

# Policies, Mechanisms, and Assurance

For any system:

Specification : What is it supposed to do?

Implementation : How does it do it?

Correctness : Does it really work?

In security:

Specification : Policy

Implementation : Mechanism

Correctness : Assurance

But:

- Some mechanisms are presented as policies.
- “One man’s policy is another man’s mechanism.” .
- Attackers will not politely respect abstraction layers.
- Assurance issues can guide policies and mechanisms.

# Policies

## Security properties

The main security properties are:

- **Integrity**  
(no improper modification of information)
- **Secrecy**  
(no improper disclosure of information)
- **Availability**  
(no improper denial of service)

## Variations on integrity

**Authenticity** is often the same as integrity, with a difference only in emphasis.

Other concepts are closely related to integrity:

- **non-repudiation**,
- **accountability**.

## Variations on secrecy

Similarly, **confidentiality** is basically secrecy.

So is **privacy**, though it is applied differently, particularly in the context of personal information.

**Anonymity** is basically an instance of secrecy.

**Pseudonymity** is anonymity plus linkability.

**Plausible deniability** is the contrary of non-repudiation and might be viewed as a weak form of secrecy.

## Security policies

Security properties are combined into security policies.

For example, a bank may want:

- authenticity of clients at ATMs and on the Web,
- non-repudiation of transactions,
- integrity of the books,
- integrity of the messaging systems,
- secrecy for client data and for internal data,
- availability of the alarm system.

The conjunction of these properties may be the bank's security policy.

## Security policies (cont.)

The bank's security policy may include less standard properties:

- exclusivity of duties to avoid conflicts of interest (Chinese Wall policies),
- dual control for sensitive transactions.

## Security policies (cont.)

There is no unique definition of security.

- Different security policies are appropriate for different users, organizations, and systems.
- In a composite system, we may find conflicting desires, e.g., non-repudiation and plausible deniability.

Policies are not always articulated precisely.

- For privacy of personal data.
- Against email flooding and spam.
- Against denial-of-service attacks.



Security is a joy born of the idea of a future or past thing, concerning which the cause of doubting has been removed.

*Spinoza.*

*Ethics, Book III (out of context)*

## Common themes

- Interaction with an uncertain environment.  
(Contrast with mutual exclusion.)
- Some security even against lucky, powerful, or persistent attackers.
  - Even if the attacker controls the network.
  - Even if a secret is compromised.
  - Even if an insider is dishonest.
- Doing without full functional correctness.  
E.g., message origin, not message correctness.

# Principals

A lot of security properties concern principals or subjects:

- users,
- computers,
- services.

The notion of principal varies (dangerously) across systems and abstraction layers:

- IP addresses,
- the computers at those addresses,
- the people who control the computers,
- ∴

## Information-flow security policies (a.k.a. multilevel security)

- Classify data and subjects into levels  
(e.g., public, secret, top secret),  
(e.g., trusted, untrusted).
- The levels need not be linearly ordered.  
They should form a lattice.

(what is a lattice?)

## Noninterference

A “low” output should not depend on “high” inputs.

More precisely, suppose that:

- Each subject  $s$  executes a (deterministic) function  $f_s$ .
- Each input into a system has a security level.

If  $s$  has level  $l$ , then the result of  $f_s$  may depend only on the inputs at level  $\leq l$ .

Extensions deal with infinite computations, probabilities, nondeterminism, . . . .

## Noninterference (cont.)

Suppose that input  $x$  has level  $k$ ,  $s$  has level  $l$ , and  $k \not\leq l$ .

Let:

$$f_s(\dots, x, \dots) = \text{if } x \text{ then } 1 \text{ else } 0$$

This function does not satisfy noninterference.

There is an “implicit” flow of information.

## Controlling information flows

The Bell-LaPadula model for secrecy:

- No subject may read data at a higher level.
- No subject may write data to a lower level.

The second condition is the  $\star$ -property and serves to thwart Trojan horses.

These are mandatory requirements on information flow (not at the discretion of subjects!).

## Controlling information flows (cont.)

The Biba model for integrity:

- No subject may read data at a lower level.
- No subject may write data to a higher level.

## Reading

A paper by McLean:

- “Security models”  
at [citeseer.ist.psu.edu/mclean94security.html](http://citeseer.ist.psu.edu/mclean94security.html).

A paper by Joshi and Leino:

- “A semantic approach to secure information flow”  
reachable from  
[citeseer.ist.psu.com/joshi00semantic.html](http://citeseer.ist.psu.com/joshi00semantic.html).

## On the reading

Sometimes I will introduce a subject in just a few minutes (e.g., information-flow control). The reading should help.

I expect that it constitutes a fair amount of work.

I expect you to prioritize (skim everything, study carefully only what seems most important).

If you would like help or complementary material:

- talk to others in the course,
- come see me,
- read Anderson's book.