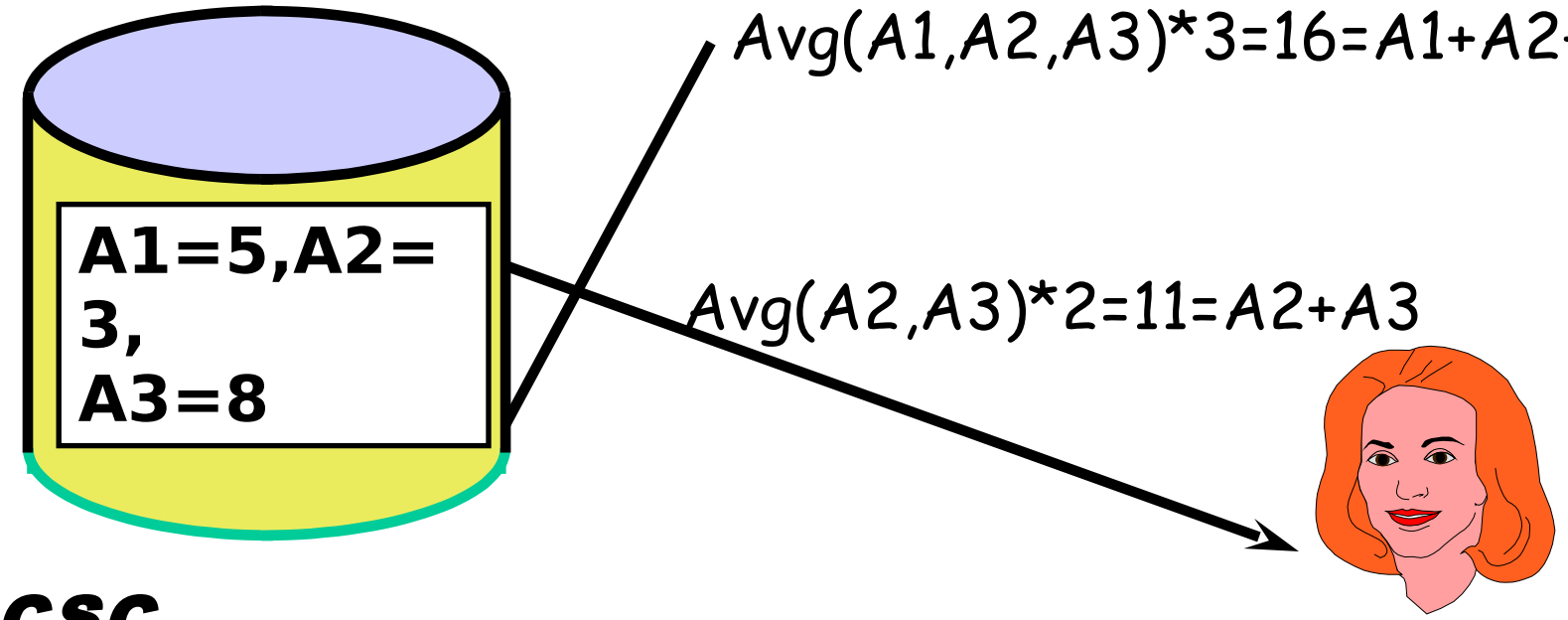


ucsc
kasun@cmb.ac.lk

Data Dependency

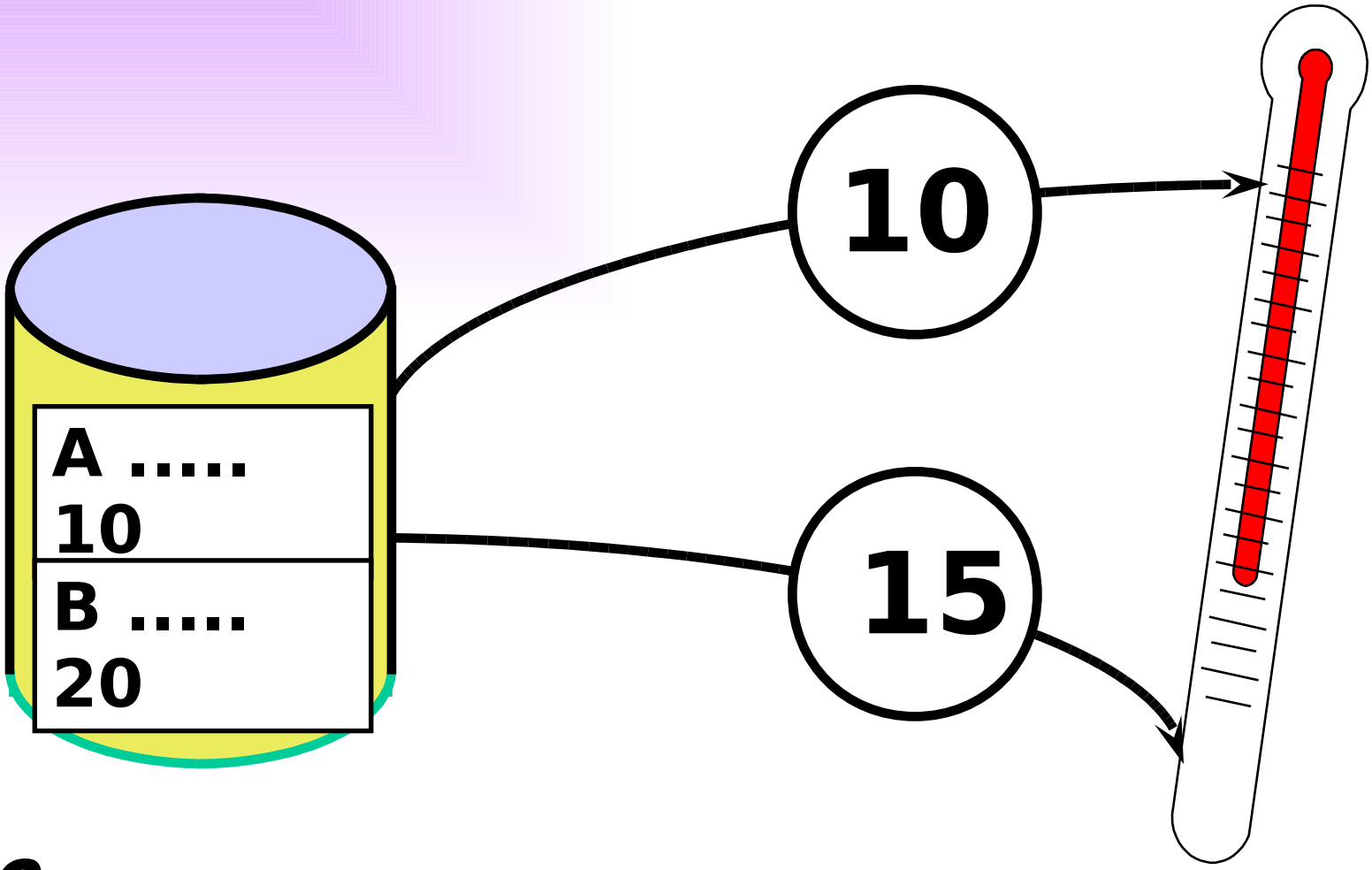


For example, suppose that an attacker wishes to find out the salary of A1. He can do this by asking for the average salaries of A1, A2 and A3 and of A2, A3.



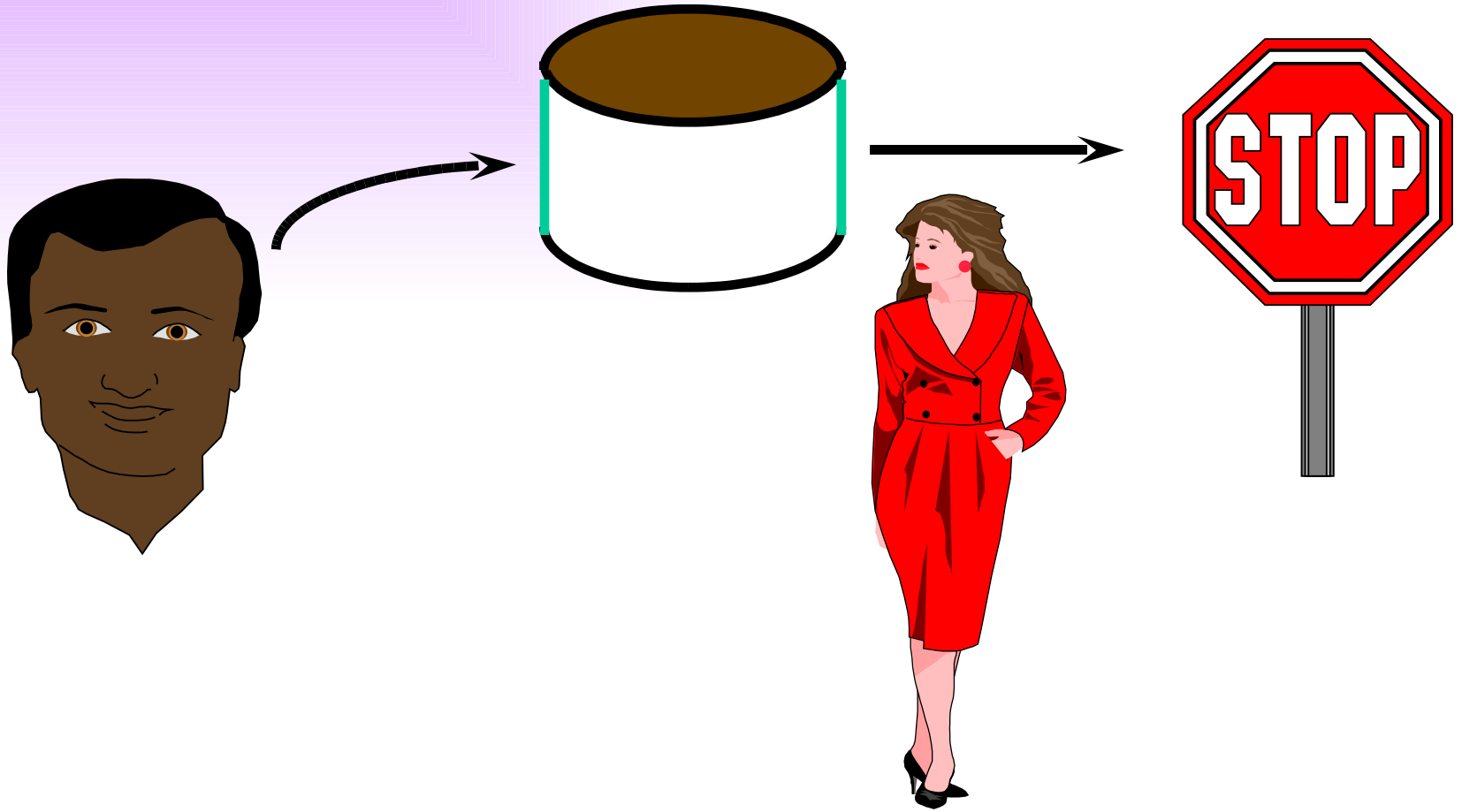
kasun@cmb.ac.lk

Data Classification



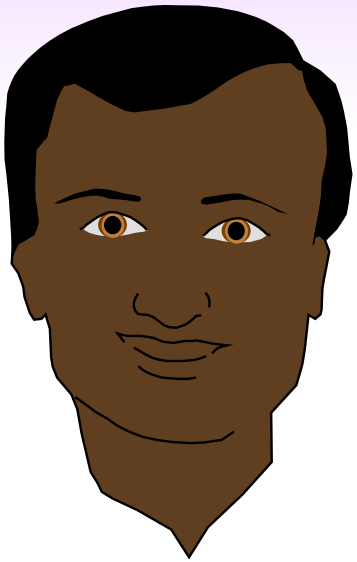
kasun@cmb.ac.lk

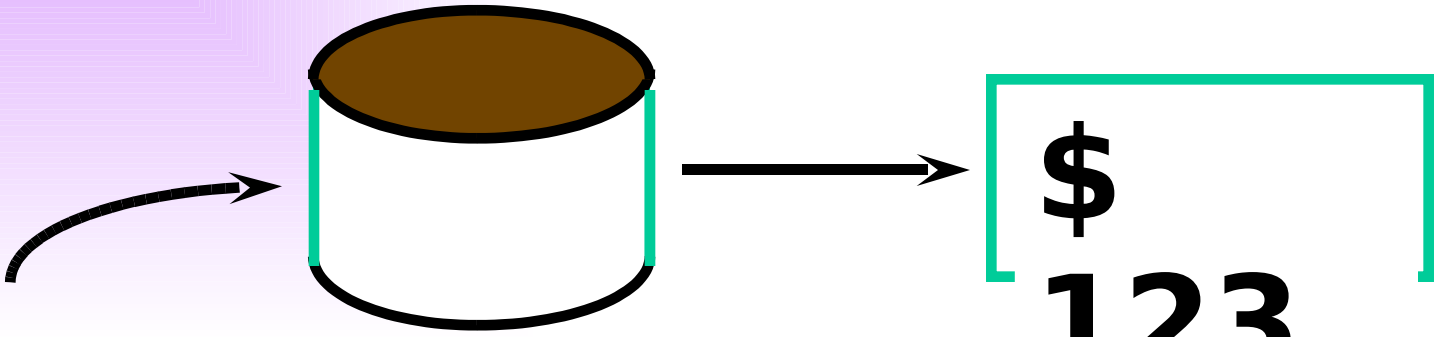
Individual Salary – High Classification



UCSC
kasun@cmb.ac.lk

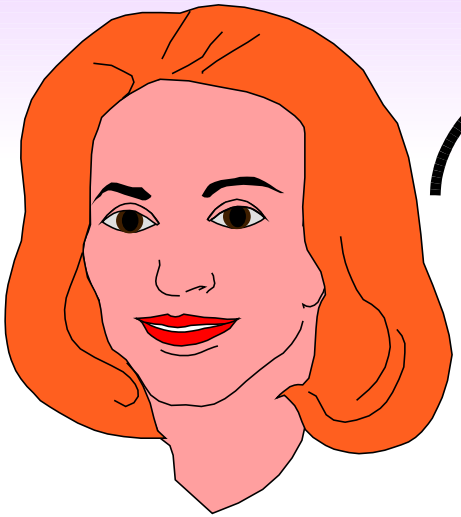
Average Salary – Low Classification

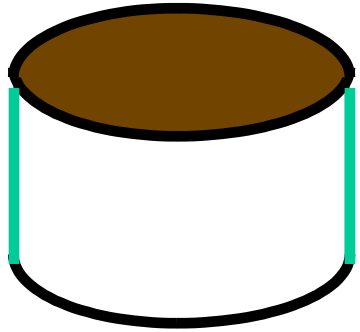




UCSC
kasun@cmb.ac.lk

Database Queries

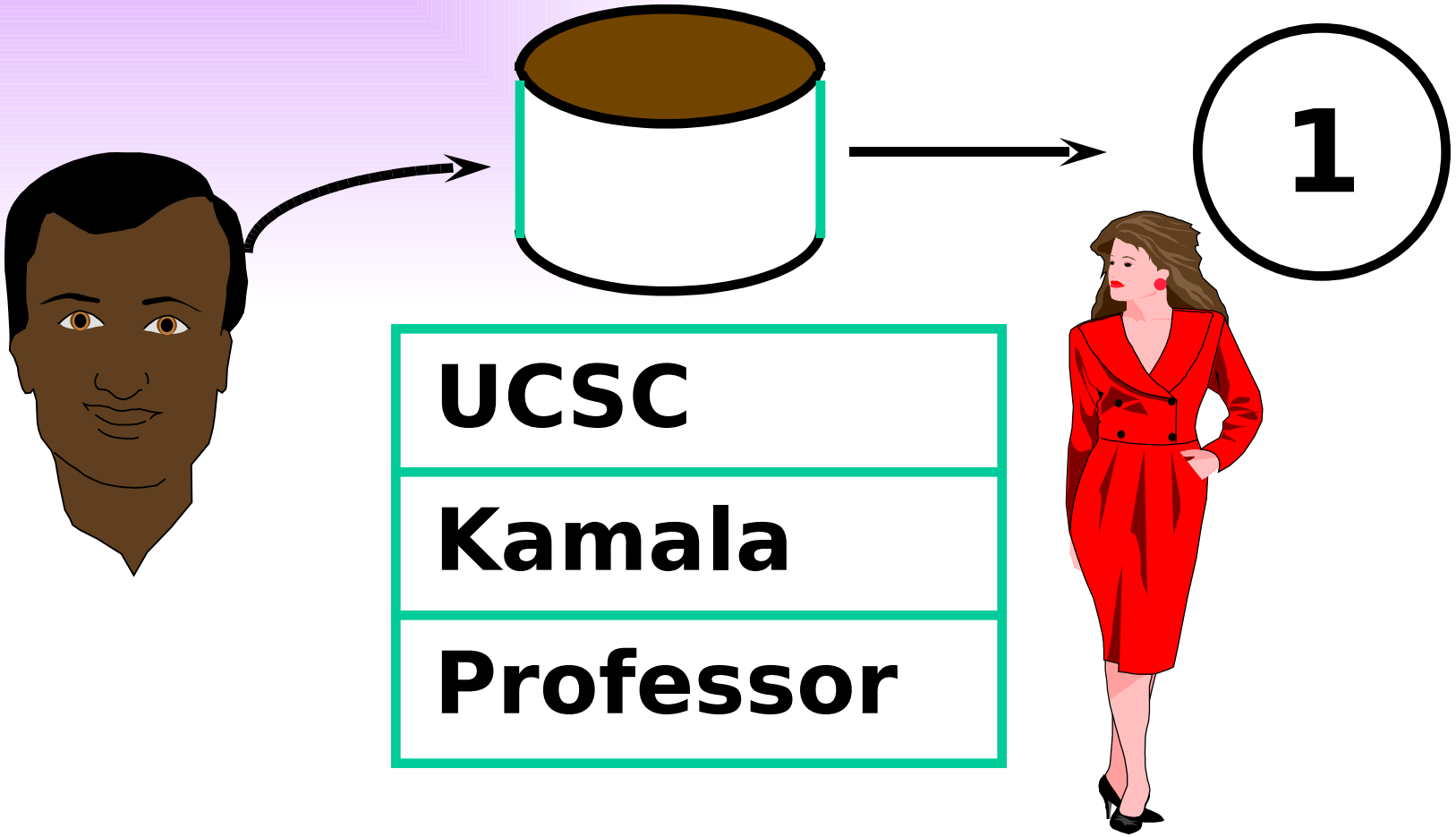




123	John	$
4		123

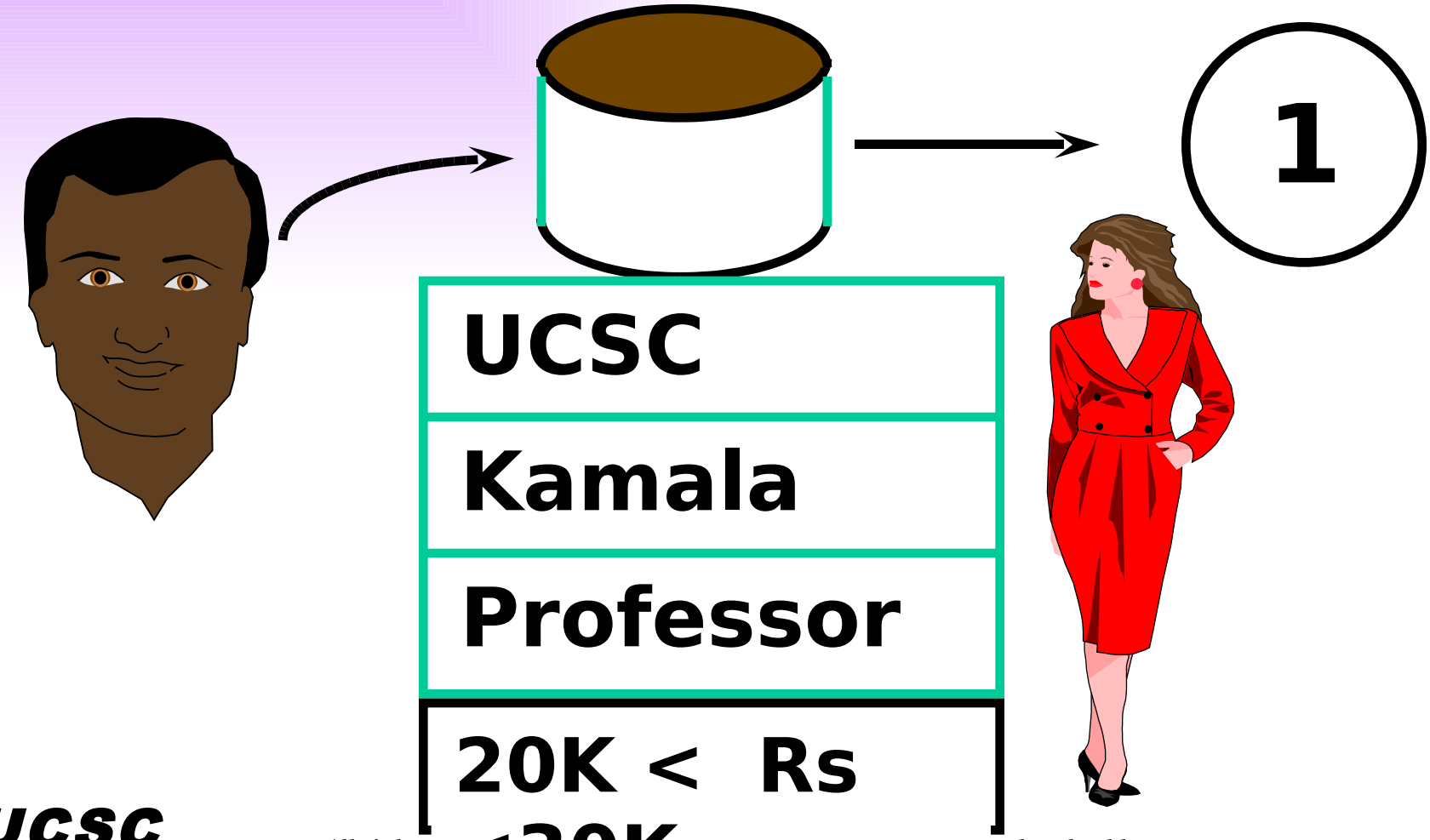
ucsc
kasun@cmb.ac.lk

Small Query Window



UCSC
kasun@cmb.ac.lk

Additions to Small Query Window



ucsc
kasun@cmb.ac.lk

To avoid disasters:

- Back-up files
- Store backups securely:
  - off-site, locked
- Twin sites for computer equipment
- Insure with a disaster recovery firm - loan of equipped premises
- Disaster Recovery plan essential
  - Prioritise programs to be up & running first
  - Notification to staff of procedure
  - Test the planning (like fire practice)!

Preventing your own disasters

- no eating or drinking near the computer
- no extremes of environment (heat, smoke)
- clean & lock your computer!
- backup hard disk files and floppy disks
- store backups safely
- write protect disks

kasun@cmb.ac.lk

Why backup?

- disks don't last forever
- liquids + disks = disaster
- magnetic fields damage disks
- files are damaged if a power failure happens during saving or loading
- accidental deletion by YOU!
- network crashes

kasun@cmb.ac.lk

Disaster Recovery Plans

- Single system or device failures - Includes a network device, disk, motherboard, network interface card, or component failure.
- Data center events - Provides procedures for a major event within a data center.
- Site events - Identifies the critical capabilities that need to be restored.
- Testing the DRP - Identifies key employees and performs walkthroughs of the plan periodically.

Backup Policy

- **Frequency of backups - Identifies how often backups actually occur.**
- **Storage of backups - Defines how to store backups in a secure location. It also states the mechanism for requesting and restoring backups.**
- **Information to be backed up - Identifies which data needs to be backed up more frequently.**

Operating Systems, Database and Program Security

4.3 Program Security

- Kinds of Malicious Code
- How Viruses Attach and Gain Control
- Homes for Viruses
- Virus Signatures
- Preventing Virus Infection
- Trapdoors
- Convert Channels
- Control Against Program Threats
- Java mobile codes

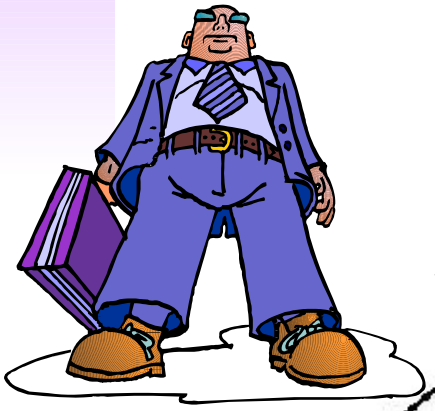
Malicious Software

- Malicious code often masquerades as good software or attaches itself to good software
- Some malicious programs need host programs
  - Trojan horses, logic bombs, viruses
- Others can exist and propagate independently
  - Worms, automated viruses
- There are many infection vectors propagation mechanisms

Image courtesy of: Tech Tips.com

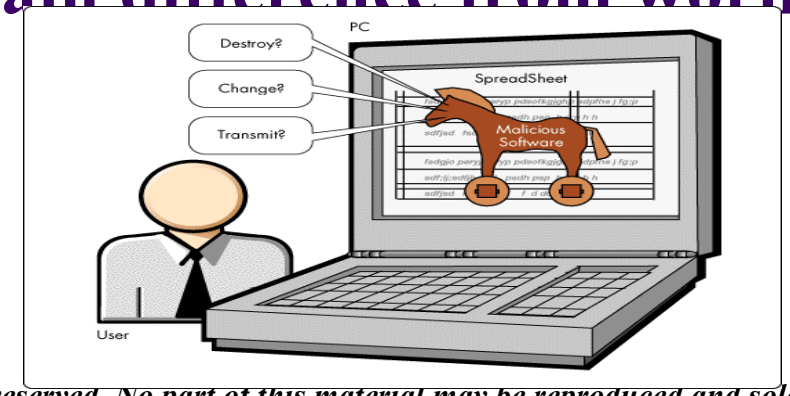
UCSC
kasun@cmb.ac.lk

Malicious Software



Trojan Horses

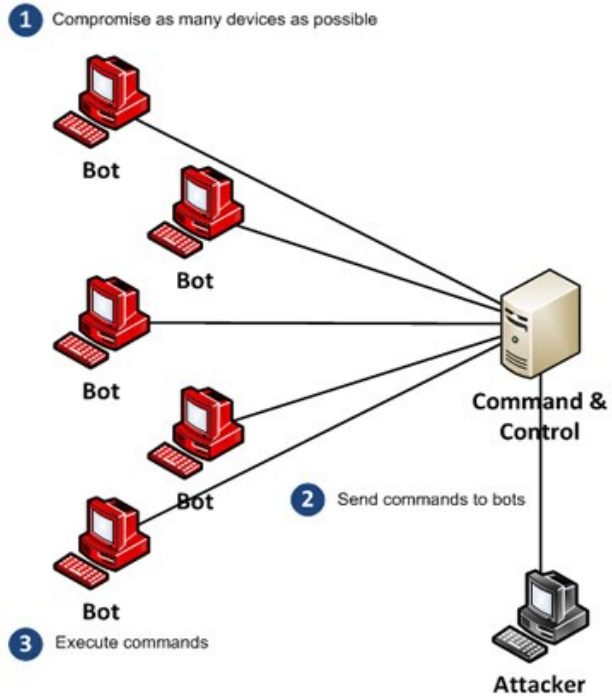
- A trojan horse is malicious code hidden in an apparently useful host program
- When the host program is executed, trojan does something harmful or unwanted
- Trojans do not replicate
  - This is the main difference from worms and viruses



ucsc
kasun@cmb.ac.lk

Zombie

- program which secretly takes over another networked computer
- then uses it to indirectly launch attacks
- often used to launch distributed denial of service (DDoS) attacks
- exploits known flaws in network systems



What is a computer virus?

- A program that reproduces itself
- Affects software, not the actual machine
- Many different types:
  - file:
  - macro:
  - boot or partition, etc.
- May lie dormant
- Can strike at any time

kasun@cmb.ac.lk

Viruses

- **Virus propagates by infecting other programs**
  - **Automatically creates copies of itself, but to propagate, a human has to run an infected program**
    - Self-propagating malicious programs are usually called worms
- **Viruses employ many propagation methods**
  - **Insert a copy into every executable (.COM, .EXE)**
  - **Insert a copy into boot sectors of disks**
    - "Stoned" virus infected PCs booted from infected floppies, stayed in memory and infected every floppy inserted into PC
  - **Infect TSR (terminate-and-stay-resident) routines**
    - By infecting a common OS routine, a virus can always stay in memory and infect all disks, executables, etc.

ucsc
kasun@cmb.ac.lk

What do viruses do?

- Non-destructive viruses
  - Print unexpected message, make a sound
  - Examples: Peace, Red Cross, Bubbleboy
- Destructive viruses
  - Destroy data and files
  - Examples: Michelangelo, Dark Avenger, Joshi, Stealth (makes hard disk inoperable) etc.
- Millions of viruses and hundreds of different effects
- Non-destructive viruses still waste time

kasun@cmb.ac.lk

Protection against viruses

- Few computer systems are totally secure
- Buy from a reputable computer dealers
- Buy "shrink-wrapped" software
- Always suspect any other software
- Make executables read only
- Write protect diskettes
- Careful with internet & email files
- Use scanning software (anti-virus) to check for viruses

If you find a virus

- DON'T PANIC
- Work systematically. DON'T RUSH
- Tell system manager - IT Services
- Clean up
  - Disinfect and Retrieve clean files
  - Destroy disk
  - Ask for help if you are not sure!

Cryptovirus

- A cryptovirus is a virus embedding and using a public-key (http://www.cryptovirology.com/)

First Technique

- Use the private key to encrypt the payload

Second Technique

- Use a symmetric key to encrypt payload
- Use the private key to encrypt the symmetric key

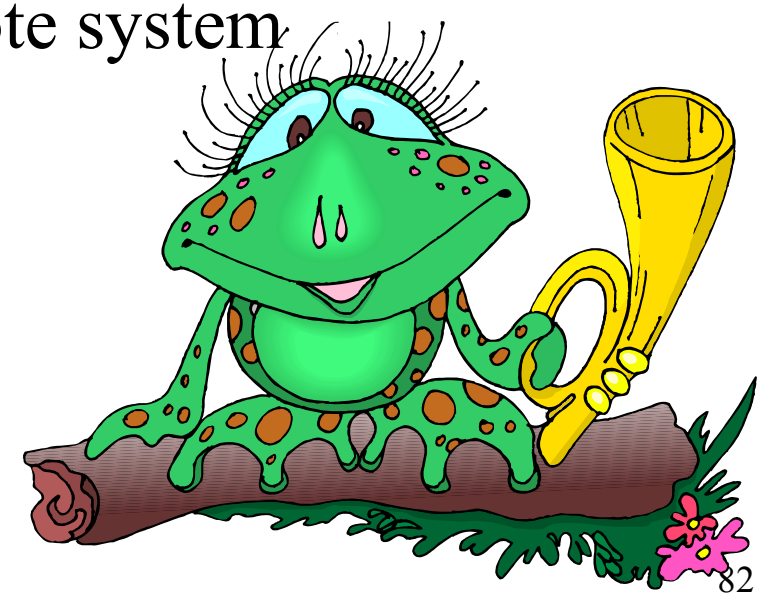
kasun@cmb.ac.lk

Worms

- replicating but not infecting program
- typically spreads over a network
  - cf Morris Internet Worm in 1988, led to creation of CERTs
- using users distributed privileges or by exploiting system vulnerabilities
- widely used by hackers to create **zombie PC's**, subsequently used for further attacks, esp DoS
- major issue is lack of security of permanently connected systems, esp PC's

Virus (and Worm) Phases

1. Dormant
2. Propagation
   * search for other systems to infect
   * establish connection to target remote system
   * replicate self onto remote system
1. Trigger
2. Execute



UCSC
kasun@cmb.ac.lk

Anti-Virus Technologies

- **Simple anti-virus scanners**
  - **Look for signatures (fragments of known viruses)**
  - **Heuristics for recognizing code associated with viruses**
    - For example, polymorphic viruses often use decryption loops
  - **Integrity checking to find modified files**
    - Record file sizes, checksums, MACs (keyed hashes of contents)
- **Generic decryption and emulation scanners**
  - **Goal: detect polymorphic viruses with known body**
  - **Emulate CPU execution for a few hundred instructions, virus will eventually decrypt, can recognize known body**
    - Does not work very well against metamorphic viruses and viruses not located near beginning of infected executable

kasun@cmb.ac.lk

Possible Counter Measures

- Update all softwares like operating system, drivers all softwares that use the internet and update anti virus and anti spyware
- Install inbound and outbound firewall
- Encrypt important data
- Backup the data regularly
- Install third party registry editor, traffic monitoring software
- Disable autorun feature
- Hope antivirus vendors find a cure for it in near future
- Use open source software and operating systems

kasun@cmb.ac.lk

Applets

- An applet is a typically small program embedded in another application, generally a Web browser that provides a JVM.
- An applet's host program provides an *applet context* in which the applet executes.
- An applet is generally launched from an HTML document with an `APPLET` tag that specifies the URL for the applet bytecodes

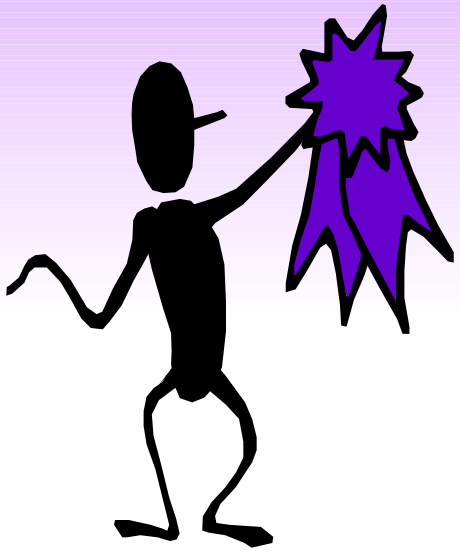
Applet security

- Applets typically execute under a strict security manager that prevents an applet from
  - Accessing the local disk to read, write, delete, or execute files.
  - Loading nonstandard libraries.
  - Opening connections to arbitrary hosts.

Applet security

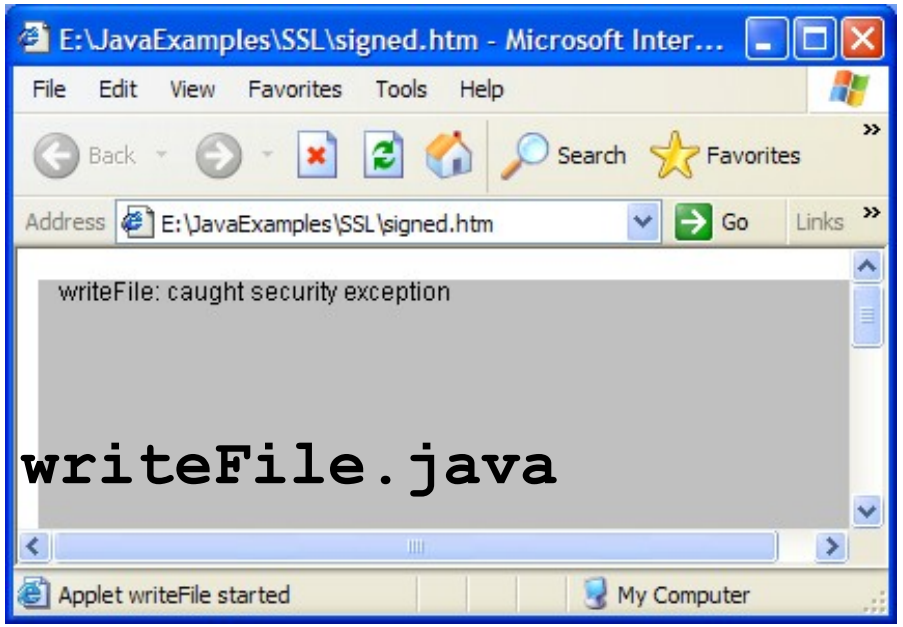
- The tight applet security is sometimes described as *sandbox security* to suggest that an applet must "play" within a confined area from which it must not venture.
- An applet is allowed to open a socket to the server from which is downloaded, thus enabling socket-based communications.

Java Signed Applet:



- Compile the applet
- Create a JAR file
- Generate Keys
- Sign the JAR file
- Export the Public Key Certificate
- Import the Certificate as a Trusted Certificate
- Create the policy file
- Run the applet

Creating a Jar file:




```
E:\JavaExamples>javac writeFile.java
```

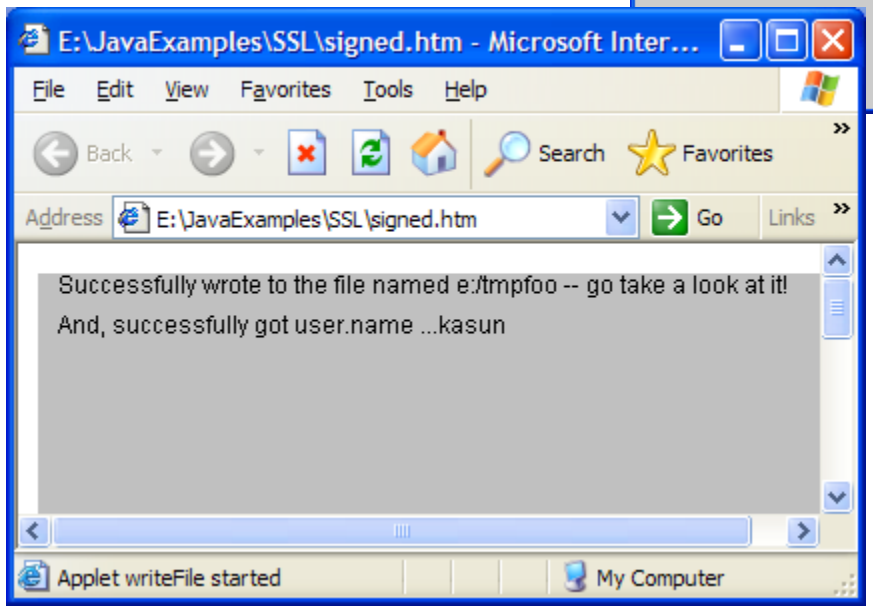
```
E:\JavaExamples>jar -cvf writeFile.jar
writeFile.class

added manifest adding: writeFile.class(in =
1747) (out= 984)(deflated 43%)
```

Signing a Applet:

```
E:\JavaExamples\SSL>keytool -genkey -alias
kasun -keystore writeFile


E:\JavaExamples\SSL>jarsigner -keystore
writeFile writeFile.jar kasun

Enter Passphrase for keystore: writeFile


E:\JavaExamples\SSL>jarsigner -verify
writeFile.jar

jar verified.
```

Run a Signed Applet:

```
<applet code="writeFile.class"
        archive="writeFile.jar"
        width=400 height=400>
</applet>
```



ucsc
kasun@cmb.ac.lk

Questions?