Operating Systems & Security

Stefan Köpsell
(Slides [mainly] created by Andreas Pfitzmann)
Technische Universität Dresden, Faculty of Computer Science, D-01062 Dresden
Nöthnitzer Str. 46, Room 3067
Phone: +49 351 463-38272, e-mail: sk13@inf.tu-dresden.de, http://dud.inf.tu-dresden.de/

Lehrveranstaltungen im Bereich Sicherheit / technischer Datenschutz

Lehrveranstaltung	*Lehrende(r)*	*SWS*
Security & Cryptography I / II	Köpsell	2/2
Kryptographie und -analyse	Franz	2/0
Datensicherheit	Franz	2/1
Steganographie und Multimedia-Forensik	Franz	2/1
Kanalkodierung	Schönfeld	2/2
Informations- und Kodierungstheorie	Schönfeld	2/1
Einführung in das Datenschutzrecht	Wagner	2/0
Hauptseminar: Technischer Datenschutz	Clauß, Köpsell	0/2
Hauptseminar: Sicherheit in ubiquitären Systemen	Borcea-Pfitzmann	0/2
Praktikum: Kryptographie und Datensicherheit	Clauß	0/4
Praktikum: Datenschutzfreundl. Technologien im Internet	Clauß, Köpsell	0/4
Praktikum: Sicherheit in ubiquitären Systemen	Borcea-Pfitzmann	0/4
Proseminar Sicherheit in Computersystemen	Clauß	0/2
Proseminar Kryptogra. Grundlagen der Datensicherheit	Köpsell	0/2
Informatik und Gesellschaft	Köpsell	2/0

Lehrgebiete

- Mehrseitige Sicherheit, insbesondere Sicherheit durch verteilte Systeme
- Datenschutz & Datenschutzfreundliche Technologien
- Kryptographie
- Steganographie
- Multimedia-Forensik
- Informations- und Kodierungstheorie

Forschungsgebiete

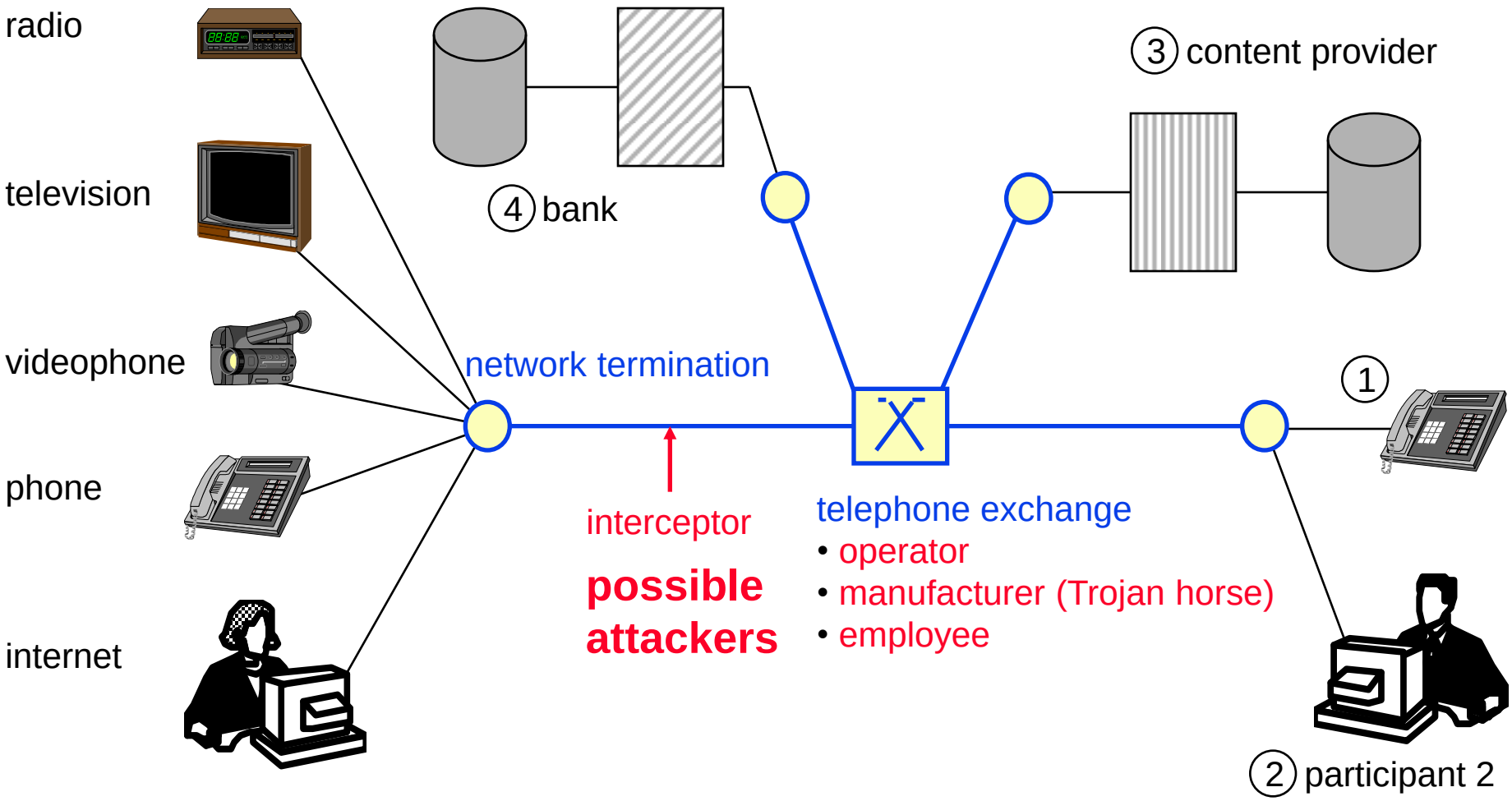
- Anonymer Webzugriff (Projekt: AN.ON)
- Identitätsmanagement (Projekte: PRIME, PrimeLife, FIDIS)
- Multimedia-Forensik (Erkennung von Bildeingabegeräten & Bildmanipulationen)
- Datenschutzgerechtes Datamining (GeneCloud)
- Sicherheit bei Network Coding
  - Teilprojekt im DFG SFB HAEC – Highly Adaptive Energy-Efficient Computing, erste Phase von 2011 - 2015
- Sicherheit & Datenschutz in Smart Grids
  - Projekt „TrueGrid“ – www.truegrid.eu

Table of Contents (1)

1 Introduction
- 1.1 What are computer networks (open distributed systems) ?
- 1.2 What does security mean?
  - 1.2.1 What has to be protected?
  - 1.2.2 Protection against whom?
  - 1.2.3 How can you provide for security?
  - 1.2.4 Protection measures – an overview
  - 1.2.5 Attacker model
- 1.3 What does security in computer networks mean?

2 Security in single computers and its limits
- 2.1 Physical security
  - 2.1.1 What can you expect – at best?
  - 2.1.2 Development of protection measures
  - 2.1.3 A negative example: Smart cards
  - 2.1.4 Reasonable assumptions on physical security
- 2.2 Protecting isolated computers against unauthorized access and computer viruses
  - 2.2.1 Identification
  - 2.2.2 Admission control
  - 2.2.3 Access control
  - 2.2.4 Limitation of the threat “computer virus” to “transitive Trojan horse”
  - 2.2.5 Remaining problems

Part of a Computer Network



example. ⑤ monitoring of patients, ⑥ transmission of moving pictures during an operation

Why are legal provisions (for security and data protection) not enough ?

Important Terms

computers interconnected by **communication network**
= **computer network** (of the first type)

computers providing switching in **communication network**
= **computer network** (of the second type)

distributed system
spatial
control and implementation structure

open system ≠ **public** system ≠ **open source** system

service integrated system

digital system

Threats and corresponding protection goals

threats:	example: medical information system	protection goals:
1) unauthorized access to information	computer company receives medical files	confidentiality
2) unauthorized modification of information	undetected change of medication	integrity ≅ partial correctness
3) unauthorized withholding of information or resources	detected failure of system	availability for authorized users

≥ total correctness

no classification, but pragmatically useful

example: unauthorized modification of a program

1)	cannot be detected, but can be prevented;	cannot be reversed
2)+3)	cannot be prevented, but can be detected;	can be reversed

Threats and corresponding protection goals

threats: — example: medical information system — protection goals:

1) unauthorized access to information
   computer company receives medical files
   → **c**onfidentiality

2) unauthorized modification of information
   undetected change of medication
   → **i**ntegrity ≅ partial correctness

3) unauthorized withholding of information or resources
   detected failure of system
   → **a**vailability for authorized users

≥ total correctness

no classification, but pragmatically useful

example: unauthorized modification of a program

1)	cannot be detected, but can be prevented;	cannot be reversed
2)+3)	cannot be prevented, but can be detected;	can be reversed

Definitions of the protection goals

confidentiality

Only authorized users get the information.

integrity

Information are correct, complete, and current
or this is detectably not the case.

availability

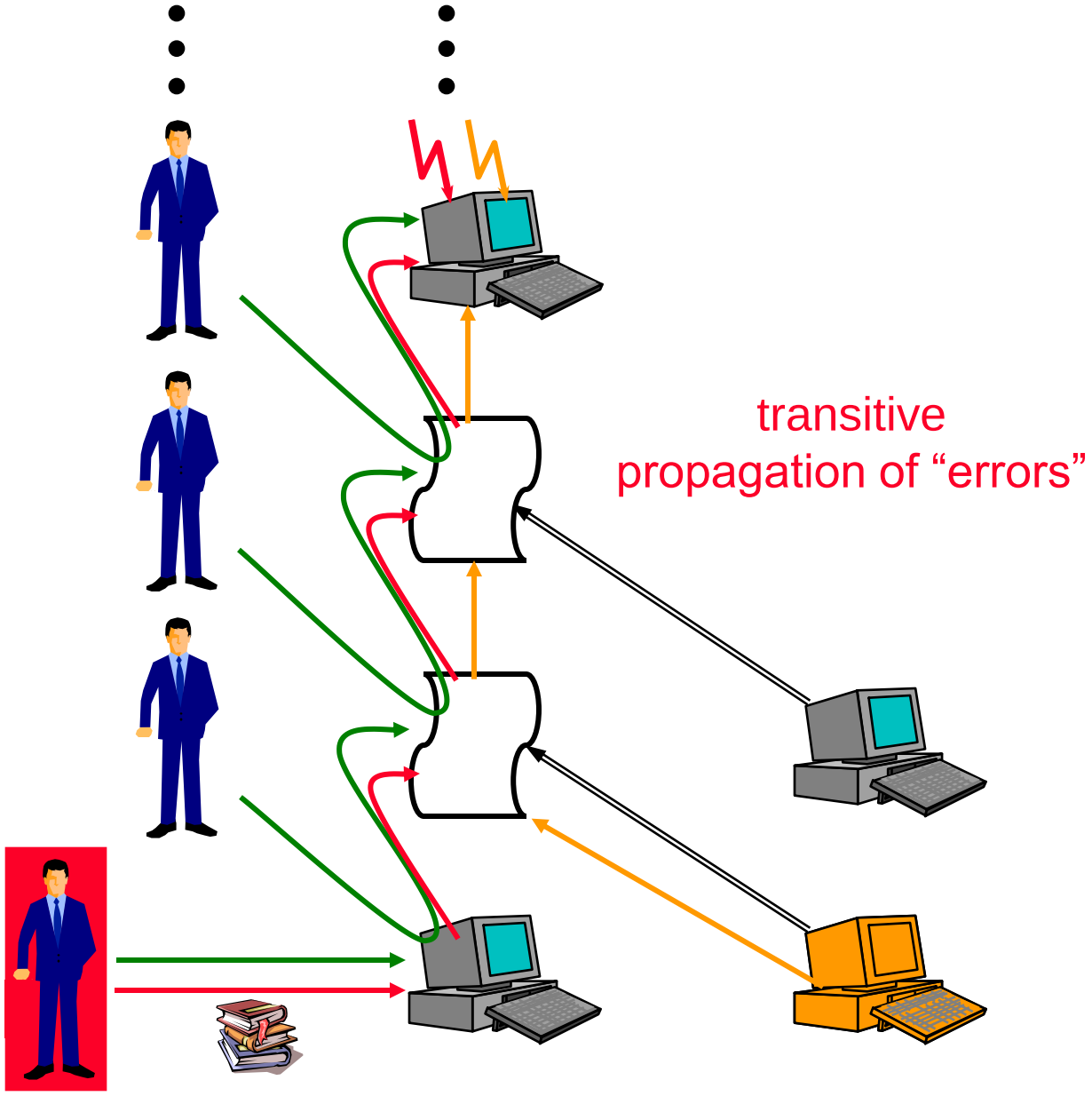
Information and resources are accessible where and
when the authorized user needs them.

- subsume: data, programs, hardware structure

- it has to be clear, who is authorized to do what in which situation

- it can only refer to the inside of a system

Transitive propagation of errors and attacks

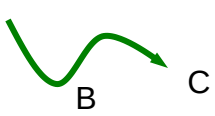


symbol explanation

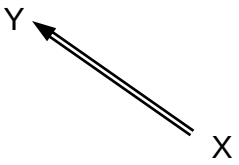
computer

program

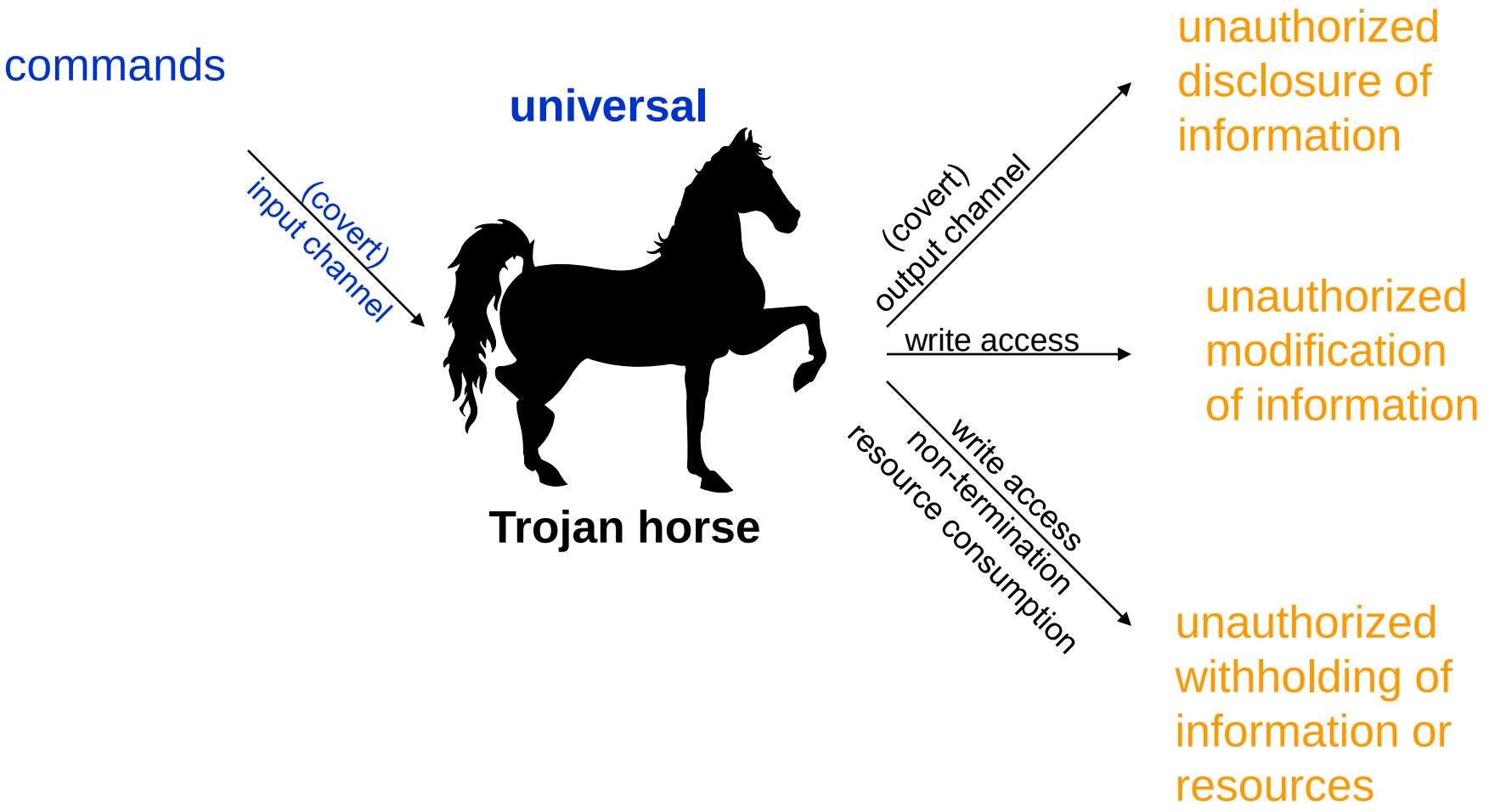
A used B to design C



machine X executes program Y



universal Trojan horse



Protection against whom ?

Laws and forces of nature

- components are growing old
- excess voltage (lightning, EMP)
- voltage loss
- flooding (storm tide, break of water pipe, heavy rain)
- change of temperature ...

fault tolerance

Human beings

- outsider
- user of the system
- operator of the system
- service and maintenance
- producer of the system
- designer of the system
- producer of the tools to design and produce
- designer of the tools to design and produce
- producer of the tools to design and produce the tools to design and produce
- designer ... includes user, operator, service and maintenance ... of the system used

Trojan horse
- universal
- transitive

Which protection measures against which attacker ?

protection concerning / protection against	**to achieve the intended**	**to prevent the unintended**
designer and producer of the tools to design and produce	intermediate languages and intermediate results, which are analyzed independently	
designer of the system	see above + several independent designers	
producer of the system	independent analysis of the product	
service and maintenance	control as if a new product, see above	
operator of the system		restrict physical access, restrict and log logical access
user of the system	physical and logical restriction of access	
outsiders	protect the system physically and protect the data cryptographically from outsiders	

Which protection measures against which attacker ?

protection concerning protection against	**to achieve the intended**	**to prevent the unintended**
designer and producer of the tools to design and produce	intermediate languages and intermediate results, which are analyzed independently	
designer of the system	see above + several independent designers	
producer of the system	independent analysis of the product	
service and maintenance	control as if a new product, see above	
operator of the system		restrict physical access, restrict and log logical access
user of the system	physical and logical restriction of access	
outsiders	protect the system physically and protect data cryptographically from outsiders	

physical distribution and redundance

confidentiality, unobservability, anonymity, unlinkability:

avoid the ability to gather “unnecessary data”

Considered maximal strength of the attacker

attacker model

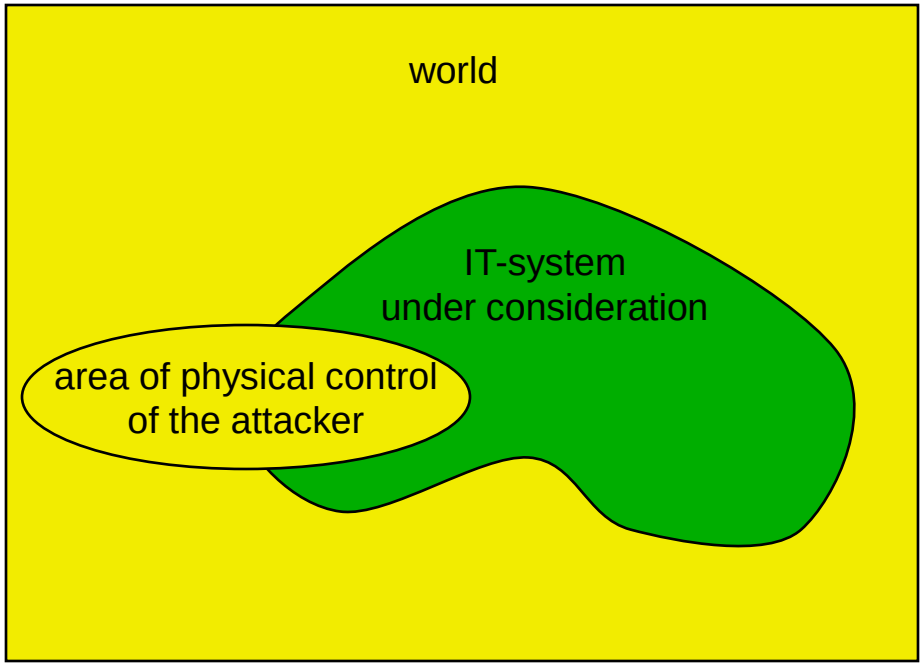
It‘s not possible to protect against an omnipotent attacker.

- roles of the attacker (outsider, user, operator, service and maintenance, producer, designer …), *also combined*
- area of physical control of the attacker
- behavior of the attacker
  - passive / active
  - observing / modifying (with regard to the agreed rules)
- stupid / intelligent
  - computing capacity:
    - not restricted: computationally unrestricted
    - restricted: computationally restricted

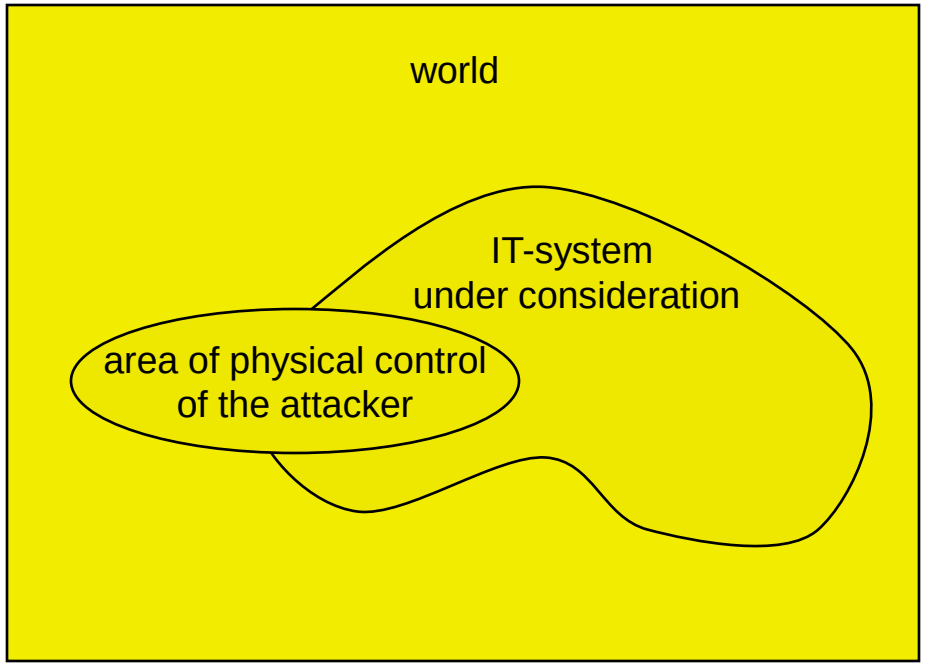
money

time

Observing vs. modifying attacker



observing attacker



modifying attacker

acting according to
the agreed rules

possibly breaking
the agreed rules

Strength of the attacker (model)

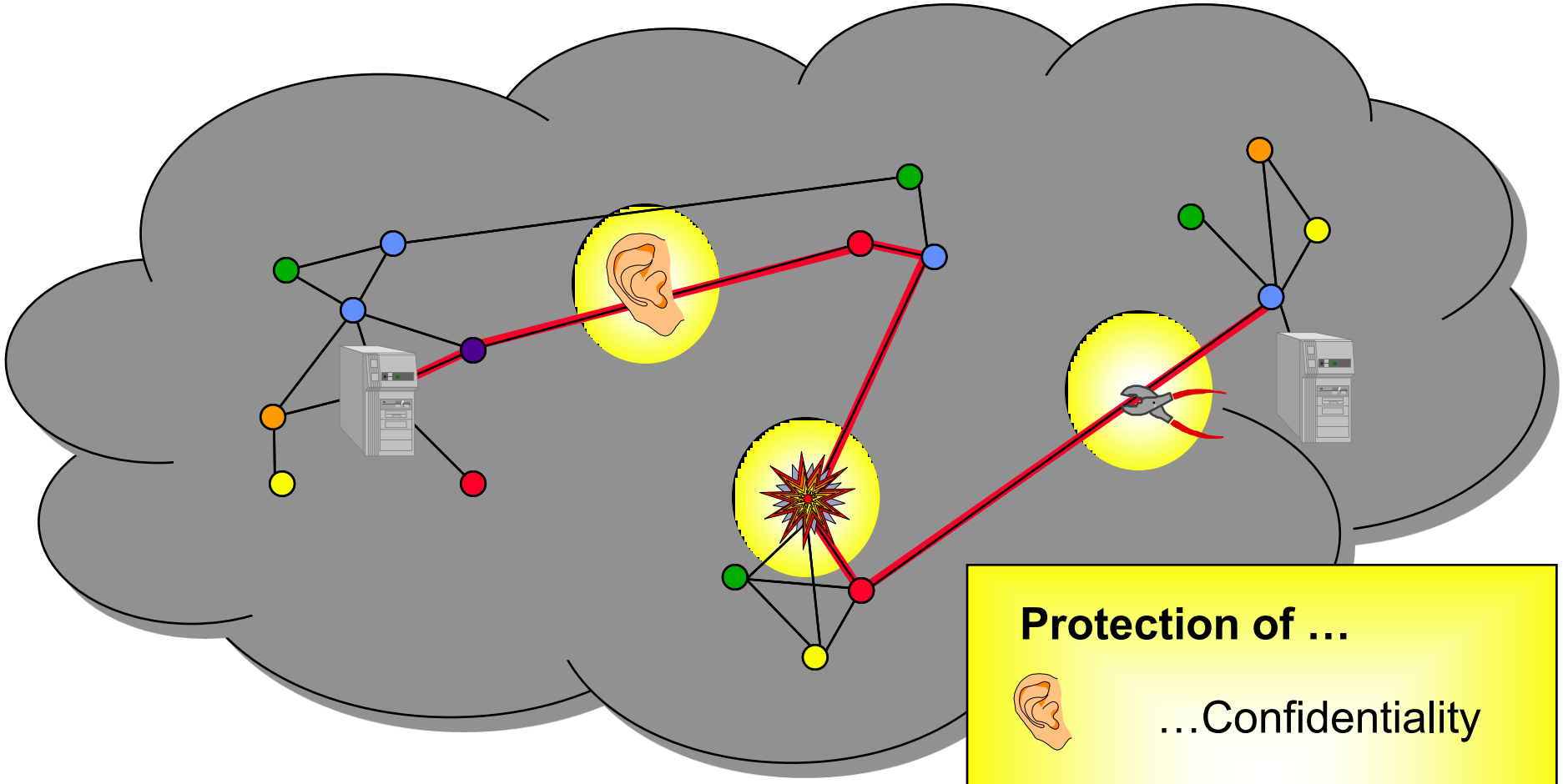
Attacker (model) *A* is stronger than attacker (model) *B*, iff *A* is stronger than *B* in at least one respect and not weaker in any other respect.

Stronger means:

- set of roles of *A* $\supset$ set of roles of *B*,
- area of physical control of *A* $\supset$ area of physical control of *B*,
- behavior of the attacker
  - active is stronger than passive
  - modifying is stronger than observing
- intelligent is stronger than stupid
  - computing capacity: not restricted is stronger than restricted
- more money means stronger
- more time means stronger

Defines partial order of attacker (models).

The Internet



Protection of …

…Confidentiality

…Integrity

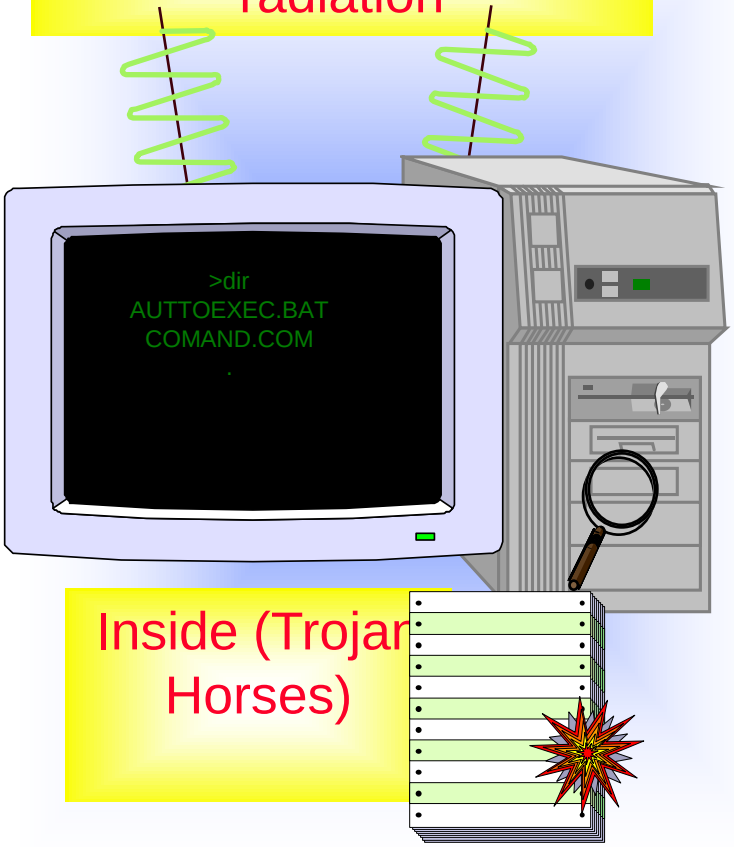
…Avaliablity

- Telecommunication networks:
  - many operators
  - many users

> “Access points”

Computer

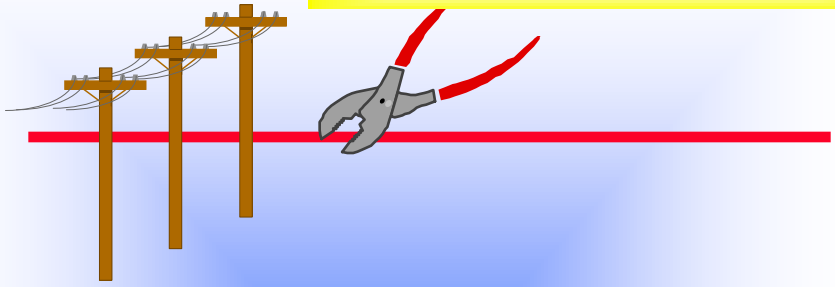
Electromagentic radiation



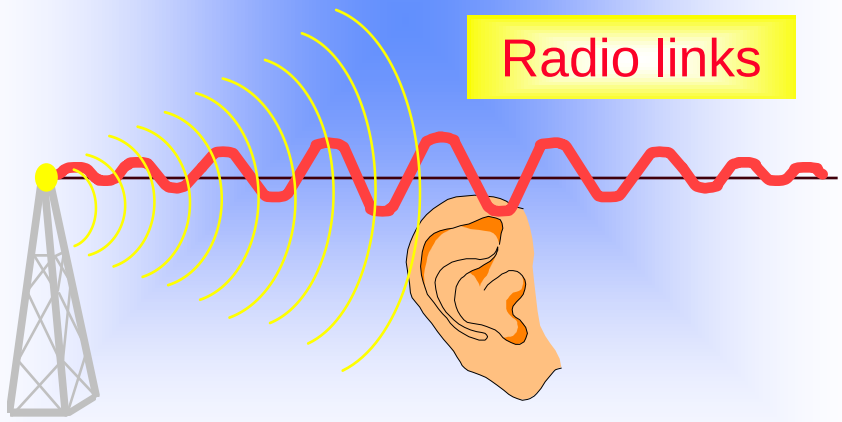
Inside (Trojan Horses)

Transmission

Copper or fiber lines



Radio links



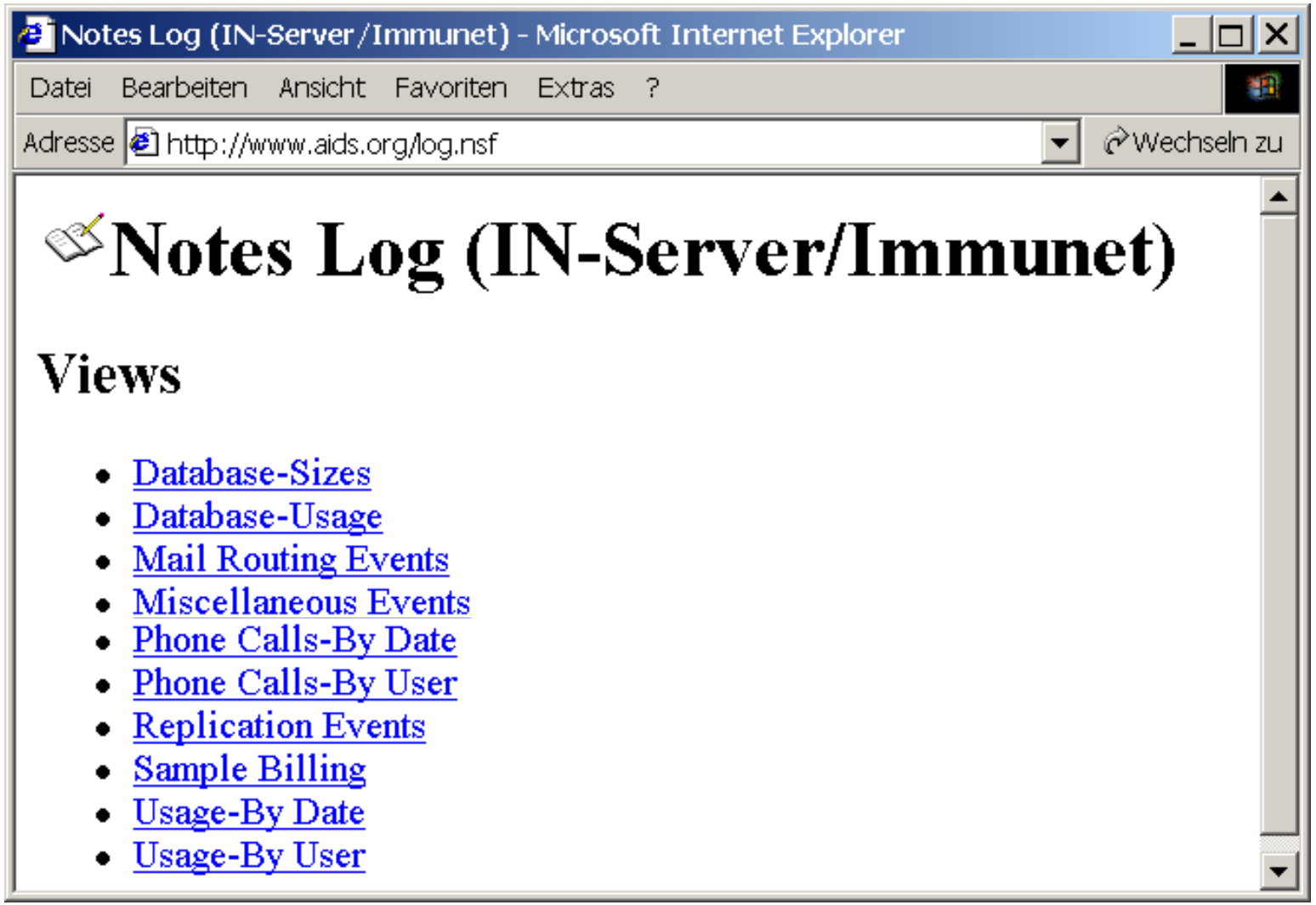
Protection against whom ?

EU-Parliament about the global interception system ECHELON:

„... there can now be no doubt that the **purpose of the system is to intercept**, at the very least, **private and commercial communications**, and not military communications,...“

„... Calls on the Commission and the Member States to **inform their citizens and firms about the possibility that their international communications may**, under certain circumstances, **be intercepted**; insists that this information should be accompanied by **practical assistance in designing and implementing comprehensive protection measures, including the security of information technology**; ...“

Why should I protect myself... ?



... because you cannot always rely on others!

Security in computer networks

confidentiality

- message content is confidential — **end-to-end encryption**
- **place** • sender / recipient anonymous — **mechanisms to protect traffic data**

integrity

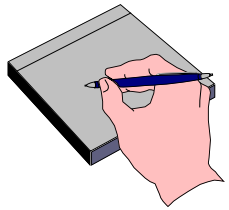
- detect forgery — **authentication system(s)**
- **time** • recipient can prove transmission — **sign messages**
- **time** • sender can prove transmission — **receipt**
- ensure payment for service — **during service by digital payment systems**

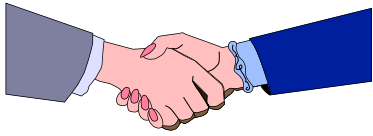
availability

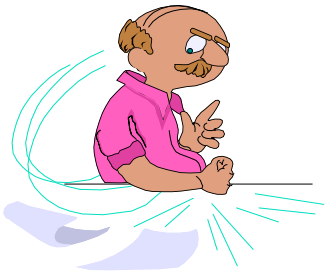
- enable communication — **diverse networks; fair sharing of resources**

Multilateral security

- Each party has its particular protection goals.
- Each party can formulate its protection goals.
- Security conflicts are recognized and compromises negotiated.
- Each party can enforce its protection goals within the agreed compromise.



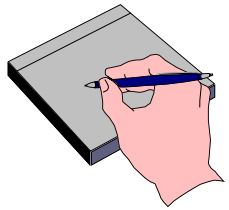


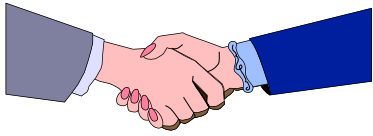


Security with minimal assumptions about others

Multilateral security (2nd version)

- Each party has its particular goals.
- Each party can formulate its protection goals.
- Security conflicts are recognized and compromises negotiated.
- Each party can enforce its protection goals within the agreed compromise.



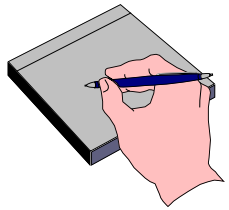


Security with minimal assumptions about others

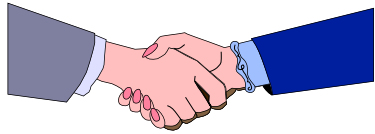
Multilateral security (3rd version)

- Each party has its particular goals.

- Each party can formulate its protection goals.



- Security conflicts are recognized and compromises negotiated.



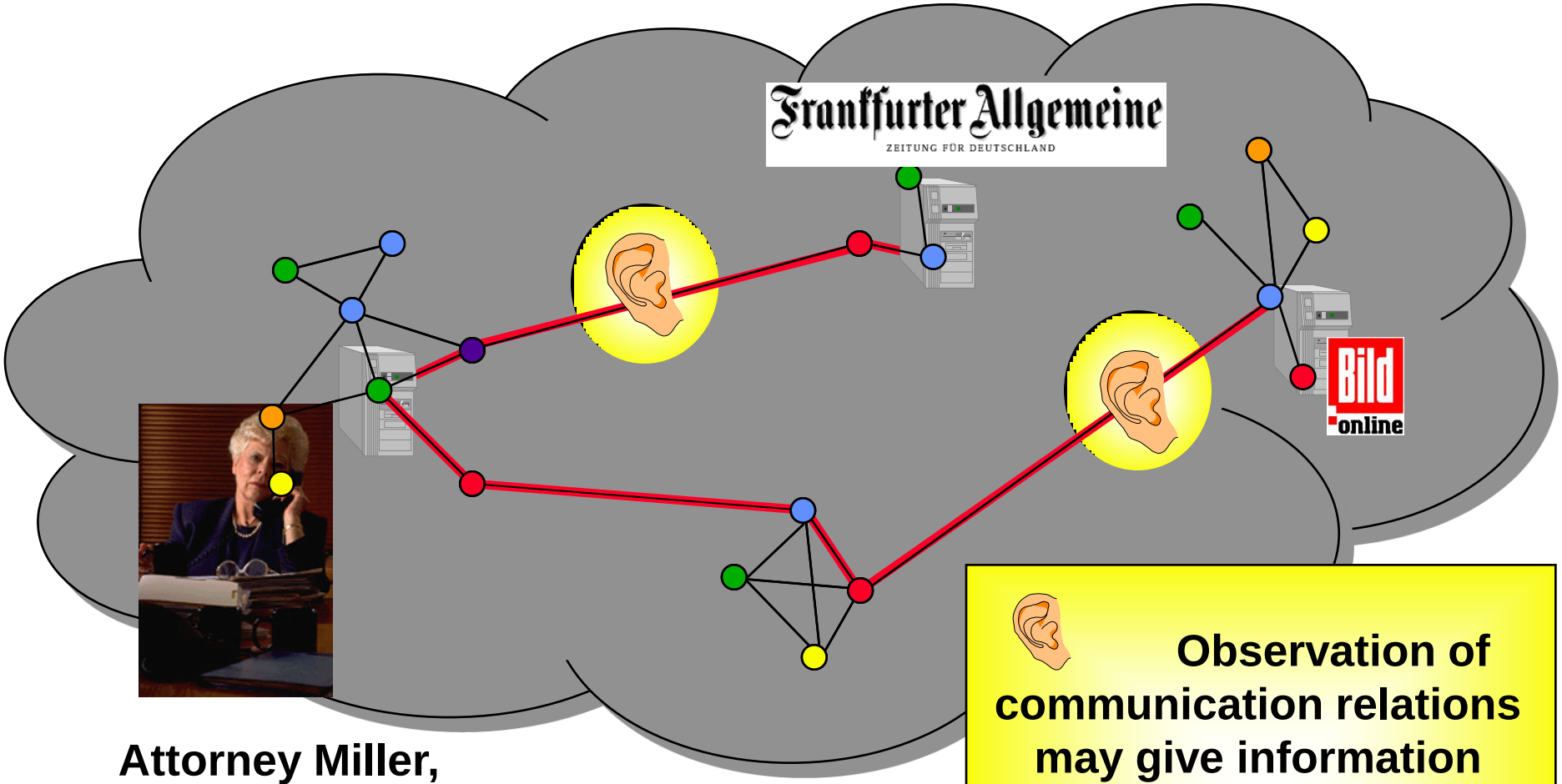
- Each party can enforce its protection goals within the agreed compromise. As far as limitations of this cannot be avoided, they equally apply to all parties.

Security with minimal assumptions about others

Protection Goals: Sorting

	Content	Circumstances
Prevent the unintended	Confidentiality Hiding	Anonymity Unobservability
Achieve the intended	Integrity	Accountability
	Availability	Reachability Legal Enforceability

> Why encryption is not enough



Attorney Miller, specialized in mergers

Observation of communication relations may give information about contents

Protection Goals: Definitions

Confidentiality ensures that nobody apart from the communicants can discover the content of the communication.

Hiding ensures the confidentiality of the transfer of confidential user data. This means that nobody apart from the communicants can discover the existence of confidential communication.

Anonymity ensures that a user can use a resource or service without disclosing his/her identity. Not even the communicants can discover the identity of each other.

Unobservability ensures that a user can use a resource or service without others being able to observe that the resource or service is being used. Parties not involved in the communication can observe neither the sending nor the receiving of messages.

Integrity ensures that modifications of communicated content (including the sender's name, if one is provided) are detected by the recipient(s).

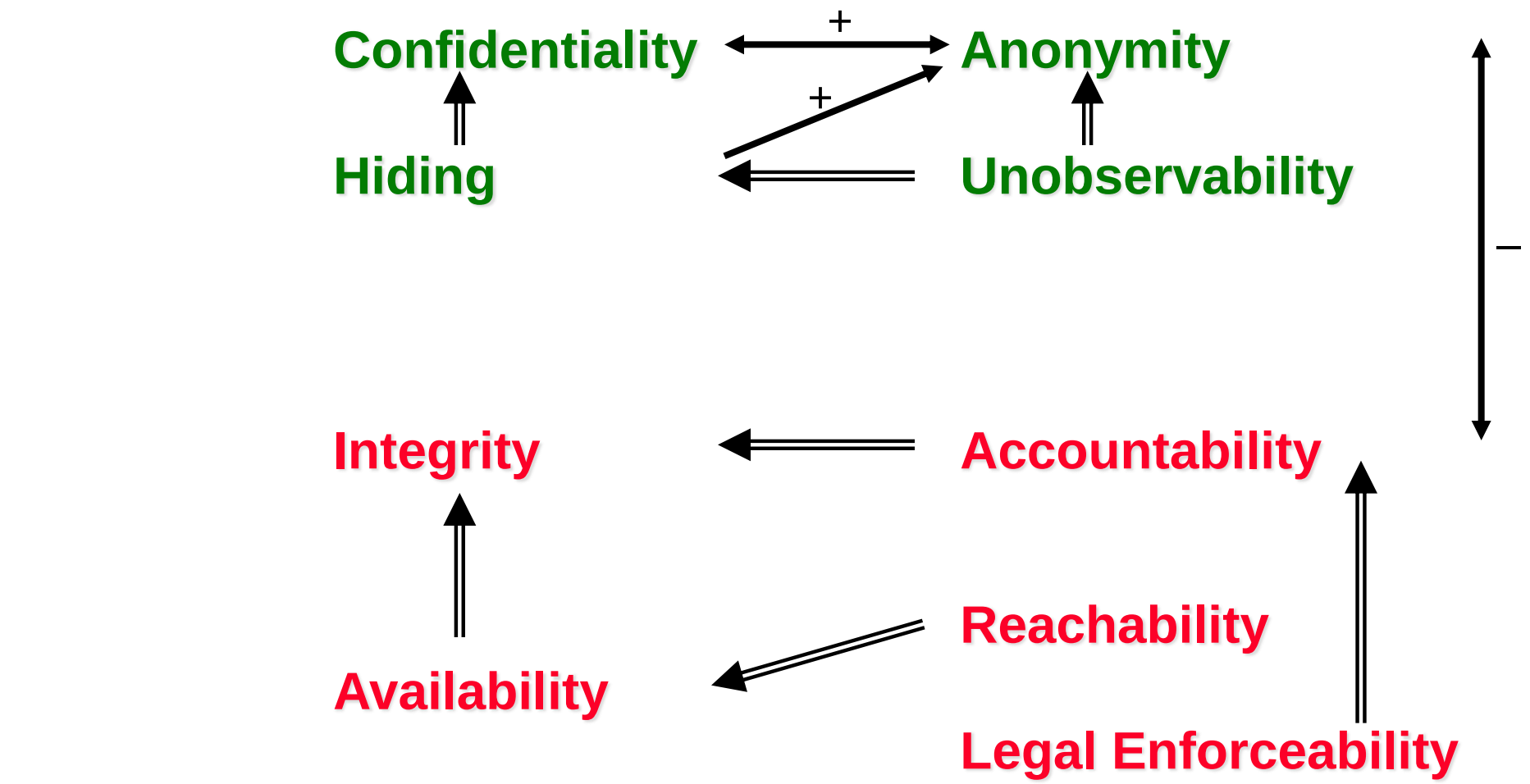
Accountability ensures that sender and recipients of information cannot successfully deny having sent or received the information. This means that communication takes place in a provable way.

Availability ensures that communicated messages are available when the user wants to use them.

Reachability ensures that a peer entity (user, machine, etc.) either can or cannot be contacted depending on user interests.

Legal enforceability ensures that a user can be held liable to fulfill his/her legal responsibilities within a reasonable period of time.

Correlations between protection goals

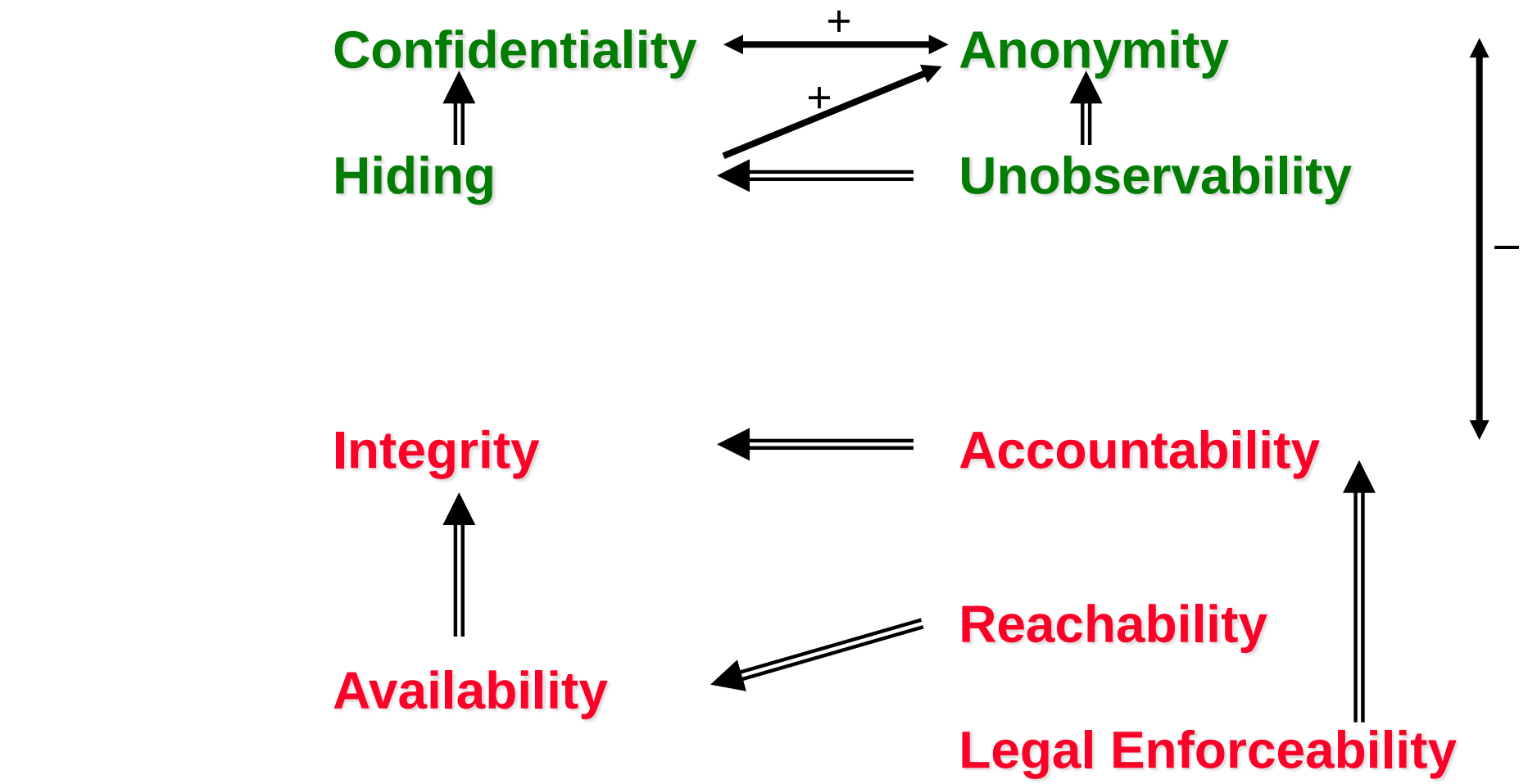


implies

+ strengthens

– weakens

Correlations between protection goals



Transitive closure to be added

implies

+ strengthens

– weakens

Physical security assumptions

Each technical security measure needs a physical “anchoring” in a part of the system which the attacker has neither read access nor modifying access to.

Range from “computer centre X” to “smart card Y”

What can be expected at best ?

Availability of a locally concentrated part of the system cannot be provided against *realistic* attackers

→ **physically distributed system**

… hope the attacker cannot be at many places at the same time.

Distribution makes **confidentiality** and **integrity** more difficult**.** But physical measures concerning confidentiality and integrity are more efficient: Protection against *all realistic* attackers seems feasible. If so, physical distribution is quite ok.

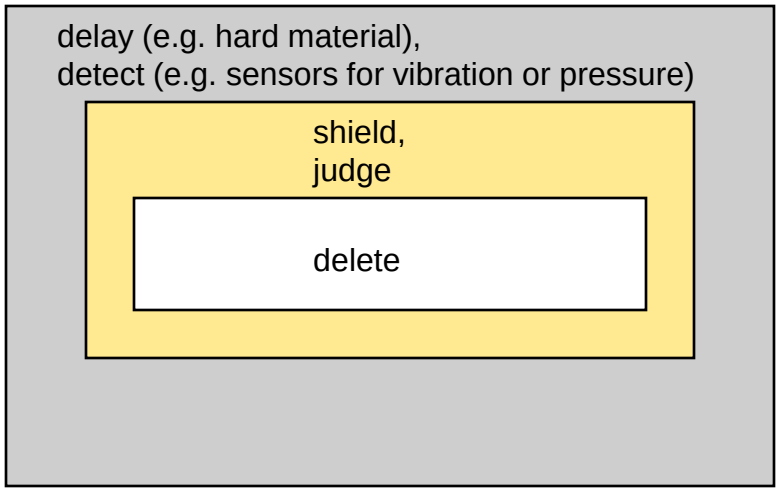
Tamper-resistant casings

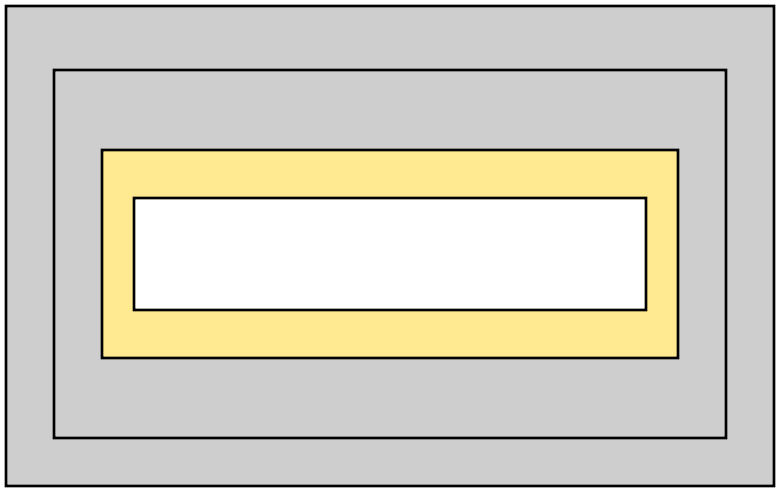
Interference: detect
judge

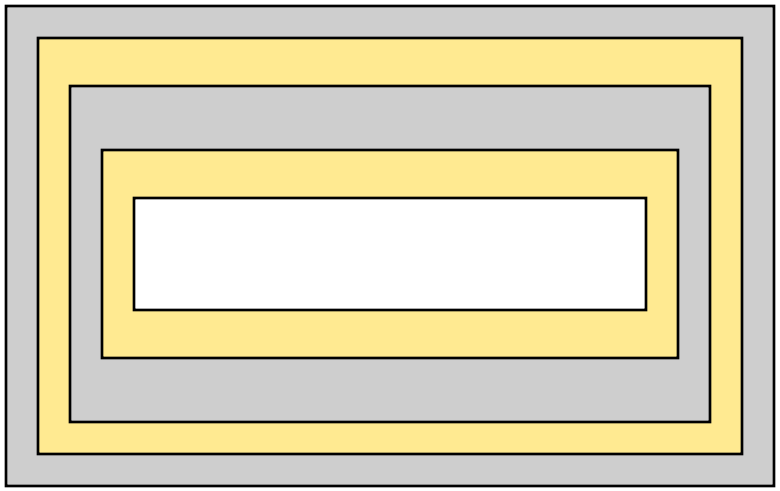
Attack: delay
delete data (etc.)

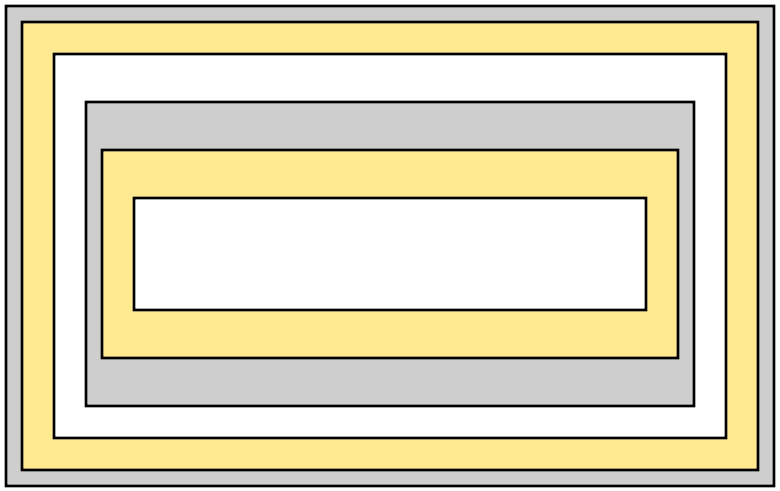
Possibility: several layers, shielding 

Shell-shaped arrangement of the five basic functions









Tamper-resistant casings

Interference: detect
judge

Attack: delay
delete data (etc.)

Possibility: several layers, shielding

Problem: validation ... credibility

Negative example: smart cards

- no detection (battery missing etc.)
- shielding difficult (card is thin and flexible)
- no deletion of data intended, even when power supplied

Golden rule

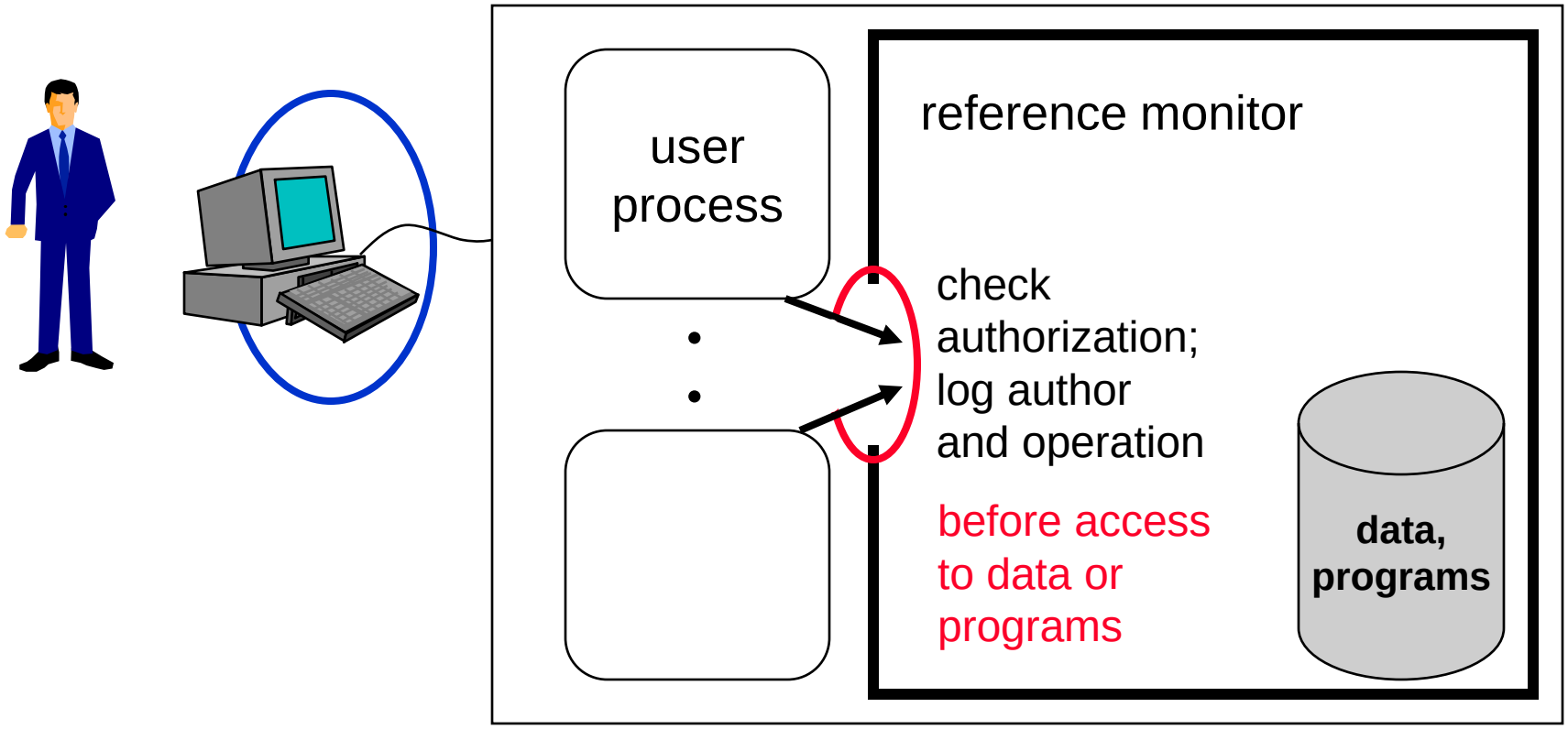
Correspondence between organizational and IT structures

AUTHENTICATION

Slides are from Andreas Pfitzmann, Andreas Westfeld, Sebastian Clauß, Mike Bergmann and myself (Stefan Köpsell)

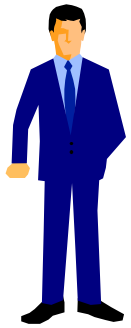
Lookahed: Why autentication: Admission and access control

Admission control communicate with authorized partners only



Access control subject can only exercise operations on objects if authorized.

Identification of human beings by IT-systems



What one *is*
- hand geometry
- finger print
- picture
- hand-written signature
- retina-pattern
- voice
- typing characteristics

has
- paper document
- metal key
- magnetic-strip card
- smart card (chip card)
- calculator

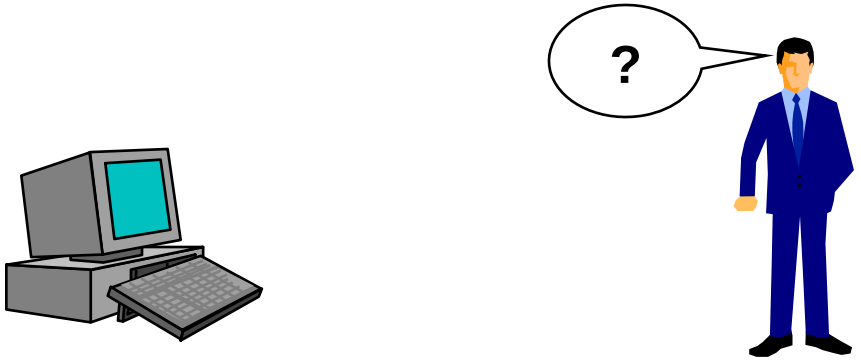
knows
- password, passphrase
- answers to questions
- calculation results for numbers

→ **eID-card**

New German eID Card

PIN protects access to chip

Identification of IT-systems by human beings

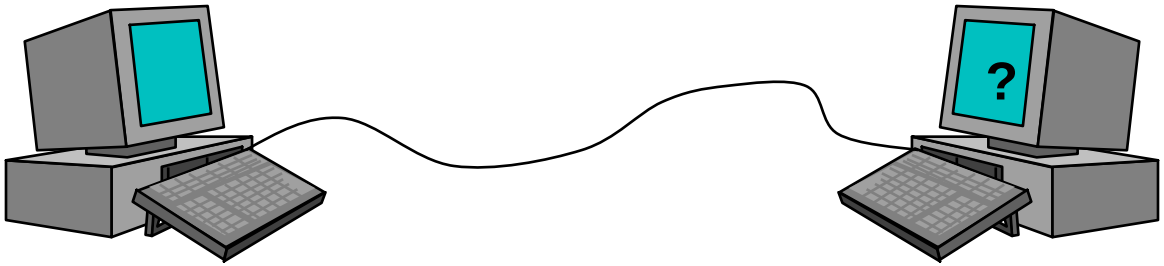


What it *is*
- casing
- seal, hologram
- pollution

knows
- password
- answers to questions
- calculation results for numbers

Where it *stands*

Identification of IT-systems by IT-systems



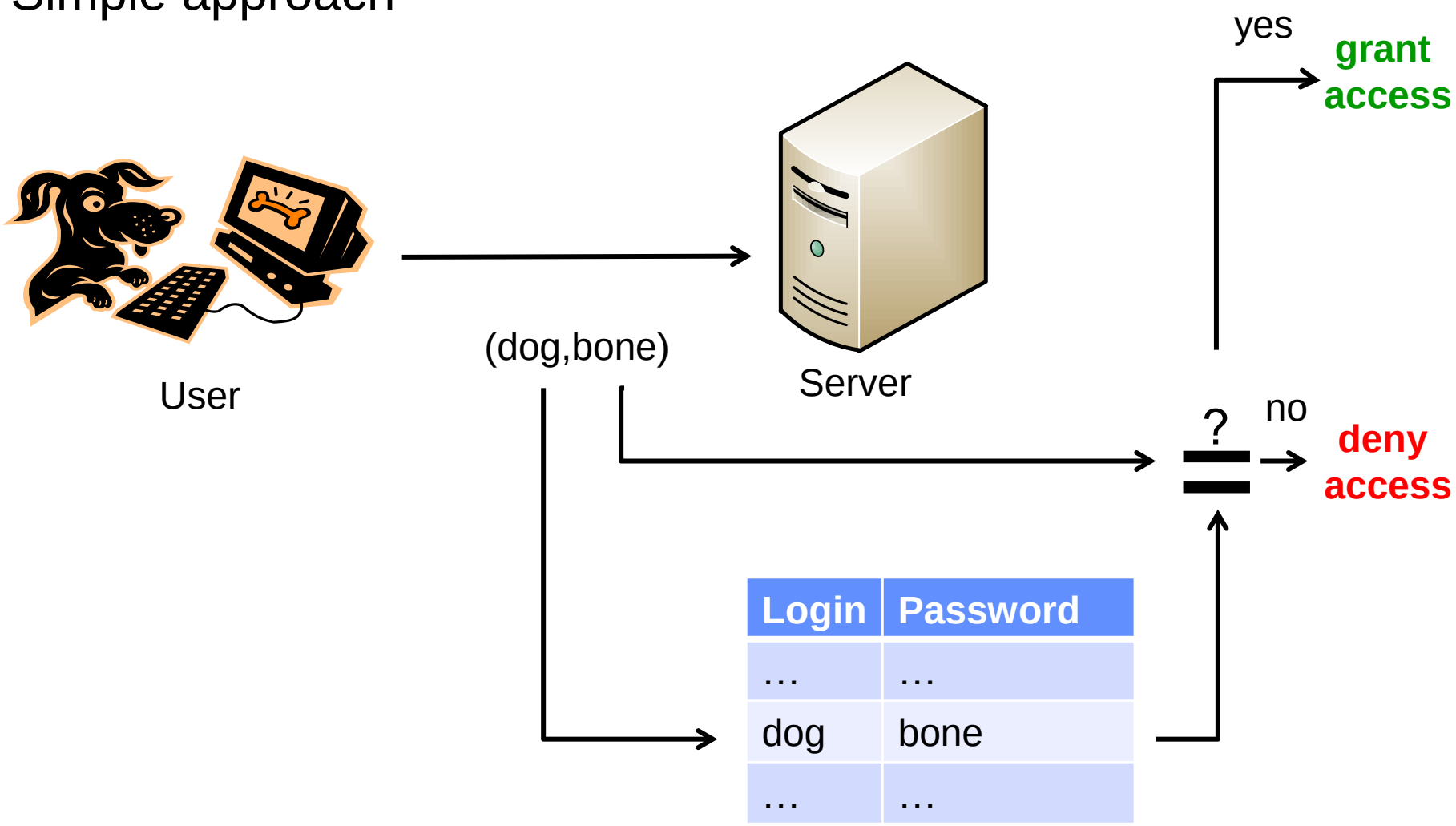
What it *knows*

- password
- answers to questions
- calculation results for numbers
- **cryptography**

Wiring *from where*

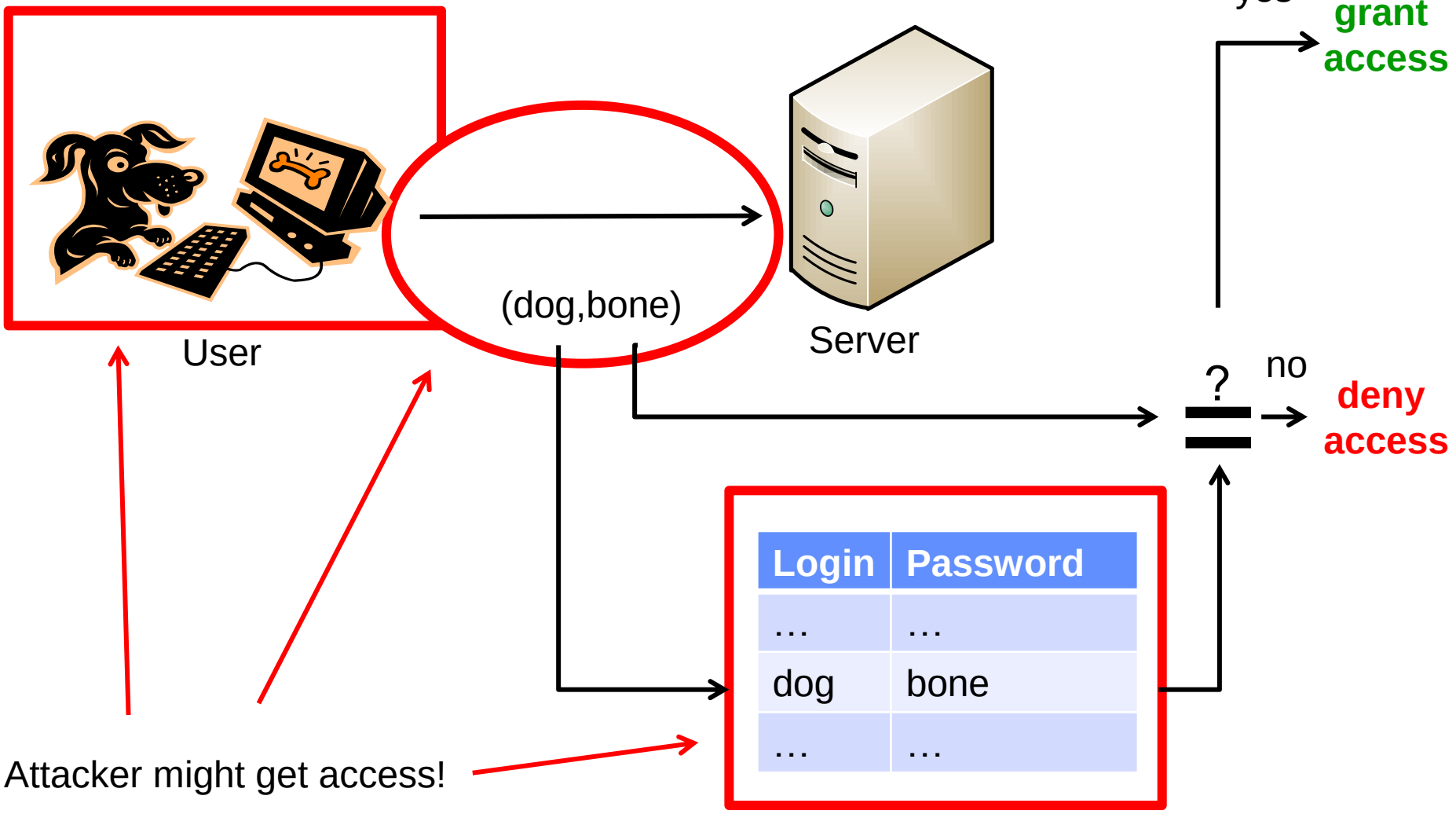
Password based authentication

- Simple approach



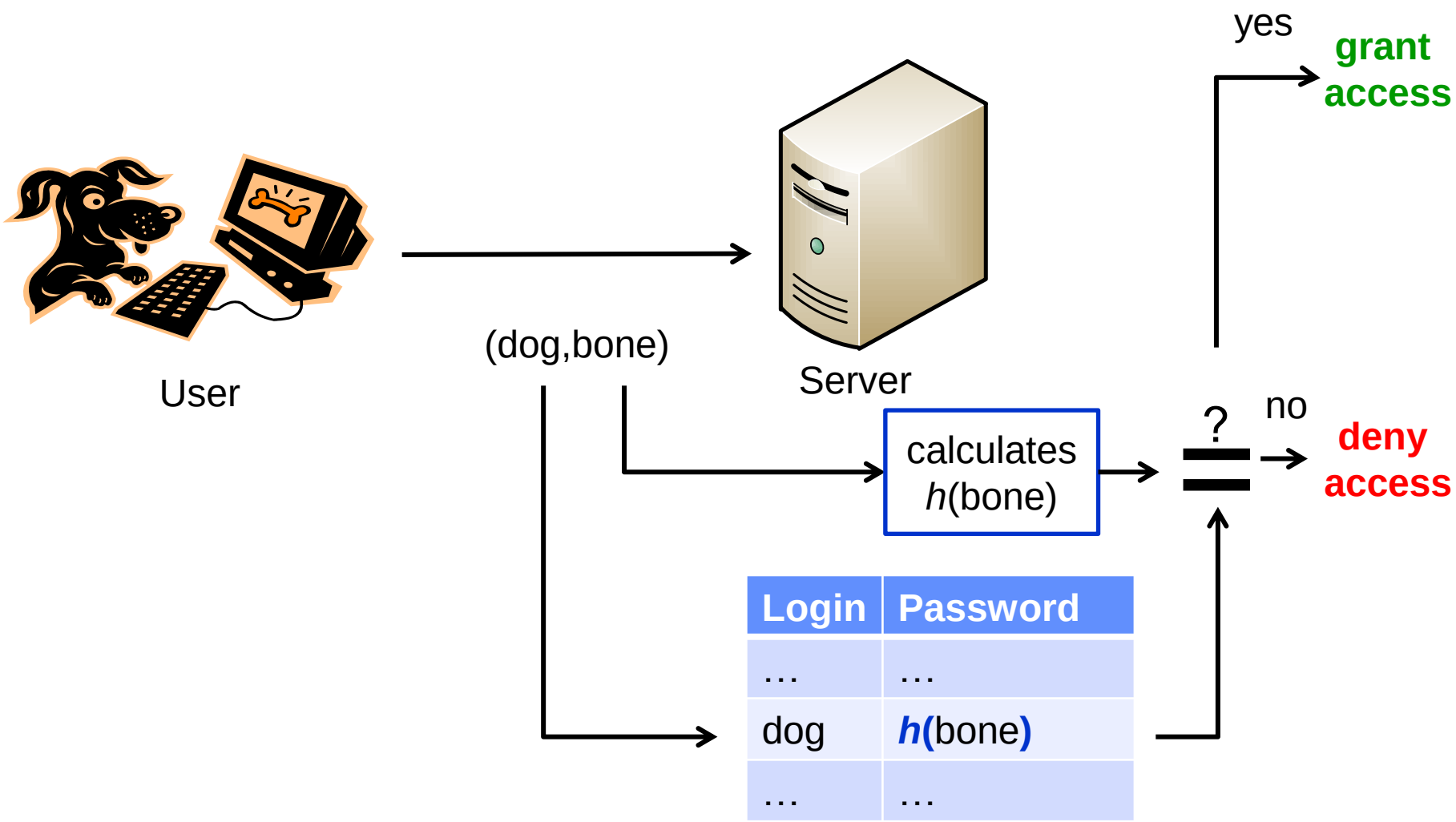
Password based authentication

- Simple approach – **security problems**



Password based authentication

- Enhanced approach using one way (hash) functions



One-way functions – cryptographic hash functions

- One-way function *f:*
  - calculating $f(x)=y$ is easy
  - calculating $f^{-1}(y)=x$ is hard
    - computation / storage
  - open question: Do one-way functions exist?
- Cryptographic hash function *h*
  - might have different properties depending on the use case
  - collision resistance:
    - it is hard to find $h(y)=h(x)$ with $y \neq x$
    - note: *h* is usually not *collision free*, because $|h(x)| \ll |x|$
  - preimage resistance / one-way function / secrecy
    - given $h(x)$ it is hard to find x
  - second-preimage resistance / weak collision resistance / binding
    - given $h(x)$ it is hard to find $h(y)=h(x)$ with $y \neq x$
  - Note:
    - *h* is not necessarily a “random extractor”
    - only one of “secrecy” and “binding” can be information theoretic secure

Examples for cryptographic hash functions

- MD5
  - Message-Digest Algorithm
  - developed by Ronald Rivest (April 1992)
  - produces 128 bit hash values
  - can process arbitrary long inputs
  - **today MD5 is broken!**
- SHA-1
  - Secure Hash Standard
  - published 1993 as FIPS PUB 180 by US NIST
  - produces 160 bit hash values
  - **today SHA-1 is insecure!**
- SHA-2
  - set of hash functions, with hash values of 224, 256, 384, 512 bit
  - published 2001 as FIPS PUB 180-2 by NIST
  - **SHA-2 hash functions are believed to be secure**
- SHA-3
  - will be the result of the NIST Cryptographic Hash Algorithm Competition started November 2007
  - 3 selection rounds, 5 finalists
  - result expected late 2012

MD5 Hash in the Wild

- United States Cyber Command (USCYBERCOM)
  - mission statement:= "*USCYBERCOM plans, coordinates, integrates, synchronizes and conducts activities to: direct the operations and defense of specified Department of Defense information networks and; prepare to, and when directed, conduct full spectrum military cyberspace operations in order to enable actions in all domains, ensure US/Allied freedom of action in cyberspace and deny the same to our adversaries.*"

MD5 Hash in the Wild

mission statement:= "*USCYBERCOM plans, coordinates, integrates, synchronizes and conducts activities to: direct the operations and defense of specified Department of Defense information networks and; prepare to, and when directed, conduct full spectrum military cyberspace operations in order to enable actions in all domains, ensure US/Allied freedom of action in cyber-space and deny the same to our adversaries.*"

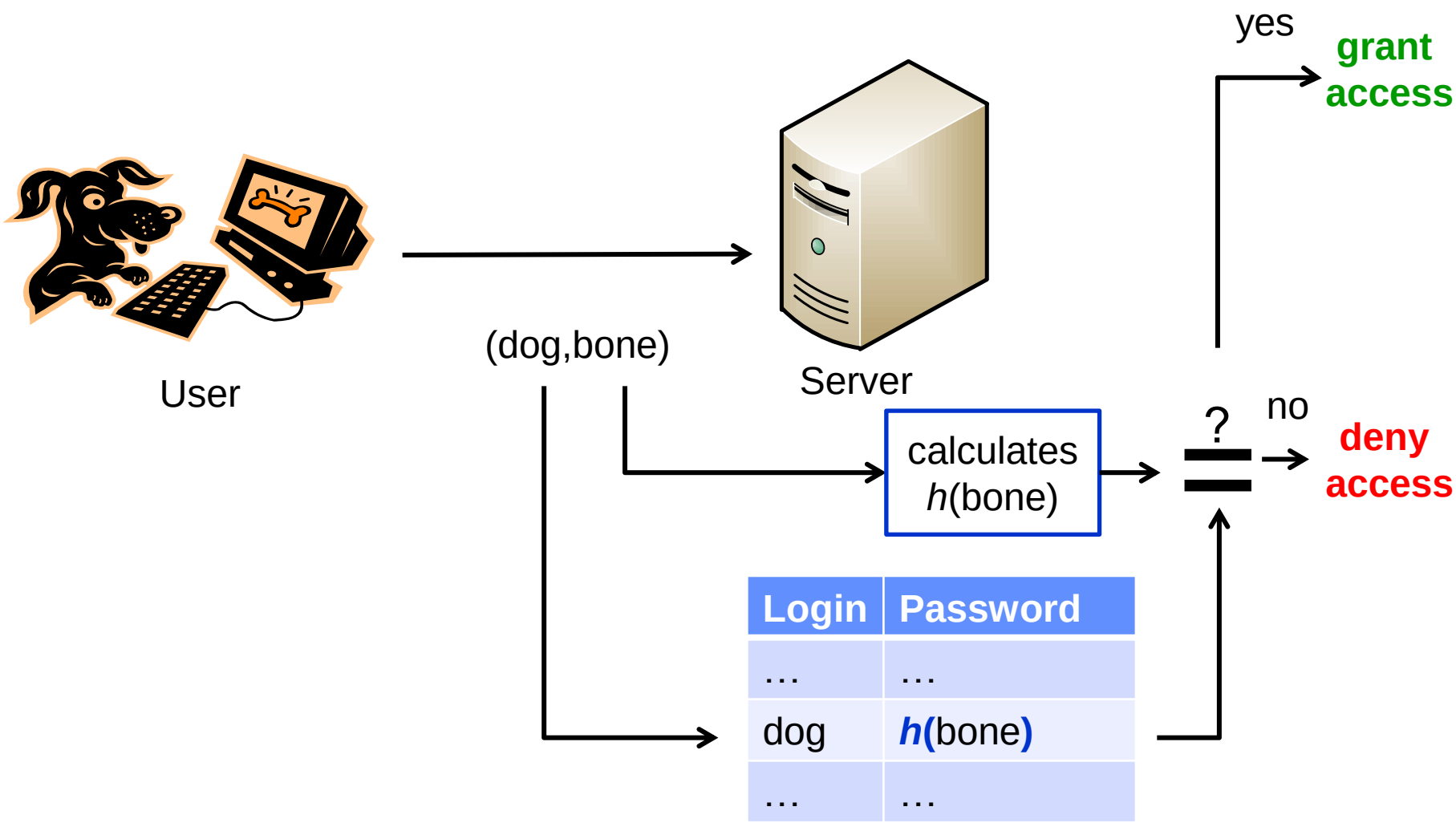
mission statement:= "*USCYBERCOM plans, coordinates, integrates, synchronizes and conducts activities to: direct the operations and defense of specified Department of Defense information networks and; prepare to, and when directed, conduct full spectrum military cyberspace operations in order to enable actions in all domains, ensure US/Allied freedom of action in cyber-space and deny the same to our adversaries.*"

MD5(mission statement)=
9ec4c12949a4f31474f299058ce2b22a

(Remember: MD5 is broken → find other interesting mission statements…)

Password based authentication

- Enhanced approach using one way (hash) functions

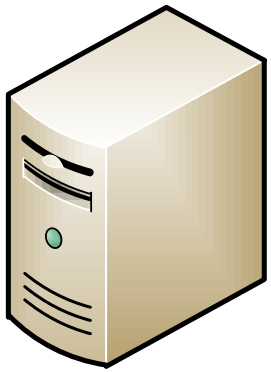


Password based authentication

- Enhanced approach using one way (hash) functions

User

(dog,bone)



Server

yes

grant access

calculates *h*(bone)

?

no

deny access

Login	Password
…	…
dog	*h*(bone)
…	…

Slightly reduced risk, if attacker gets access.

Remaining problems of password based authentication based one way functions

- Brute Force attack
  - function *h*() is public
  - value of *h*(*x*) is known to the attacker

  ➔ try all possible values for *x*

  Considerations:
    - usually >> 1 Mio. *h*(x)/s on ordinary hardware
    - assumption: password uses only small letters
    - password length = 8

Login	Password
…	…
dog	*h*(bone)
…	…

time needed: $\frac{26^8}{1\,000\,000 \cdot 60 \cdot 60} \approx 58\text{h}$

- first countermeasures:
  - limit false attempts
- first password rules:
  - use a large alphabet (small and capitalised letters, numbers, specials)
  - use a long password

Remaining problems of password based authentication based one way functions

- first password rules:
  - use a large alphabet
    - (small, capitalised letters, numbers, specials)
    - time needed: $\frac{(26+26+10+30)^8}{1\,000\,000 \cdot 60 \cdot 60 \cdot 24 \cdot 365.25} \approx 162a$
  - use a long password

Login	Password
…	…
dog	*h*(bone)
…	…

- remaining possible attacks:
  - increase in computation power
    - distributed approach
    - GPU
    - Moore's law
  - pre-computation:
    - attacker creates lockup table
    - search time (example above):

      $\mathrm{ld}\left((26+26+10+30)^8\right) < 53$ comparisons

Remaining problems of password based authentication based one way functions

- remaining possible attack:
  - pre-computation:
    - attacker creates lockup table
    - search time (example above):

      $\mathrm{ld}\left((26+26+10+30)^8\right)<53$ comparisons
    - problem: storage space
    - Example:
      - size of MD5 hash: 128 bit
      - number of hashes: $(26+26+10+30)^8$=5 132 188 731 375 616
      - storage
        - space: 5 132 188 731 375 616 · 128 bit ≈ 75 000 TByte
        - price: ≈ 2 Mio €
    - solution: storage space / computation tradeoff
      - Martin E. Hellman: “A Cryptanalytic Time – Memory Trade-Off”

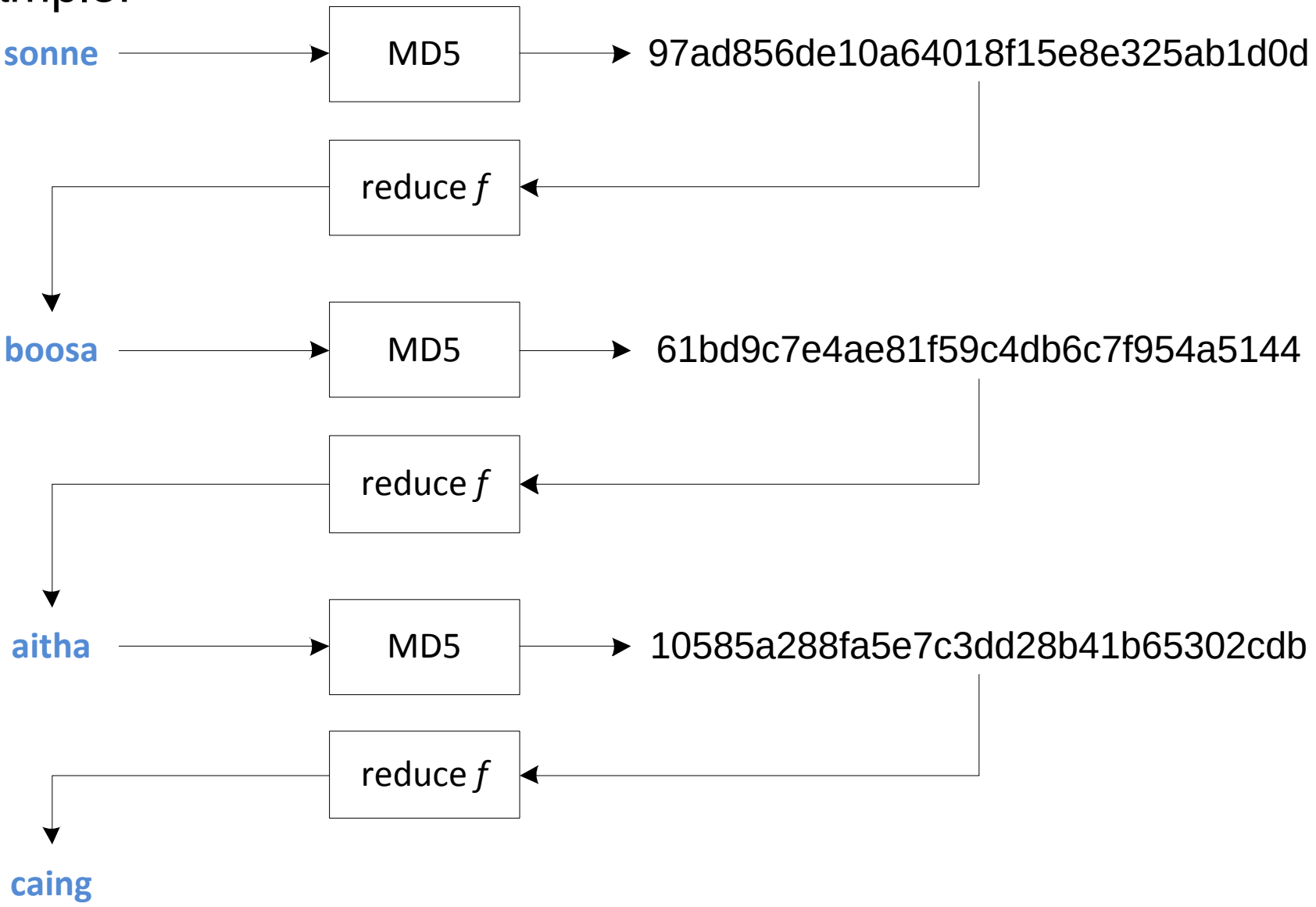
Login	Password
…	…
dog	*h*(bone)
…	…

Cryptanalytic Time – Memory Trade-Off

- main idea:
  - store only certain parts of the lookup table
  - regenerate the missing parts on demand
- requires “reduce” function *f*
  - *f*: H → P (H: set of hash values, P: set of passwords)
  - note: *f* is NOT the inverse of *h*
- general procedure:
  - calculate a chain of hash and reduce function calls
    - p → h() → f() → h() → f() → h () … → f() → p’
  - store first and last value in a table
    - sort by the last value
  - length of chain influences Time – Memory trade-off

Cryptanalytic Time – Memory Trade-Off

- Example:



Cryptanalytic Time – Memory Trade-Off

- 2nd example
  - breaking of PINs
  - *h*(x):= (x·7807) mod 16157
  - *f*(x) := x mod 9000 + 1000

- PIN-table:

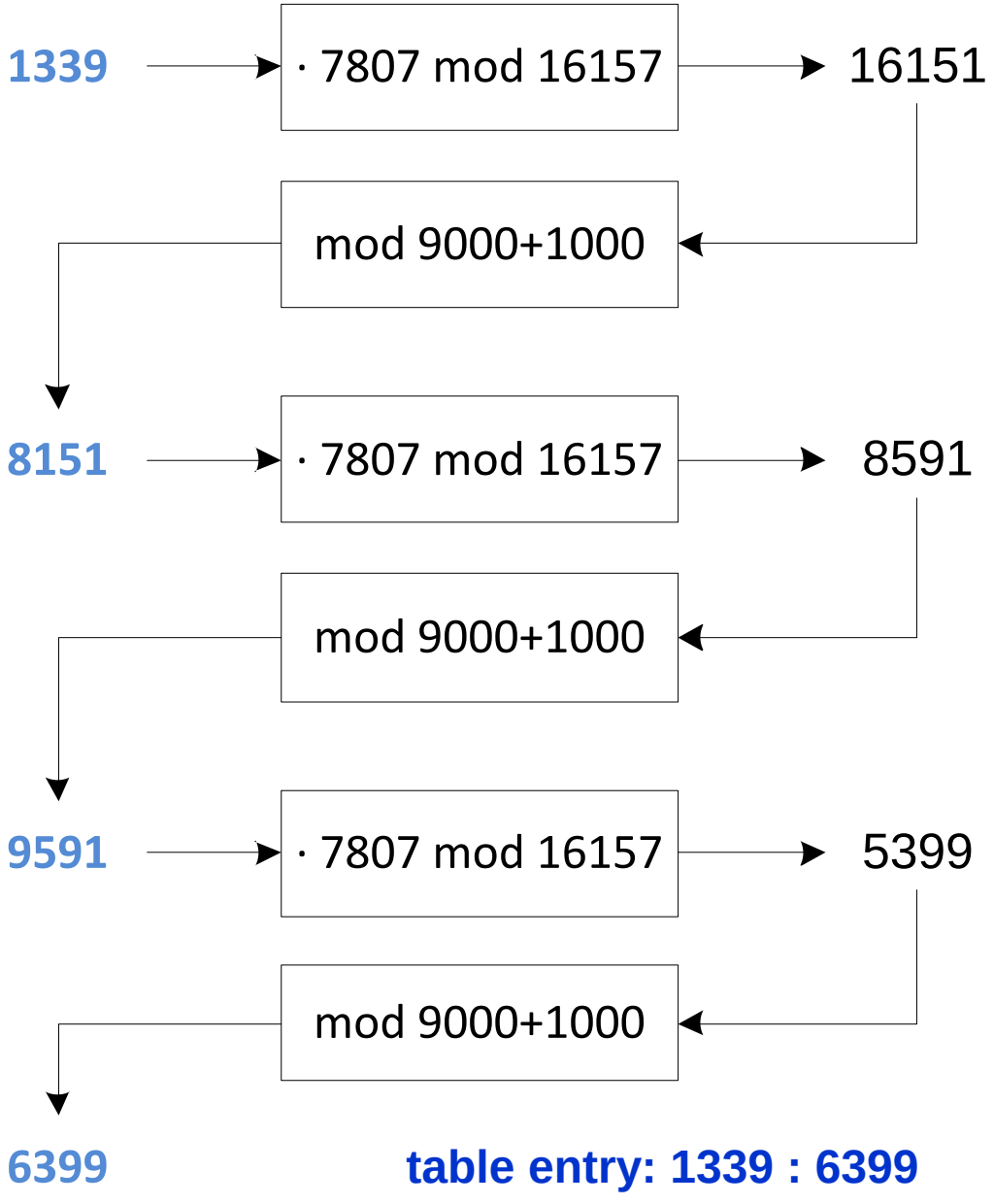
1309–9139–7018–**2139**

2439–9327–4447–**4493**

1084–4677–6676–**5207**

1339–8151–9591–**6399**

3128–8069–6697–**7584**



Cryptanalytic Time – Memory Trade-Off

- PIN-table:

1309–9139–7018–**2139**

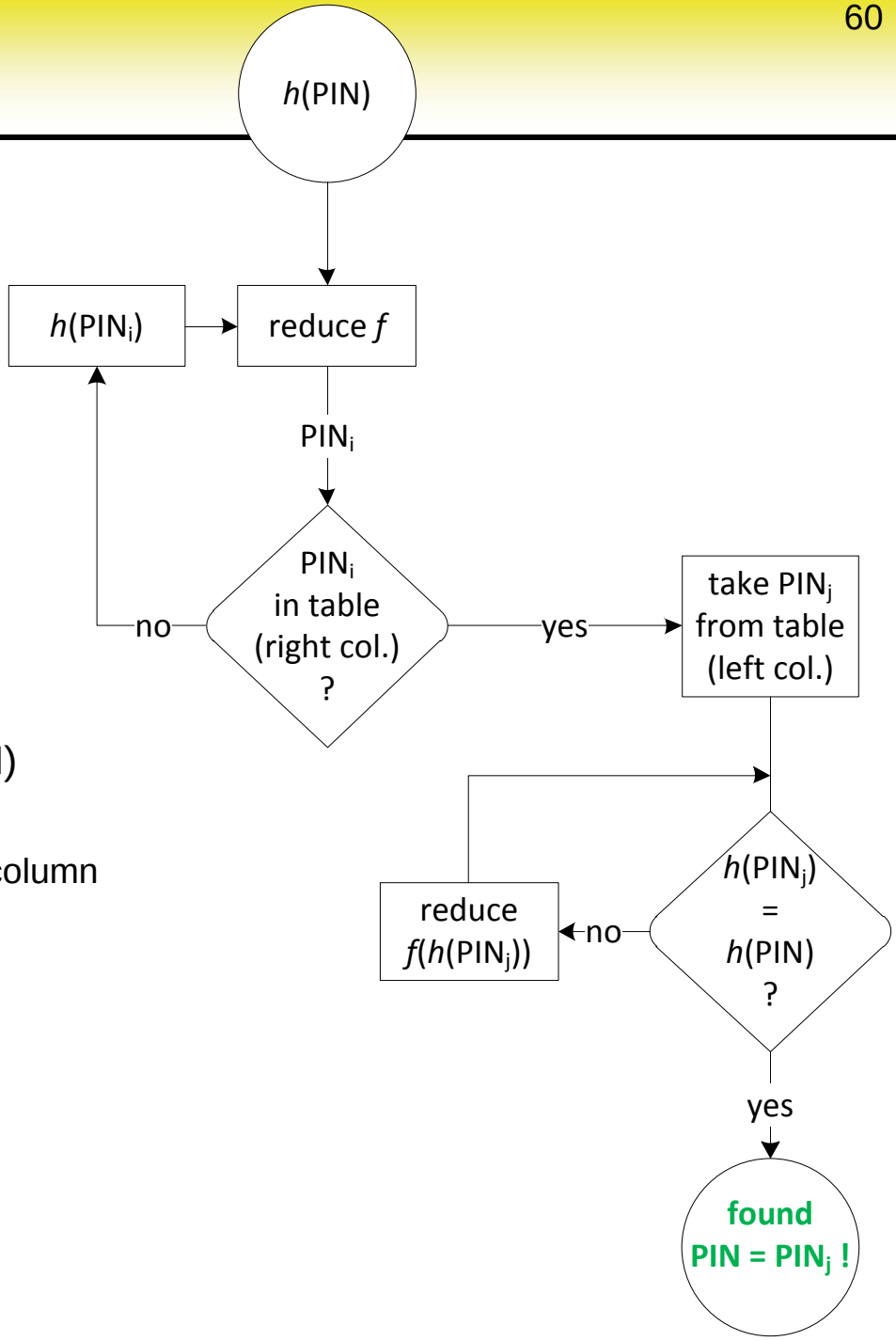
2439–9327–4447–**4493**

1084–4677–6676–**5207**

1339–8151–9591–**6399**

3128–8069–6697–**7584**

- Breaking a PIN:
  - Goal: find PIN for given hash value *h*(PIN)
  - Algorithm:
    - 1. hash / reduce until value is in the right column
    - 2. take left column value
    - 2. hash / reduce until PIN is found



Remaining problems of password based authentication based one way functions

- remaining possible attack:
  - pre-computation:
    - attacker creates lockup table
    - search time (example above):

      $\mathrm{ld}\left((26+26+10+30)^8\right)<53$ comparisons
    - problem: storage space
    - Example:
      - size of MD5 hash: 128 bit
      - number of hashes: $(26+26+10+30)^8$=5 132 188 731 375 616
      - storage
        - space: 5 132 188 731 375 616 · 128 bit ≈ 75 000 TByte
        - price: ≈ 2 Mio €
    - solution: storage space / computation tradeoff
      - Martin E. Hellman: “A Cryptanalytic Time – Memory Trade-Off”
    - today: use the power of the Internet / Cloud
      - http://www.freerainbowtables.com/

Login	Password
…	…
dog	*h*(bone)
…	…

Remaining problems of password based authentication based one way functions

- remaining possible attack:
  - pre-computation
- countermeasure:
  - salt!
  - *h*(x) → *h*(**salt**,x)
  - salt:
    - long (e.g. 128 bit) random value
    - some part is unique for the system (i.e. 104 bit)
    - some part is randomly chosen by the system for each entry in the password table (i.e. 24 bit)
      - NOT stored at the system
    - verification: iterate over all possible salt values

Login	Password
…	…
dog	*h*(bone)
…	…

➔ pre-computation has to be done *for each possible salt*

Remaining problems of password based authentication based one way functions

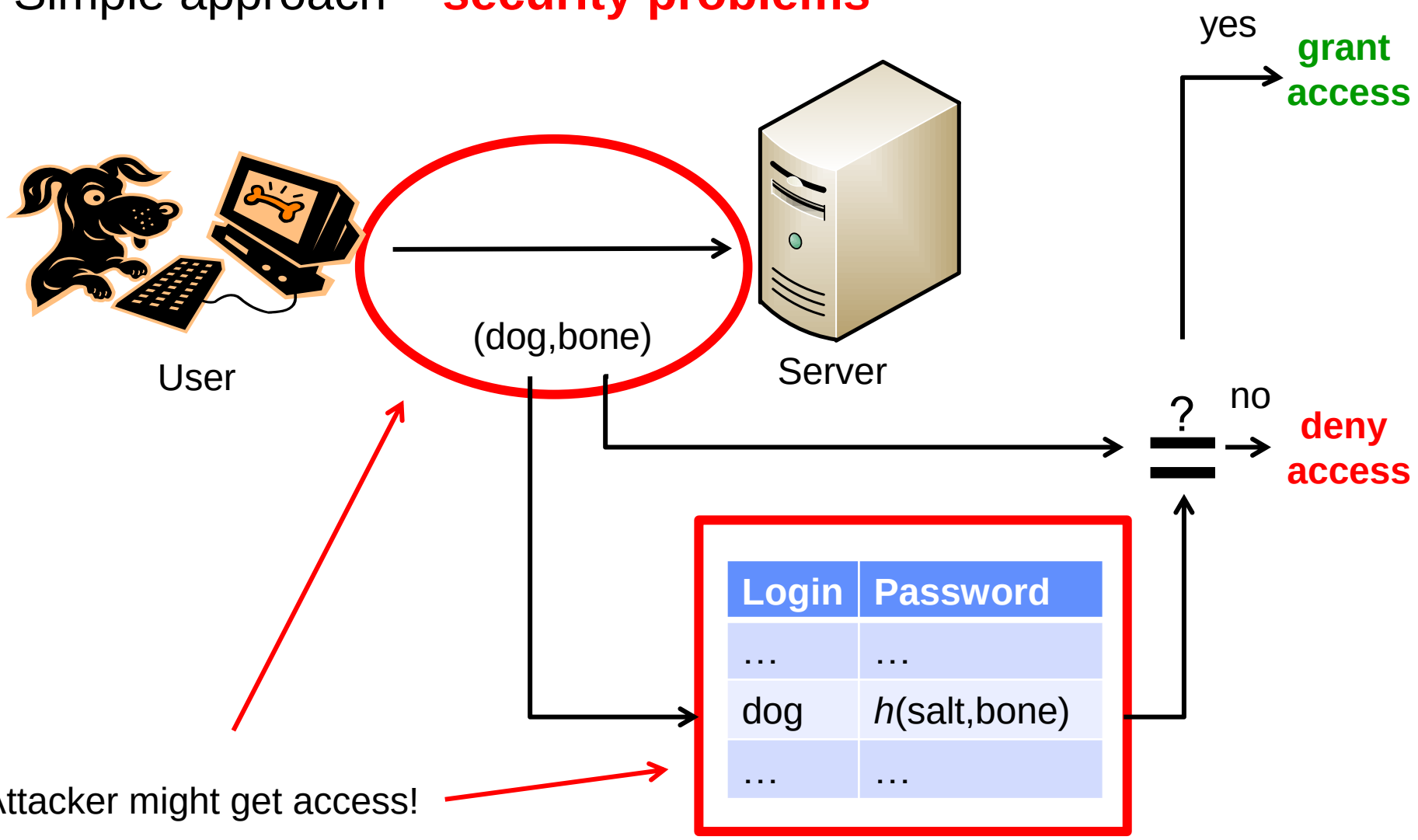
- remaining possible attack:
  - **dictionary attack**
  - problem: people do not chose passwords **randomly**
  - often names, words or predictable numbers are used
  - http://www.whatsmypass.com/the-top-500-worst-passwords-of-all-time
  - attacker uses dictionaries for brute force attack
  - prominent program: *John the Ripper*
    - supports dictionary attacks and password patterns

Login	Password
…	…
dog	*h*(salt,bone)
…	…

- possible solutions:
  - enforce password rules
    - consider usability
  - pre-check passwords (e.g. using John)
  - train people to “generate” good passwords
    - Example: sentence → password
    - “This is the password I use for Google mail” → “TitpIu4Gm”

Password based authentication

- Simple approach – **security problems**

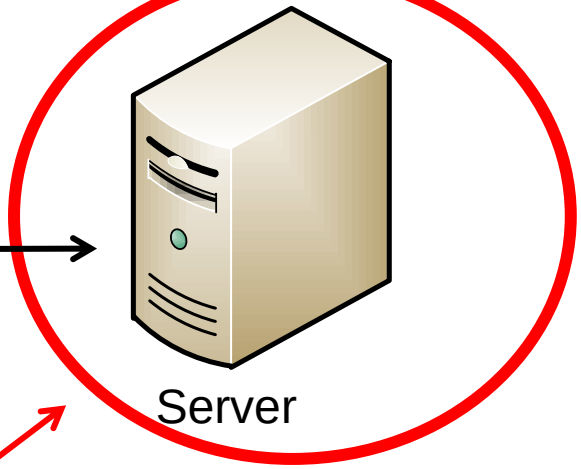


Password based authentication

- Simple approach – **security problems**

User

(dog,bone)



Server

yes

grant access

?

no

deny access

Login	Password
…	…
dog	*h*(salt,bone)
…	…

Might be the attacker!

The Server as Attacker

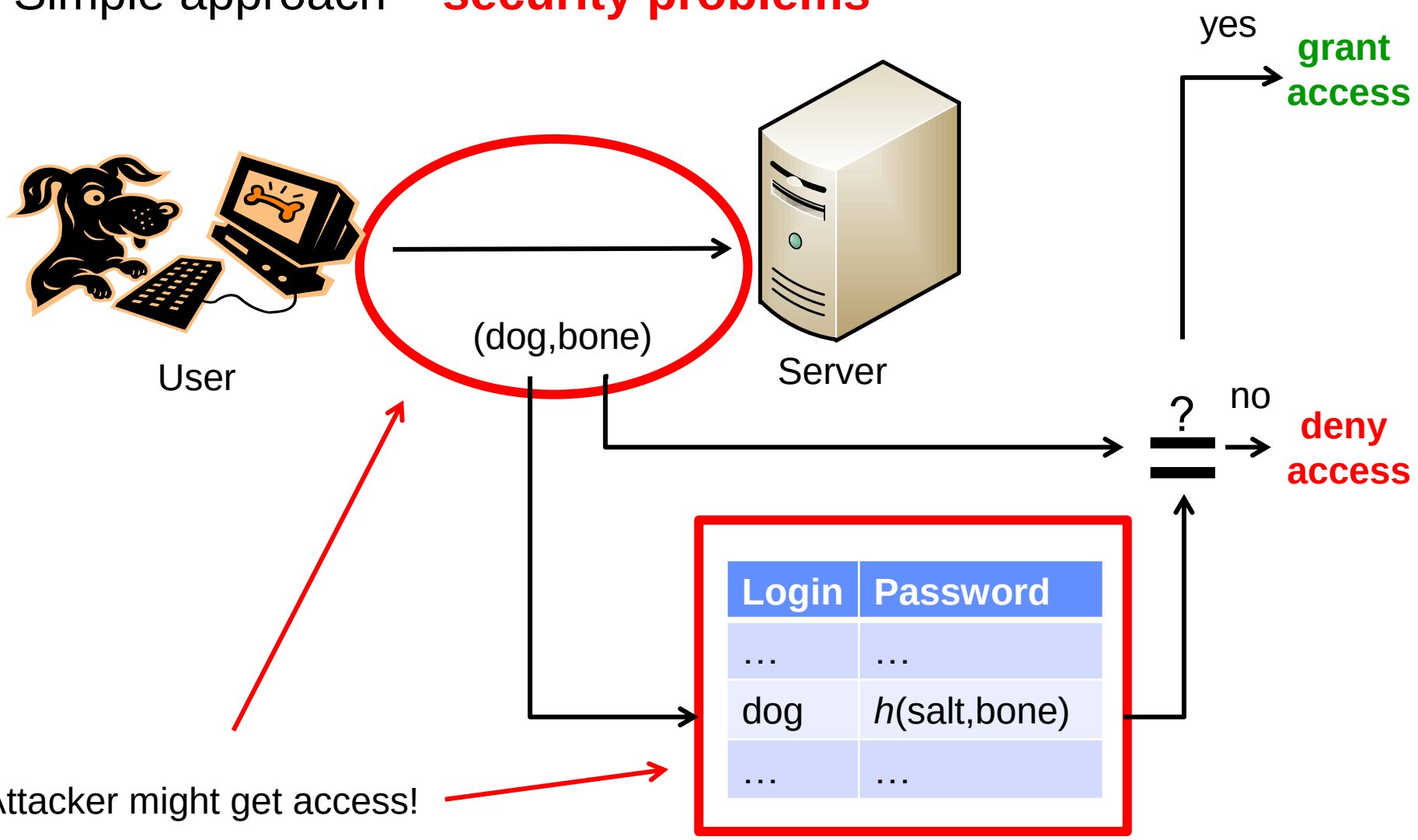
- … a new Web 2.0 service
- … for people which like city journeys
- … find cool cities and places like shops, restaurants, hotels etc.
- ... information from globe-trotters for globe-trotters
- … they can share their knowledge after secure login
- So that's wrong?

➔ It collects (username,password) and tries to login into other popular services like Gmail, Twitter, eBay, Amazon etc.

password rule: never "reuse" passwords!

Password based authentication

- Simple approach – **security problems**



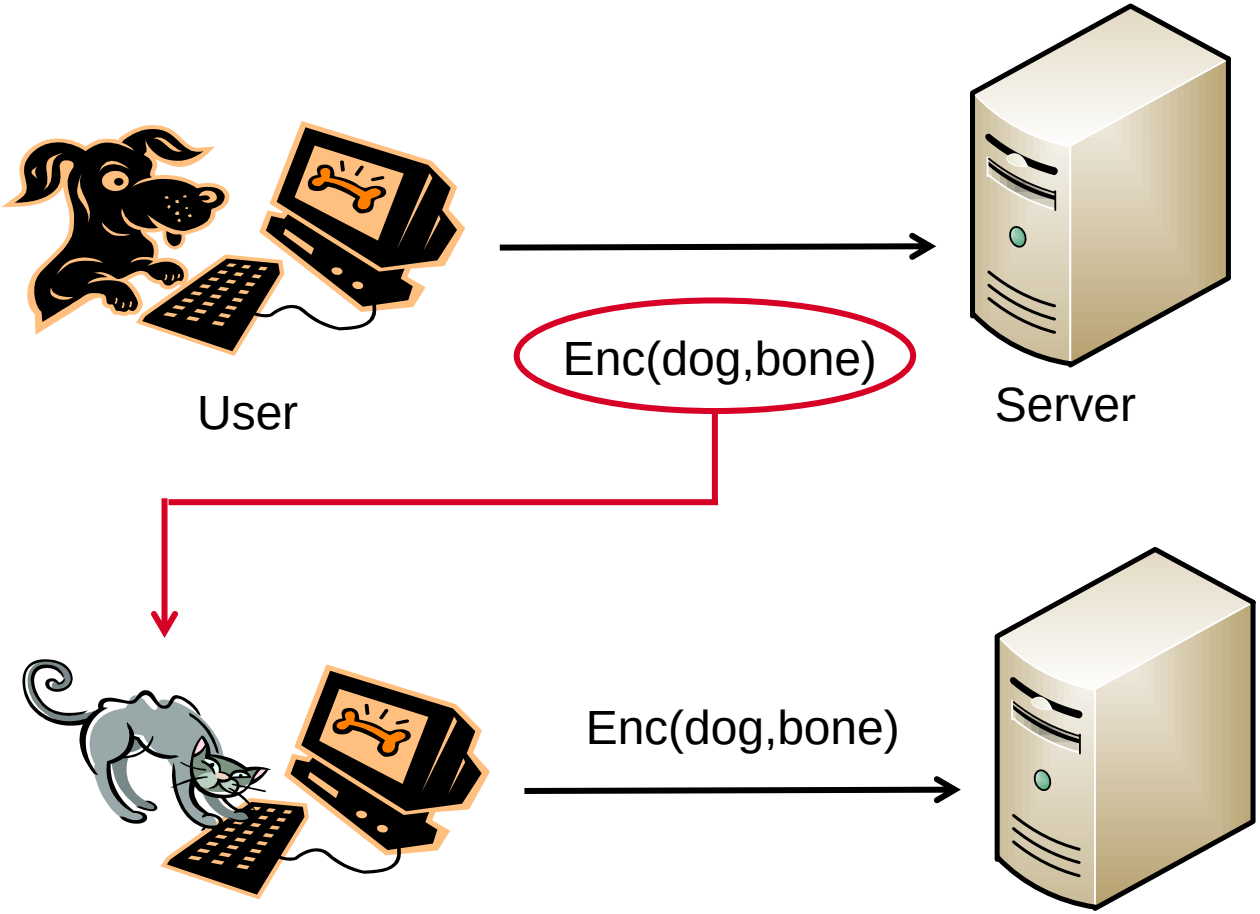
Password based authentication

- **security problems**

- possible solution:
  - encrypt communication
- remaining problems:
  - not always possible
  - replay attack

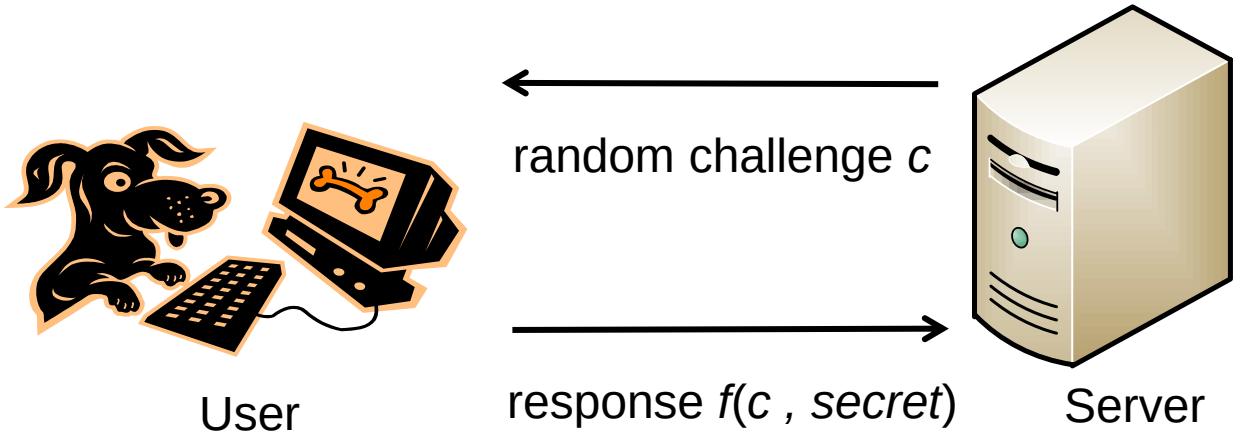
Password based authentication

- **security problem – replay attack**



Password based authentication

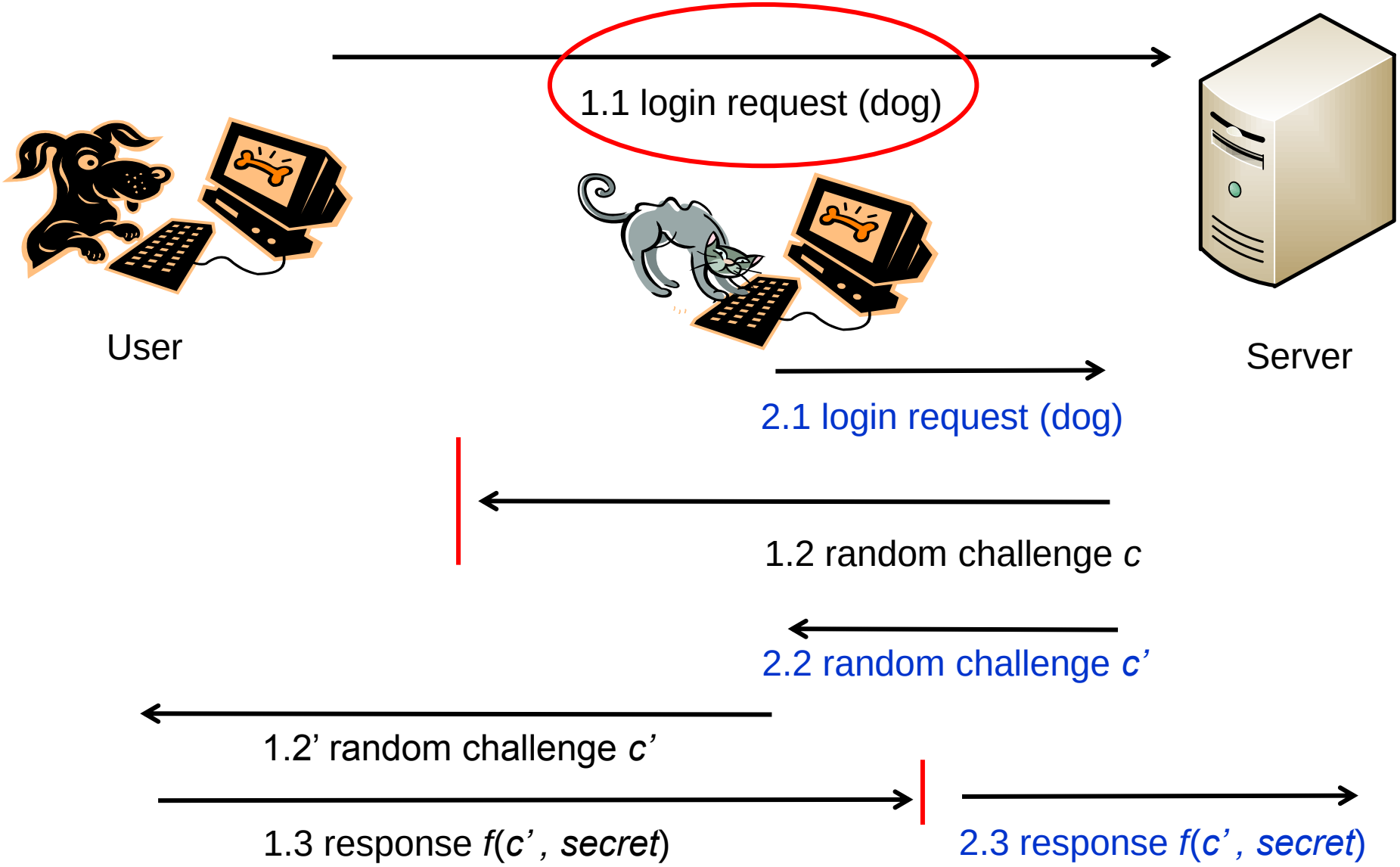
- **security problem – replay attack**
- **possible solution: challenge-response protocol**



  - tries to ensure *freshness*
- remaining problems:
  - Man-in-the-middle attacks
  - parallel protocol runs

Password based authentication

- **security problem – MITM / parallel protocol runs**



Password based authentication

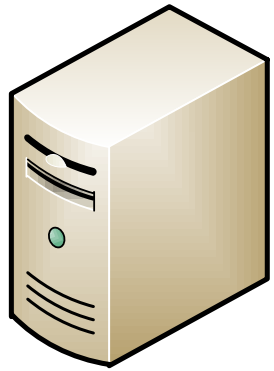
- **security problem – MITM / parallel protocol runs**
- **possible solutions:**
  - disallow parallel login protocol runs for the same user
  - make protocol runs distinguishable

Password based authentication

- **possible solution: distinguishable protocol runs**

User

1.1 login request (dog,r_{dog})



Server

- will not accept the same *r* multiple times

2.1 login request (dog,r_{cat})

1.2 random challenge *c*

2.2 random challenge *c'*

1.2' random challenge *c'*

1.3 response *f*(*c'* , *secret,* r_{dog})

2.3 response *f*(*c'* , *secret,* r_{dog})

Password based authentication

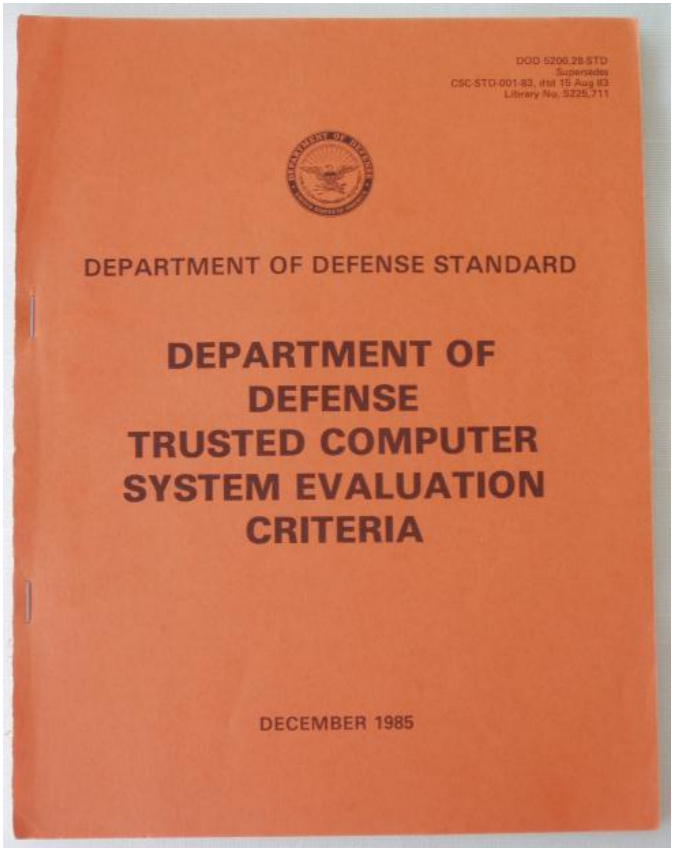
- **security problem – MITM / parallel protocol runs**
- **possible solutions:**
  - disallow parallel login protocol runs for the same user
  - make protocol runs distinguishable
- remaining security problems:
  - …

  (ok I will stop here – if you are interested in many more problems / solutions I recommend: Colin Boyd, Anish Mathuria: “Protocols for Authentication and Key Establishment”, Springer, 2003.)

Password based authentication

- **(non protocol related) security problems:**
  - phising, i.e. faked UI for entering secret information
  - today: mostly Internet based attacks
  - but: local attacks possible as well
    - faked login / lock screen
    - solution: “trusted path” / Secure Attention Key

      3.2.2.1.1 The TCB [Trusted Computing Base] shall support a trusted communication path between itself and user for initial login and authentication. Communications via this path shall be initiated exclusively by a user.
      [Department of Defense: “Trusted Computer System Evaluation Criteria”, CSC-STD-001-83, 15. August 1983 – called “Orange Book”]
    - well known implementations:
      - Windows: Ctrl+Alt+Del
      - Linux: Ctrl+Alt+Pause
        - could be freely chosen in principle



[http://en.wikipedia.org/wiki/File:Orange-book-small.PNG]

One time password

- One Time Password
  - only used to authenticate a single transation
- Advantage
  - abuse of OTP becomes harder for the attacker
- Implementations
  - list of OTPs
    - known from online banking: TAN, iTAN
  - on the fly generated and transmitted over a second channel
    - mTAN
  - time-synchronized (hardware) tokens:
    - token knows a secret *s*
    - OTP= *f*(s,time)
  - hash chain based

One time password

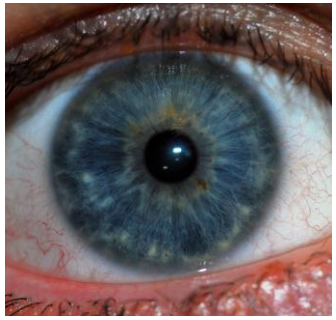
- OTP Implementations
  - hash chain based
    - Leslie Lamport: “Password Authentication with Insecure Communication”
    - users generates hash chain:
      - $h^n(\ldots h^3(h^2(h^1(\text{password}))))$
    - users sends $\boldsymbol{h^n()}$ as his “password” during register procedure
    - next login user sends $\boldsymbol{h^{n-1}()}$
    - server verifies: h($\boldsymbol{h^{n-1}()}$) = $\boldsymbol{h^n()}$
    - server now stores: $\boldsymbol{h^{n-1}()}$

Biometrics for Authentication

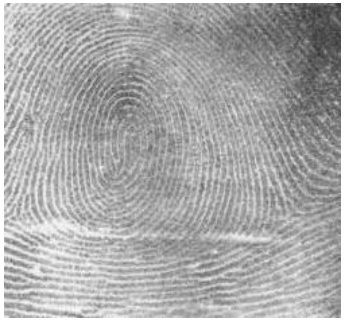
- *Physiological* or *behavioural* characteristics (of a human being) are measured and compared with reference values to
  - **verify,** that a given subject is the one it claimed to be
    - claimed “identity” is known to the system by other means
  - **identify**, a subject within a given set of (known) subjects
    - “identity” should be derived from biometrics
    - usually more difficult compared to verification

Biometrics: Physiological / Behavioural Characteristics

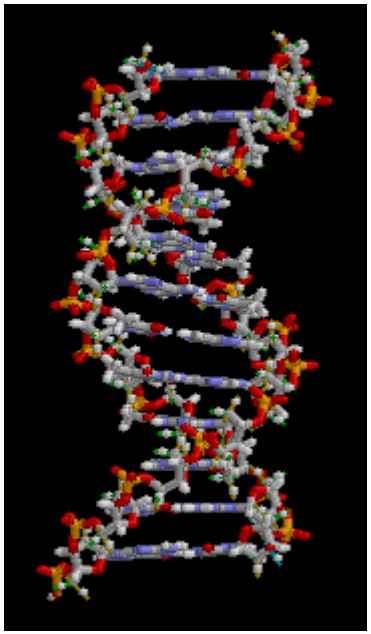
[Pictures are mostly from Wikipedia]



Iris / Retina



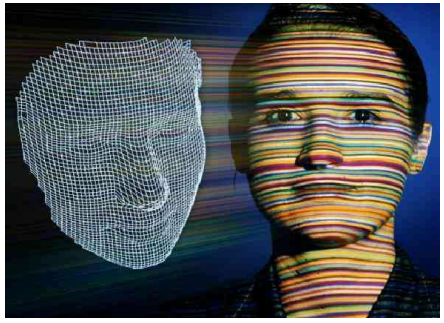
Fingerprint



DNA

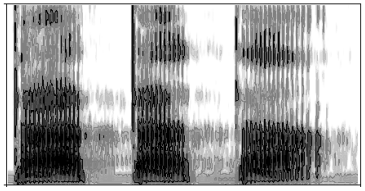
Thermography: facial thermograms

Hand geometry



[http://www.bromba.com/knowhow/IBS2005.pdf]

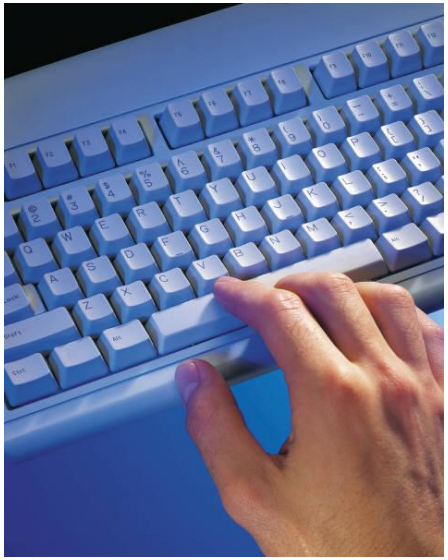
(3D) Face geometry



Voice spectrogram

Handwriting: appearance, dynamics of writing

Gait



Key strokes: dynamics of writing (speed, pressure etc.)

Biometric characteristics: Requirements

- universal: everyone has it
- unique
- stable over time
- measurable
- acceptable
- analysable
- resistant against cloning / faking

Biometric Systems: General Architecture

- Enrolment phase:

- Verification:

- Identification:

Biometrics: Pros and Cons

- Pros:
  - Cannot be divulged or lost/forgotten
  - can be utilised “on the fly”
  - Hard to copy
- Cons:
  - Cannot be renewed
  - Person related data requires special protection (privacy)
  - Invasion (of privacy)
  - Error rate

Biometrics: Pros and Cons

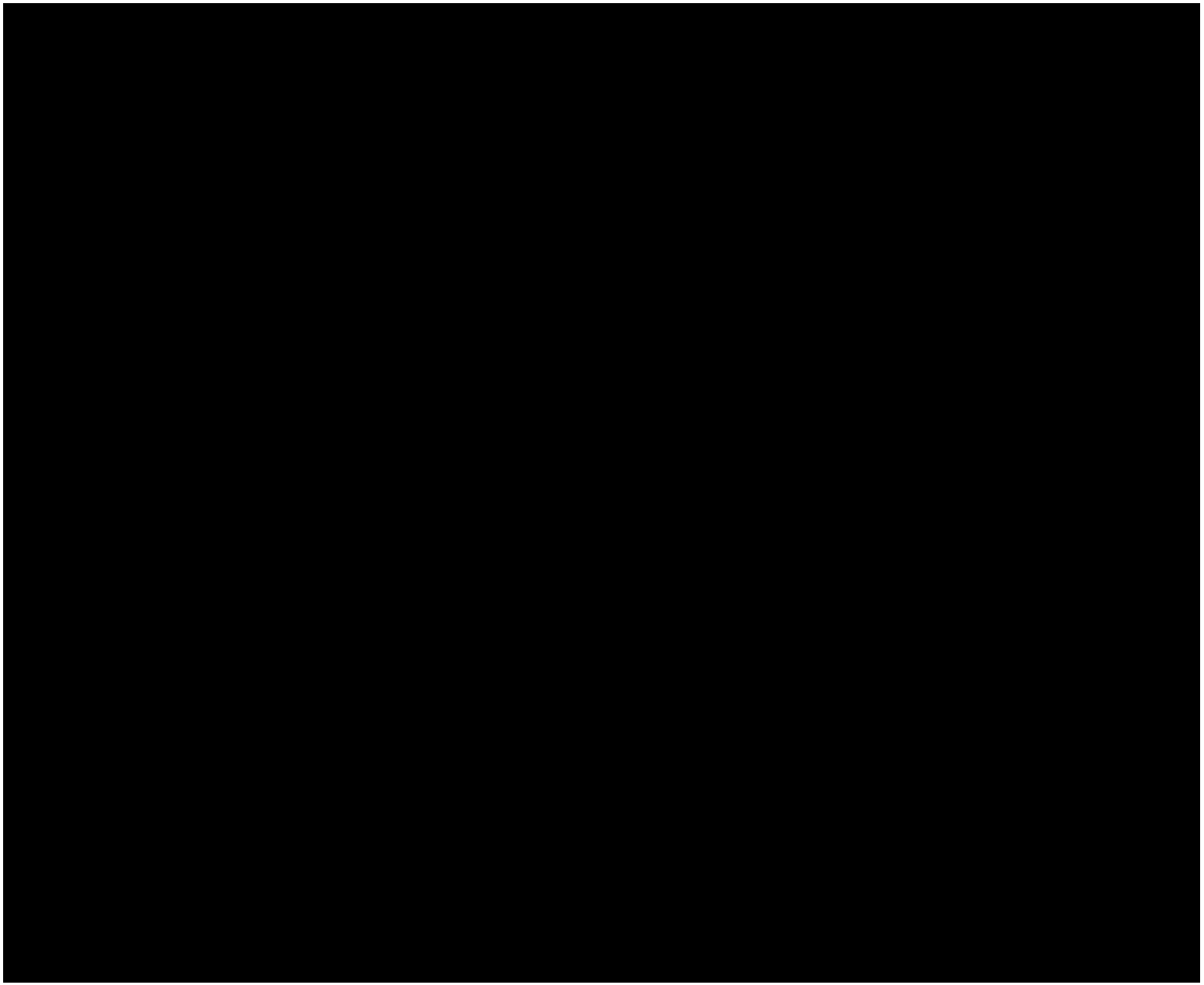
- Pros:
  - Cannot be divulged or lost/forgotten
    - but could be stolen

Safety Risks of Biometrics

Biometrics: Pros and Cons

- Pros:
  - Cannot be divulged or lost/forgotten
    - but could be stolen:
      - http://news.bbc.co.uk/2/hi/asia-pacific/4396831.stm
    - could become „unusable“ due to
      - ageing
      - incidents
      - disease
  - can be utilised “on the fly”
    - privacy problems (unnoticeable measurement of Biometrics)
  - Hard to copy
    - depends on the Biometric system used
    - many systems are easy to cheat
    - ftp://ftp.ccc.de/pub/documentation/Fingerabdruck_Hack/fingerabdruck.mpg

Demonstration of Fingerprint Cloning by CCC



Biometrics: Pros and Cons

- Pros:
  - Cannot be divulged or lost/forgotten
    - but could be stolen:
      - http://news.bbc.co.uk/2/hi/asia-pacific/4396831.stm
    - could become „unusable“ due to
      - ageing
      - incidents
      - disease
  - can be utilised “on the fly”
    - privacy problems (unnoticeable measurement of Biometrics)
  - Hard to copy
    - depends on the Biometric system used
    - many systems are easy to cheat
    - ftp://ftp.ccc.de/pub/documentation/Fingerabdruck_Hack/fingerabdruck.mpg
    - cloning of e.g. fingerprints might be in the interest of law enforcement
      - access to biometrically secured devices

Biometric Systems: Types of Failures

- False Accept Rate (FAR) / False Match Rate (FMR):
  - **Security problem!**
- False Reject Rate (FRR) / False nonmatch Rate (FNR):
  - Usability / acceptance problem
- Receiver Operating Characteristic (ROC):
  - curve of FAR against FRR
- Equal Error Rate (EER):
  - rate for FAR=FRR
  - can be seen from the ROC curve

ROC Curve and Security Problems of Biometrics

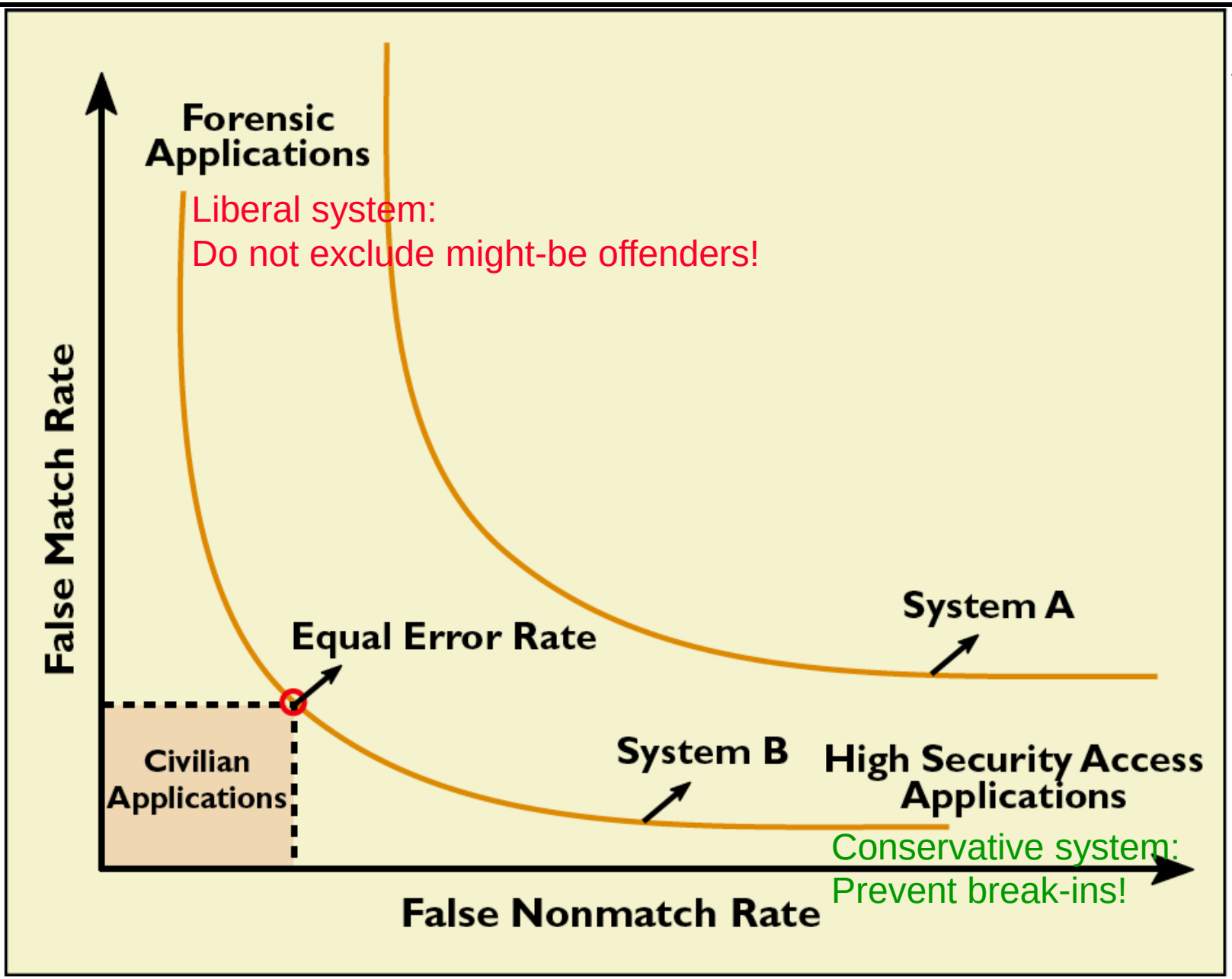


Figure taken from:
Anil Jain, Lin Hong, Sharath Pankanti: Biometric Identification; Communications of the ACM 43/2 (2000) 91-98

Low FMR causes high FNR and vice versa !

Biometric Systems: Types of Failures

- False Accept Rate (FAR):
  - **Security problem!**
- False Reject Rate (FRR):
  - Usability / acceptance problem
- Receiver Operating Characteristic (ROC):
  - curve of FAR against FRR
- Equal Error Rate (EER):
  - error rate for FAR=FRR
  - can be seen from the ROC curve
- Failure To Enroll Rate (FTE):
  - Usability / acceptance problem
- Failure To Capture Rate (FTC):
  - Usability / acceptance problem

Enhanced Security: Multi-biometric Systems

Multi-sample

capacitive

Multi-sensor

Multi-modal

Hybrid

optical

Multi-algorithm

Multi-instance

Minutiae

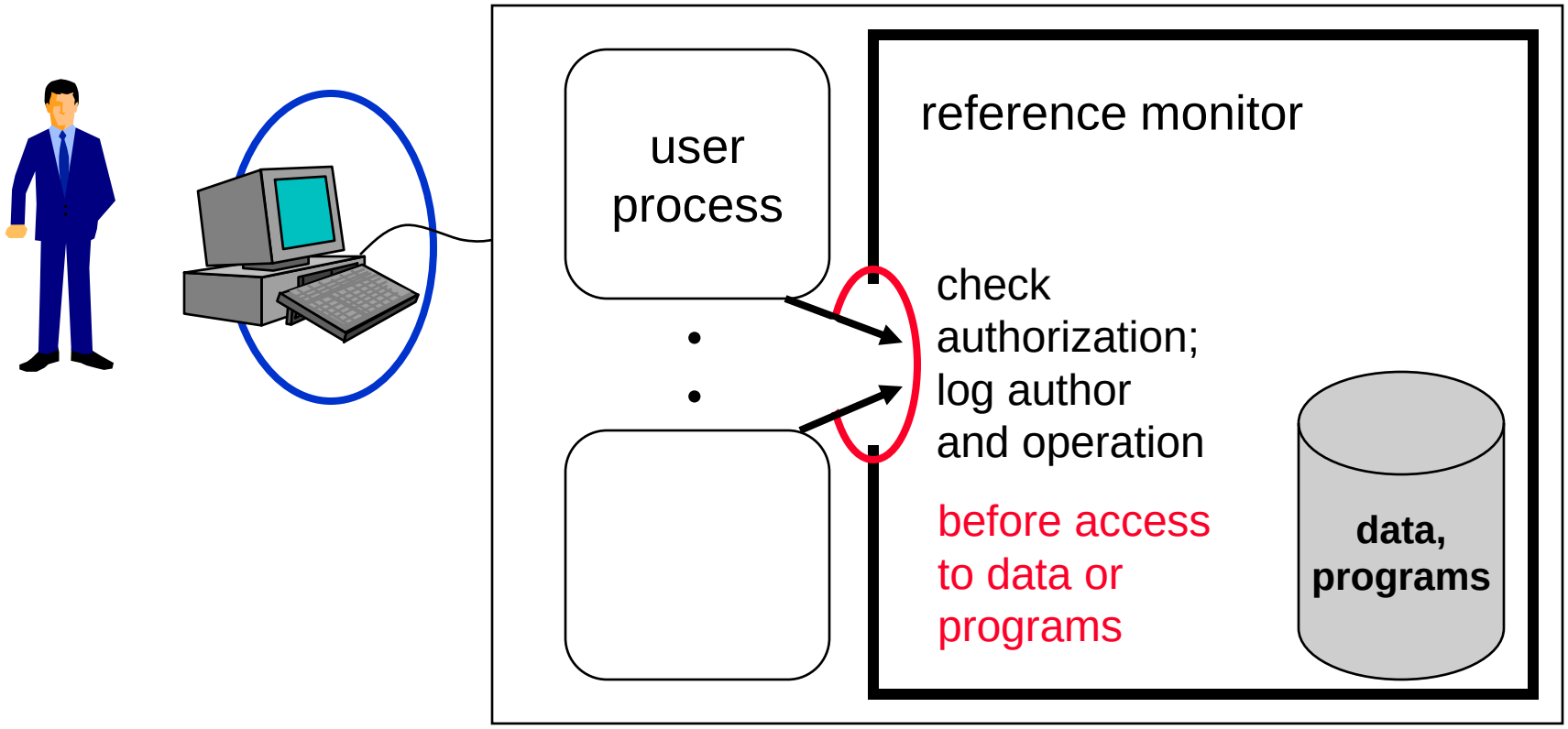
structure

right eye

left eye

Admission and access control

Admission control communicate with authorized partners only

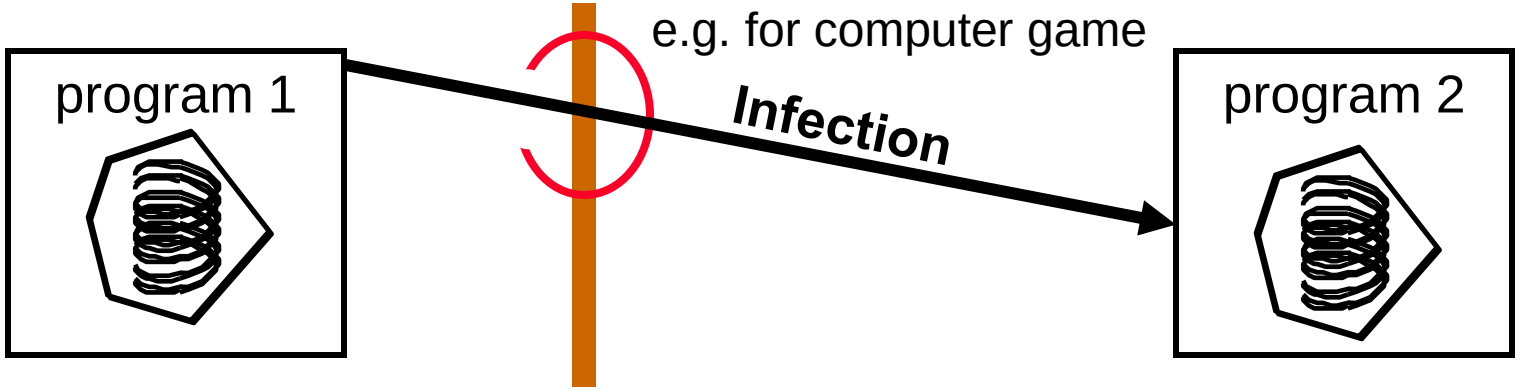


Access control subject can only exercise operations on objects if authorized.

Computer virus vs. transitive Trojan horse

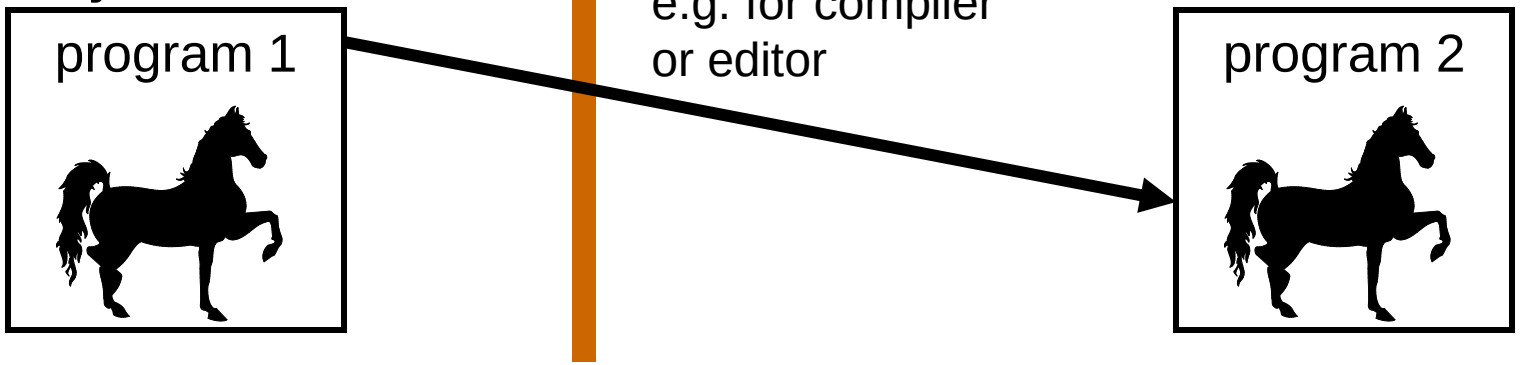
computer virus

unnecessary write access,
e.g. for computer game



transitive Trojan horse

necessary write access,
e.g. for compiler
or editor



Access control

Limit spread of attack by as little privileges as possible:
Don‘t grant unnecessary access rights!

➡ No computer viruses, only transitive Trojan horses!

Basic facts about Computer viruses and Trojan horses

Other measures fail:

1. Undecidable if program is a computer virus
   proof (indirect) assumption: decide (•)

```
program counter_example
if decide (counter_example)    then  no_virus_functionality
                               else  virus_functionality
```

2. Undecidable if program is Trojan horse

Better be too careful!

3. Even known computer viruses are not efficiently identifiable
   self-modification → ~~virus scanner~~

4. Same for: Trojan horses

5. Damage concerning data is not ascertainable afterwards
   function inflicting damage could modify itself

Further problems

1. Specify exactly what IT system is to do and what it is *not* to do. ?
2. Prove *total correctness* of implementation. **today** ?
3. Are all *covert channels* identified? ?

Golden Rule

Design and realize IT system as *distributed* system, such that a limited number of attacking computers cannot inflict significant damage.

Distributed System

Aspects of distribution

physical distribution
distributed control and implementation structure

distributed system:

no entity has a global view on the system

Security in distributed systems

Trustworthy terminals

Trustworthy	only to user
	to others as well

Ability to communicate

Availability by	redundancy and diversity

Cryptography

Confidentiality by	encryption
Integrity by	message authentication codes (MACs) or digital signatures

Availability

Infrastructure with the least possible complexity of design

Connection to completely diverse networks

- different frequency bands in radio networks
- redundant wiring and diverse routing in fixed networks

Avoid bottlenecks of diversity

- e.g. radio network needs same local exchange as fixed network,
- for all subscriber links, there is only one transmission point to the long distance network

Diffie-Hellman key agreement (1)

practically important: patent exhausted before that of RSA
→ used in PGP from Version 5 on

theoretically important: steganography using public keys

based on difficulty to calculate **discrete logarithms**

Given a prime number ***p*** and ***g*** a generator of Z_p^*

$$g^x = h \bmod p$$

x is the **discrete logarithm** of ***h*** to basis ***g*** modulo ***p***:

$$x = \log_g(h) \bmod p$$

discrete logarithm assumption

Discrete logarithm assumption

$\forall$ PPA $\mathcal{DL}$ (probabilistic polynomial algorithm, which tries to calculate discrete logarithms)

$\forall$ polynomials Q

$\exists\, L \;\forall\; \mathcal{l} \geq L$: (asymptotically holds)

If p is a random prime of length $\mathcal{l}$

thereafter g is chosen randomly within the generators of Z_p^*

x is chosen randomly in Z_p^*

and $g^x = h \mod p$

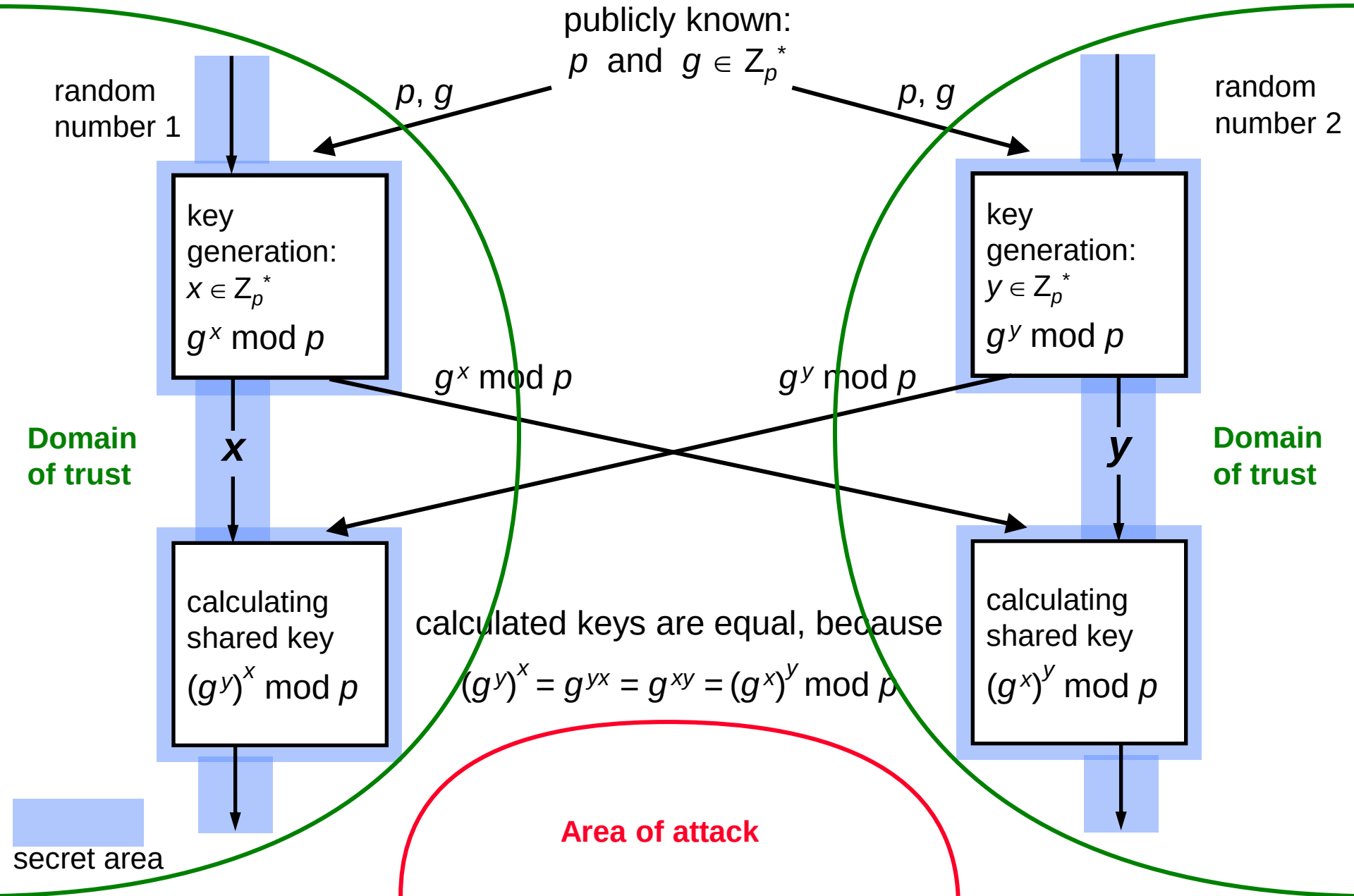
$$\mathcal{W}(\mathcal{DL}(p,g,h)=x) \leq \frac{1}{Q(\mathcal{l})}$$

(probability that $\mathcal{DL}$ really calculates the discrete logarithm, decreases faster than $\frac{1}{\text{any polynomial}}$)

trustworthy ??

practically as well analyzed as the assumption factoring is hard

Diffie-Hellman key agreement (2)



Diffie-Hellman assumption

Diffie-Hellman (DH) assumption:

Given p, g, $g^x \bmod p$ and $g^y \bmod p$

Calculating $g^{xy} \bmod p$ is difficult.

DH assumption is stronger than the discrete logarithm assumption

- Able to calculate discrete Logs $\Rightarrow$ DH is broken.
  Calculate from p, g, $g^x \bmod p$ and $g^y \bmod p$ either x or y. Calculate $g^{xy} \bmod p$ as the corresponding partner of the DH key agreement.
- Until now it couldn't be shown:
  Using p, g, $g^x \bmod p$, $g^y \bmod p$ and $g^{xy} \bmod p$ either x or y can be calculated.

Find a generator in cyclic group Z_p^*

Find a generator of a cyclic group Z_p^*

Factor $p-1 =: p_1^{e_1} \bullet p_2^{e_2} \bullet \ldots \bullet p_k^{e_k}$

1. Choose a random element g in Z_p^*

2. For i from 1 to k:

$$b := g^{\frac{p-1}{p_i}} \bmod p$$

If $b=1$ go to 1.