Computer Security — Part 2: Resource Security

Dusko Pavlovic

Oxford
Michaelmas Term 2008

Outline

Outline

Recall from Lecture 1

Resource security (access control)

- **authorization:** "bad *resource calls* don't happen"
- **availability:** "good *resource calls* do happen"

In an operating or a computer system

- all resource constraints are security properties

What is a resource?

A resource is whatever we (humans, animals, organisms) compete for.

What is a resource?

A resource is whatever we (humans, animals, organisms) compete for.

Examples

- territory, food, storage, CPU. . .
- axe, printer, program. . .
- money, information, reputation. . .

What is a resource?

A resource is an **object** used in computation
or in social interaction.

What is a resource?

A resource is an **object** used in computation
or in social interaction.

A computer system or a social group

consists of

- subjects $\mathcal{S}$: people, users, agents, voters. . .
- objects $\mathcal{O}$: goods, files, devices, candidates. . .

Security 2: Resource Security
Dusko Pavlovic
Authorization
Resources
Access control
Multi level security
Security models
Availability
Summary

What is a resource?

A resource is anything that can be **secured**.

Security 2: Resource Security
Dusko Pavlovic
Authorization
Resources
Access control
Multi level security
Security models
Availability
Summary

What is a resource?

A resource is anything that can be **secured**.

Simplest resource security requirements

- privately owned: requires authorization
  - den, shelter, home, account. . .
- publicly shared: requires availability
  - well, path, printer, Internet. . .

Security 2: Resource Security
Dusko Pavlovic
Authorization
Resources
Access control
Multi level security
Security models
Availability
Summary

What is a resource?

A resource is anything that can be **secured**.

Simplest resource security requirements

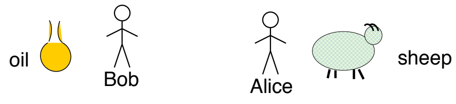
- privately owned: requires authorization
  - den, shelter, home, account. . .
- publicly shared: requires availability
  - well, path, printer, Internet. . .

Resource use in social and computational systems
is based on complex combinations of owning and sharing.

Security 2: Resource Security
Dusko Pavlovic
Authorization
Resources
Access control
Multi level security
Security models
Availability
Summary

Access control

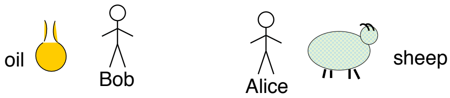
Privately owned resources



Security 2: Resource Security
Dusko Pavlovic
Authorization
Resources
Access control
Multi level security
Security models
Availability
Summary

Access control

Privately owned resources

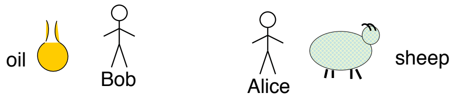


q_0		
	sheep	oil
Alice	use	∅
Bob	∅	use

Table: Permission matrix

Access control

. . . can be traded, jointly owned, partially shared etc.



q_1		
	sheep	oil
Alice	{milk, wool}	cup oil
Bob	cup milk	use

Table: Permission matrix

Permission matrix

For the given sets

- $\mathcal{S}$ of subjects
- $\mathcal{O}$ of objects
- $\mathcal{A}$ of actions

a *permission matrix* at a state q is an assignment

$$\mathcal{S} \times \mathcal{O} \xrightarrow{M^q} \wp\mathcal{A}$$

- of the pairs $\langle u, i\rangle \in \mathcal{S} \times \mathcal{O}$ to
- to the sets (possibly empty) of actions $M^q_{ui} \subseteq \mathcal{A}$

which the subject u is permitted to execute on the object i.

Access matrix

For the given sets

- $\mathcal{S}$ of subjects
- $\mathcal{O}$ of objects
- $\mathcal{A}$ of actions

an *access matrix* at a state q is an assignment

$$\mathcal{S} \times \mathcal{O} \xrightarrow{B^q} \wp\mathcal{A}$$

- of the pairs $\langle u, i\rangle \in \mathcal{S} \times \mathcal{O}$ to
- to the sets (possibly empty) of actions $B^q_{ui} \subseteq \mathcal{A}$

which the subject u attempts to execute on the object i.

Authorization

Access control is thus enforced by

- preventing the accesses in B^q_{ui}
- that are not permitted in M^q_{ui}.

Authorization

Access control is thus enforced by

- preventing the accesses in B^q_{ui}
- that are not permitted in M^q_{ui}.

The operating system makes sure at every state q that

$$B^q_{ui} \subseteq M^q_{ui}$$

holds for every subject u and every object i.

Access control implementations

In UNIX-like operating systems,

- $\mathcal{S}$ = users
- $\mathcal{O}$ = files
- $\mathcal{A} = \{r, w, x\}$, i.e., read, write and execute

Access control implementations

In UNIX-like operating systems,

- $\mathcal{S}$ = users
- $\mathcal{O}$ = files
- $\mathcal{A} = \{r, w, x\}$, i.e., read, write and execute

Access Control Lists (ACL)

UNIX does not maintain large global matrices

$$\mathcal{S} \times \mathcal{O} \xrightarrow{M,B} \wp\mathcal{A}$$

but smaller object-based Access Control Lists

$$\mathcal{O} \longrightarrow (\wp\mathcal{A})^U$$

where $U = \{\mathtt{u}, \mathtt{g}, \mathtt{o}\}$, with $\mathtt{u} \in \mathcal{S}$, $\mathtt{g} \subseteq \mathcal{S}$ and $\mathtt{o} = \mathcal{S}$.

Access control implementations

In UNIX-like operating systems,

- $\mathcal{S}$ = users
- $\mathcal{O}$ = files
- $\mathcal{A} = \{r, w, x\}$, i.e., read, write and execute

Capabilities

Symbian does not maintain large global matrices

$$\mathcal{S} \times \mathcal{O} \xrightarrow{M,B} \wp\mathcal{A}$$

but smaller *subject*-based *Capabilities*

$$\mathcal{S} \longrightarrow \wp(\mathcal{O} \times \mathcal{A})$$

where each subject stores cryptographically protected capability tags $\langle i, a \rangle$.

Access control implementations

Homework

Read the about UNIX permission matrices (ACLs) in your favorite UNIX reference. What do the commands `chmod`, `setacl` and `getacl` do?

Access control implementations

Homework

Read the about UNIX permission matrices (ACLs) in your favorite UNIX reference. What do the commands `chmod`, `setacl` and `getacl` do?

Compare the UNIX access control with the Windows access control.

Access control implementations

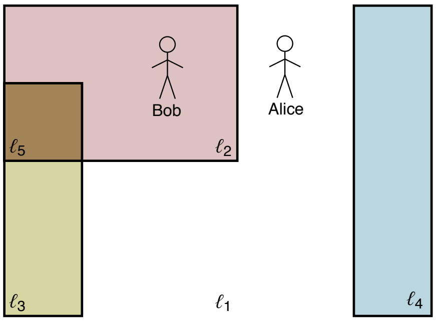
Homework

Read the about UNIX permission matrices (ACLs) in your favorite UNIX reference. What do the commands `chmod`, `setacl` and `getacl` do?

Compare the UNIX access control with the Windows access control. The paper *"Windows access control demystified"* by Govindavjahala and Appel may help.

Multi level security

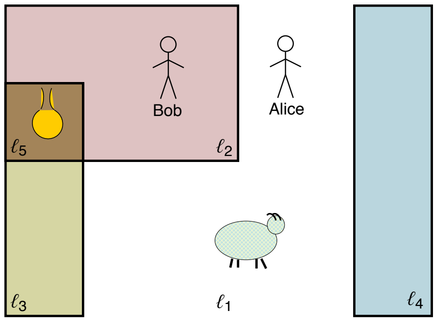
In the meantime, at the dawn of Neolithic, Bob builds protected vaults ℓ_2 and ℓ_3, with a secure chamber ℓ_5.



Multi level security

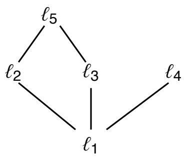
Security 2: Resource Security
Dusko Pavlovic
Authorization
Resources
Access control
Multi level security
Security models
Availability
Summary

In the meantime, at the dawn of neolithic, Bob builds protected vaults ℓ_2 and ℓ_3, with a secure chamber ℓ_5.



Security levels

Security 2: Resource Security
Dusko Pavlovic
Authorization
Resources
Access control
Multi level security
Security models
Availability
Summary



$\ell \leq c$		
	location ℓ	clearance c
Alice	ℓ_1	ℓ_1
Bob	ℓ_2	ℓ_5
sheep	ℓ_1	
oil	ℓ_5	

Clearance structure

Security 2: Resource Security
Dusko Pavlovic
Authorization
Resources
Access control
Multi level security
Security models
Availability
Summary

For the given

- set $\mathcal{S}$ of subjects
- set O of objects
- partially ordered set $\mathbb{L}$ of security levels

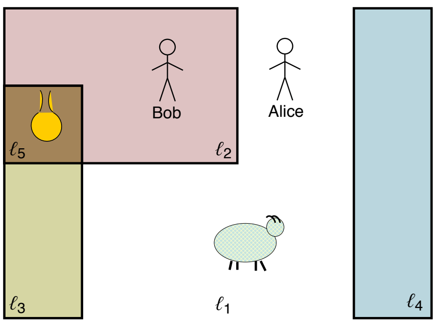
a *clearance structure* at a state q consists of the maps

- $c^q : \mathcal{S} \longrightarrow \mathbb{L}$ of *clearances*
- $\ell^q_{\mathcal{S}} : \mathcal{S} \longrightarrow \mathbb{L}$ of *subject locations*
- $\ell^q_O : O \longrightarrow \mathbb{L}$ of *object locations* (or *classifications*)

Maintaining multi level security

Security 2: Resource Security
Dusko Pavlovic
Authorization
Resources
Access control
Multi level security
Security models
Availability
Summary

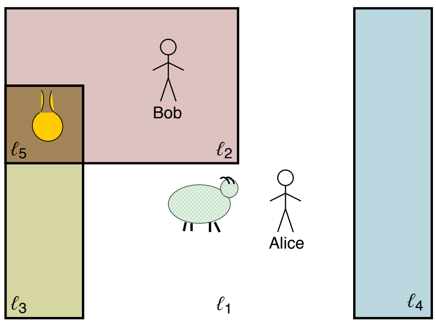
In the meantime, Alice and Bob agree



Maintaining multi level security: state q_0

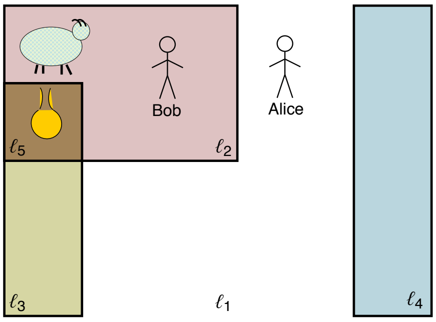
Security 2: Resource Security
Dusko Pavlovic
Authorization
Resources
Access control
Multi level security
Security models
Availability
Summary

In the meantime, Alice and Bob agree
to store Alice's sheep in Bob's protected vault ℓ_2.



Maintaining multi level security: state q_1

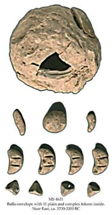
In the meantime, Alice and Bob agree
to store Alice's sheep in Bob's protected vault ℓ_2.



Security 2: Resource Security
Dusko Pavlovic
Authorization
Resources
Access control
Multi level security
Security models
Availability
Summary

Maintaining multi level security: state q_1

As a receipt for the deposit of her sheep into Bob's vault,
Alice gets a *secure token* in a clay envelope.



Security 2: Resource Security
Dusko Pavlovic
Authorization
Resources
Access control
Multi level security
Security models
Availability
Summary

Maintaining multi level security: state q_1

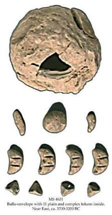
As a receipt for the deposit of her sheep into Bob's vault,
Alice gets a *secure token* in a clay envelope.

- To take the sheep, Alice must give the token.

Security 2: Resource Security
Dusko Pavlovic
Authorization
Resources
Access control
Multi level security
Security models
Availability
Summary

Maintaining multi level security: state q_1

As a receipt for the deposit of her sheep into Bob's vault,
Alice gets a *secure token* in a clay envelope.



- To take the sheep, Alice must give the token.
- To give the sheep, Bob must take the token.

Security 2: Resource Security
Dusko Pavlovic
Authorization
Resources
Access control
Multi level security
Security models
Availability
Summary

Maintaining multi level security: state q_1

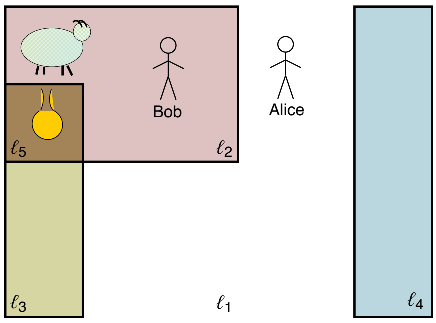
As a receipt for the deposit of her sheep into Bob's vault,
Alice gets a *secure token* in a clay envelope.

- To take the sheep, Alice must give the token.
- To give the sheep, Bob must take the token.
- Anyone who gives the token can take the sheep.

Security 2: Resource Security
Dusko Pavlovic
Authorization
Resources
Access control
Multi level security
Security models
Availability
Summary

No-read-up: state q_1

Alice cannot take ("read") the sheep out of the vault,
because she cannot enter there.



No-read-up: state q_1

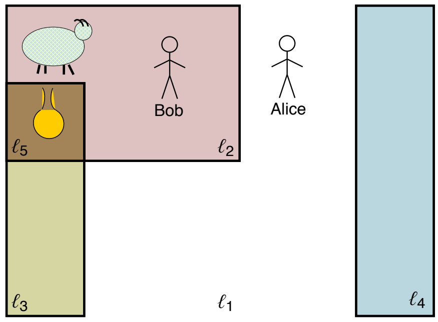
Only a subject cleared to enter the vault can take ("read") an object from there

$$\mathrm{r} \in B_{ui} \implies \boldsymbol{c}(u) \geq \ell(i)$$

No-write-down: state q_1

Security 2: Resource Security
Dusko Pavlovic
Authorization
Resources
Access control
Multi level security
Security models
Availability
Summary

Bob cannot give ("write") the sheep out of the vault while he is in there.



No-write-down: state q_1

Security 2: Resource Security
Dusko Pavlovic
Authorization
Resources
Access control
Multi level security
Security models
Availability
Summary

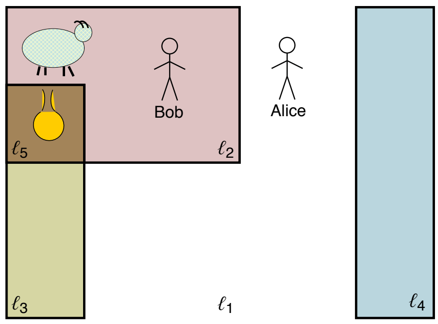
Only a subject who is outside the vault can give ("write") an object there

$$\mathrm{w} \in B_{ui} \implies \ell(u) \leq \ell(i)$$

Maintaining multi level security: state q_1

Security 2: Resource Security
Dusko Pavlovic
Authorization
Resources
Access control
Multi level security
Security models
Availability
Summary

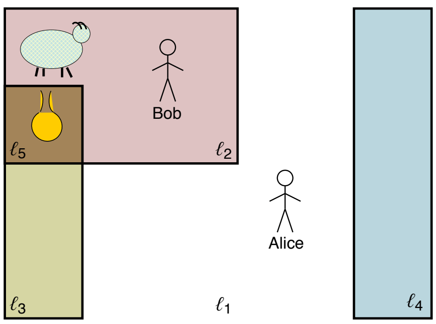
When Alice wants to take ("read") her sheep,



Maintaining multi level security: state q_1

Security 2: Resource Security
Dusko Pavlovic
Authorization
Resources
Access control
Multi level security
Security models
Availability
Summary

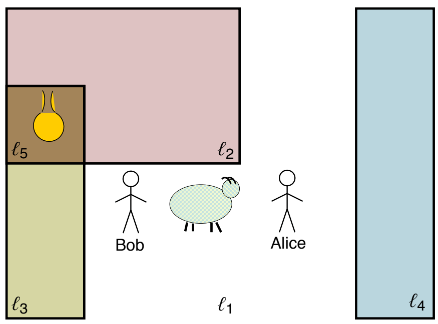
When Alice wants to take ("read") her sheep,



Maintaining multi level security: state q_2

Security 2: Resource Security
Dusko Pavlovic
Authorization
Resources
Access control
Multi level security
Security models
Availability
Summary

When Alice wants to take ("read") her sheep, Bob comes out, breaks the token, and gives ("writes") the sheep.



History of multi level security

- This security protocol goes back to Uruk (Irak), 4000 B.C.

History of multi level security

- This security protocol goes back to Uruk (Irak), 4000 B.C.
- More robust security tokens and promisory notes were made not only of clay, but also of horn, ivory, copper, silver, gold.

History of multi level security

- This security protocol goes back to Uruk (Irak), 4000 B.C.
- More robust security tokens and promisory notes were made not only of clay, but also of horn, ivory, copper, silver, gold.
- Security annotations on clay tokens evolved into cuneiform pictograms, the earliest writing and numeral system.

History of multi level security

- This security protocol goes back to Uruk (Irak), 4000 B.C.
- More robust security tokens and promisory notes were made not only of clay, but also of horn, ivory, copper, silver, gold.
- Security annotations on clay tokens evolved into cuneiform pictograms, the earliest writing and numeral system.
- **Writing and arithmetic have evolved from resource security protocols.**

History of multi level security

- This security protocol goes back to Uruk (Irak), 4000 B.C.
- More robust security tokens and promisory notes were made not only of clay, but also of horn, ivory, copper, silver, gold.
- Security annotations on clay tokens evolved into cuneiform pictograms, the earliest writing and numeral system.
- **Writing and arithmetic have evolved from resource security protocols.**
- In computers, banks, companies and governments Access Control and Multi Level Security are still organized around the same security model.

Outline

Security model

Bell-LaPadula, Biba, Clark-Wilson

Given a state machine Q, describing the computation with

- a set $\mathcal{S}$ of subjects
- a set $\mathcal{O}$ of objects
- a set $\mathcal{A}$ of actions
- a poset $\mathbb{L}$ of security levels

a security model consists of the following data for each state $q \in Q$

- a permission matrix $M^q : \mathcal{S} \times \mathcal{O} \longrightarrow \mathcal{A}$
- an access matrix $B^q : \mathcal{S} \times \mathcal{O} \longrightarrow \mathcal{A}$
- a clearance map $\boldsymbol{c}^q : \mathcal{S} \longrightarrow \mathbb{L}$
- a location map $\boldsymbol{\ell}^q : \mathcal{S} + \mathcal{O} \longrightarrow \mathbb{L}$

Secure states

A state $q \in Q$ is said to be secure with respect to a model $\langle M, B, \boldsymbol{c}, \boldsymbol{\ell} \rangle$ if the following conditions are satisfied for all subjects $u \in \mathcal{S}$ and objects $i \in \mathcal{O}$

- authorization: $B^q_{ui} \subseteq M^q_{ui}$,
- clearance: $\boldsymbol{\ell}^q(u) \leq \boldsymbol{c}^q(u)$
- no-read-up: $\mathtt{r} \in B^q_{ui} \Longrightarrow \boldsymbol{c}^q(u) \geq \boldsymbol{\ell}^q(i)$
- no-write-down: $\mathtt{w} \in B^q_{ui} \Longrightarrow \boldsymbol{\ell}^q(u) \leq \boldsymbol{\ell}^q(i)$

where $\mathtt{r}, \mathtt{w} \in \mathcal{A}$ are distinguished actions.

Security 2: Resource Security
Dusko Pavlovic
Authorization
Security models
Availability
Summary

Secure states

Homework

Formalize the details of the described sheep bank protocol with in terms of the multi level security model. Do not forget to include the clay token in the model, or else Bob may release the sheep to Eve.

Can Alice sell the sheep while in the vault?

Describe a similar protocol for digital content instead of the sheep.

Security 2: Resource Security
Dusko Pavlovic
Authorization
Security models
Availability
Summary

Secure states

Warning

The terminology of "security models" and "secure states" can be misleading.

The modeling methodology itself does not guarantee security. There are models where the formally secure states are intuitively insecure.

Security 2: Resource Security
Dusko Pavlovic
Authorization
Security models
Availability
Summary

Secure states

Warning

The terminology of "security models" and "secure states" can be misleading.

The modeling methodology itself does not guarantee security. There are models where the formally secure states are intuitively insecure.

Example: McLean's System Z

Every security model can be extended by the transitions to the state z with

$$\begin{aligned} \boldsymbol{c}^z(u) &= \top \\ \boldsymbol{\ell}^z(u) = \boldsymbol{\ell}^z(i) &= \bot \end{aligned}$$

where $\bot$ is the lowest and $\top$ the highest security level.

Security 2: Resource Security
Dusko Pavlovic
Authorization
Security models
Availability
Summary

Secure states

Warning

The terminology of "security models" and "secure states" can be misleading.

The modeling methodology itself does not guarantee security. There are models where the formally secure states are intuitively insecure.

Comment

The state z corresponds to a situation where all security constraints are eliminated. Such situations do happen, and sometimes need to be described.

A good language does not disallow false statements, but allows recognizing them.

Security 2: Resource Security
Dusko Pavlovic
Authorization
Security models
Availability
Summary

Secure states

Solution

In order to control

- downgrading of objects, and
- authorization of subjects

the state transitions must be constrained.

Secure states

Solution

In order to control

- downgrading of objects, and
- authorization of subjects

the state transitions must be constrained.

This leads to the distinction of

- discretionary access control,
  - where the authorizations can be delegated
- mandatory access control
  - where the authorizations are centrally managed

Secure states

Solution

In order to control

- downgrading of objects, and
- authorization of subjects

the state transitions must be constrained.

This leads to the distinction of

- discretionary access control,
  - where the authorizations can be delegated
- mandatory access control
  - where the authorizations are centrally managed

Many practical access control systems combine the two.

Outline

Denial of Service (DoS) attacks

Bob and Charlie go to Alice's restaurant. They did not book a table in advance. They don't get a table.

Annoyed, Bob and Charlie call next day, and book a lot of tables at Alice's. Through the evening, Alice turns back many guests. Bob and Charlie don't show up at all.

Distributed Denial of Service (DoS) attacks

In the future, Alice attempts to prevent bogus bookings by authenticating the callers: she asks for a callback number. This makes booking a table more complicated.

If he is very motivated, Bob can still *distribute* the task of booking tables among his friends.

As a final step, Alice can *deter* bogus bookings by requiring a credit card number with each booking. To authenticate the cards, she has to authorize a small amount on each of them before the visit.

DoS attack on TCP: SYN flooding

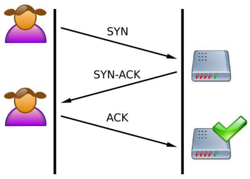


Figure: Normal 3-way handshake in TCP

DoS attack on TCP: SYN flooding

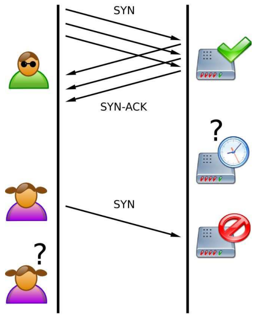


Figure: SYN flood: half open connections lock the server

Commons: publicly shared resources

For centuries, Alice, Bob and Charlie have been sharing an **open field system**.

Commons: publicly shared resources

For centuries, Alice, Bob and Charlie have been sharing an **open field system**.

Commons: publicly shared resources

In England, such open fields were called *Commons*.

Alice, Bob and Charlie alternated different crops with grazing, and maintained the land together.

Commons: publicly shared resources

In England, such open fields were called *Commons*.

Alice, Bob and Charlie alternated different crops with grazing, and maintained the land together.

Two remarkable social processes ensued:

- Tragedy of the Commons, and
- Enclosure Movement

Tragedy of the Commons

Charlie realized that it was in his *rational* interest to invest

- all effort into exploiting the public resource, and
- no effort into maintaining it.

Charlie became a *free rider*.

Tragedy of the Commons

Security 2: Resource Security
Dusko Pavlovic
Authorization
Security models
Availability
Denial of Service
Free-riding
Enclosure
Summary

Charlie realized that it was in his *rational* interest to invest

- all effort into exploiting the public resource, and
- no effort into maintaining it.

Charlie became a *free rider*.

Alice and Bob realized that it was in their *rational* interest

- to stop maintaining the resource for Charlie, and
- to hurry to exploit the resource too.

Tragedy of the Commons

Security 2: Resource Security
Dusko Pavlovic
Authorization
Security models
Availability
Denial of Service
Free-riding
Enclosure
Summary

Charlie realized that it was in his *rational* interest to invest

- all effort into exploiting the public resource, and
- no effort into maintaining it.

Charlie became a *free rider*.

Alice and Bob realized that it was in their *rational* interest

- to stop maintaining the resource for Charlie, and
- to hurry to exploit the resource too.

A *race to the bottom* ensued. The resource got depleted.

Tragedy of the Commons

Security 2: Resource Security
Dusko Pavlovic
Authorization
Security models
Availability
Denial of Service
Free-riding
Enclosure
Summary

Unrestricted access to a resource causes the race to the bottom.

Tragedy of the Commons

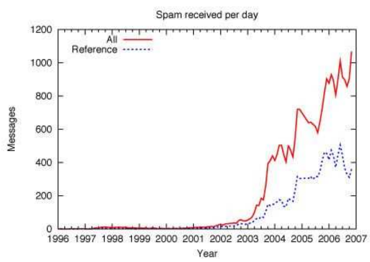
Security 2: Resource Security
Dusko Pavlovic
Authorization
Security models
Availability
Denial of Service
Free-riding
Enclosure
Summary

Fair sharing of public resources is a security problem.

Tragedy of the Commons

Security 2: Resource Security
Dusko Pavlovic
Authorization
Security models
Availability
Denial of Service
Free-riding
Enclosure
Summary

The Internet is a common resource.
Spam is a symptom of the Tragedy of the Commons.



Security policies

Security policies are both technical and political tools.

Security policies

Security 2: Resource Security
Dusko Pavlovic
Authorization
Security models
Availability
Denial of Service
Free-riding
Enclosure
Summary

Security policies are both technical and political tools.

They regulate computation and social life,
as processes of sharing and distributing resources.

Regulation

Security 2: Resource Security
Dusko Pavlovic
Authorization
Security models
Availability
Denial of Service
Free-riding
Enclosure
Summary

Charlie the free-rider drew more value out of the land,
and *enclosed* it, away from Alice and Bob.

Regulation

Security 2: Resource Security
Dusko Pavlovic
Authorization
Security models
Availability
Denial of Service
Free-riding
Enclosure
Summary

Charlie the free-rider drew more value out of the land,
and *enclosed* it, away from Alice and Bob.

In England, this happened in XV–XVII centuries.
(The Colleges were among the notable beneficiaries.)

Enclosure

Security 2: Resource Security
Dusko Pavlovic
Authorization
Security models
Availability
Denial of Service
Free-riding
Enclosure
Summary

The law locks up the man or woman
Who steals the goose from off the common
But leaves the greater villain loose
Who steals the common from off the goose.

The law demands that we atone
When we take things we do not own
But leaves the lords and ladies fine
Who take things that are yours and mine.

The poor and wretched donÕt escape
If they conspire the law to break;
This must be so but they endure
Those who conspire to make the law.

The law locks up the man or woman
Who steals the goose from off the common
And geese will still a common lack
Till they go and steal it back.

Anonymous, England, XVII century

Enclosure

Security 2: Resource Security
Dusko Pavlovic
Authorization
Security models
Availability
Denial of Service
Free-riding
Enclosure
Summary

Homework

Read the article *"The Second Enclosure Movement and the Construction of the Public Domain"* by James Boyle.

Discuss and contrast the possible technical and political solutions of the security problems arising around modern Commons.

Outline

Summary

- Resource security is among the oldest and the deepest layers of social structure.
  - Already microorganisms compete to secure resources.
  - The first security protocols date back to 4000 B.C. They led to the invention of money and writing.
  - Our banks, our governments and our operating systems use similar security models.

Summary

- The problems of resource security are both technical and political:
  - public availability vs private ownership,
  - the Commons vs the Enclosure.

Summary

- The problems of resource security are both technical and political:
  - public availability vs private ownership,
  - the Commons vs the Enclosure.
- Security policies are engineering problems.

Summary

- The problems of resource security are both technical and political:
  - public availability vs private ownership,
  - the Commons vs the Enclosure.
- Security policies are engineering problems.
- Security engineering is a political tool.

Summary

- The problems of resource security are both technical and political:
  - public availability vs private ownership,
  - the Commons vs the Enclosure.
- Security policies are engineering problems.
- Security engineering is a political tool. (For better or for worse.)

Summary

- The problems of resource security are both technical and political:
  - public availability vs private ownership,
  - the Commons vs the Enclosure.
- Security policies are engineering problems.
- Security engineering is a political tool.
  (For better or for worse.)

- Cryptography (the next part of the course) is much simpler ;)