COMPUTER SECURITY

3. Security policies

```
The company's computers should be used only for work.
```

```
xhost +maq1 -maq2
```

```
Grant {
      permission java.net.SocketPermission
      "*:1024-65535", "connect,accept"; };
```

```
<Directory /usr/share/doc>
      Order deny,allow
      Deny from all
      Allow from fe.up.pt
</Directory>
```

```
Only users of group ADMINISTRATOR can install system software.
```

Definition of security policy

- specification of requirements for considering a system as secure
- statement of system's states, splitting them in:
  - secure (or authorized) and
  - insecure (or unauthorized)
- the set of all elementary security policies (as stated above)

Setting up security

- specify security policy -> who can do what, how and when
- use security mechanisms -> enforce the defined policy

So, in a secured system:

- the policy states which are its secure states (and its insecure ones)
  - the specification should be complete and correct
- the mechanisms enforcing the policy should prevent the system from entering an insecure state
  - the implementation should be complete and correct

Secure system:

- starting from an authorized state, never enters an unauthorized one

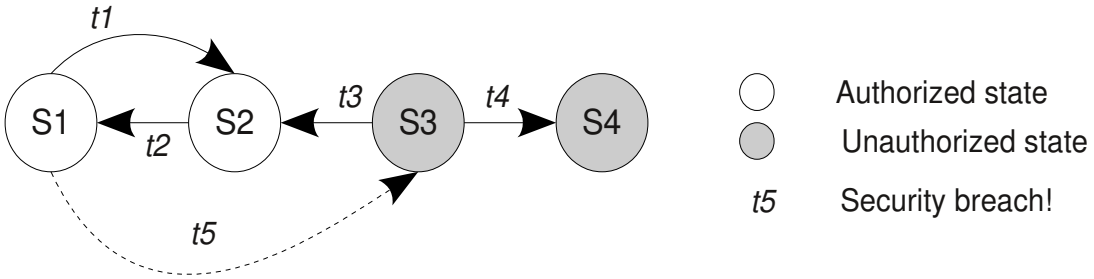

Fig. Example of an insecure system.

Important issue:

- Can you assure a system is secure?
  - in general, NO.
  - in some cases, yes.

A classification for security mechanisms

- secure
  - always keep the system in secure states
  - *are the most convenient for the administrator* (why?)
- precise
  - always keep the system in any of the possible secure states
  - are the most convenient for the user (why?)
- broad (portuguese: *latos*)
  - keep the system in states that can be secure or not
  - *are the most common* (why?)

...A classification for security mechanisms (cont.)

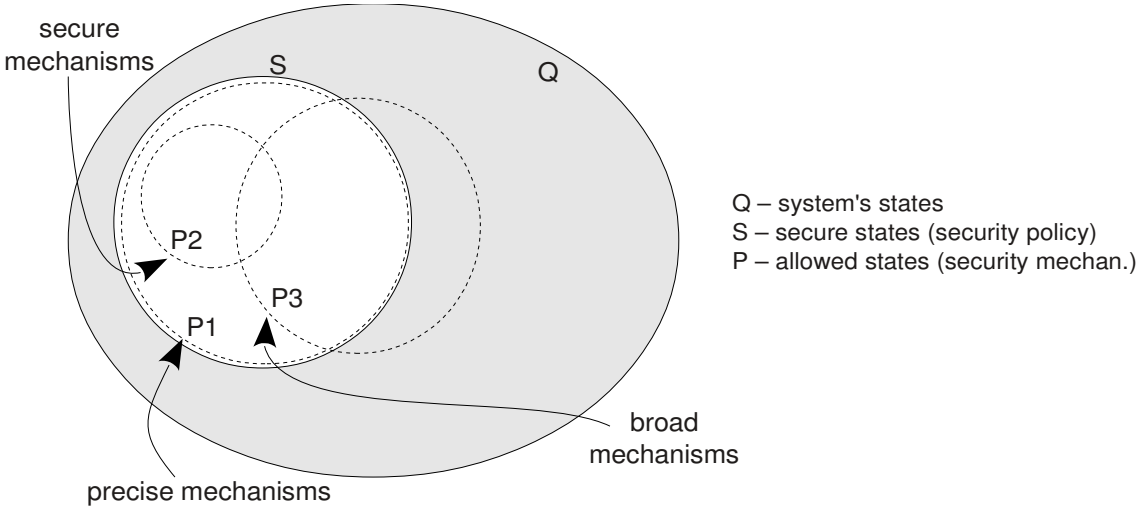

Fig. Types of security mechanisms and states of a system.

Exercise (policies and security mechanisms):

A Faculty states that all students who attend must have an honest academic conduct.

a) Consider the situation in terms of policy, mechanisms to implement it, and authorization states.

b) Present some situations where there may be incidents of dishonesty and suggest ways of dealing with them.

c) Rate and lay out the situations presented in the previous paragraph in terms of authorization states and state transitions. Include the ways you have suggested for the Faculty to deal with the situations.

d) How would you classify the penalties for misconduct?

Another classification for security mechanisms (and policies)

- discretionary or identity-based (*individual*)
  - control is determined by the info's or resource's owner
- mandatory or rule-based (*geral*)
  - usually, control is imposed by the system
- combined!...
  - the system enforces certain controls and the owner is allowed to exercise others...

Exercise:

Present examples of each type of situations, eventually based on your experience as a student of this Faculty, if you which.

Languages for expressing policies

- normal *versus* technical
- high-level *versus* low-level

High-level

- kind of abstract language (even mathematical)

Low-level

- language particular to the situations (and even associated to specific security mechanisms)

Examples of policies

```
xhost +maq1 -maq2
```

```
Grant {
      permission java.net.SocketPermission
      "*:1024-65535", "connect,accept"; };
```

```
<Directory /usr/share/doc>
      Order deny,allow
      Deny from all
      Allow from fe.up.pt
</Directory>
```

```
The company's computers should be used only for work.
```

Exercise:

Classify the policies' languages presented in the above examples and explain, in general terms, what would be meant to achieve with each policy.

Exercise:

Present a computer security policy for FEUP, with half a dozen items.
Suggest some mechanisms that could be used to enforce the policies.

Types of security policies

- confidentiality
  - military
- integrity
  - commercial
- availability
  - quality of service
- combined
  - most common in real systems!

Notation:

- *S* : subject or group of subjects (*entidade*)
  - active system's agent (user -> process...)
- *O* : object or class of objects
  - what undergoes the action of *S* (in other contexts, could also be a subject)
- *I* : information kept in *O*
  - content or state or attribute of *O* to be protected

- general admissible access actions[1]
  - "read" (*R*) Object by Subject, so acquiring its Information
  - "write" (*W*) Object by Subject, so modifying its Information (just writing, no reading implied!)
  - "execute" (X) Object by Subject, so potentially modifying its Information state (just executing, no reading implied!)

1 beware of variants dependent on security models!

Confidentiality policies

Definition

- policies that aim to prevent unauthorized disclosure of information
- *I* of *O* is confidential relative to *S* if:
  - *S* cannot get to know *I*

Bell-LaPadula's confidentiality model

- reference in modeling of computer security (multi-level systems)
- is specially used in military installations
- define ordered security *levels* (that may contain unordered *categories*[1])
- determines the membership of entities and objects in levels (and categories)
- specifies rules for subjects accessing (observing or altering) objects

1 also called *compartments*

Security levels

- "quantitative" classifications of subjects and objects
- ordered by the security "importance" (degree of confidentiality) that the system grants them
- *Examples*: TOP SECRET, SECRET, CONFIDENTIAL, UNCLASSIFIED

Security categories

- qualitative classifications of kinds of information
- are independent of the classification by security levels
- the inclusion relation of set theory is used here
- *Examples* of categories: elements of {ARMY, NAVY, AIR FORCE} and their combinations, for example, {ARMY, NAVY}.
- *Examples* of relations: the category {ARMY, NAVY} encompasses the category {ARMY}

...Security levels and categories (cont.): illustration

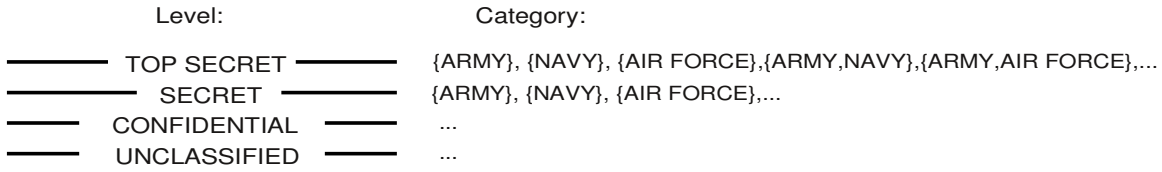

Fig. Illustration of security levels and categories, according to Bell-LaPadula's model.

... Security levels and categories (cont.): alternative illustration

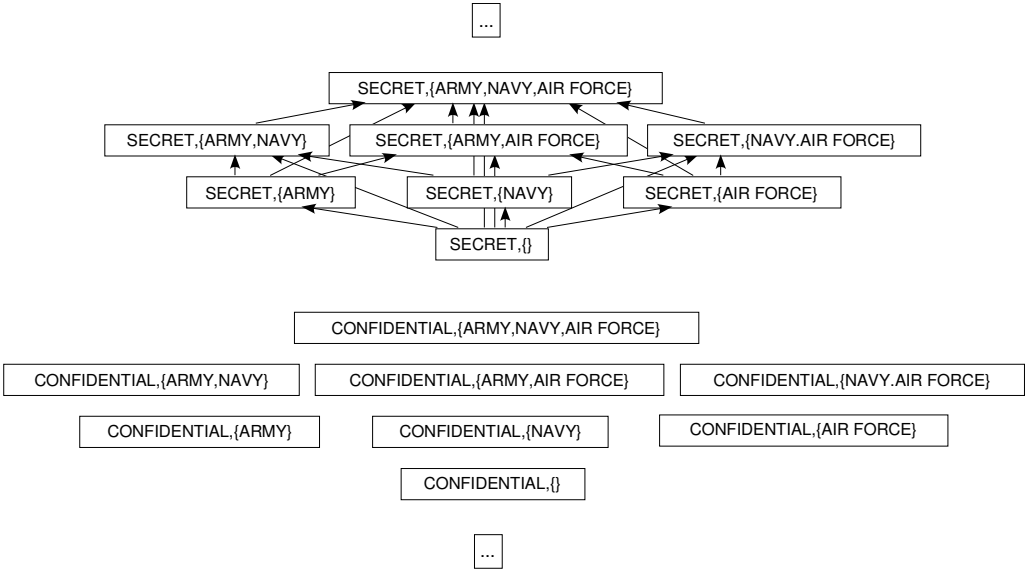

Fig. The arrows (shown only for level SECRET) represent the inclusion relations between categories (X includes Y: X ¬ Y).

Example of usage of security levels and categories:

- Subject “Adam” *is in* (or *has clearance at*) level (SECRET,{AIRFORCE})
- Document “Report X” *is in* (or *is at*) level CONFIDENTIAL, {NAVY,AIRFORCE})
- Can “Adam” access “Report X”?

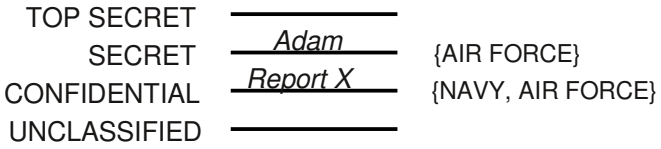

Fig. What type of access will “Adam” have relative to “Report X”?

Security policies (Bell-LaPadula's model)

- Simple security condition
- Star condition
- Discretionary security condition

Notation:

- classification of S: (L_S,C_S) -> (level, category)!
- classification of O: (L_O,C_O) -> idem!
- *observe* action corresponds to previously defined *reading* (R)
- *alter* action corresponds to previously defined *writing* (W)

Comment:

- In a less formal presentation, only the two first conditions are mentioned; but they imply the third!
- This is because only the first two policies are *mandatory*, and specific of the model; the last one is *discretionary*, and common to other models.

...Security policies of Bell-LaPadula's model (cont.)

Simple security condition (ss - *property*)

- subject *S* may *observe* object *O* (so, acquiring its information *I*) if and only if
  - $L_S \geq L_O$, $C_S \supseteq C_O$
- informally: you can only read documents in a security level equal or inferior to your own level and of equal or narrower scope

Star condition (* - *property*, *star property*)

- *S* may *alter O* (so, changing its information *I*) only if
  - $L_S \leq L_O$, $C_S \subseteq C_O$
- informally: you can only write documents in a security level equal or superior to your own level and of equal or broader scope

D*iscretionary security*, ds – *property* (*Condição individual de segurança*)

- subject *S* may access object *O* if and only if
  - *S* has the corresponding individual access permission to *O*

... Security policies of Bell-LaPadula's model (cont.)

Comments:

- The model's main concern is to prevent the unauthorized knowledge (observation) of an objects' information;
  - that explains the model's specific use of “alteration” control: preventing the unauthorized insertion of sensitive information in objects
  - there is no other worrying about the integrity control of objects!
- In the original model,
  - “read” permission (*R*) encompasses “observation” but not “alteration”
  - “write” permission (*W*) encompasses both “observation“ and “alteration”;
  - “append” permission encompasses “alteration” but not “observation”;
  - “execute” permission encompasses neither “observation“ nor “alteration”;
    - this clashes with current computer operation as execution of a file will give away information of the file (the results of its code's execution) and invocation of a routine might change the state of its associated object!
    - each implementation of the model will have to clarify this.
- Here, as said, we used: *R* – just reading; and *W* – just writing.

Exercises:

- Answer the question raised above, «*Will "Adam" be able to read or write in "Report X"?*».
- And if "Report X" was " TOP SECRET", with the same category?
- (Compare your answer with the one resulting from the picture below.)

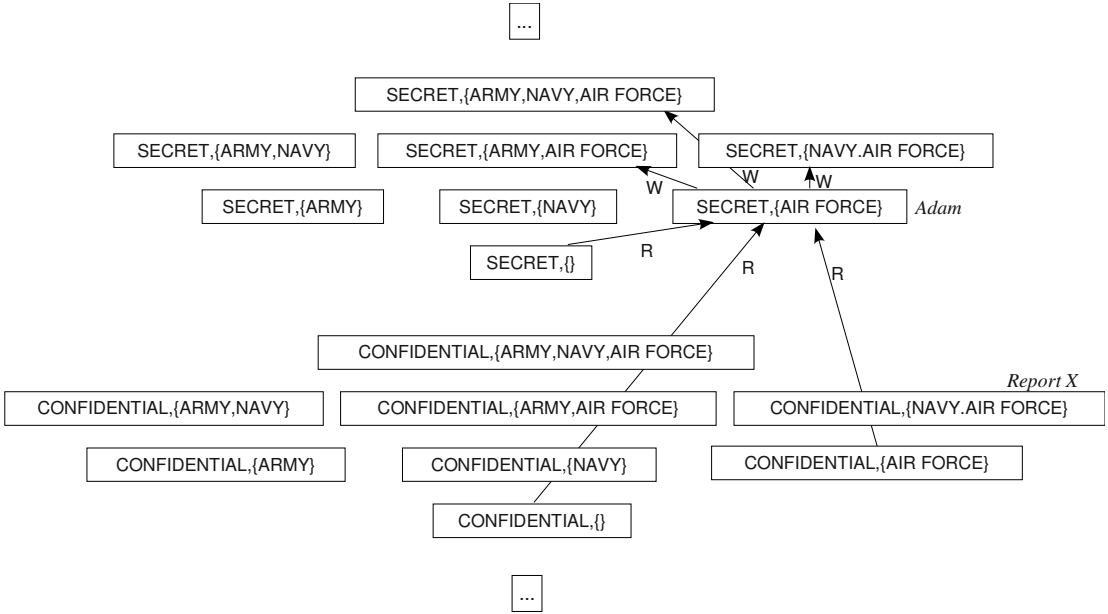

Fig. Overall picture of subject “Adam” and object “Report X” in the security level's tree. It is seen what “Adam” may read (R) and write (W) in levels SECRET and CONFIDENTIAL.

Exercise:

- A Colonel has clearance at (SECRET, {NUC, EUR})
- A Lieutenant has clearance at (SECRET, {EUR})
- The Lieutenant may communicate with the Colonel ("talk" to him and "be heard")
- But the Colonel may not communicate with the Lieutenant ("talk" to him and "be heard")
- What is absurd!
- Solution? See ahead...

Implications of the Bell-LaPadula model

Comment: some terminology and concepts of the model are here simplified, updated and even ignored (such is the case of “state”, “rule”, “action”...)

Secure states:

- satisfy all of the stated conditions

Safe state transitions:

- preserve the satisfaction of all of the stated conditions

Basic security theorem (Bell-LaPadula)

- A system is secure if it implements correctly the policies of Bell LaPadula's model.
- useful to a formal verification:
  - if a system starts from a secure state and if
  - the allowed state's transitions are always safe
    - then the system will always be secure
    - (no matter what it receives as input).

Alteration of security levels

- Are necessary to deal with some situations: e.g. to produce documents that may be “disclosed”.
- The system should provide appropriate mechanisms for transition of levels and level control. For instance:
  - permission for the alteration (reduction?) of a subject's privileges, that will have allocated (and controlled!):
    - a maximum security level
    - a current security level
  - acceptance of existance of trusted subjects, that may violate some security properties

Exercise:

Will the system now remain secure? Will the basic security theorem be useful now?

Objections and controversy over the model

- Some people, namely McLean, objected that, starting from a few “special” assumptions, the basic theorem could be induced to find secure a state that clearly was not.
  - (McLean presented a system that, with state's transitions allowed by Bell-LaPadula's model, could be conducted to a state in which all subjects had minimum privileges and all accesses were allowed!
- LaPadula objected that McLean's example did not contravene the base assumptions of his model – it is focused only on confidentiality and assumes that the discretionary control is essentially static.
  If the state's transitions used in McLean's example were necessary to the application at hand, the model should include them; otherwise, they should not be implemented.
  - the “tranquility condition”, implicit in the model, specifies the state's transitions that are allowed.

...Objections and controversy over the model (cont.)

Principle of tranquility

- Strong:
  - security levels do not change during the lifetime of a system
- Weak:
  - security levels do not change in a way that violates the security policies
- -> largely prevents McLean's objection

Results of the controversy

- Above all, the controversy showed that the base assumptions and the definition of security itself are essential for establishing a secure system.

Exercise:

Do exercise n. 2 of chapter 5 of Bishop's book (the big one), p. 150.

Integrity policies

Definitions

- policies that are aimed to assure the correction of the information, preventing its non-authorized alteration
- *I* of *O* is integer relative to *S* if:
  - *S* can trust *I* (believing it is correct)
- types of information:
  - content and attributes of *O*
  - origin, including place and originator subject
  - behavior, or operation, of an “active” object

Biba's integrity Model

- already takes into account the execution of programs that may endanger the integrity of the system to protect

Policies:

- of strict integrity (Biba's model)
- ... (*not covered here*)

Integrity levels

- Security ratings for subjects and objects
- the higher the level, the greater is the trust on an object and on the actions of a subject
- qualitative categories can also be defined and used, just like in Bell-LaPadula's model

Strict integrity policy (Biba's model)

- a subject may only read from objects rated at a superior or equal level
- a subject may only write to objects rated at an inferior or equal level
- a subject may only execute operations on objects rated at an inferior or equal level
- -> dual of LaPadula's policy

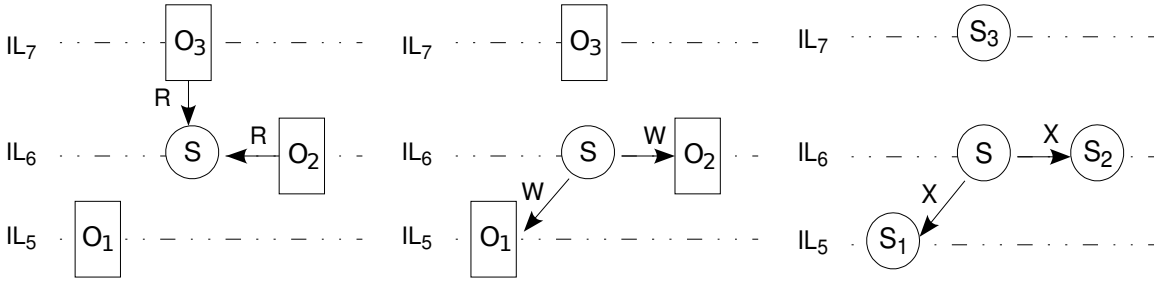

Fig. The different parts of the **strict integrity policy** (the so called **Biba's model**). (The only difference to other policies (also related to the model) lies in the left side picture!.

Exercise:

The security classification of the subjects and objects of a system should, in general, differ, depending on the type of security protection (confidentiality or integrity). But let us suppose that in a given system the classification was equal; for example, the confidentiality level, LC_O, of an object was always equal to the its integrity level, LI_O.

- In what conditions could an entity S read from an object O?
- And in what conditions could an entity S' write to an object O?

Lipner and the integrity of a system

- he considers that the integrity protection of a commercial system, needs it to be organized in a specific structural and functional way
- he recommends there should be
  - separation of duty (*separação de funções - de responsabilidades ou de pessoas*)
    - for instance, production and development tasks should be performed by different subjects
  - separation of function (*separação de ambientes*)
    - for instance, production and development should have different working environments and data
  - auditing
    - for certification and control, performed by “external” subjects
- he formulates a set of specific integrity requirements for the protection of a system (Lipner's requirements)

Hybrid security policies

Definition

- policies that encompass both confidentiality and integrity concerns in the system's protection specification

Comments

- In fact, some previous models already encompass both concerns!
- In complex systems, with several environments, the subjects may be under different types of policies, depending on which environments they are operating on, at given times.

The Chinese Wall model

- very close to the enterprise world
- very close to current legislation issues
- the “acquired knowledge” assumes here a very important role

Definitions

- **Object (*O*)**
  - information's item related to a company
- **Company dataset (CD)**
  - objects related to a single company
- **Conflict of interest class (COI)**
  - group of datasets of a company (that competes with other companies)

...The Chinese Wall model: graphical representation

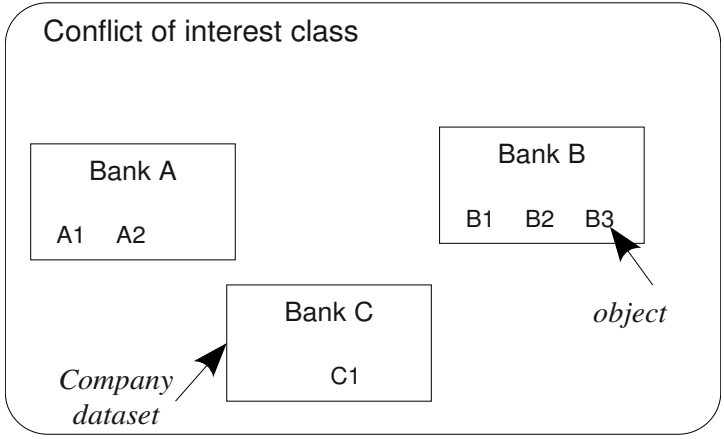

Fig. The Chinese Wall model: objects, datasets and conflict of interest class.

Policies for the Chinese Wall model

Simple security condition (ss - *property*)

- Subject *S* may read object *O*, if and only if satisfies <u>any</u> of the following conditions:
  - *S* has never read objects of other datasets, in the same conflict of interest class
    - [*S* has never read *O'* such as $CD(O') \neq CD(O)$ and $COI(O') = COI(O)$]
  - *S* has only read objects of other conflict of interest classes
    - [*S* has only read any *O'* such that $COI(O') \neq COI(O)$]
  - object *O* is sanitized (is purged of sensitive data)

Exercises:

1. Present examples of constrains, derived from this policy, for a subject that is working with Bank A (see above picture)
2. A marketing company works with banks A, B and C. Knowing that each marketing campaign needs at least 2 professionals, which is necessarily the minimum number of professionals that the company must use?

Star condition (* - *property*)

- Subject *S* may write to object *O*, if and only if satisfies all of the following conditions:
  - *S* satisfies the "simple security condition"
  - *S* cannot read sensitive objects of other datasets, beyond those in the dataset in where it intends to write on
    - [*S* cannot read any sensitive *O'* such that $CD(O') \neq CD(O)$]
    - Comment: this condition just says that a subject wanting to write sensitive data, is confined to a single dataset (for writing and reading).

...Policies for the Chinese Wall model (cont.)

Exercise:

- Suppose that S1 works with Bank A and that S2 works with Bank B. Both S1 and S2 work with Insurance Company C of a different class of conflict of interest than the bank's. Without peeking at the picture below:
  - explain how there could be a security breach, by information leakage, if the only policy taken into account was the “simple security condition” of the Chinese wall model.
  - Show how that leak could be prevented, by taking into account (and enforcing) the “star condition” policy
- (Look at the picture below, only after considering your own answer to this problem.)

...Policies for the Chinese Wall model (cont.)

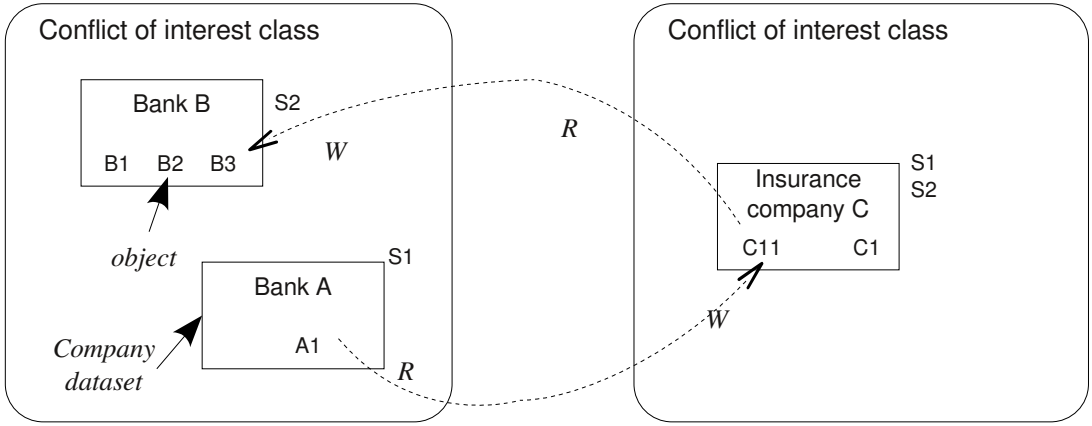

Fig. Illustration of the importance of the “star condition” in the Chinese wall model.

Originator's control policies (or “copyrights' policies“)

Definitions

- originator (or author or creator) – subject that created an object or, somehow, detains special rights over it
- holder (or owner) – entity in possession of the object
- user – subject that can use the object (in predefined ways)

Rules (policies) of the model of originator's control

- the current holder of the object cannot alter its access attributes
- copies of the object keep its original access constraints
- the originator of the object can alter its access attributes (in terms of use and of rights of access (!...)

Prob.: how to perform this type of control?
Note: acronym: ORCON, ORiginator CONtrolled...

Role-based policies (*políticas baseadas em funções*)

- The possibility of a subject to access (reading, writing...) an object depends on the current role of the subject (its current duties – *funções ou papel que desempenha*).
- So, what matters is the role, not the subject.

Definitions

- Transaction, task (*transacção, tarefa*) - activity that corresponds to the execution of a certain procedure or set of procedures (or program or set of programs)
- Role (*função, papel*) – duties given to a subject so that, by execution of a certain set of tasks, a goal can be achieved
- Active role (*função activa*) – current subject's role
- Authorized roles – set of roles that an entity might be invested with

Rules (policies) of the role-based model

- Assignment of tasks – a subject may execute a task only when given the appropriate role
- Authorization of active roles – an entity may only be given an active role from the set of authorized roles
- Authorization of tasks – a task may only be executed by a subject in an appropriate role
- Incompatibility of roles – an entity may only assume roles that are not mutually exclusive; the set of mutually exclusive roles will have to be stated and is system-dependent (principle of separation of duties - *princípio de separação de funções)*

Model's Difficulties

- What about when a subject changes role? Will there not be a security breach, due to the previously acquired knowledge?...
- Subjects are humans, right?

Noninterference and policy composition

Real systems' issues

- In large systems, structured with several sub-systems each of which with its own specific policies, it is necessary to set global policies (*policy composition)*.
- On the other hand, in certain cases, it may happen that policies are not being enforced, because the mechanisms in action do not prevent “less orthodox” uses of the system (covert channels – *canais camuflados)*.
  - such situations might reflect “interference” between the system's subjects

Base principles of composite systems

- **autonomy**: a permission granted by one of the system's components, should also be granted by the composite system's policy
- **security**: a denial of access declared by the policy of one of the components, should also be denied by the composite system's policy
- **safe defaults**: at the start, no permission is granted, but should be explicitly granted when necessary

Exercise:

In system X, Charles and Diana may access the files of both. In system Y, Alice may not access Bernard's files. In composite system (X, Y), a policy declares that Alice may access Charles' files and that Diana may access to Bernard's.

- In system (X, Y), will Charles be able to continue accessing Diana's files?
- And could now Alice access Bernard's files?
- And what can be said to Alice if she tries to access Diana's files?

Pointers...

- The "**Bell-La Padulla model paper**", 1976 – D. E. Bell and L. J. La Padula
  - csrc.nist.gov/publications/history/bell76.pdf
- The "**MacLean objection paper**", 1985 – John McLean
  - www.dtic.mil/dtic/tr/fulltext/u2/a462369.pdf
- The "**Biba model paper**", 1975 – K. J. Biba
  - seclab.cs.ucdavis.edu/projects/history/papers/biba75.pdf
- The "**Lipner integrity model paper**", 1982 – Steven B. Lipner
  - www.cs.washington.edu/research/projects/poirot3/Oakland/sp/PAPERS/00044637.PDF
- The "**Chinese wall paper**", 1989 - David F. C. Brewer and Michael J. Nash
  - www.cs.purdue.edu/homes/ninghui/readings/AccessControl/brewer_nash_89.pdf
- The "**ORCON paper**", 1989 – Richard Graubart
  - www.dtic.mil/dtic/tr/fulltext/u2/a219102.pdf
- The "**RBAC paper**", 1992 – D.F. Ferraiolo and D. R. Kuhn
  - arxiv.org/pdf/0903.2171v2.pdf