Security in Computer Networks

Multilateral Security in and by Distributed Systems

Transparencies for the Lecture:

Security and Cryptography I

(Version 2022/11/10)

Stefan Köpsell
(Slides mainly created by Andreas Pfitzmann)
Technische Universität Dresden, Faculty of Computer Science, D-01062 Dresden
Nöthnitzer Str. 46, Room 3062
Phone: +49 351 463-38272, e-mail: stefan.koepsell@tu-dresden.de, https://dud.inf.tu-dresden.de/sac1

Field of Specialization: Security and Privacy

Lectures	*Staff*	*SWS*
Security and Cryptography I, II	**Köpsell**	**2/2**
Application Security	**Köpsell**	**2/0**
Cryptography and -analysis	**Franz**	**2/1**
Information & Coding Theory	**Franz**	**2/1**
Data Security and Cryptography	**Köpsell**	**0/4**
Security Lab	**Köpsell**	**2/2**
Computers and Society	**Köpsell**	**2/0**
Seminar: Privacy and Security	**Byrenheid et al.**	**2/0**
Seminar: Security in Computer Systems	**Köpsell**	**2**
Introduction to Data Protection Law	**Wagner**	**2/0**
Resilient Networking	***Strufe***	***2/2***
Privacy Enhancing Technologies	***Strufe***	***3/1***

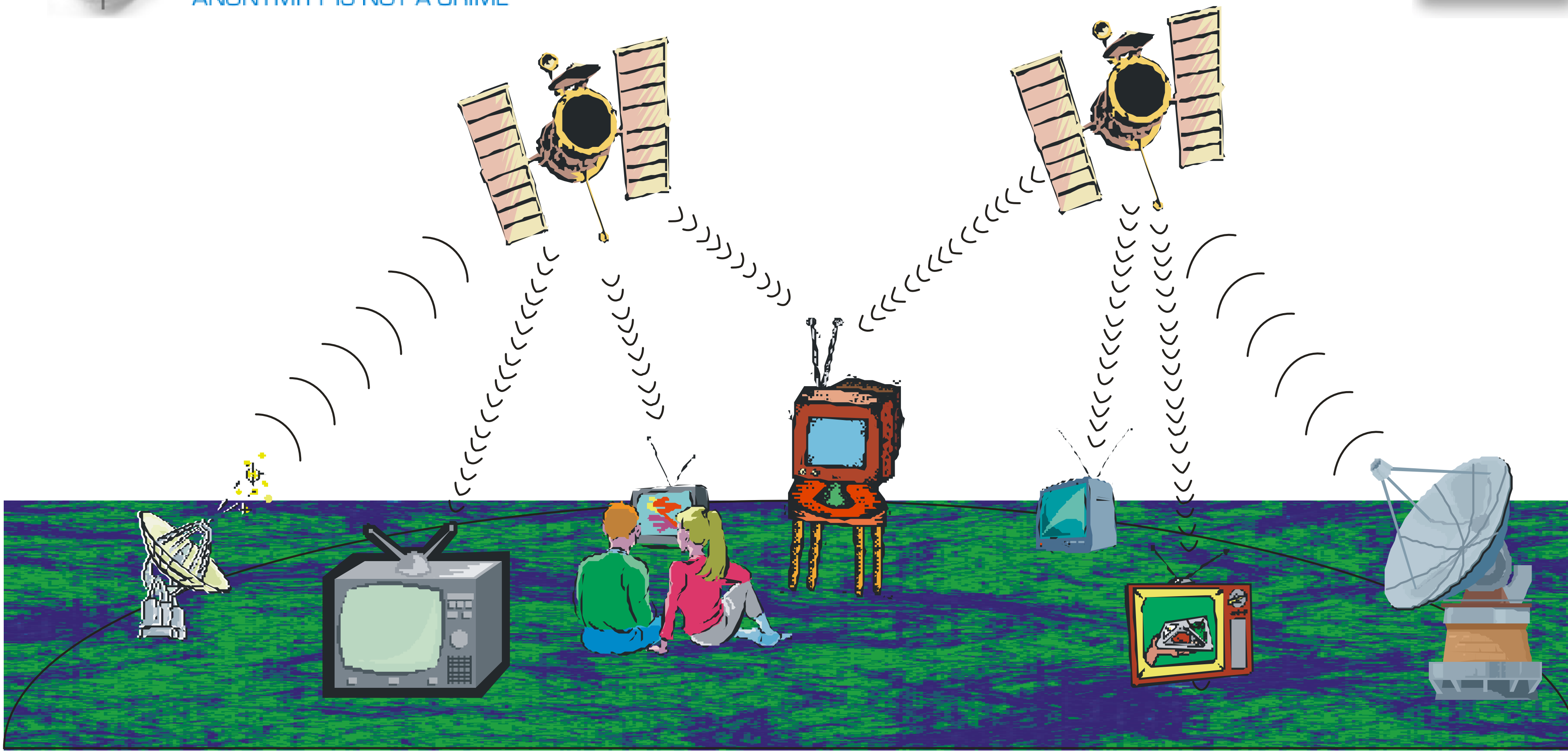
Areas of Teaching and Research

- Multilateral security, in particular security by distributed systems
- Privacy Enhancing Technologies (PETs)
- Cryptography
- Physical Layer Security
- Information- and coding theory

- Security & Privacy
  - in Vehicular Networks (Connected Driving)
  - for IoT & Cyberphysical Systems
  - industrial communication
  - focused on humans: social engineering, transparency

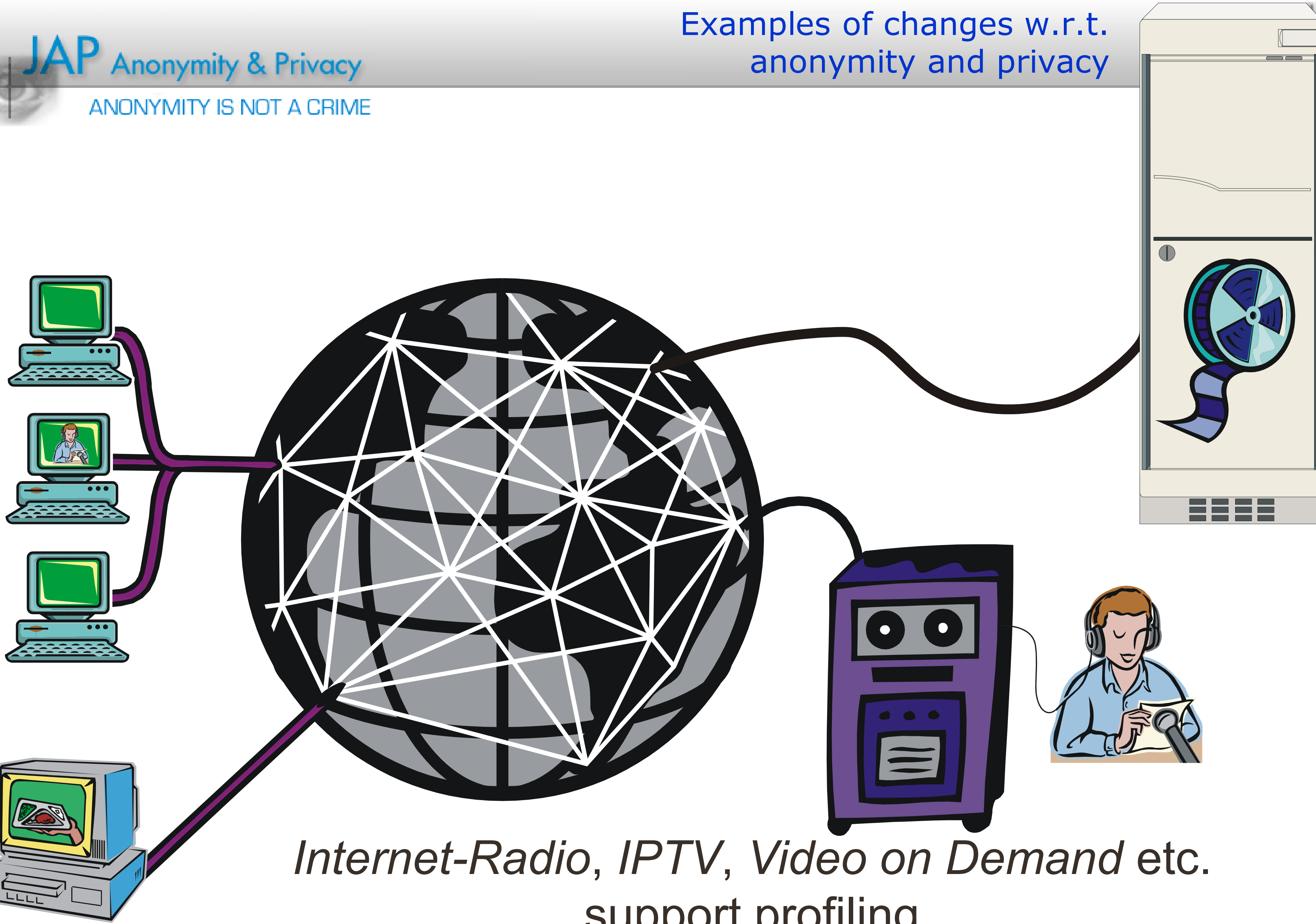
- Context-aware, Adaptive & Smart Security Solutions

Examples of changes w.r.t. anonymity and privacy



Broadcast allows recipient anonymity — it is not detectable who is interested in which programme and information

Examples of changes w.r.t. anonymity and privacy



Internet-Radio, *IPTV*, *Video on Demand* etc. support profiling

Anonymous plain old letter post is substituted by „surveillanceable" e-Mails

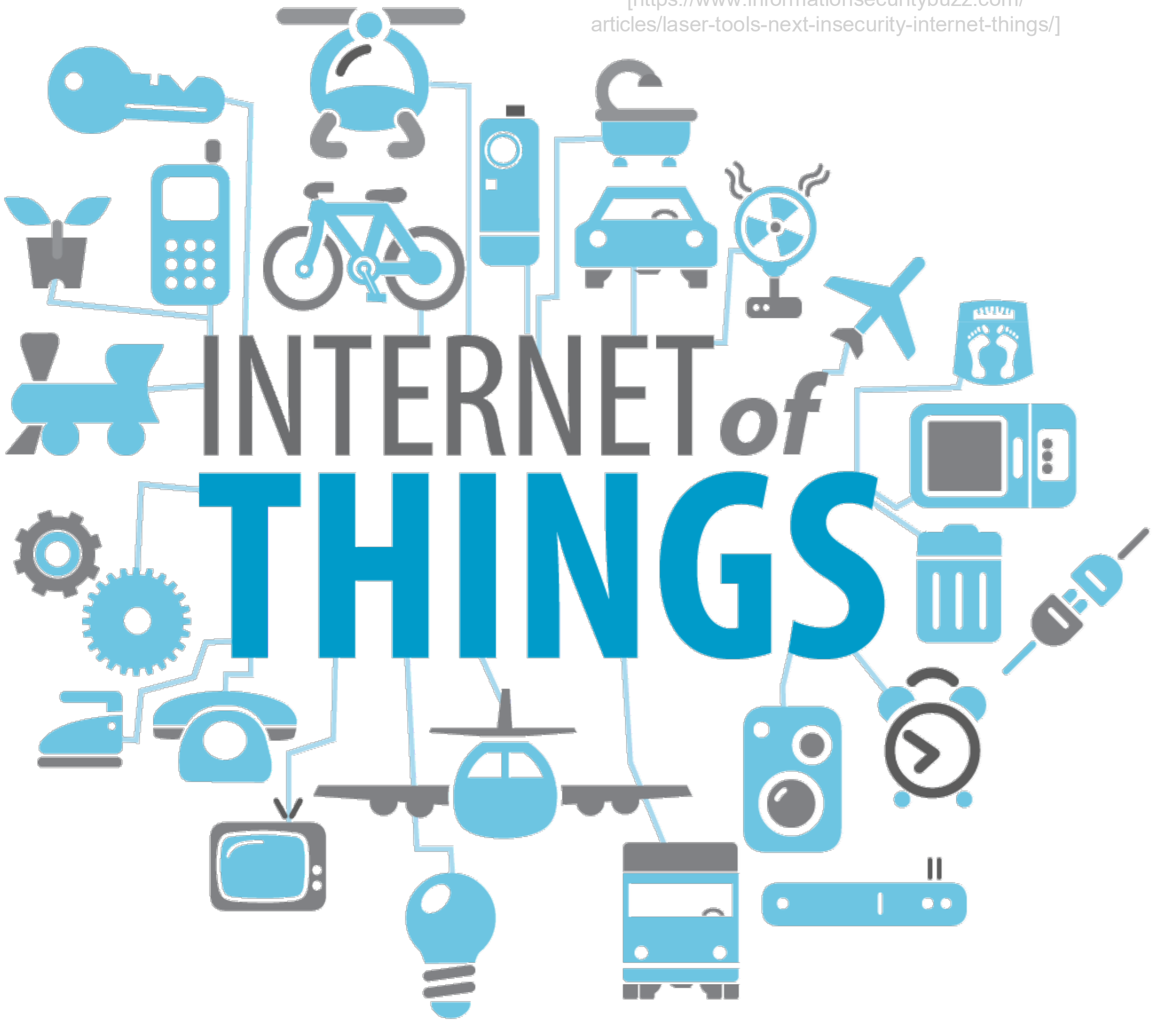
Remark: Plain old letter post has shown its dangers, but nobody demands full traceability of them …

The massmedia „newspaper“ will be personalised by means of Web, elektronic paper and print on demand

The Bystander-Challenge in Smart IoT Worlds…

- **ubiquitous sensing** undermines privacy in everybody's everyday life
  - cameras, microphones, Lidar, Radar, …
  - examples: connected driving, Smart Cities, e-Health, …
- **bystanders** are affected
  - awareness and control for uninvolved persons
  - example: voice assistants
- **reasoning about linkability and privacy** in complex and realistic systems

[https://www.informationsecuritybuzz.com/articles/laser-tools-next-insecurity-internet-things/]



Areas of Teaching and Research

- Multilateral security, in particular security by distributed systems
- Privacy Enhancing Technologies (PETs)
- Cryptography
- Physical Layer Security
- Information- and coding theory

- Security & Privacy
  - in Vehicular Networks (Connected Driving)
  - for IoT & Cyberphysical Systems
  - industrial communication
  - focused on humans: social engineering, transparency

- Context-aware, Adaptive & Smart Security Solutions

Context-aware, Adaptive & Smart Security Solutions

- **State-of-the-Art:**
  - **manual, highly application dependent engineering of security/privacy solutions by security experts**
  - → **costly**
  - → **does not scale**

- **Solution:**
  - **automation of the security development, deployment and operation processes**

- **Ingredients:**
  - **Reasoning about Situation: Context & Context Awareness**
  - **Measuring/Specifying Security: Quality of Security**
  - **Decisions: AI & ML (or something better)**

- **Result: Disruption for Digitalisation**
- **Minor issue: no clue how to achieve it…**

Aims of Teaching at Universities

Science shall clarify

How something is.

But additionally, and even more important

Why it is such

or

How could it be

(and sometimes, how should it be).

“**Eternal truths**” (i.e., knowledge of long-lasting relevance) should make up more than 90% of the teaching and learning effort at universities.

General Aims of Education in IT-security (sorted by priorities)

1. Education to **honesty** and a **realistic self-assessment**
2. Encouraging realistic **assessment of others**, e.g., other persons, companies, organizations
3. Ability to gather **security and data protection requirements**
   - Realistic protection goals
   - Realistic attacker models / trust models

Realistic protection goals/attacker models: Technical solution possible?

General Aims of Education in IT-security (sorted by priorities)

1. Education to **honesty** and a **realistic self-assessment**
2. Encouraging realistic **assessment of others**, e.g., other persons, companies, organizations
3. Ability to gather **security and data protection requirements**
   - Realistic protection goals
   - Realistic attacker models / trust models
4. **Validation** and **verification**, including their practical and theoretical **limits**
5. Security and data protection **mechanisms**
   - Know and understand as well as
   - Being able to develop

In short: ***Honest IT security experts with their own opinion and personal strength.***

General Aims of Education in IT-security How to achieve ?

1. Education to **honesty** and a **realistic self-assessment**

As teacher, you should make clear

- **your strengths and weaknesses as well as**
- **your limits.**

Oral examinations:

- **Wrong answers are much worse than "I do not know".**
- **Possibility to explicitly exclude some topics at the very start of the examination (if less than 25% of each course, no downgrading of the mark given).**
- **Offer to start with a favourite topic of the examined person.**
- **Examining into depth until knowledge ends – be it of the examiner or of the examined person.**

General Aims of Education in IT-security How to achieve ?

1. Education to **honesty** and a **realistic self-assessment**
2. Encouraging realistic **assessment of others**, e.g., other persons, companies, organizations

Tell, discuss, and evaluate case examples and anecdotes taken from first hand experience.

General Aims of Education in IT-security — How to achieve ?

1. Education to **honesty** and a **realistic self-assessment**
2. Encouraging realistic **assessment of others**, e.g., other persons, companies, organizations
3. Ability to gather **security and data protection requirements**
   - Realistic protection goals
   - Realistic attacker models / trust models

Tell, discuss, and evaluate case examples (and anecdotes) taken from first hand experience.

Students should develop scenarios and discuss them with each other.

General Aims of Education in IT-security How to achieve ?

1. Education to **honesty** and a **realistic self-assessment**
2. Encouraging realistic **assessment of others**, e.g., other persons, companies, organizations
3. Ability to gather **security and data protection requirements**
   - Realistic protection goals
   - Realistic attacker models / trust models
4. **Validation** and **verification**, including their practical and theoretical **limits**

Work on case examples and discuss them.

Anecdotes!

General Aims of Education in IT-security — How to achieve ?

1. Education to **honesty** and a **realistic self-assessment**
2. Encouraging realistic **assessment of others**, e.g., other persons, companies, organizations
3. Ability to gather **security and data protection requirements**
   - Realistic protection goals
   - Realistic attacker models / trust models
4. **Validation** and **verification**, including their practical and theoretical **limits**
5. Security and data protection **mechanisms**
   - Know and understand as well as
   - Being able to develop

Whatever students can discover by themselves in exercises should not be taught in lectures.

…but no this way!

First stupid and silly
now wise as Goethe
this has accomplished
the power of the
Nuremberg Funnel

Nuremberg Funnel
(German: **Nürnberger Trichter**)

Postcard from around 1940

Offers by the Chair of Privacy and Data Security

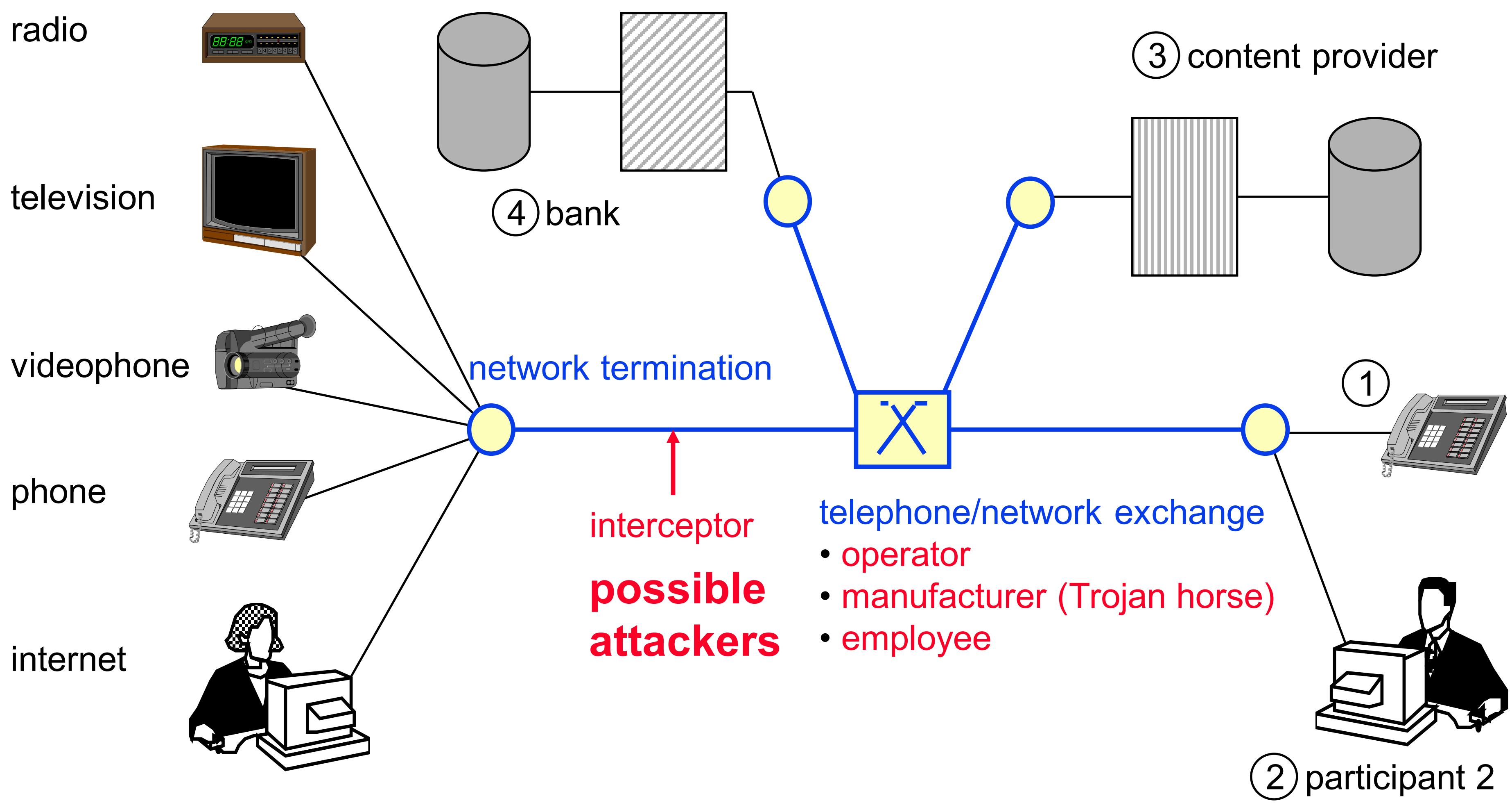
- **Interactions** between **IT-systems** and **society**, e.g., conflicting legitimate interests of different actors, privacy problems, vulnerabilities ...
- Understand **fundamental security weaknesses** of today's IT-systems
- Understand what **Multilateral security** means, how it can be characterized and achieved
- Deepened knowledge of the important tools to enable security in distributed systems based on **cryptography**
- Deepened knowledge in **error-free transmission and playback**
- Basic knowledge in **fault tolerance**
- Considerations in **building systems**: expenses vs. performance vs. security
- Basic knowledge in the relevant **legal regulations**

Aims of Education: Offers by other chairs

- Deepened knowledge **security in operating systems**
- **Verification** of OS kernels
- Deepened knowledge in **fault tolerance**
- Deepened knowledge in **trusted execution environments**

Table of Contents

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 ?

History of Communication Networks (1)

1833 First **electromagnetic telegraph**
1858 First **cable link between Europe and North America**
1876 **Phone operating across** a 8,5 km long **test track**
1881 First regional **switched phone network**
1900 Beginning of **wireless telegraphy**
1906 Introduction of **subscriber trunk dialing** in Germany, realized by two-motion selector, i.e., the first fully automatic telephone exchange through electro-mechanics
1928 Introduction of a telephone service Germany-USA, via radio
1949 First working **von-Neumann-computer**
1956 First **transatlantic telephone line**
1960 First **communications satellite**
1967 The **datex network** of the German Post starts operation, i.e., the first communication network realized particularly for computer communication (computer network of the first type). The transmission was digital, the switching by computers (computer network of the second type).
1977 Introduction of the electronic dialing system **(EWS)** for telephone through the German Post, i.e., the first telephone switch implemented by computer (computer network of the second type), but still analogue transmission

History of Communication Networks (2)

1981 First personal computer (PC) of the computer family (**IBM PC**), which is widely used in private households

1982 investments in phone network **transmission systems** are increasingly in **digital** technology

1985 Investments in telephone switches are increasingly in computer-controlled technology. Now transmission is no longer analogue, but **digital signals are switched and transmitted** (completed 1998 in Germany)

1988 Start-up of the **ISDN** (Integrated Services Digital Network)

1989 First pocket PC: **Atari Portfolio**; so the computer gets personal in the narrower sense and mobile

1993 **Cellular phone networks** are becoming a mass communication service

1994 **www** commercialization of the Internet

2000 **WAP-capable mobiles** for 77 € without mandatory subscription to services

2003 with IEEE 802.11b, **WLAN** (Wireless Local Area Network) and Bluetooth **WPAN** (Wireless Personal Area Network) find mass distribution

2004 **UMTS** starts in Germany

2005 VoIP (Voice over IP) is becoming a mass communication service

2007 first generation iPhone

2012 **LTE** with up to 300 MBit/s

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)

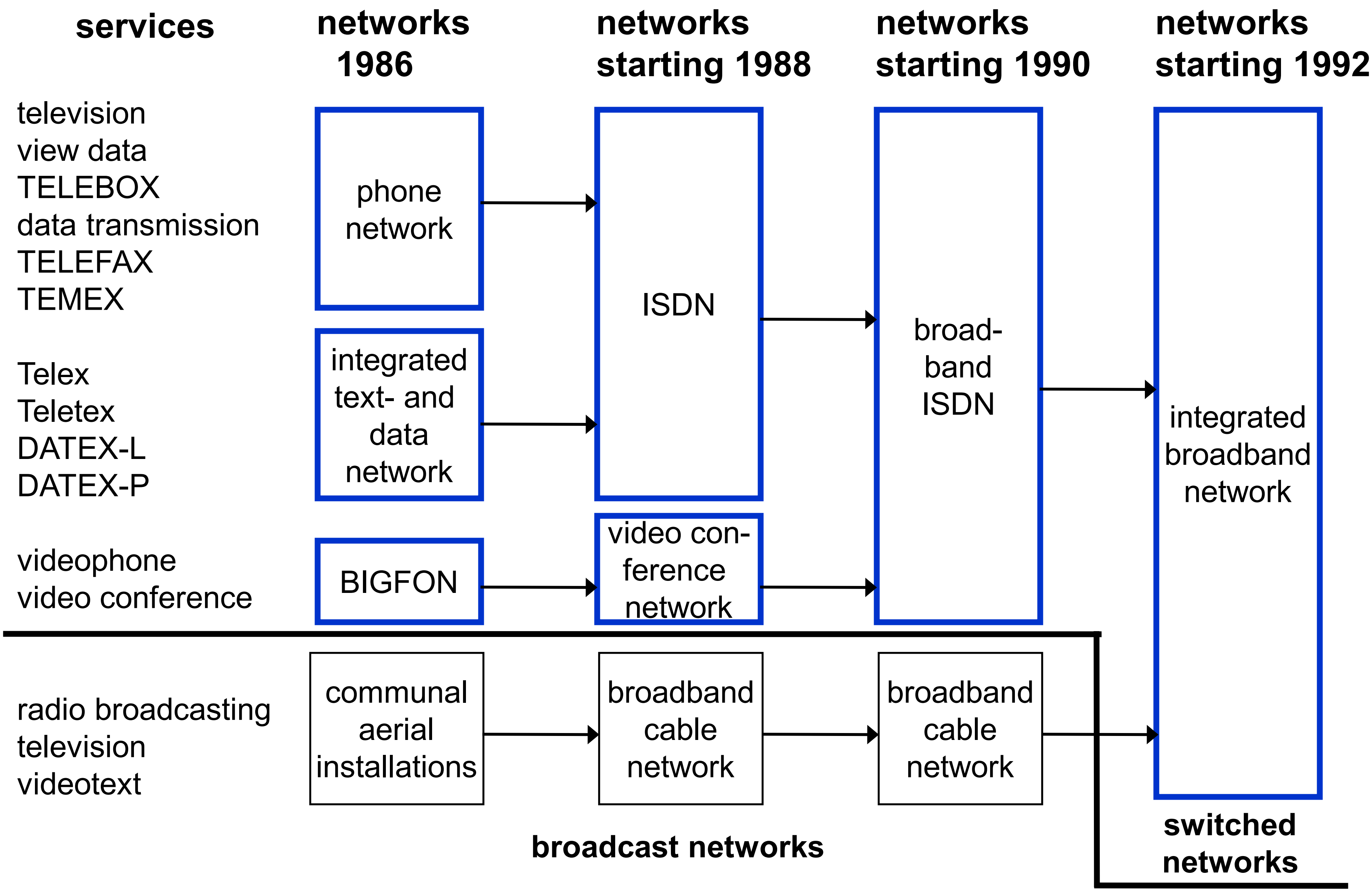
distributed system
spatial
control and implementation structure

open system $\neq$ **public** system $\neq$ **open source** system

service integrated system

digital system

Development of the fixed communication networks of the German Post (Roadmap of approx. 1982)



Threats and corresponding protection goals

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

1) unauthorized access to information — confidentiality

computer company receives medical files

2) unauthorized modification of information — integrity

undetected change of medication

≥ total correctness — ≅ partial correctness

3) unauthorized withholding of information or resources — availability for authorized users

detected failure of system

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 — confidentiality

computer company receives medical files

2) unauthorized modification of information — integrity

undetected change of medication

≥ total correctness — ≅ partial correctness

3) unauthorized withholding of information or resources — availability for authorized users

detected failure of system

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

Protection Goals: Sorting

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

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.

Unlinkability ensures that an attacker cannot sufficiently distinguish whether two or more items of interest (subjects, messages, actions, …) are related or not.

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.

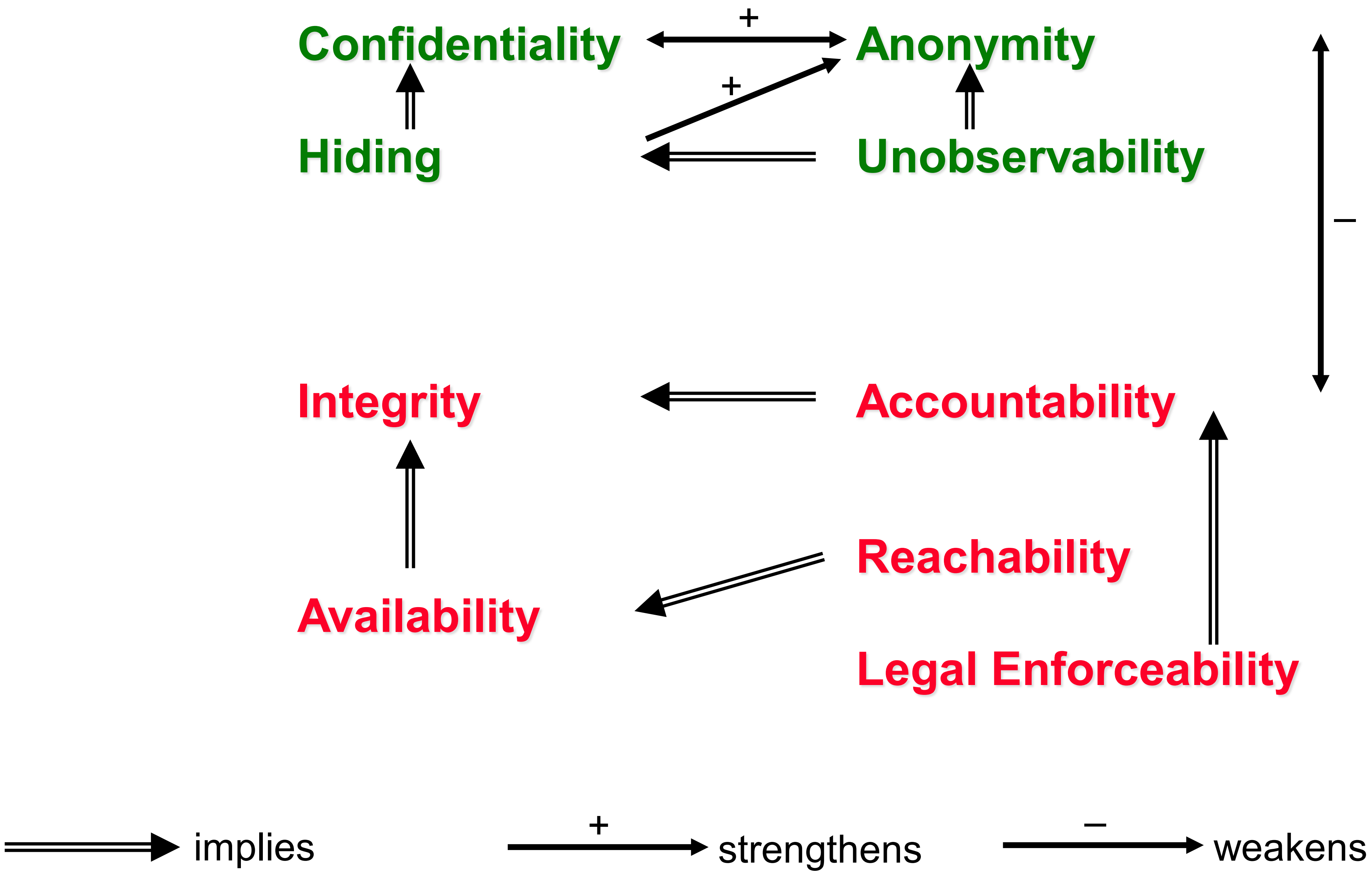
Legal enforceability ensures that a user can be held liable to fulfill his/her legal responsibilities within a reasonable period of time.

Additional Data Protection Goals: Definitions (Rost/Pfitzmann 2009)

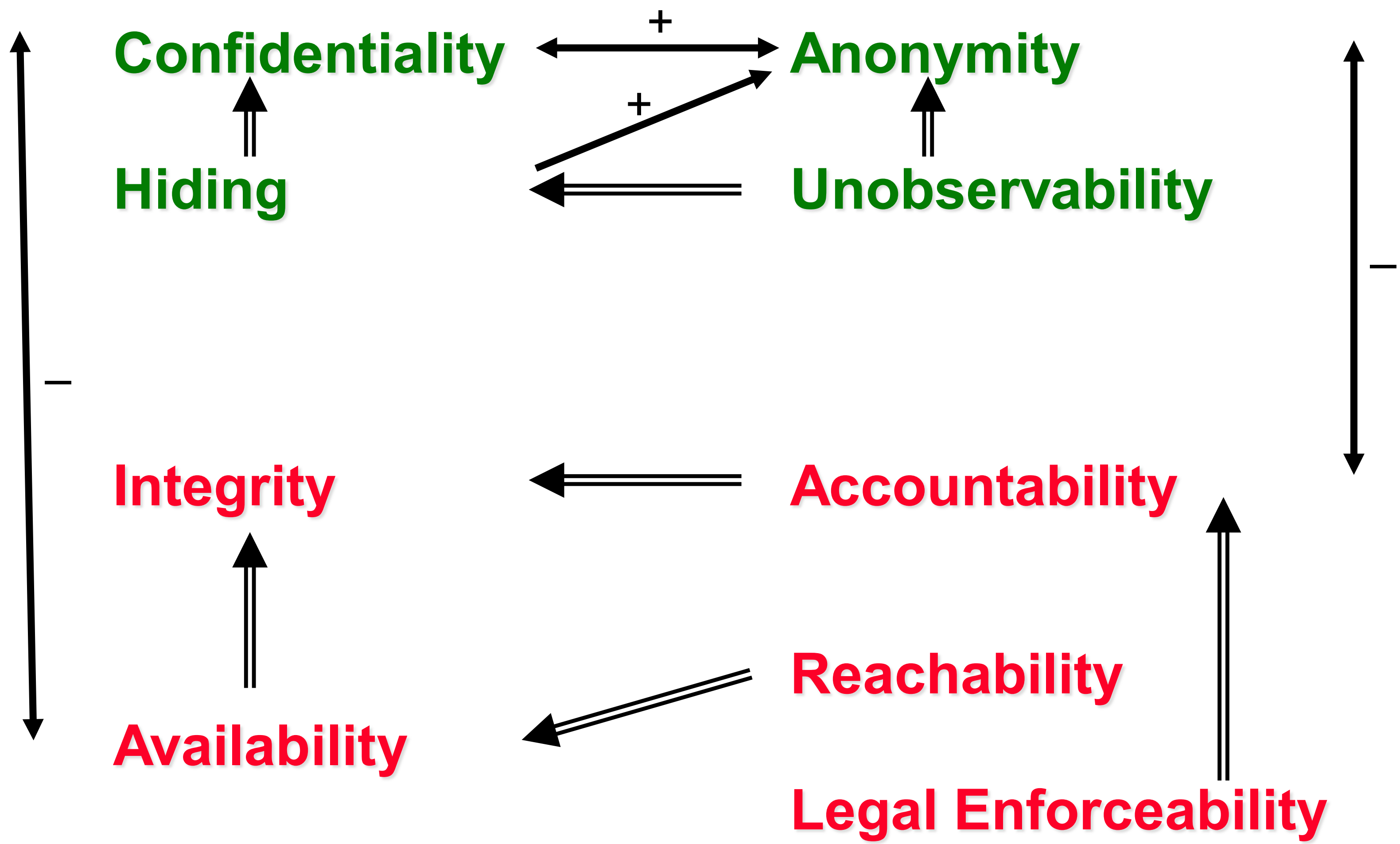
Transparency ensures that that the data collection and data processing operations can be planned, reproduced, checked and evaluated with reasonable efforts.

Intervenability ensures that the user is able to exercise his or her entitled rights within a reasonable period of time.

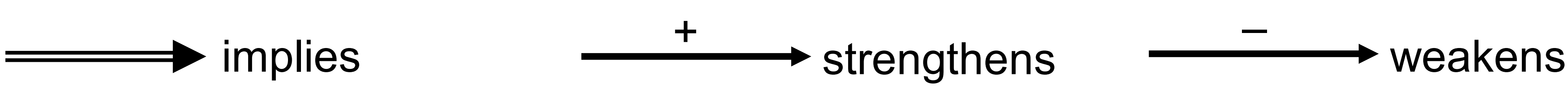
Correlations between protection goals



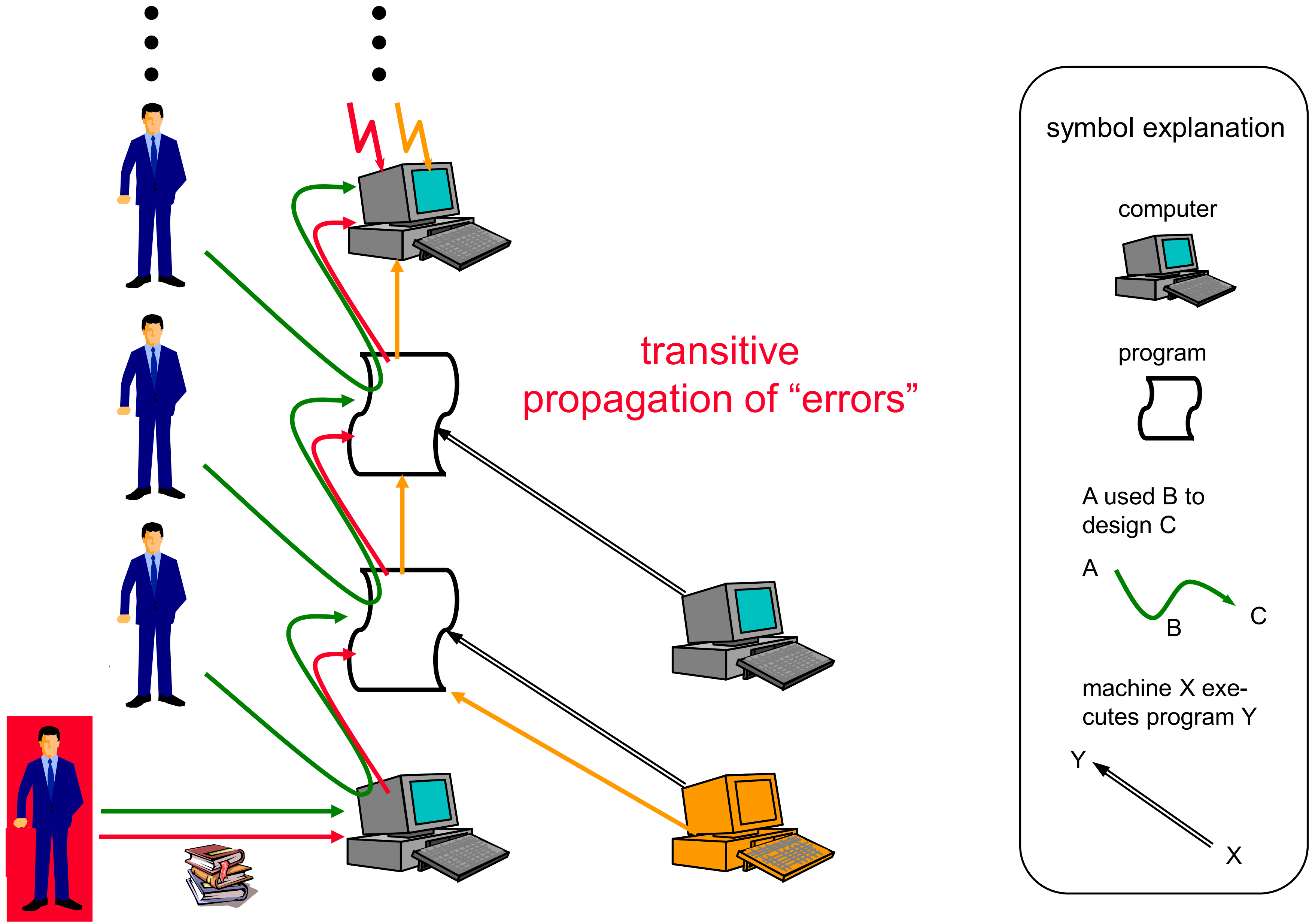
Correlations between protection goals



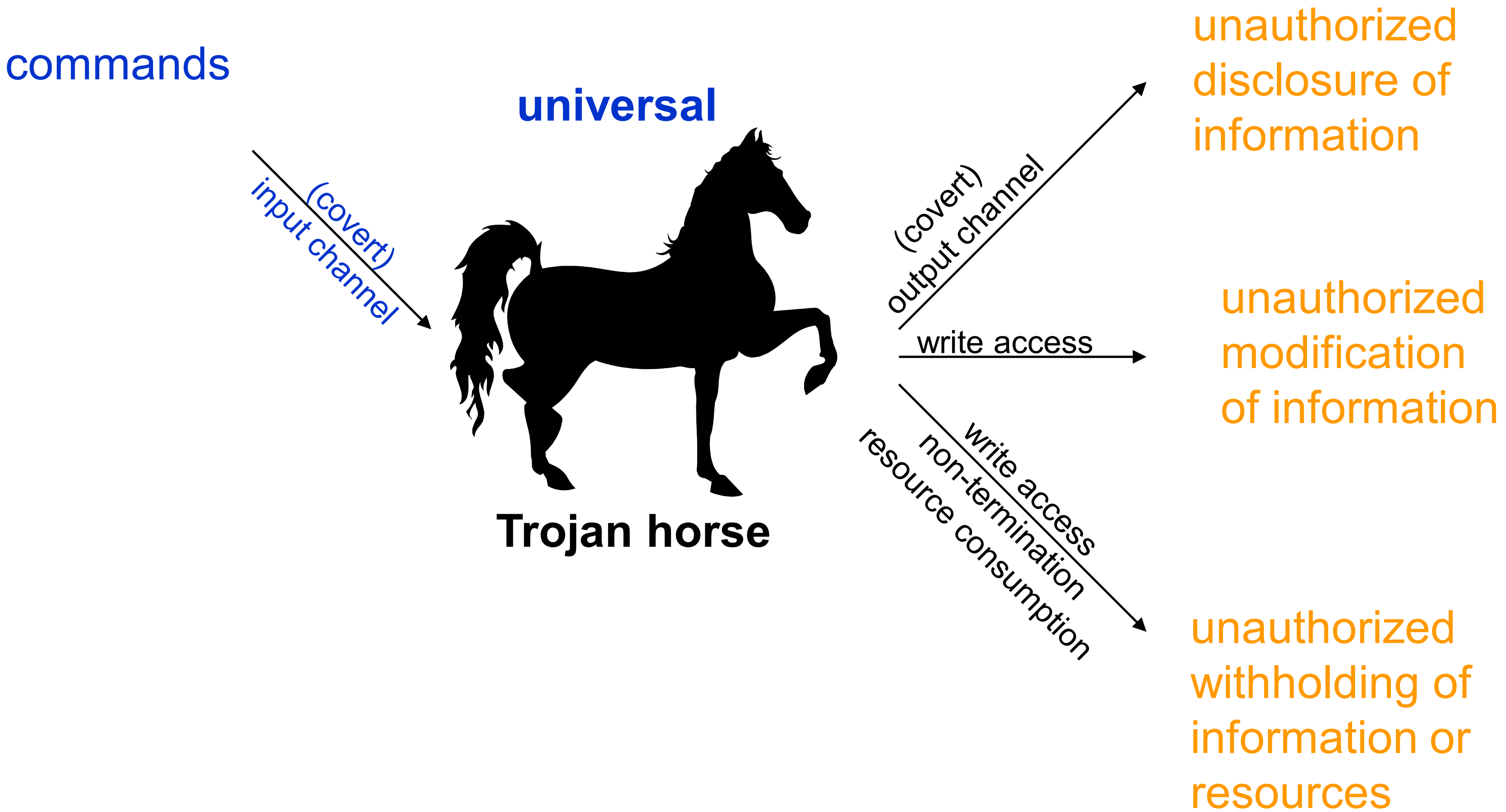
Transitive closure to be added



Transitive propagation of errors and attacks



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

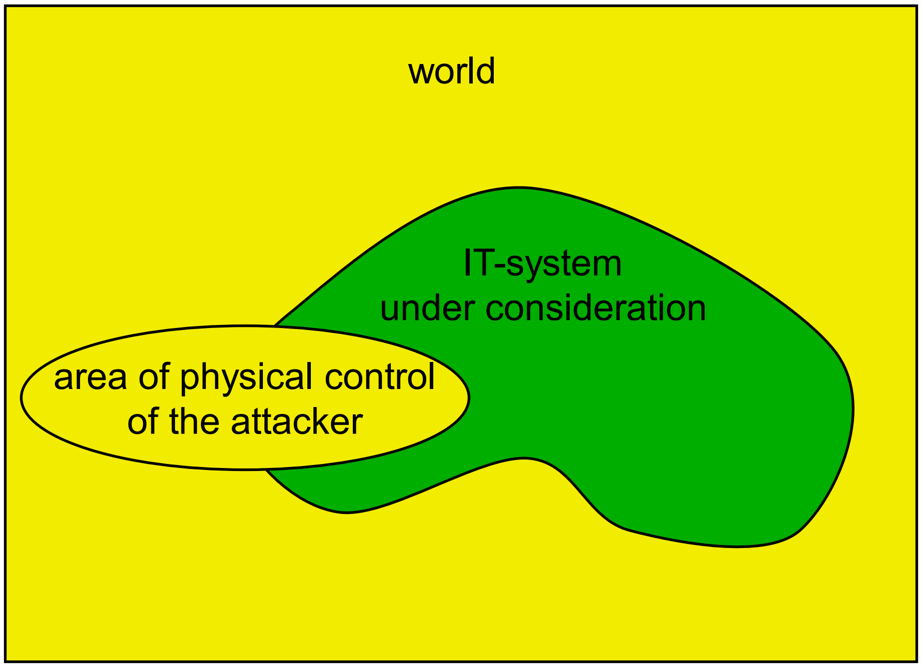
It‘s not possible to protect against an omnipotent attacker.

- 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

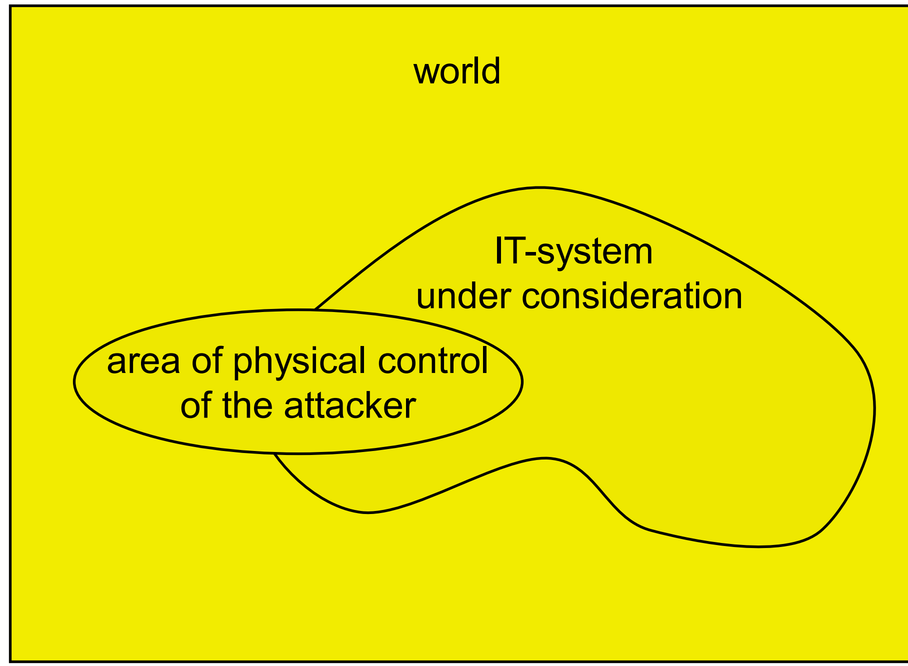
money

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).

Security in computer networks

confidentiality

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

integrity

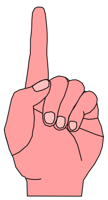
	• detect forgery	**authentication system(s)**
• **time**	• recipient can prove transmission	**sign messages**
	• sender can prove transmission	**receipt**
	• ensure payment for service	**during service by digital payment systems**

availability

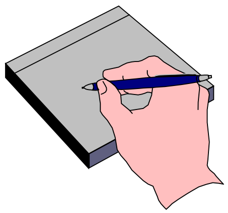
• enable communication	**diverse networks;** **fair sharing of resources**

Multilateral security

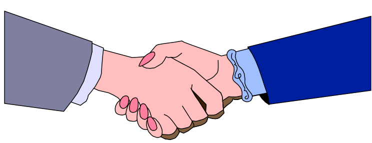
- Each party has its particular protection goals.



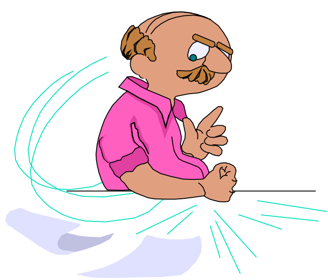
- Each party can formulate its protection goals.



- Security conflicts are recognized and compromises negotiated.



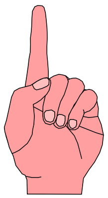
- Each party can enforce its protection goals within the agreed compromise.



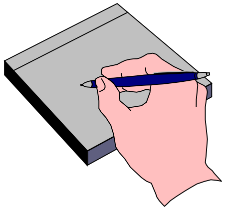
Security with minimal assumptions about others

Multilateral security (2nd version)

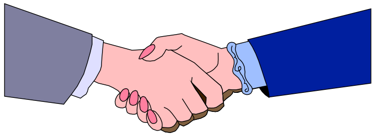
- Each party has its particular goals.



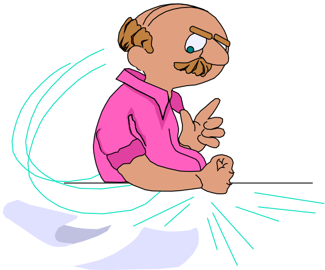
- Each party can formulate its protection goals.



- Security conflicts are recognized and compromises negotiated.



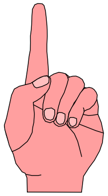
- Each party can enforce its protection goals within the agreed compromise.



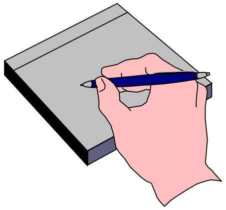
Security with minimal assumptions about others

Multilateral security (3rd version)

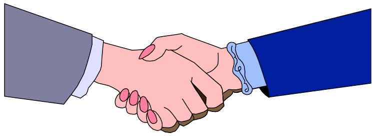
- Each party has its particular goals.



- Each party can formulate its protection goals.



- Security conflicts are recognized and compromises negotiated.



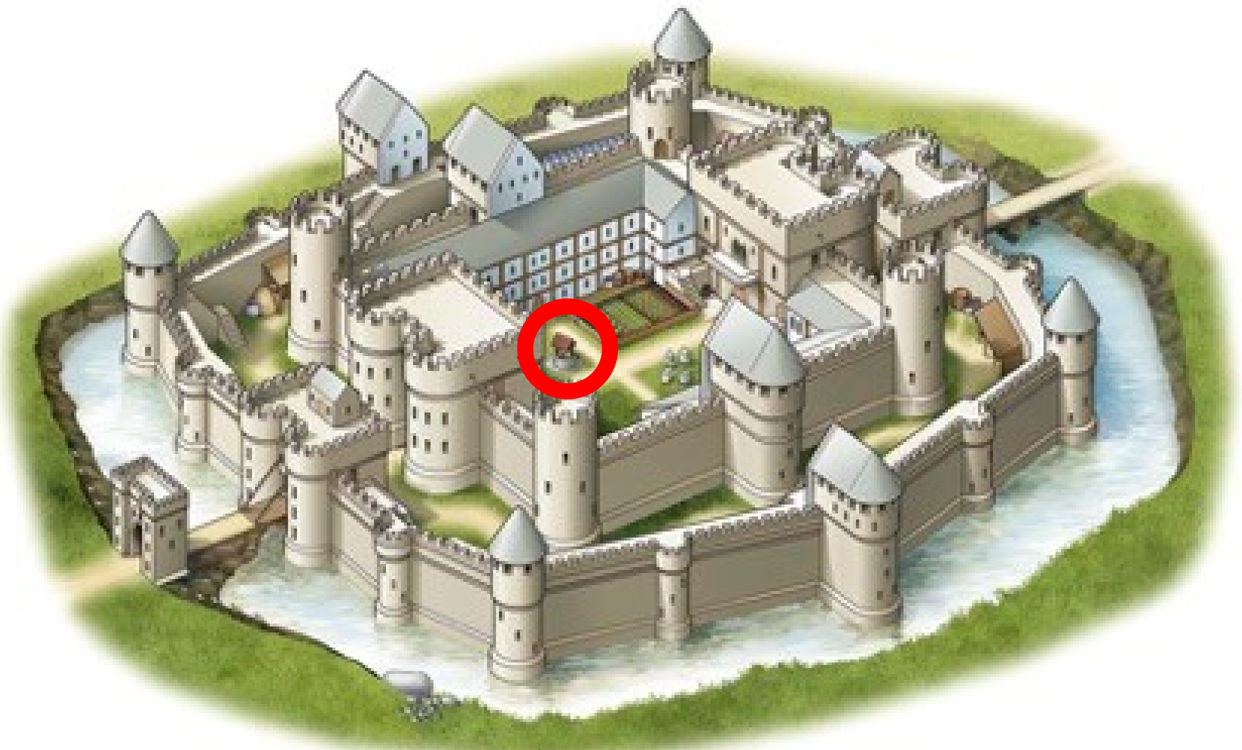
- Each party can enforce its protection goals within the agreed compromise. As far as limitations of this cannot be avoided, they equally apply to all parties.

Security with minimal assumptions about others

Multilateral Security vs. „Zero Trust“

[Rory Ward, Betsy Beyer: “BeyondCorp: A New Approach to Enterprise Security”, 2014]

- **marketing term “zero trust”**
- **fundamental idea:**
  - **trustworthy systems with minimal trust assumptions about all involved entities**
- **«Zero Trust Cybersecurity: ‘Never Trust, Always Verify’ » (NIST)**
- **current praxis:**
  - **perimeter security**
    - **firewall-like**
    - **“bad” outside**
    - **“good” inside**



[https://www.networksplus.com/defense-in-depth-a-primer/]

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**

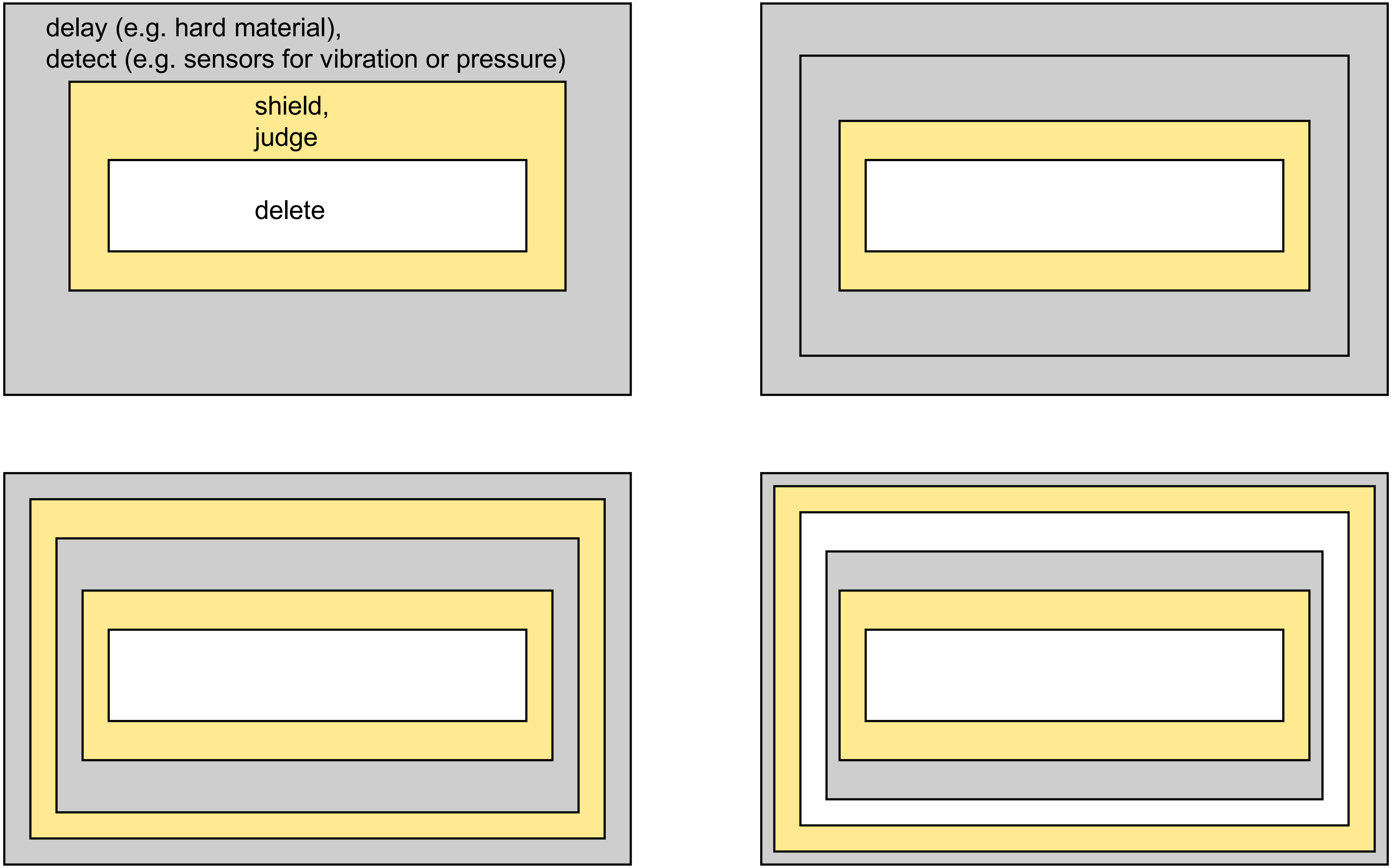
… hope the attacker cannot be at many places at the same time.

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

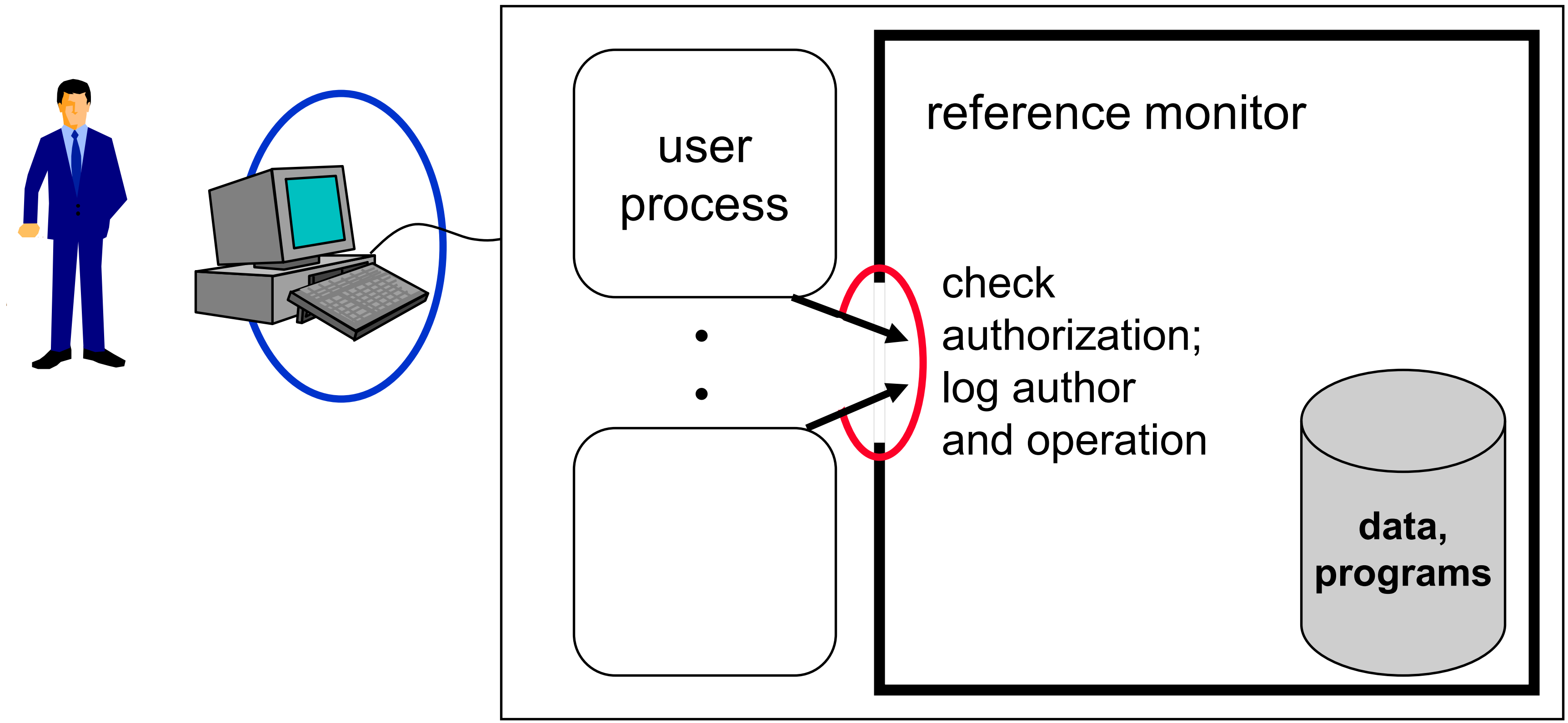
- no detection (battery missing etc.)
- shielding difficult (card is thin and flexible)
- no deletion of data intended, even when power supplied

Golden rule

Correspondence between organizational and IT structures

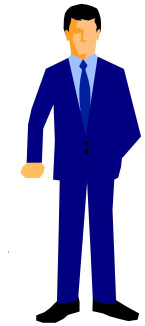
Lookahed: Why autentication: Admission and access control

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

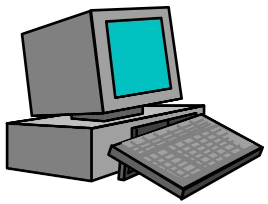
- password, passphrase
- answers to questions
- calculation results for numbers

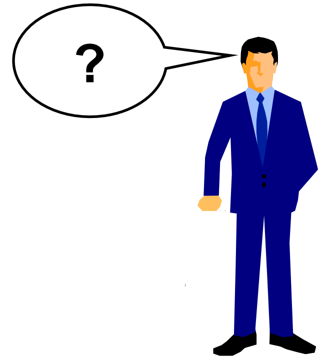
eID-card

New German eID Card

PIN protects access to chip

Identification of IT-systems by human beings



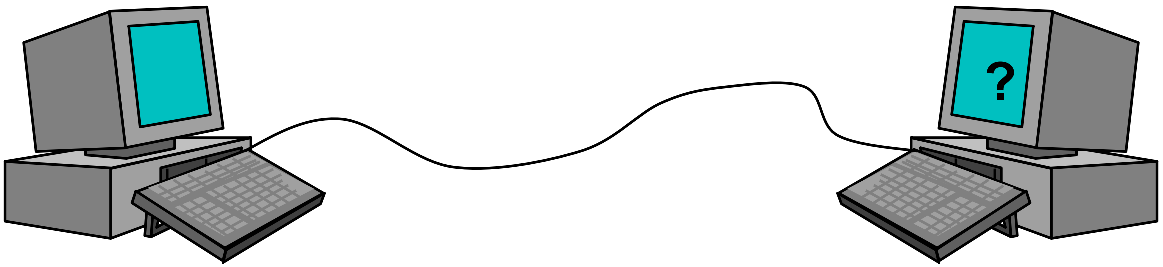


What it *is*
- casing
- seal, hologram
- pollution

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**

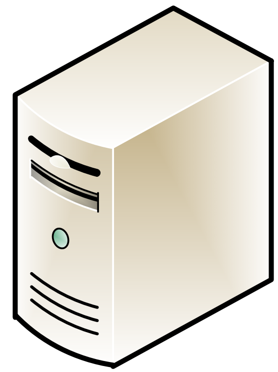
Wiring *from where*

Password based authentication

- Simple approach

User

(dog,bone)



Server

yes

grant access

?

=

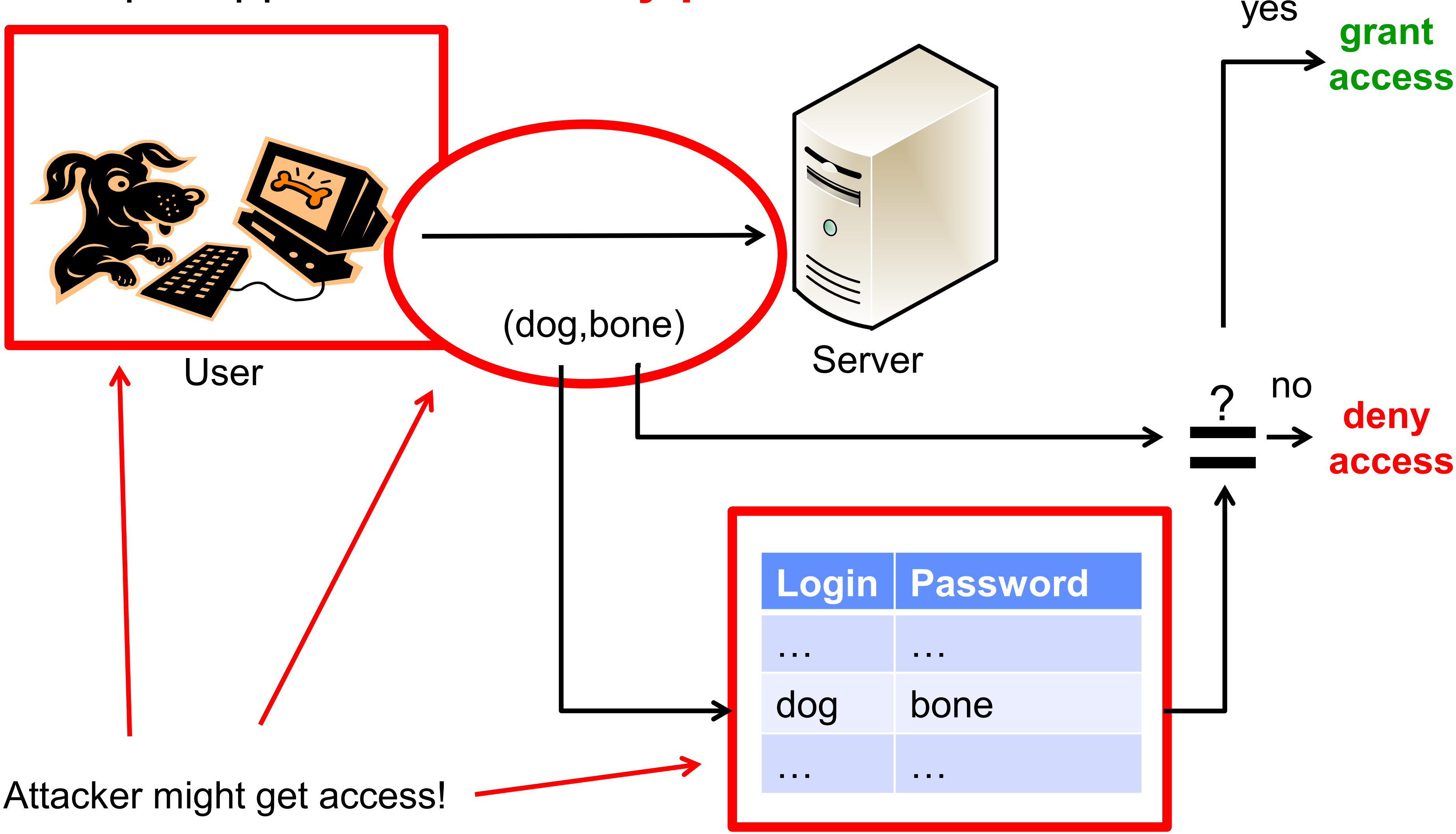
no

deny access

Login	Password
…	…
dog	bone
…	…

Password based authentication

- Simple approach – **security problems**



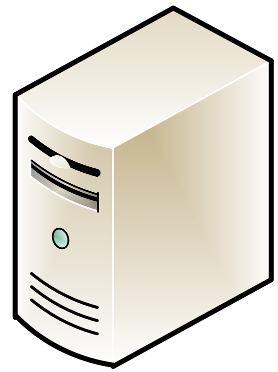
Login	Password
…	…
dog	bone
…	…

Password based authentication

- Enhanced approach using one way (hash) functions

User

(dog,bone)



Server

calculates *h*(bone)

? =

yes → **grant access**

no → **deny access**

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

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 *x*, *y* with $h(y)=h(x)$ and $y \neq x$
    - 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 x, $h(x)$ it is hard to find y with $h(y)=h(x)$ and $y \neq x$
  - 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
  - October 2012: Keccak is winner
  - FIPS 202: “SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions” (08/15)

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.*”

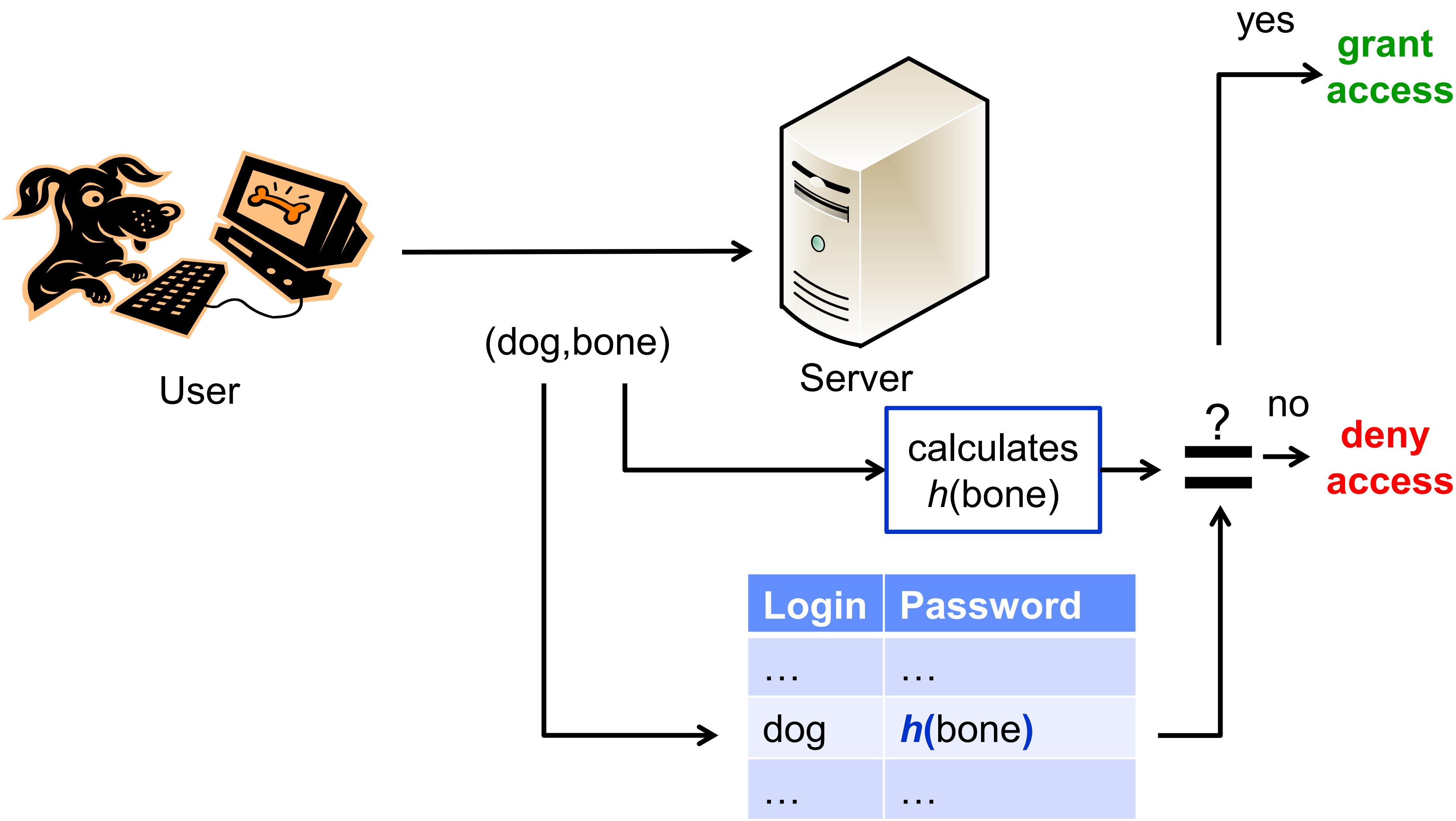
mission statement:= “*USCYBERCOM plans, coordinates, integrates, synchronizes and conducts activities to: direct the operations and defense of specified Department of Defense information networks and; prepare to, and when directed, conduct full spectrum military cyberspace operations in order to enable 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

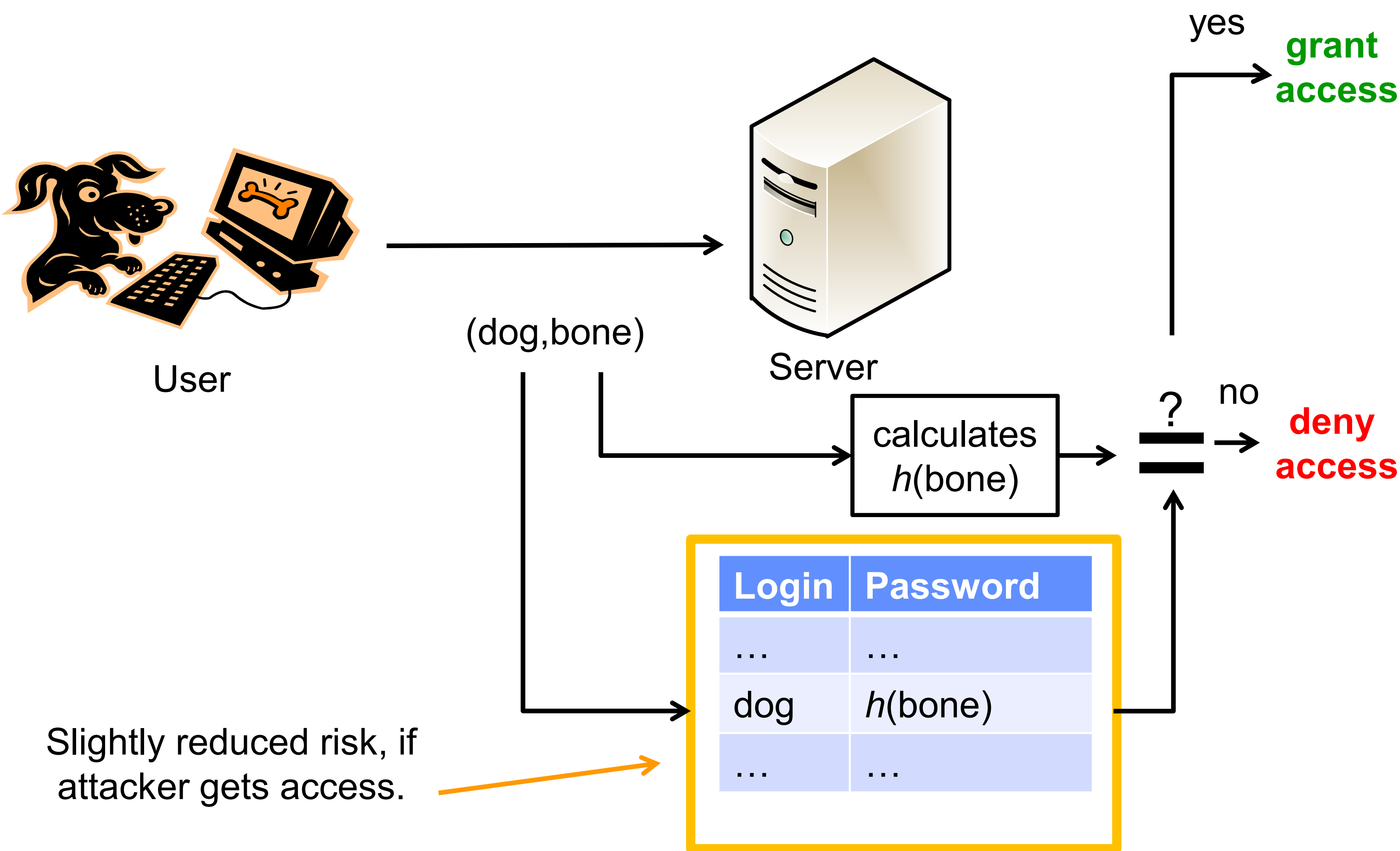
- Enhanced approach using one way (hash) functions



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

Password based authentication

- Enhanced approach using one way (hash) functions



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

Login	Password
…	…
dog	*h*(bone)
…	…

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

- 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
- countermeasure:
  - salt!
  - $h(x) \rightarrow h(\textbf{salt},x)$
  - 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

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

Login	Password
…	…
dog	*h*(bone)
…	…

Remaining problems of password based authentication based one way functions

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

- 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
- 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"

The Server as Attacker

City Whistler

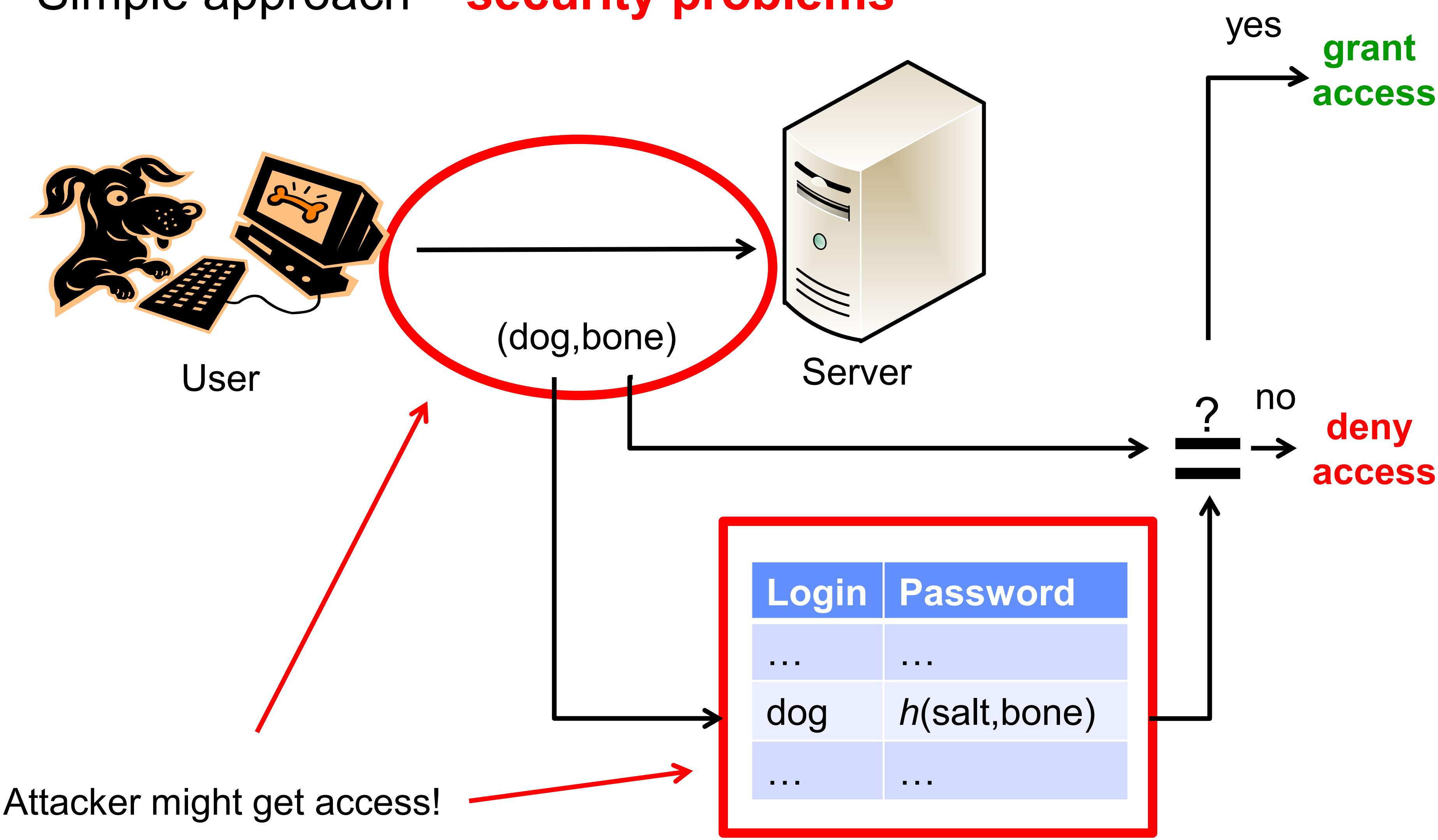
- … a new Web 2.0 service
- … for people which like city journeys
- … find cool cities and places like shops, restaurants, hotels etc.
- ... information from globe-trotters for globe-trotters
- … they can share their knowledge after secure login
- So that’s wrong?

➔ 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**



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

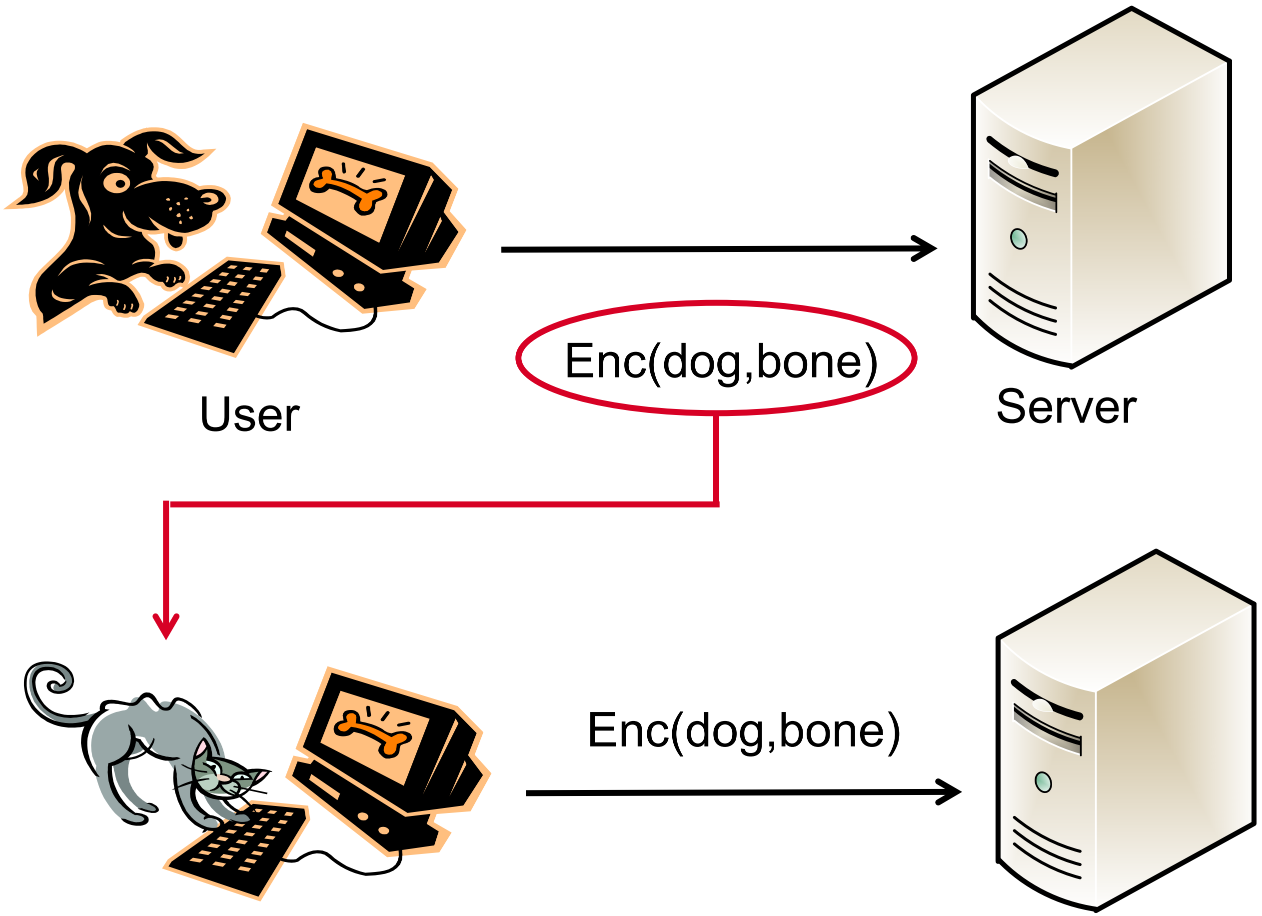
Password based authentication

- **security problems**

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

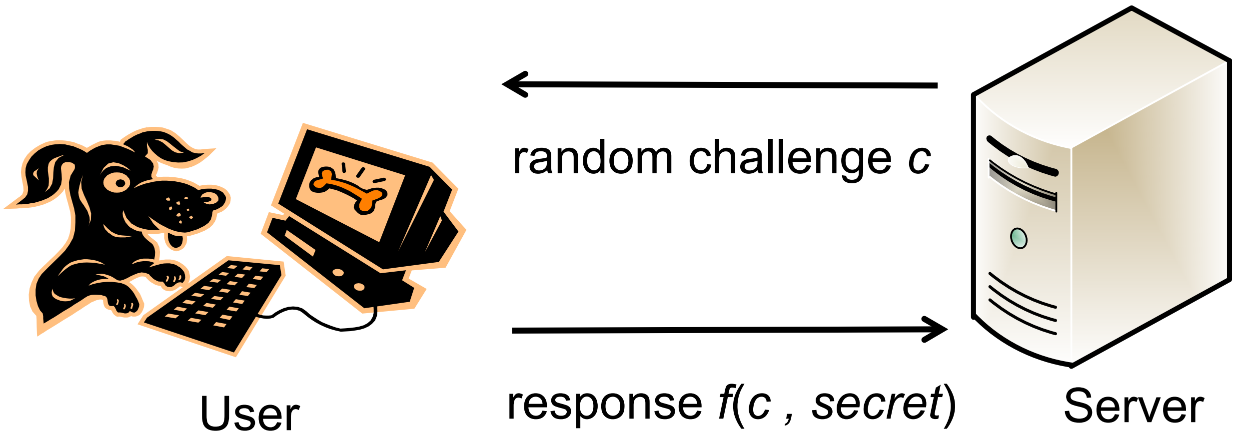
Password based authentication

- **security problem – replay attack**



Password based authentication

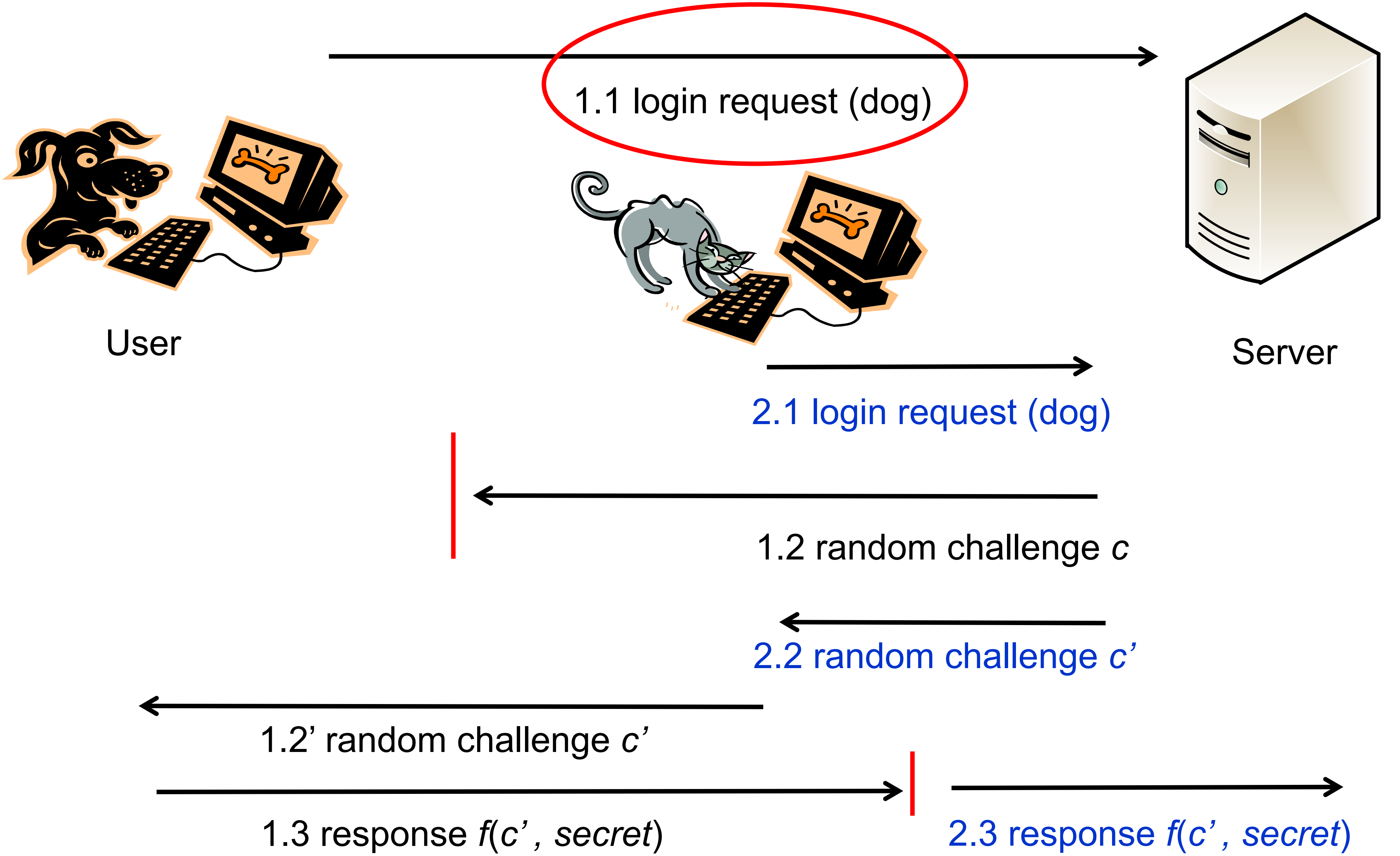
- **security problem – replay attack**
- **possible solution: challenge-response protocol**



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

Password based authentication

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

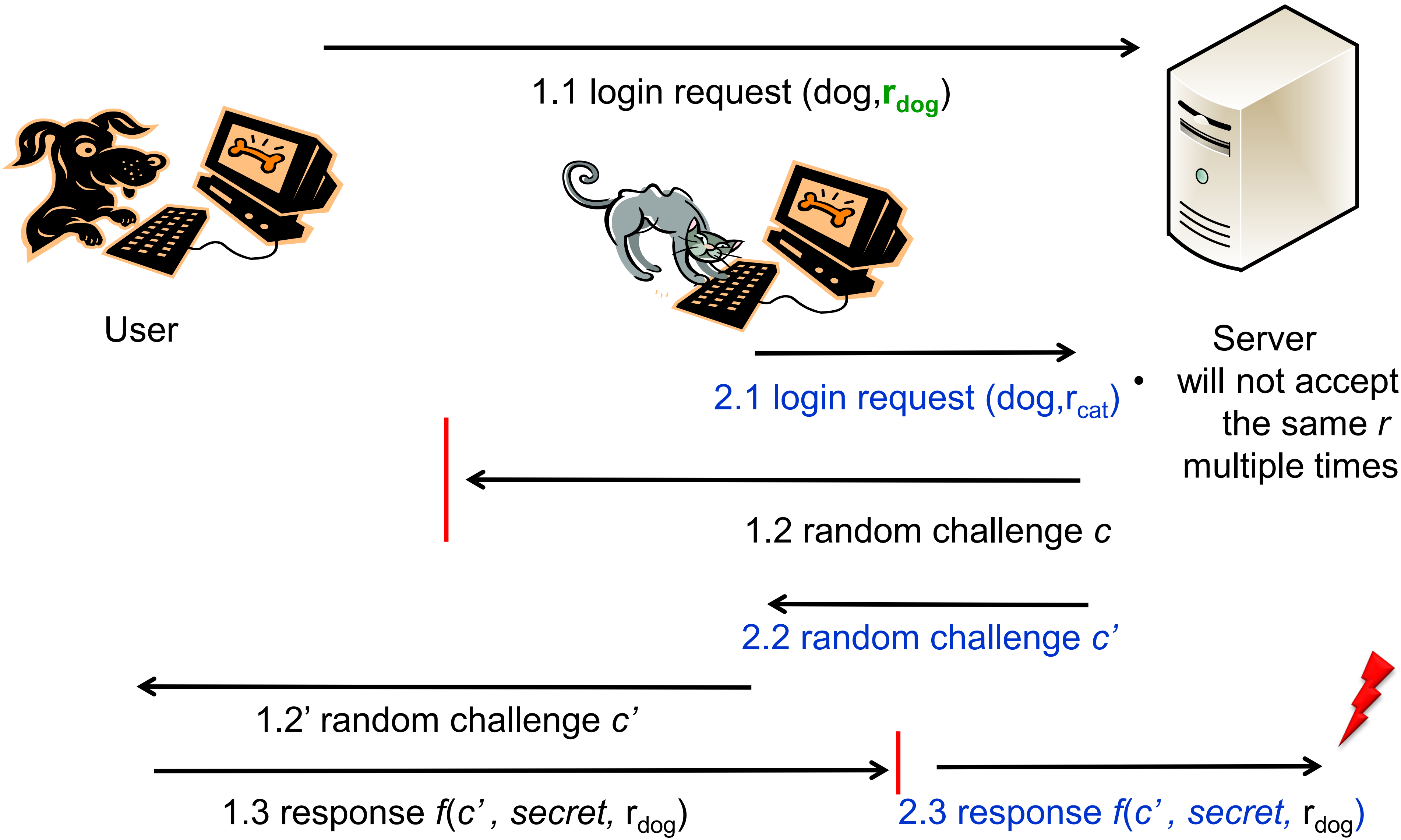


Password based authentication

- **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**



Password based authentication

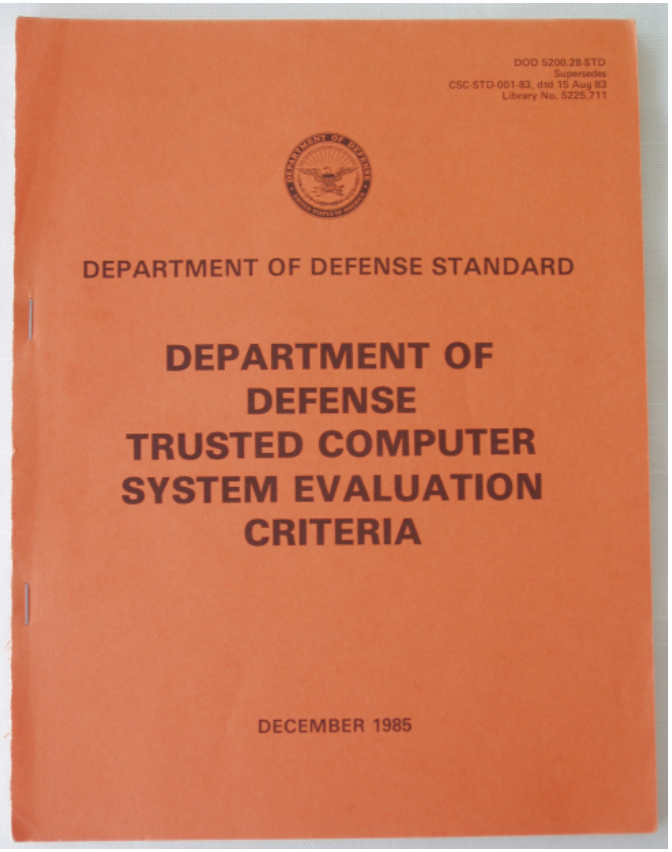
- **security problem – MITM / parallel protocol runs**
- **possible solutions:**
  - disallow parallel login protocol runs for the same user
  - make protocol runs distinguishable
- remaining security problems:
  - …

  (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

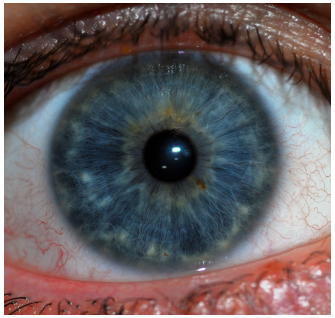
- OTP Implementations
  - hash chain based
    - Leslie Lamport: "Password Authentication with Insecure Communication"
    - users generates hash chain:
      - $h^n(\ldots h^3(h^2(h^1(\text{password}))))$
    - users sends $h^n()$ as his "password" during register procedure
    - next login user sends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

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

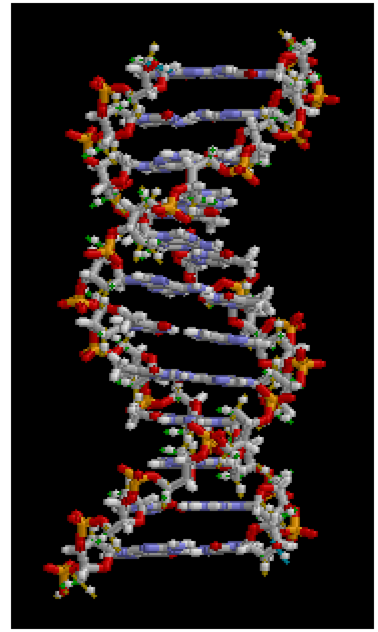
Biometrics: Physiological / Behavioural Characteristics

[Pictures are mostly from Wikipedia]



Iris / Retina

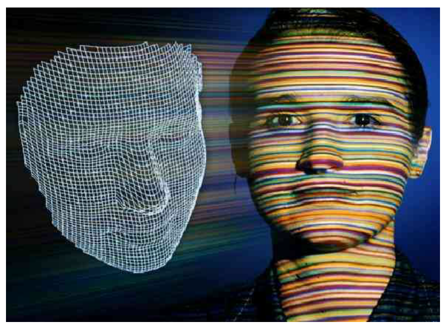
Fingerprint



DNA

Thermography:
facial thermograms

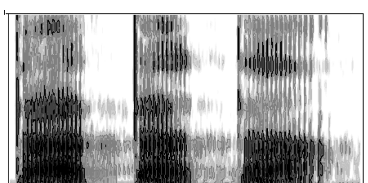
Hand geometry



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

(3D) Face geometry

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



Voice spectrogram

Handwriting:
appearance,
dynamics of writing

Gait

Biometric characteristics: Requirements

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

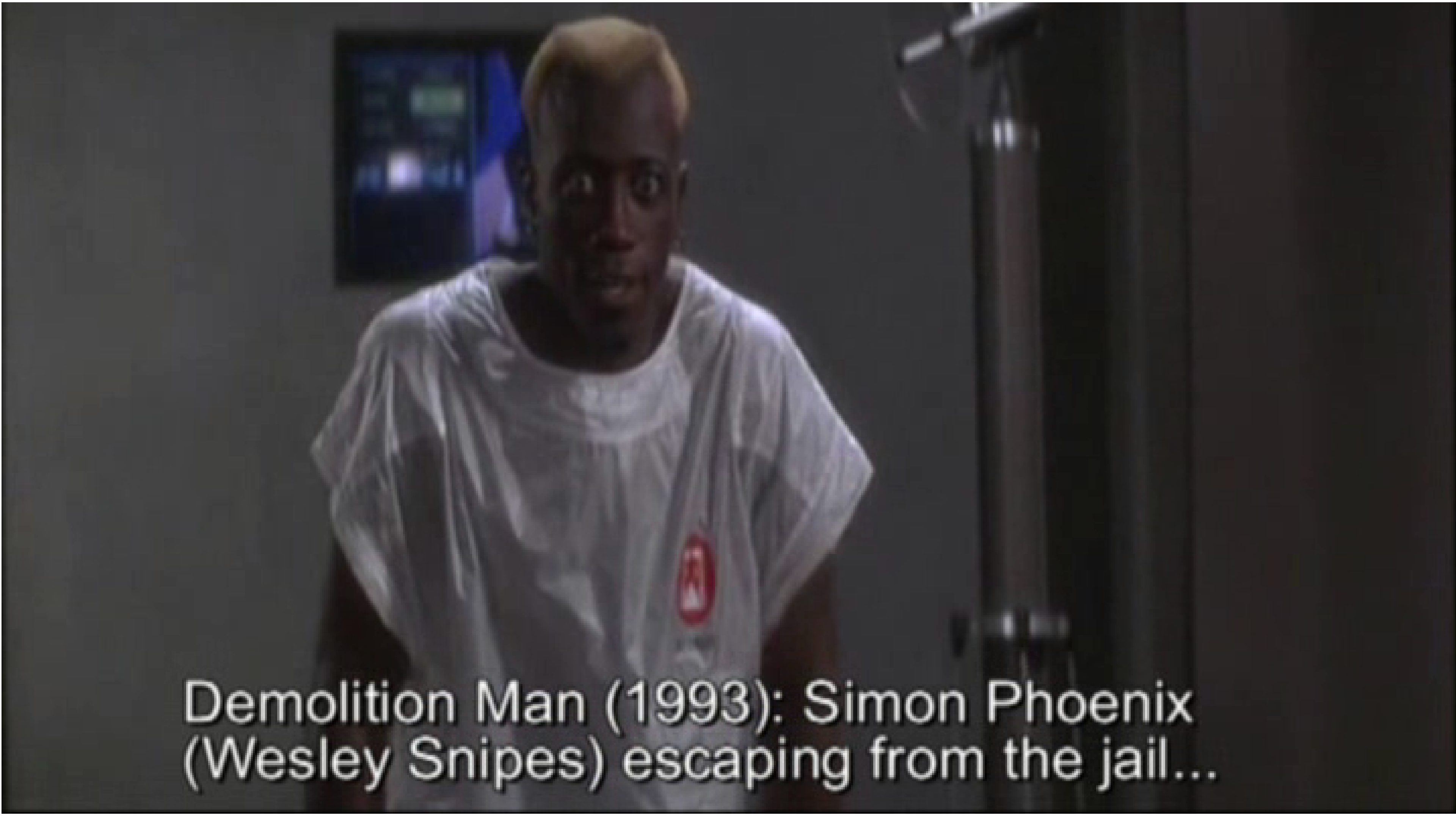
Biometrics: Pros and Cons

- 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

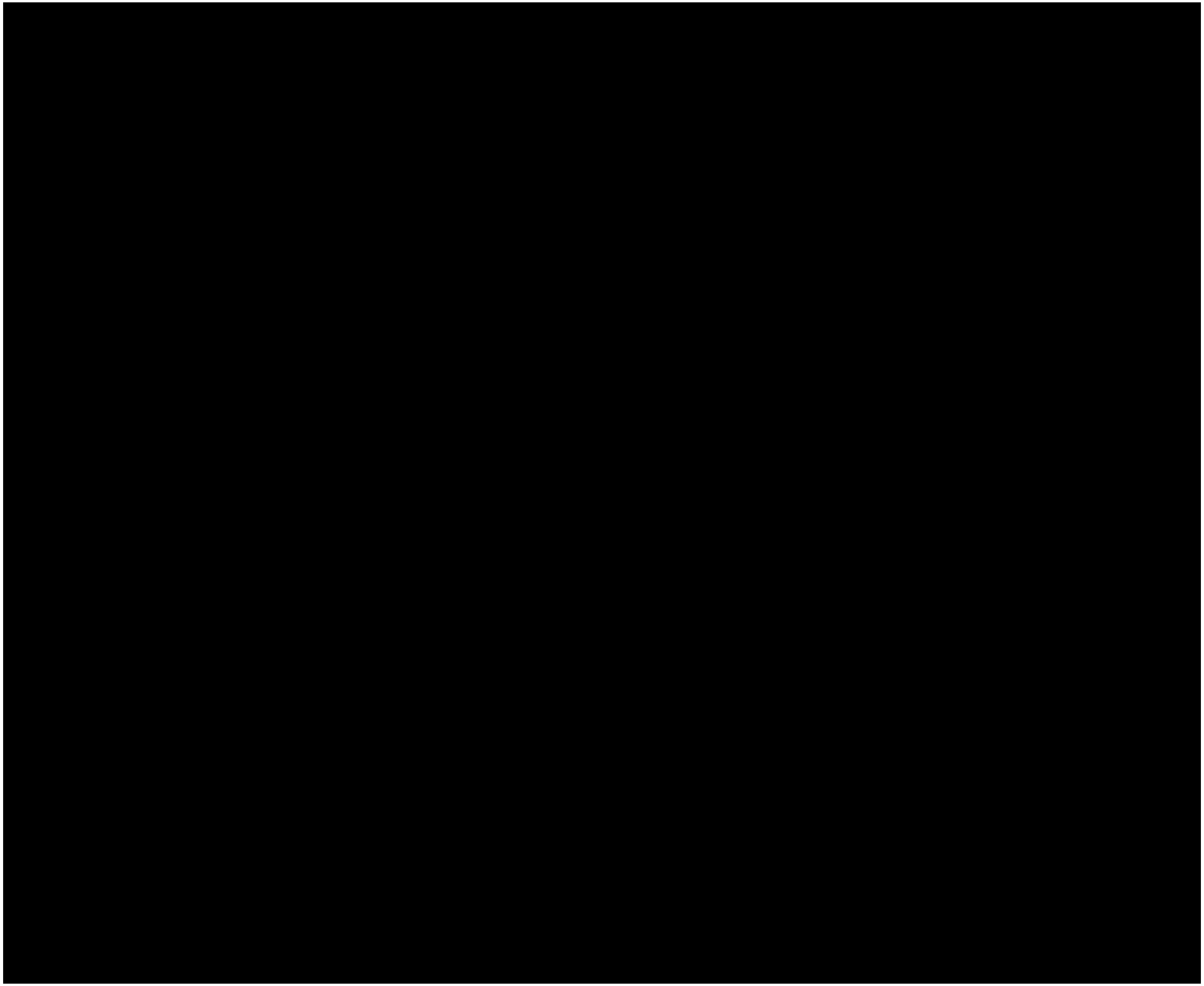
Safety Risks of Biometrics



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

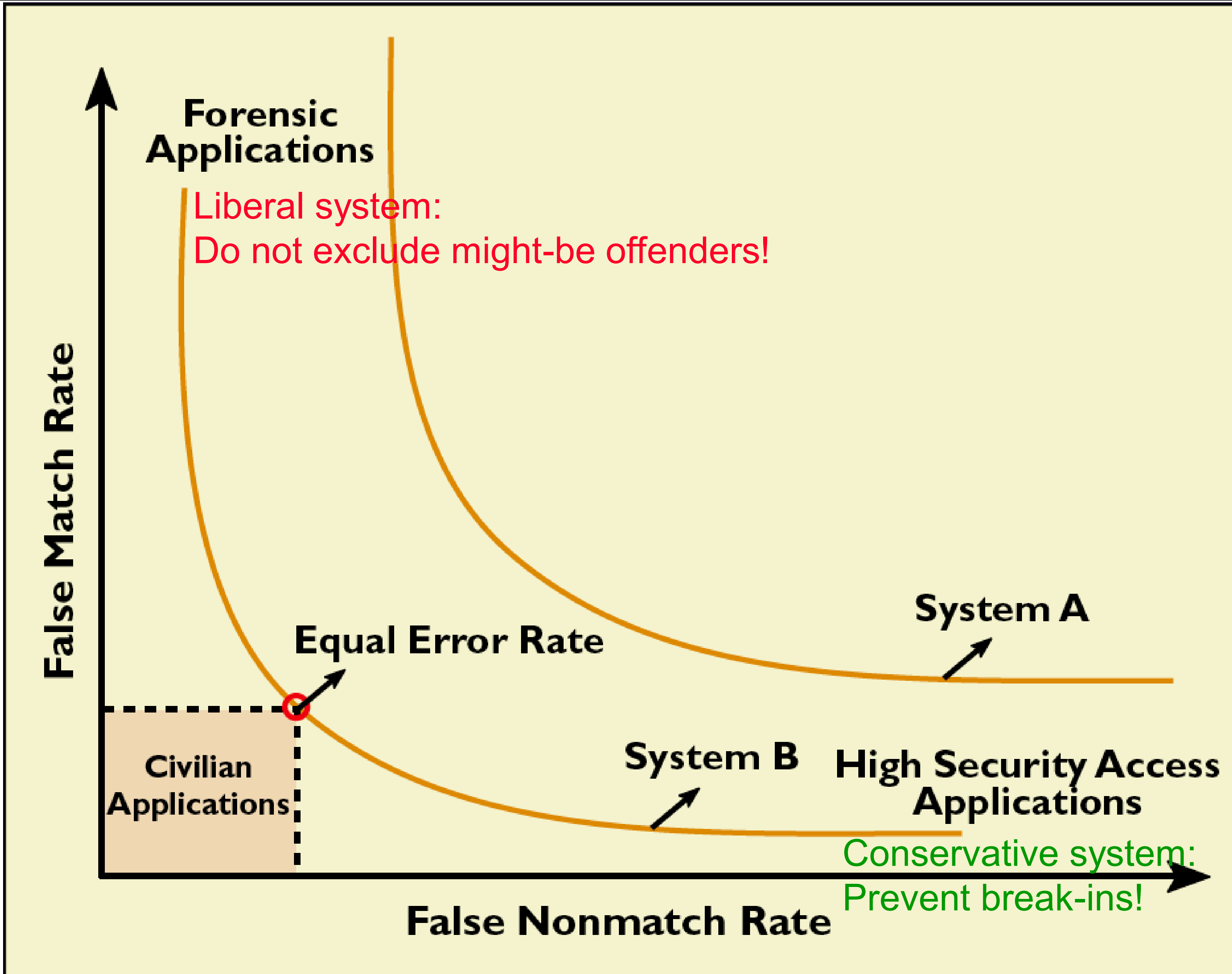


Figure taken from:

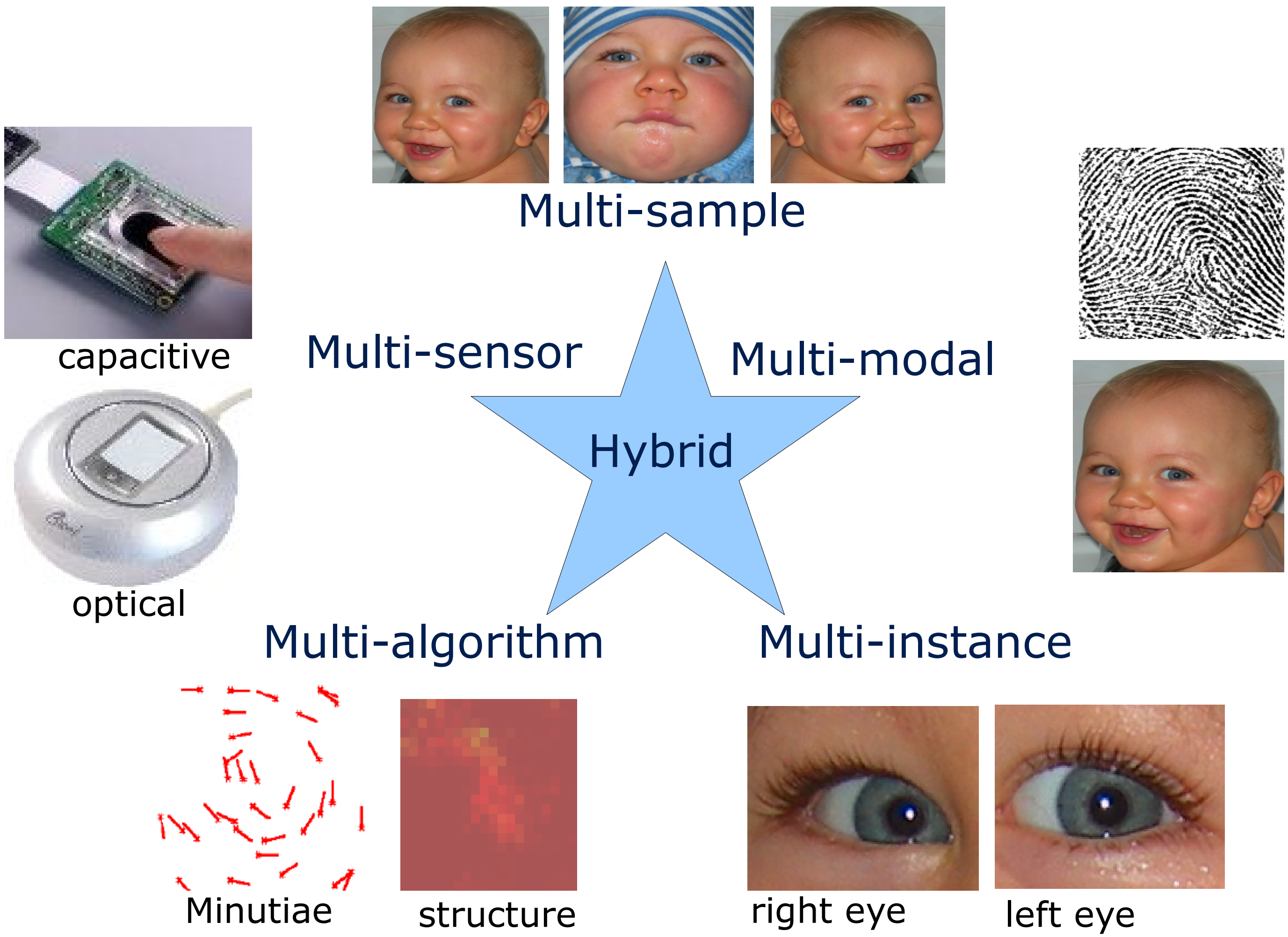
Anil Jain, Lin Hong, Sharath Pankanti: Biometric Identification; Communications of the ACM 43/2 (2000) 91-98

Low FMR causes high FNR and vice versa !

Biometric Systems: Types of Failures

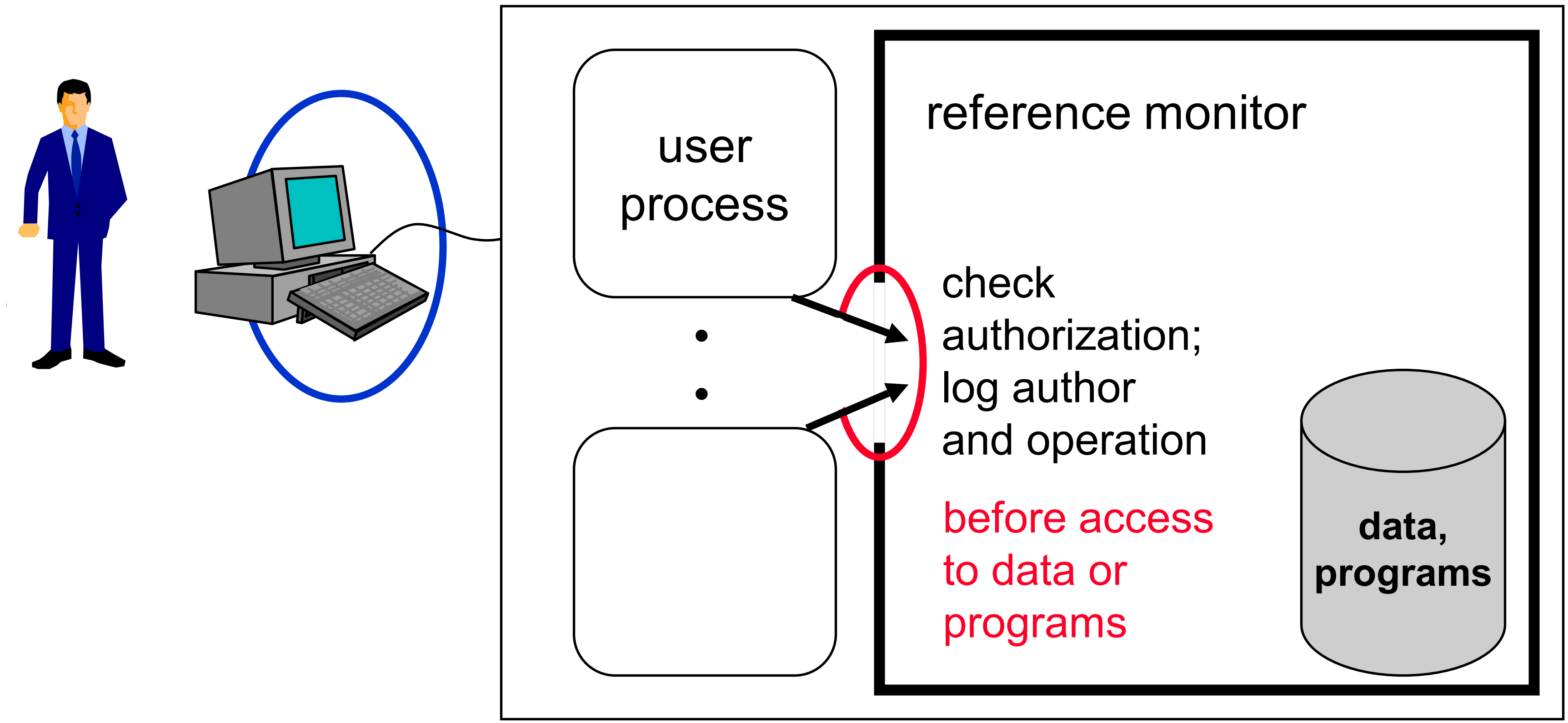
- False Accept Rate (FAR):
  - **Security problem!**
- False Reject Rate (FRR):
  - Usability / acceptance problem
- Receiver Operating Characteristic (ROC):
  - curve of FAR against FRR
- Equal Error Rate (EER):
  - error rate for FAR=FRR
  - can be seen from the ROC curve
- Failure To Enroll Rate (FTE):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

Enhanced Security: Multi-biometric Systems



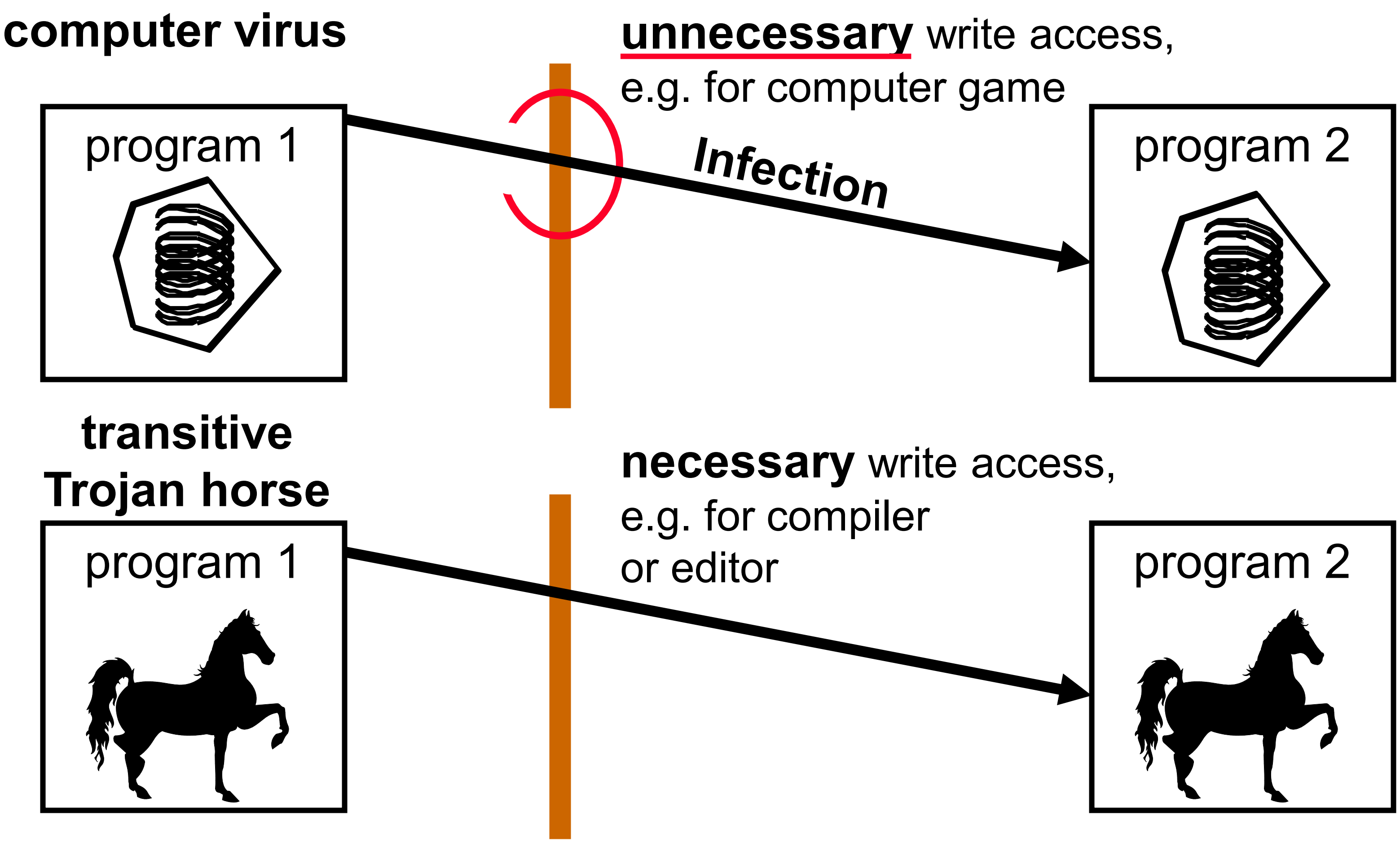
Admission and access control

Admission control communicate with authorized partners only



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

Computer virus vs. transitive Trojan horse



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 should do and what it *must not* 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

Basics of Cryptology

Achievable protection goals:

confidentiality, called **concealment**

integrity (= no *undetected* unauthorized modification of information), called **authentication**

Unachievable by cryptography:

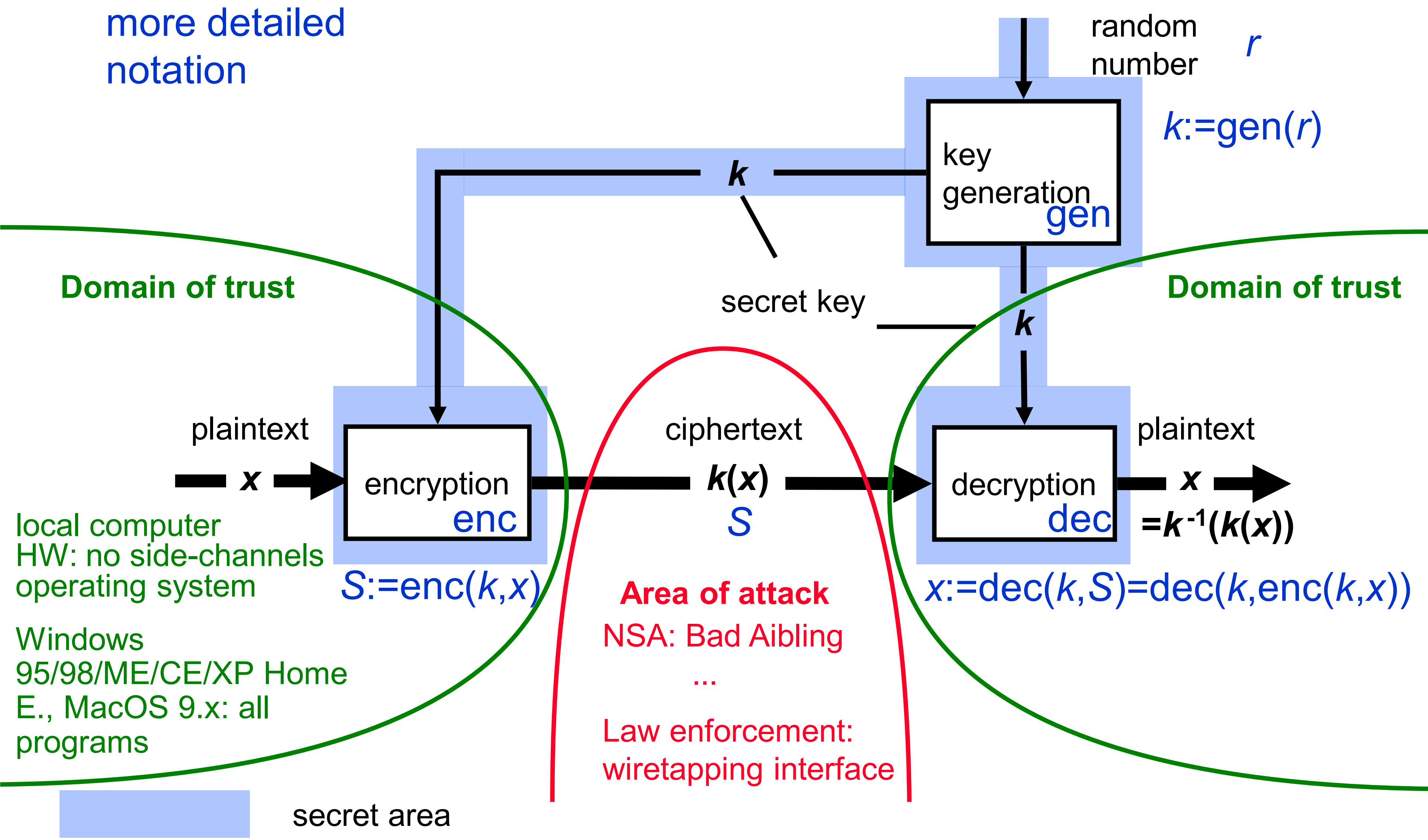
availability – at least not against strong attackers

Kerckhoff's Priniple
(1883 by Auguste Kerckhoffs: "La cryptographie militaire")

"The cipher method ***must not*** *be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience"*

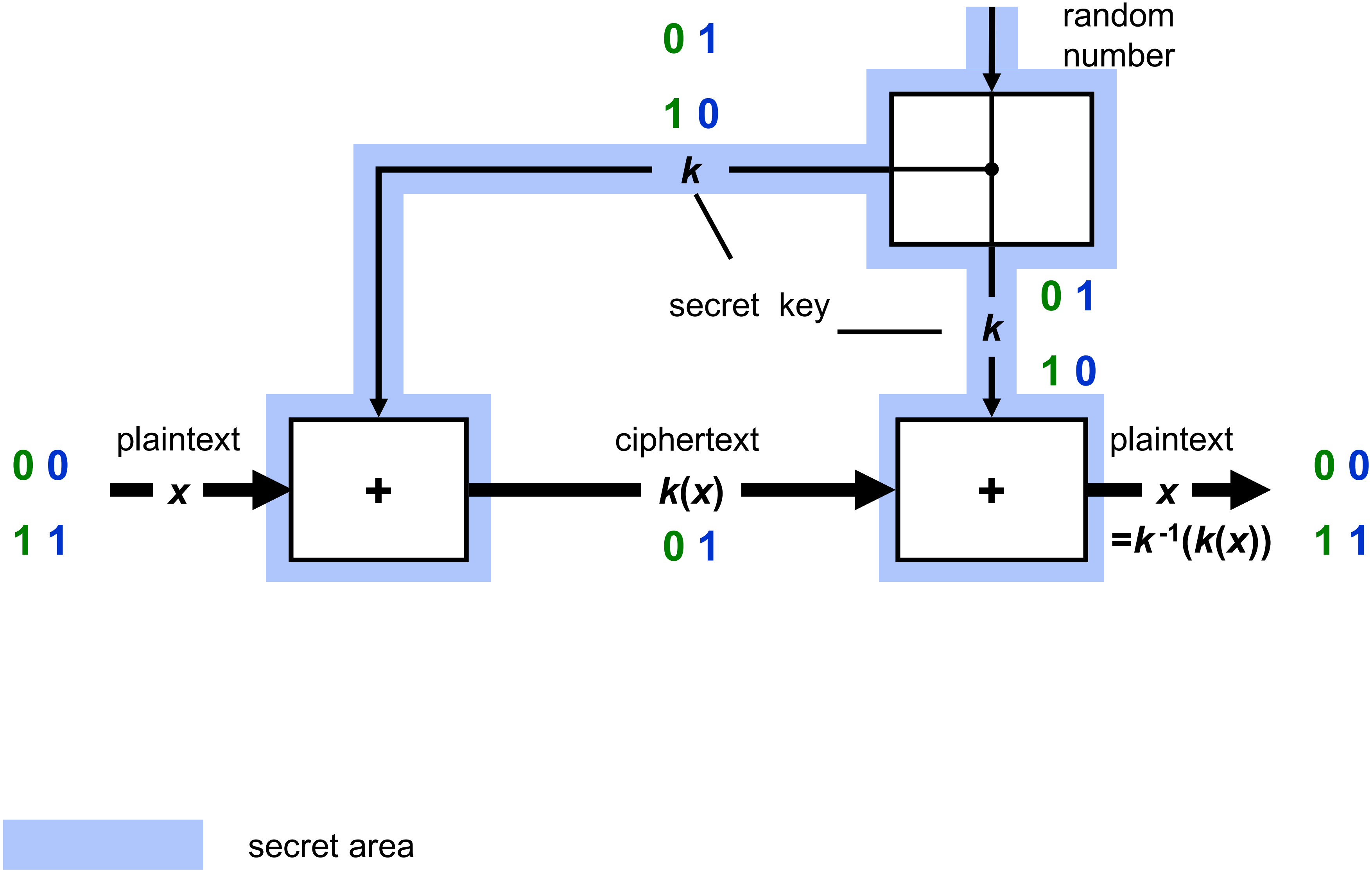
- Therefore:
  - use only publicly known and well analysed algorithms and protocols
  - do not trust "super secure" but secret algorithms
  - do not design your own algorithms (at least as longs as you are not a cryptographic expert)
  - Remember: Combining secure building blocks does not necessarily lead to a secure overall system!

Symmetric encryption system



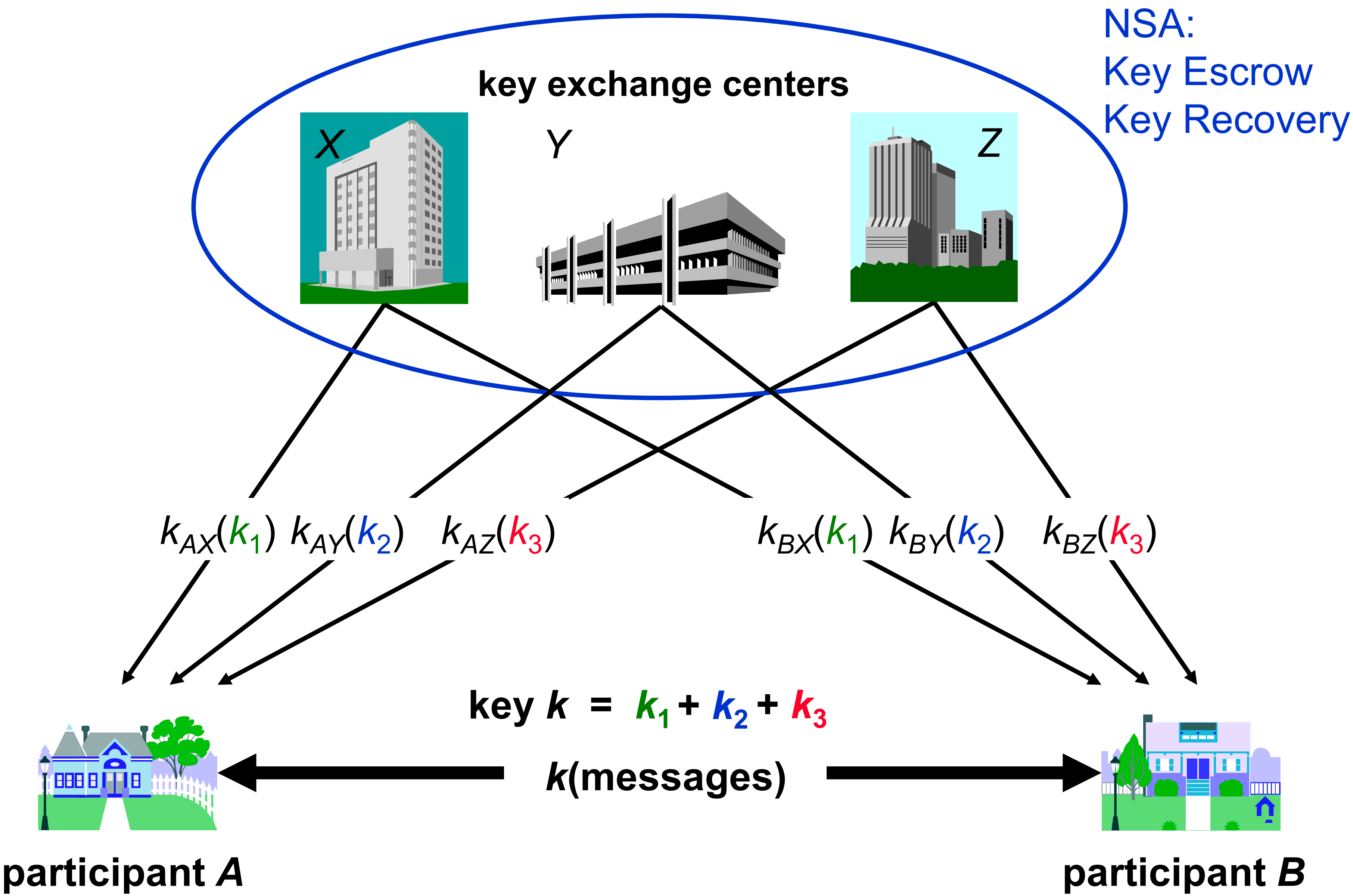
Opaque box with lock; 2 identical keys

Example: Vernam cipher (=one-time pad)

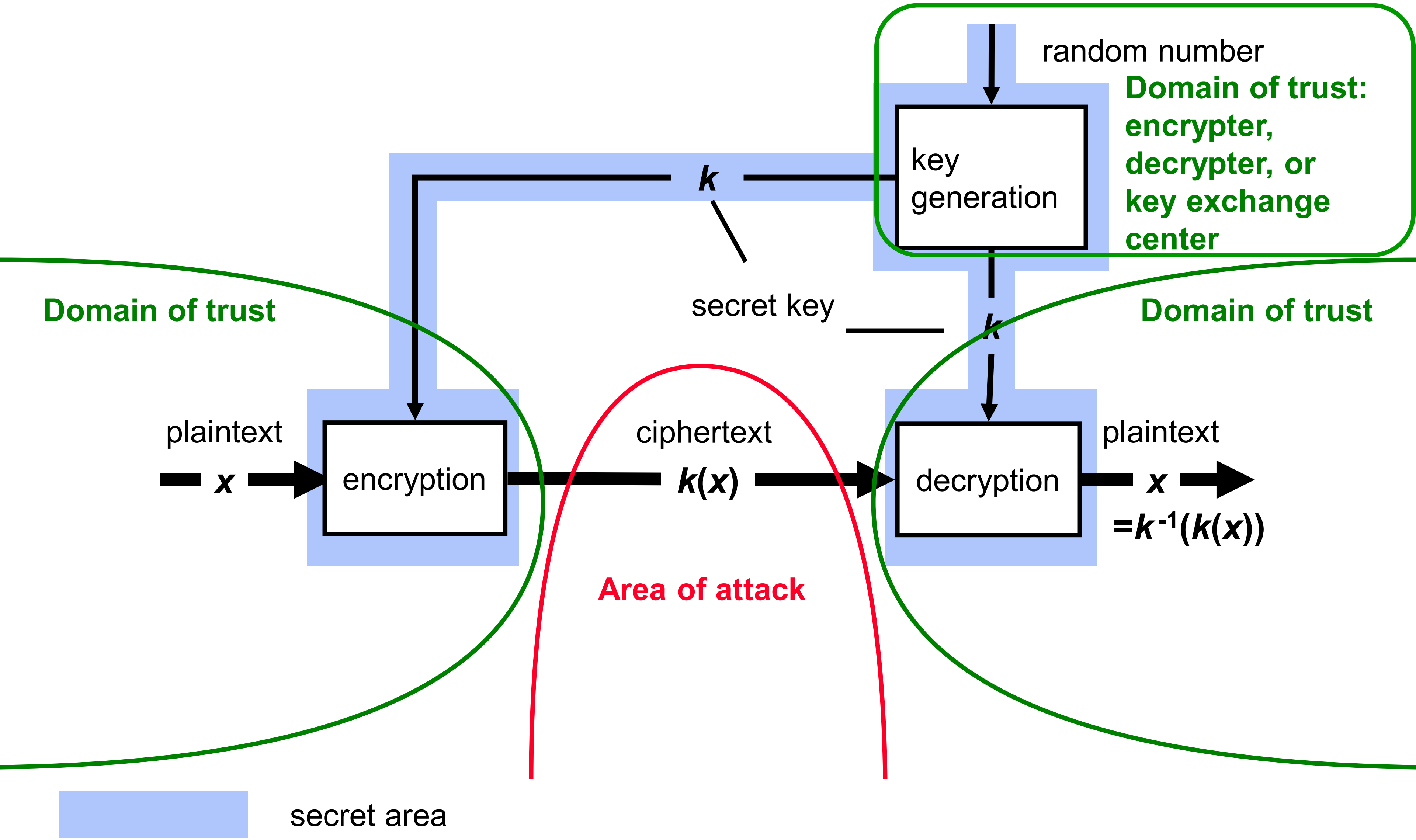


Opaque box with lock; 2 identical keys

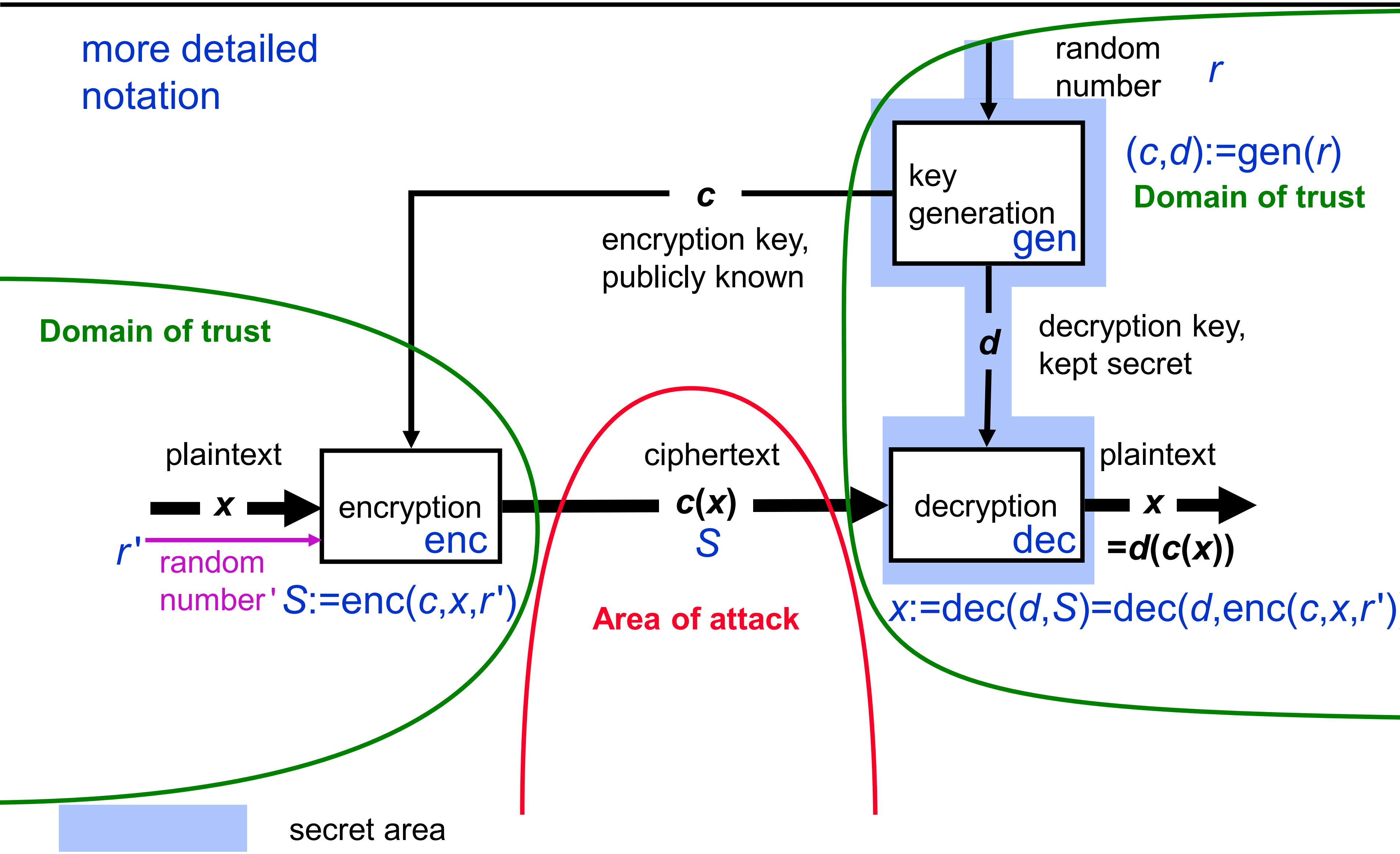
Key exchange using symmetric encryption systems



Sym. encryption system: Domain of trust key generation

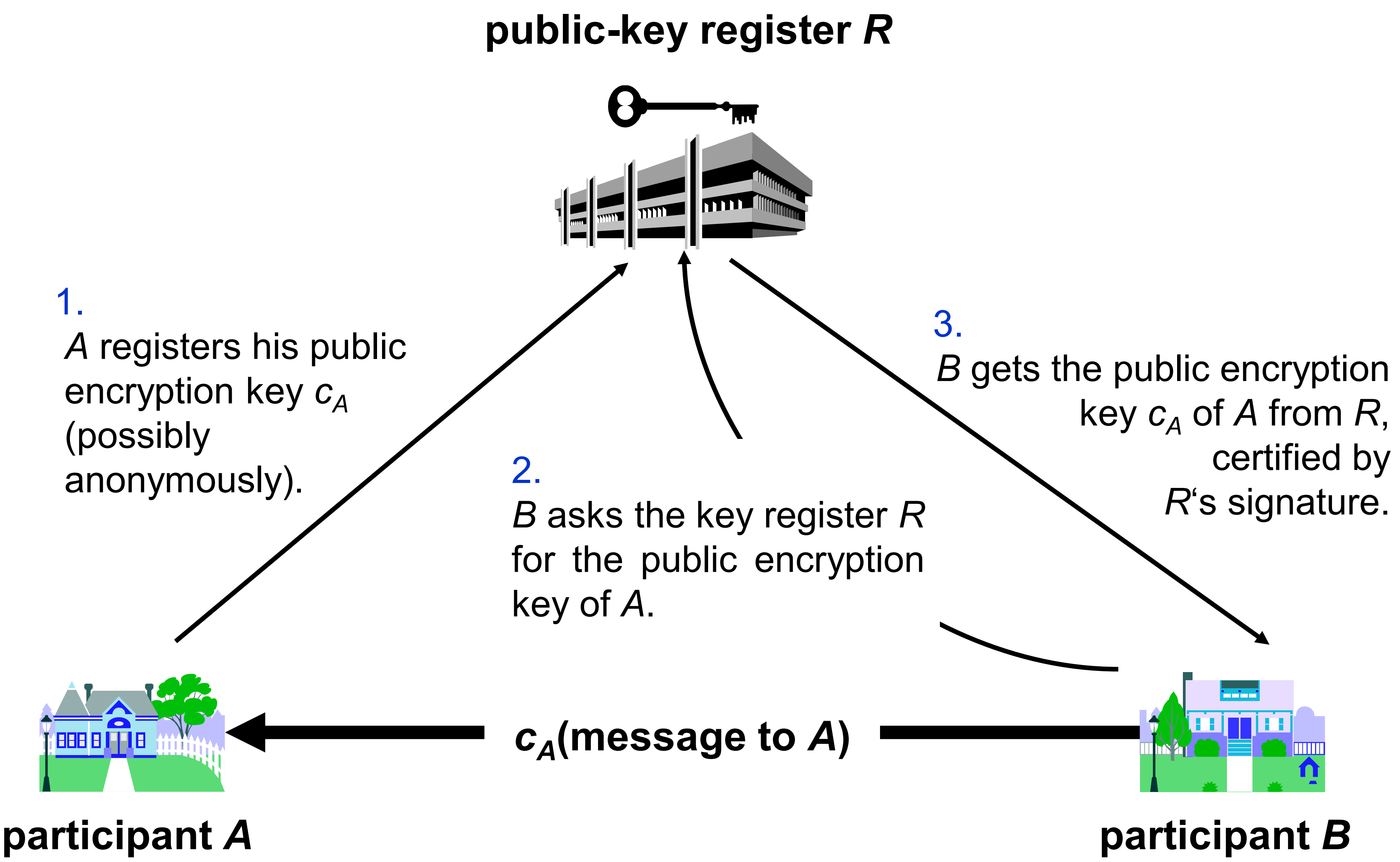


Asymmetric encryption system

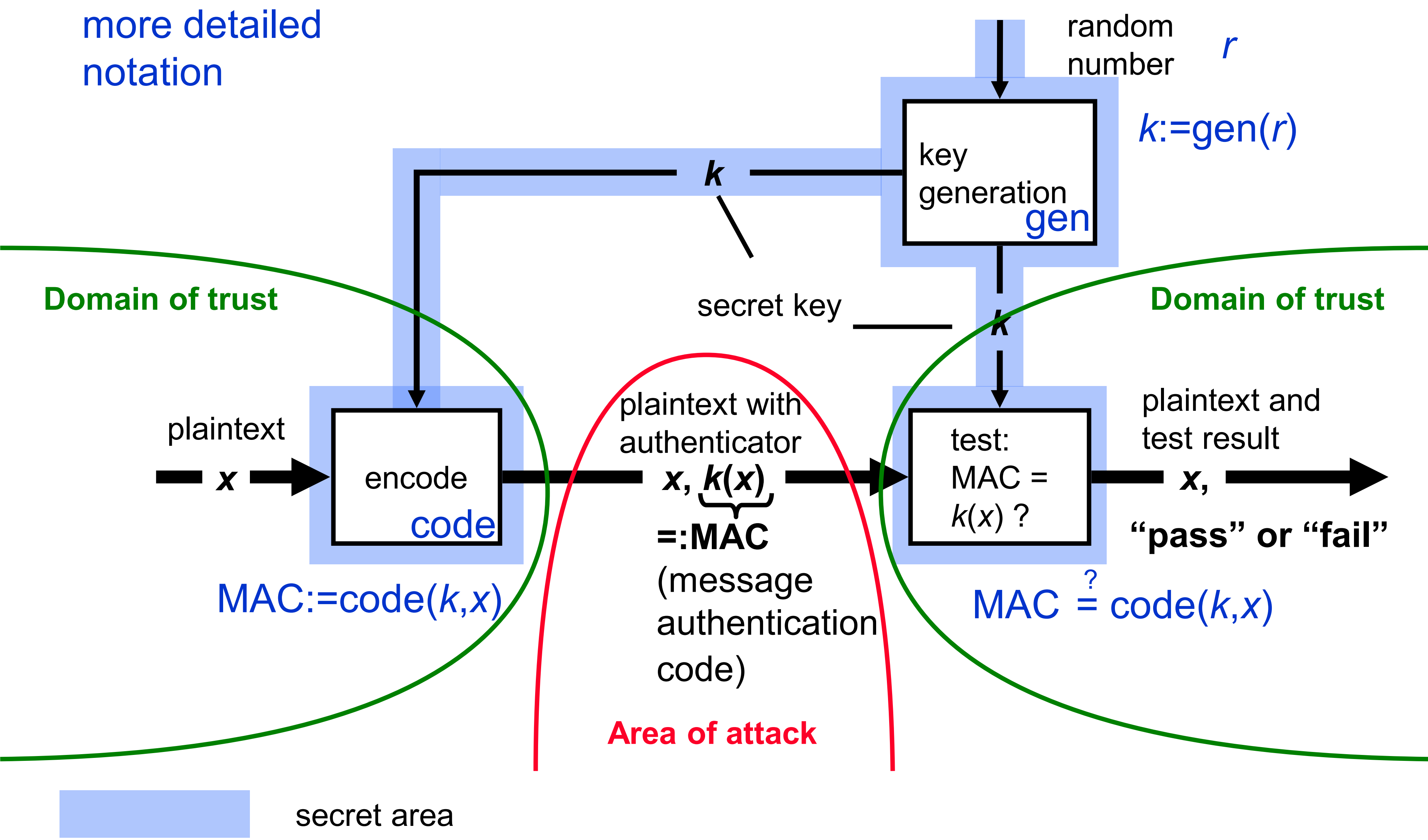


Opaque box with spring lock; 1 key

Key distribution using asymmetric encryption systems

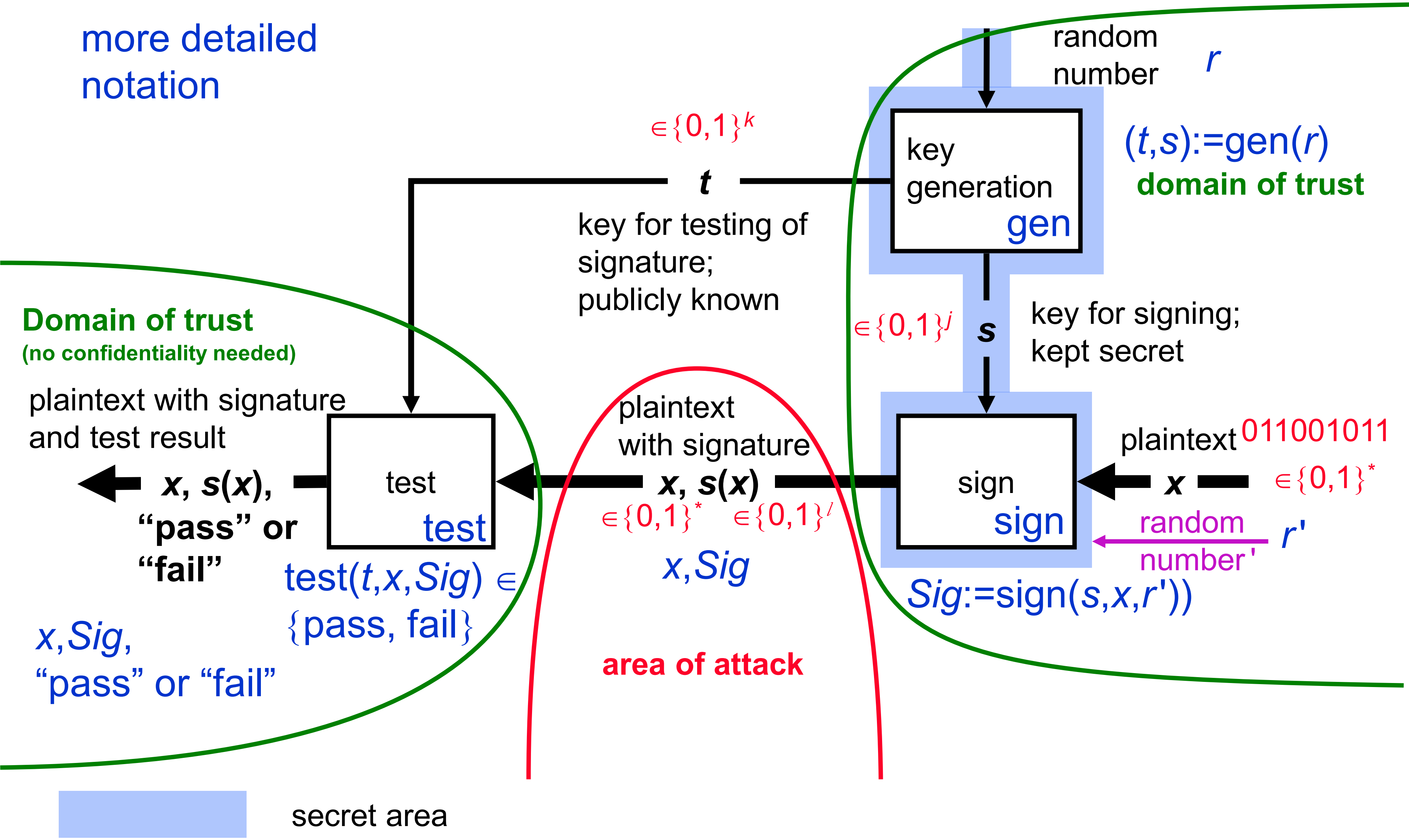


Symmetric authentication system



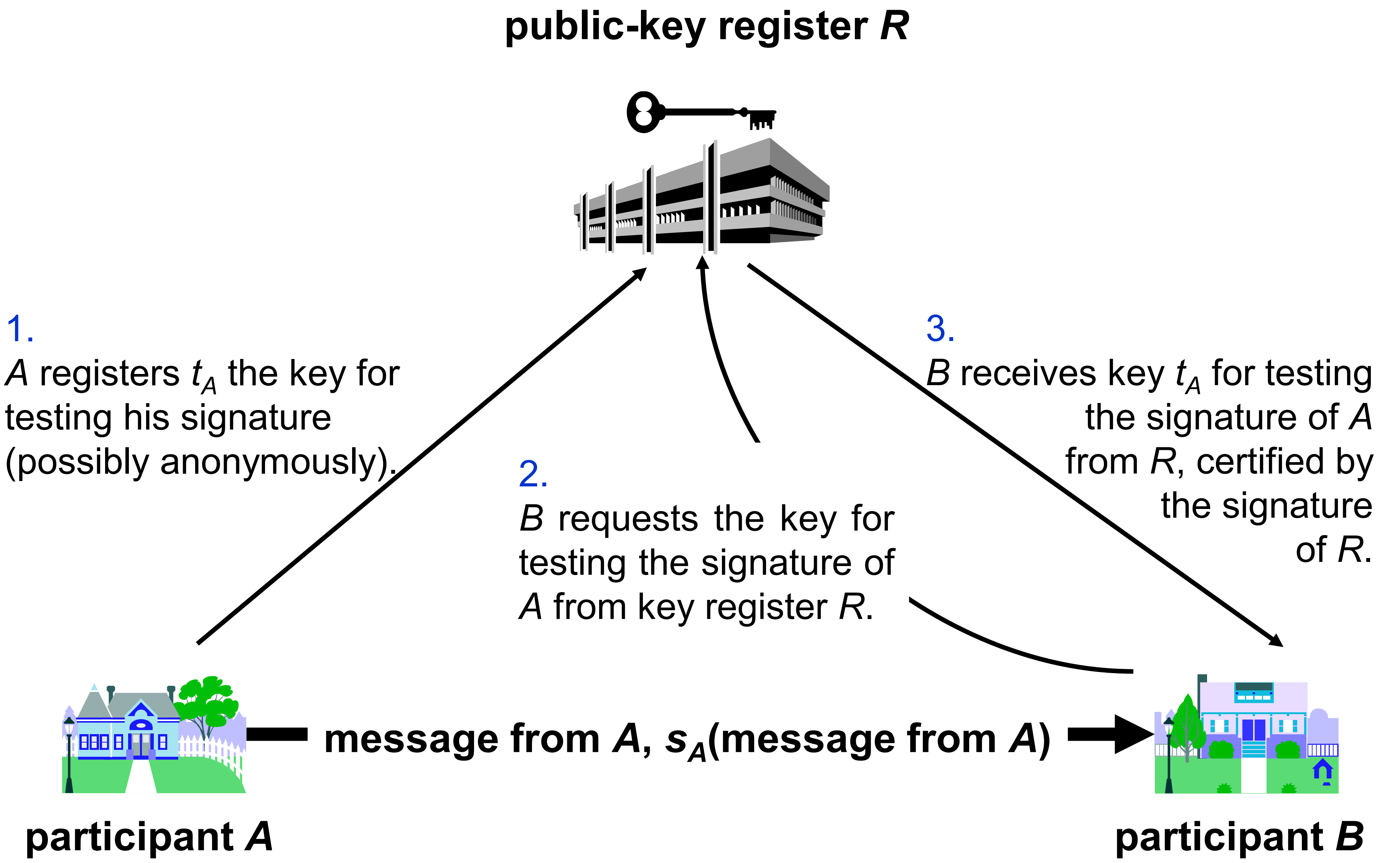
Show-case with lock; 2 identical keys

Digital signature system

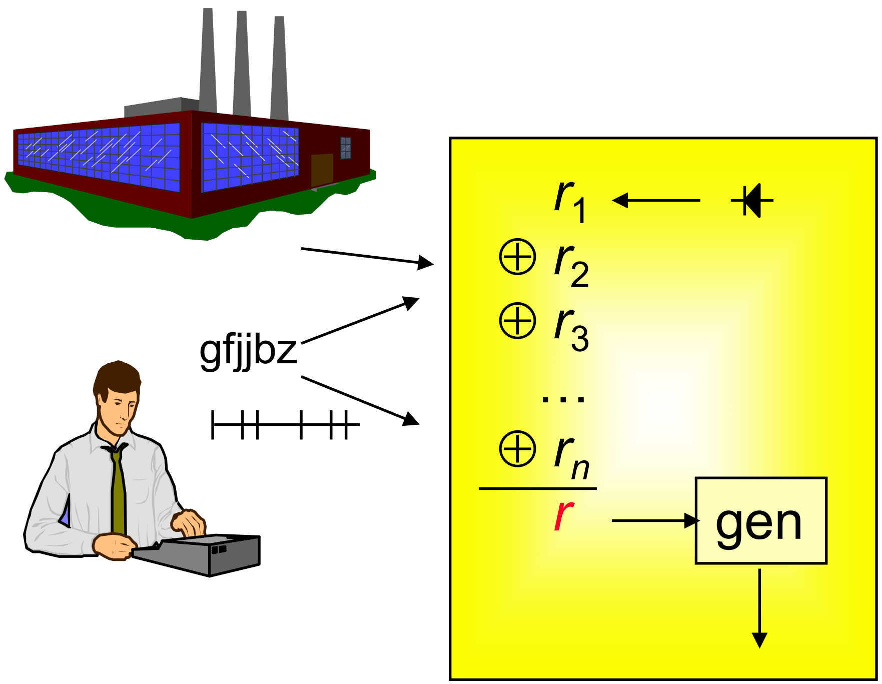


Show-case with lock; 1 key

Key distribution using digital signature systems



Key generation



generation of a random number *r* for the key generation:

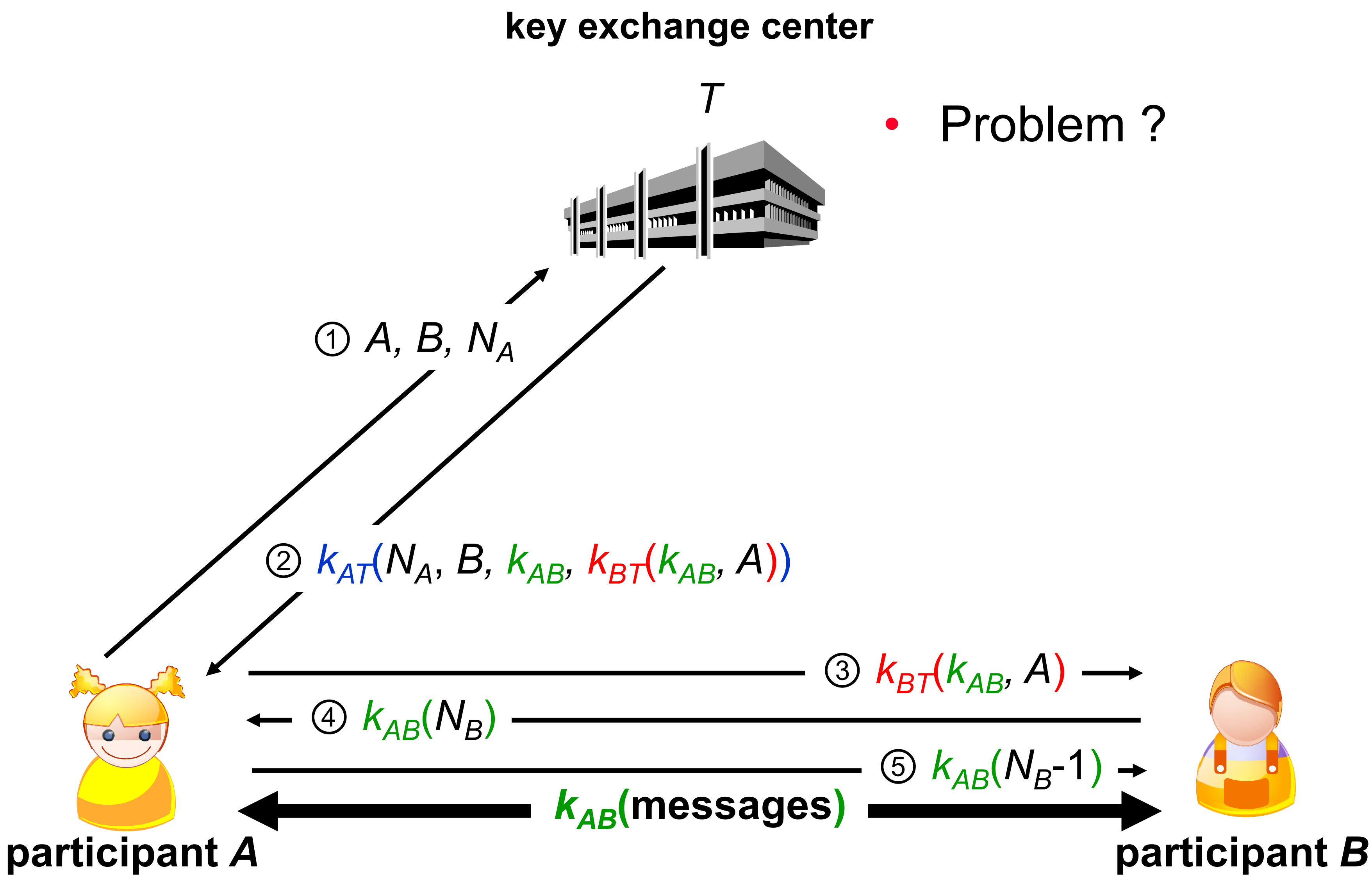
XOR of

- r_1, created in device,
- r_2, delivered by producer,
- r_3, delivered by user,
- r_n, calculated from keystroke intervals.

Needham-Schroeder-Protocol using Symmetric encryption

- from 1978
- goals:
  - key freshness:
    - key is „fresh“, i.e. a newly generated one
  - key authentication:
    - key is only known to Alice and Bob (and maybe some trusted third party)
- preconditions:
  - a trusted third party *T*
  - shared term secret keys between Alice (resp. Bob) and the trusted third party:
    - k_{AT}, k_{BT}

Needham-Schroeder-Protocol using Symmetric encryption

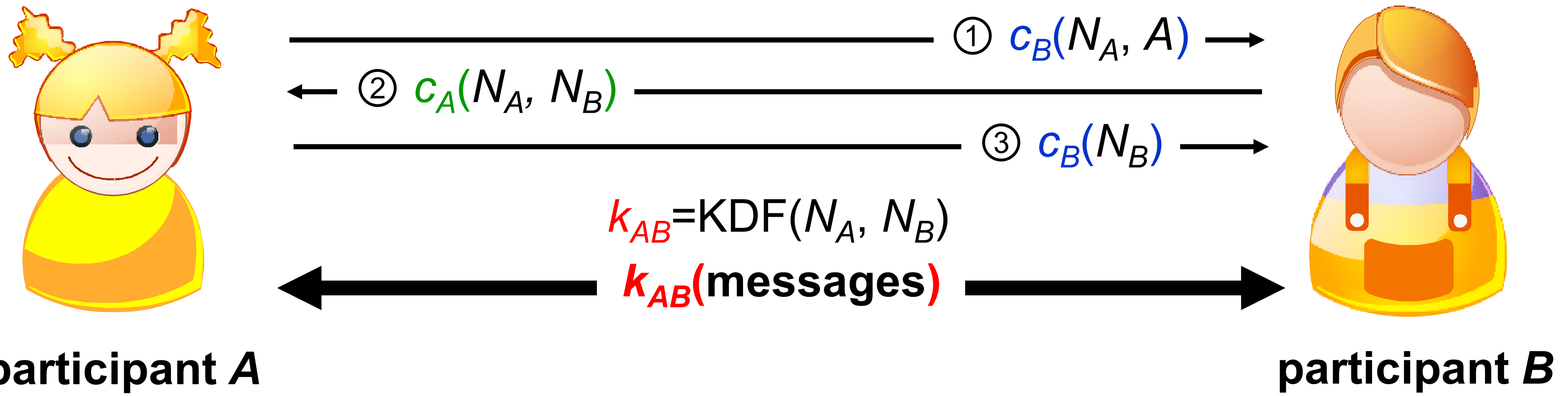


- Problem ?

Needham-Schroeder-Protocol using Asymmetric encryption

- from 1978
- goals:
  - key freshness:
    - key is „fresh“, i.e. a newly generated one
  - key authentication:
    - key is only known to Alice and Bob
- preconditions:
  - public encryption keys of Alice c_A and Bob c_B known to each other

Needham-Schroeder-Protocol using Asymmetric encryption



- Problem ?

Comments on key exchange

Whom are keys assigned to?

1. individual participants	asymmetric systems
2. pair relations	symmetric systems
3. groups	–

How many keys have to be exchanged?

n participants

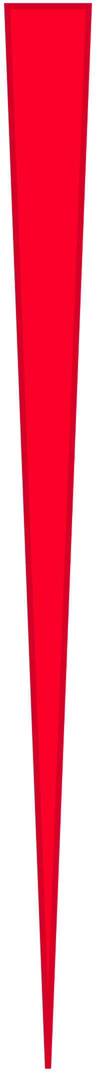
asymmetric systems	n per system
symmetric systems	$n \bullet (n\text{-}1)$

When are keys generated and exchanged?

Security of key exchange limits security available by cryptography:

execute several initial key exchanges

Goal/success of attack



a) key (total break)

b) procedure equivalent to key (universal break)

c) individual messages,

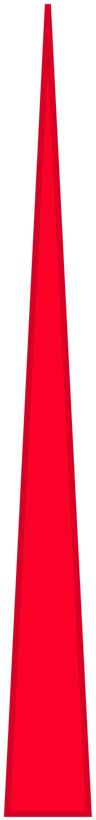
e.g. especially for authentication systems

c1) one selected message (selective break)

c2) any message (existential break)

Types of attack

severity



a) passive

- **a1) ciphertext-only attack**
- **a2) known-plaintext attack**

b) active

(according to encryption system; asym.: either b1 or b2;
sym.: b1 or b2)

- **b1) signature system: plaintext → ciphertext (signature)**
  (chosen-plaintext attack)
- **b2) encryption system: ciphertext → plaintext**
  (chosen-ciphertext attack)

adaptivity

- **not adaptive**
- **adaptive**

criterion:	action		permission
	passive attacker	≠	observing attacker
	active attacker	≠	modifying attacker

Basic facts about “cryptographically strong” (1)

If no security against computationally unrestricted attacker:

1) using of keys of constant length l:
   - attacker algorithm can always try out all 2^l keys
     (breaks asym. encryption systems and sym. systems in known-plaintext attack).
   - requires an exponential number of operations
     (too much effort for $l > 100$).

   → the best that the designer of encryption systems can hope for.

2) complexity theory:
   - mainly delivers asymptotic results
   - mainly deals with “worst-case”-complexity

   → useless for security; same for “average-case”-complexity.

   goal: problem is supposed to be difficult almost everywhere, i.e. except for an infinitesimal fraction of cases.
   - security parameter l (more general than key length; practically useful)
   - if $\underbrace{l \to \infty}_{\text{slow}}$, $\underbrace{\text{then probability of breaking} \to 0}_{\text{fast}}$.
   - hope: slow fast

Basic facts about "cryptographically strong" (2)

3) 2 classes of complexity:

   en-/decryption: easy = polynomial in l
   breaking: hard = not polynomial in l $\approx$ exponential in l
   Why?

   a) harder than exponential is impossible, see 1).

   b) self-contained: substituting polynomials in polynomials gives polynomials.

   c) reasonable models of calculation (Turing-, RAM-machine) are polynomially equivalent.

   For practice polynomial of high degree would suffice for runtime of attacker algorithm on RAM-machine.

4) Why assumptions on computational restrictions, e.g., factoring is difficult?

   Complexity theory cannot prove any useful lower limits so far.
   Compact, long studied assumptions!

5) What if assumption turns out to be wrong?

   a) Make other assumptions.

   b) More precise analysis, e.g., fix model of calculation exactly and then examine if polynomial is of high enough degree.

6) Goal of proof: If attacker algorithm can break encryption system, then it can also solve the problem which was assumed to be difficult.

Security classes of cryptographic systems

security

1. attacker assumed to be computationally unrestricted

2. cryptographically strong

3. well analyzed

4. somewhat analyzed

5. kept secret

Overview of cryptographic systems

system type / security		concealment		authentication	
		sym. $\supset$	asym.	sym. $\supset$	asym.
		sym. encryption system	asym. encryption system	sym. authentication system	digital signature system
information theoretic		Vernam cipher (one-time pad)	1	authentication codes	2
$\cap$					
cryptographically strong	active attack	pseudo one-time pad with s^2 mod n generator	3 CS	4	GMR
	$\cap$				
	passive attack	5	system with s^2 mod n generator	6	7
$\cap$					
well analyzed	mathematics	8	RSA	9	RSA
	chaos	DES	10	DES	11

Hybrid cryptosystems (1)

Combine:

- from asymmetric systems: easy key distribution
- from symmetric systems: efficiency (factor 100 ... 10000, SW and HW)

How?

use asymmetric system to distribute key for symmetric system

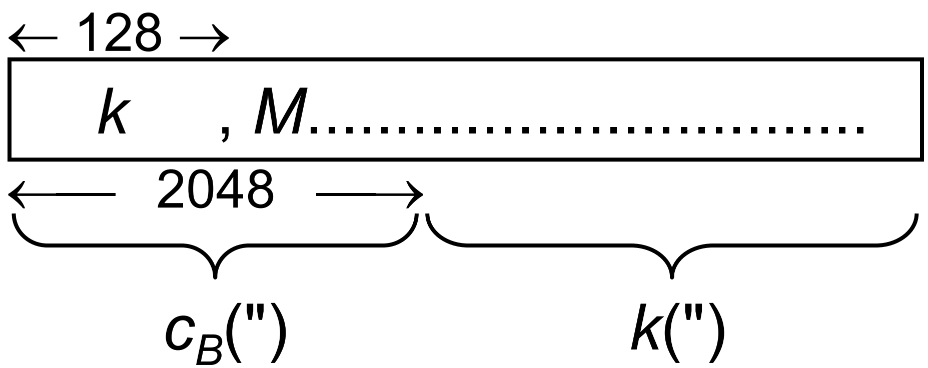
Encryption:

$$A \overset{M}{\dashrightarrow} B$$

get c_B choose k	$\xrightarrow[c_B(k),k(M)]{}$	decrypt k with d_B decrypt M with k

Hybrid cryptosystems (2)

Even more efficient: part of M in first block



If B is supposed also to use k: append $s_A(B,k)$

Authentication: k authorized and kept secret

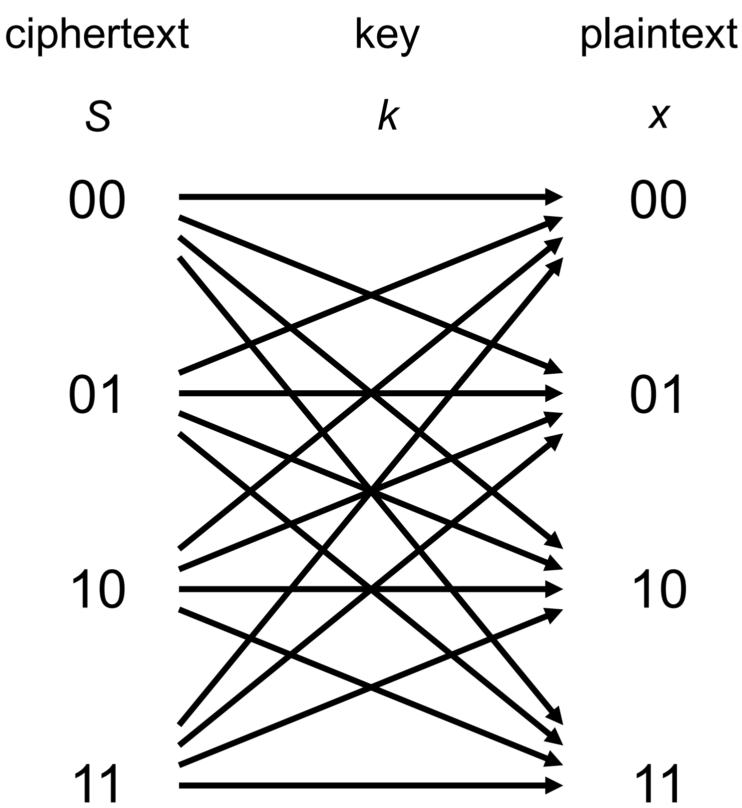
get c_B
choose k

$$M,\underbrace{k(M)}_{\text{MAC}},c_B(B,k,s_A(B,k))\longrightarrow$$

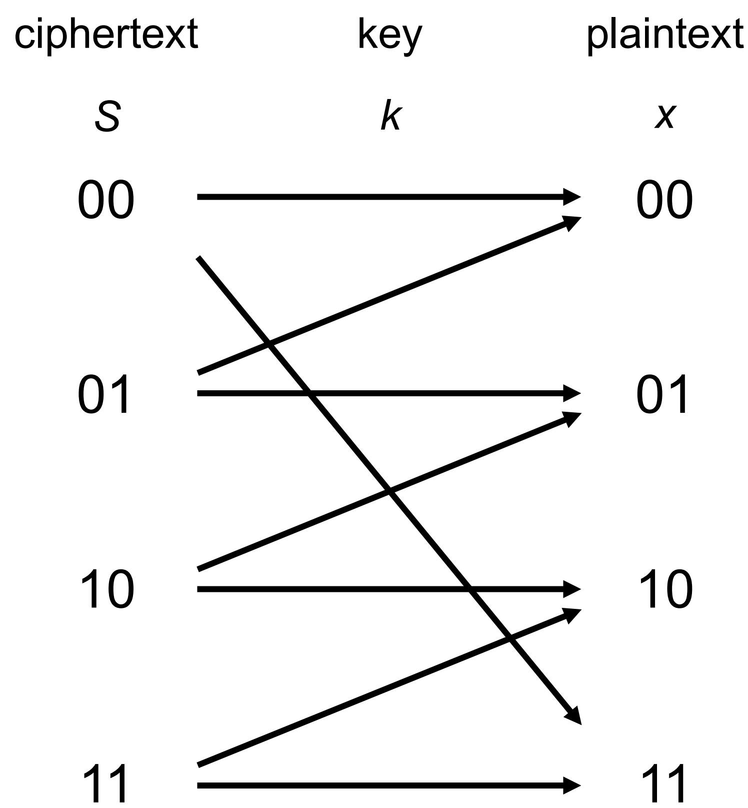
get t_A
decrypt $c_B(B,k,s_A(B,k))$
test B,k with t_A
test M with k

Information-theoretically secure encryption (1)

"Any ciphertext *S* may equally well be any plaintext *x*"



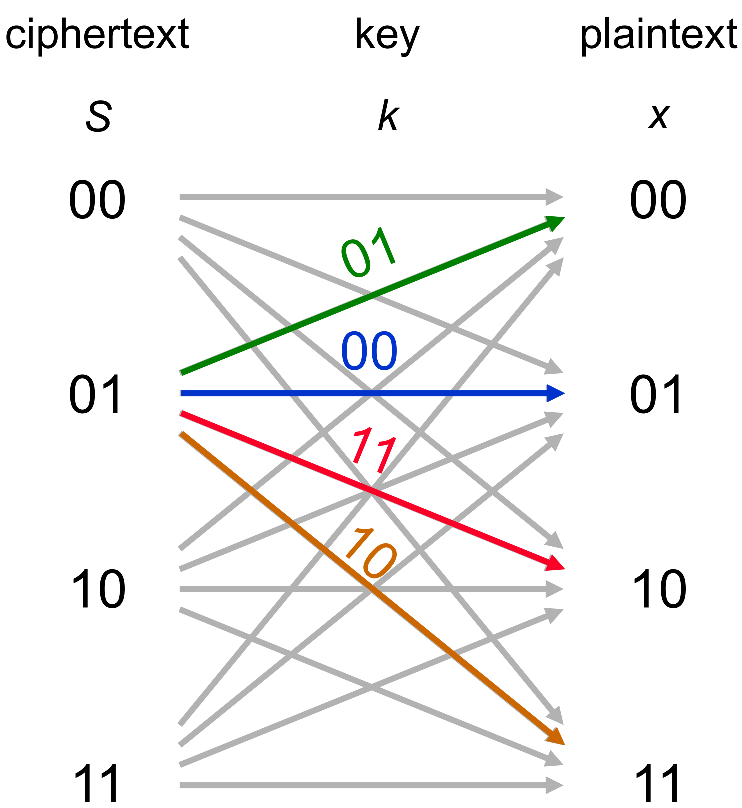
secure cipher



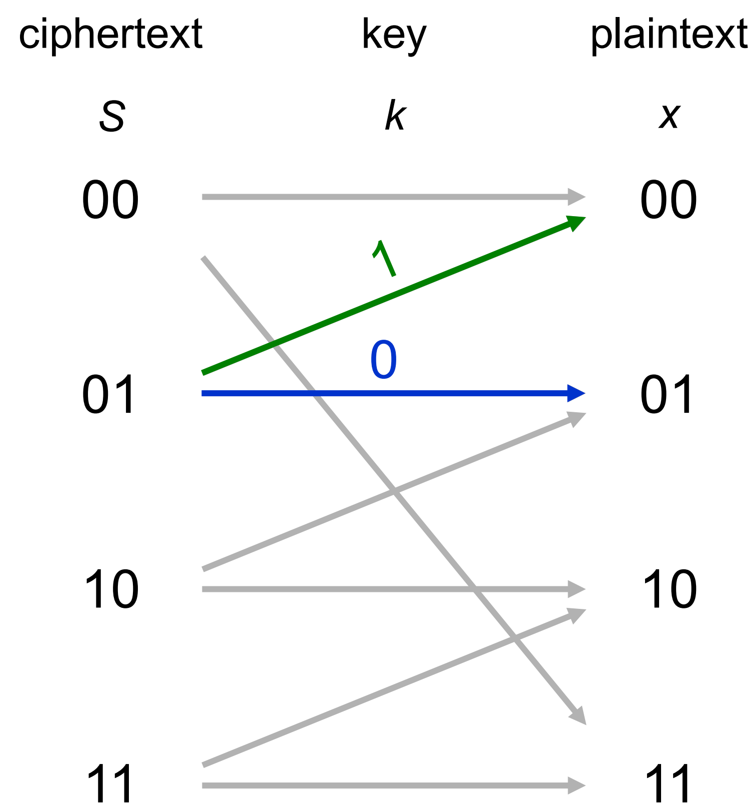
insecure cipher

Information-theoretically secure encryption (2)

“Any ciphertext *S* may equally well be any plaintext *x*”



secure cipher



insecure cipher

example : Vernam cipher mod 2

x = 00 01 00 10

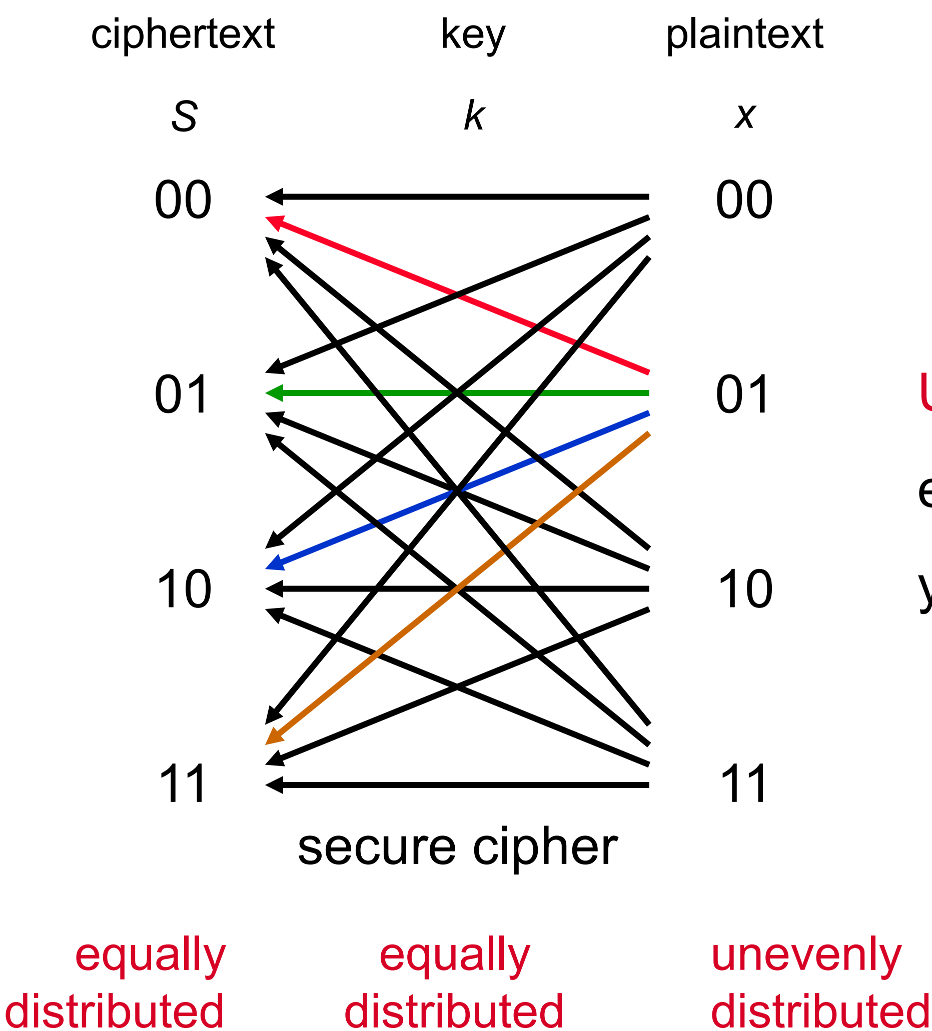
$\oplus$ k = 10 11 01 00

S = 10 10 01 10

subtraction of one key bit mod 4 from 2 plaintext bits

Information-theoretically secure encryption (3)

Different probability distributions – how do they fit?



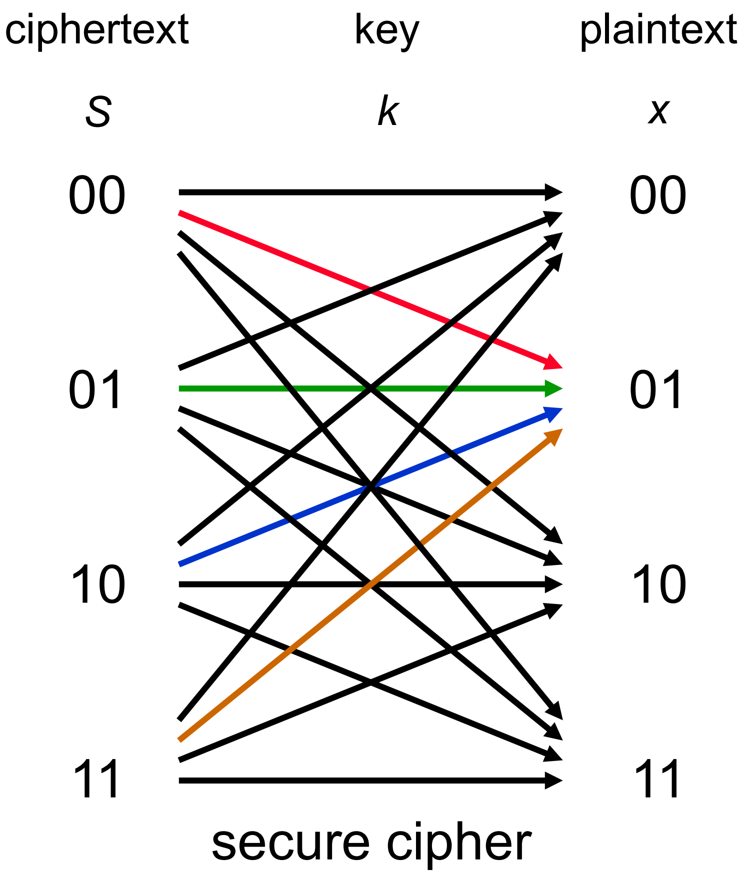
Unevenly distributed plaintexts

enciphered with equally distributed keys

yield equally distributed ciphertexts.

Information-theoretically secure encryption (4)

Different probability distributions – how do they fit?



equally distributed

equally distribu-ted, but *not* independently of the ciphertexts

unevenly distributed

Equally distributed ciphertexts deciphered with equally distributed keys can yield unevenly distributed plaintexts, iff ciphertexts and keys are *not* independently distributed, i.e., the ciphertexts have been calculated using the plaintext and the key.

Vernam cipher (one-time pad)

All characters are elements of a group G.

Plaintext, key and ciphertext are character strings.

For the encryption of a character string x of length n, a randomly generated and secretly exchanged key $k = (k_1,\ldots,k_n)$ is used.

The i^{th} plaintext character x_i is encrypted as

$$c_i \;:=\; x_i + k_i$$

It can be decrypted with

$$x_i \;:=\; c_i - k_i.$$

Evaluation: 1. secure against adaptive attacks
2. easy to calculate
3. but key is very long

Keys have to be very long for information-theoretical security

$\mathcal{K}$ is the set of keys,
$\mathcal{X}$ is the set of plaintexts, and
$\mathcal{C}$ is the set of ciphertexts, which appear at least once.

$|\mathcal{C}| \geq |\mathcal{X}|$ otherwise it can't be decrypted (fixed k)

$|\mathcal{K}| \geq |\mathcal{C}|$ so that any ciphertext might as well be any plaintext (fixed x)

therefore $|\mathcal{K}| \geq |\mathcal{X}|$.

If plaintext cleverly coded, it follows that:

The length of the key must be at least the length of the plaintext.

Definition for information-theoretical security

1. Definition for information-theoretical security

(all keys are chosen with the same probability)

$$\forall c \in C. \exists\, const \in \mathbb{N}. \forall x \in X\colon |\{k \in K \mid k(x) = c\}| = const. \qquad (1)$$

The a-posteriori probability of the plaintext x is $P(x|c)$, after the attacker got to know the ciphertext c.

2. Definition

$$\forall c \in C.\ \forall x \in X\colon P(x|c) = P(x). \qquad (2)$$

Both definitions are equivalent (if $P(x) > 0$):

According to Bayes: $\boxed{P(x|c) = \frac{P(x) \cdot P(c|x)}{P(c)}}$

Therefore, (2) is equivalent to

$$\forall c \in C.\ \forall x \in X\colon P(c|x) = P(c). \qquad (3)$$

We show that this is equivalent to

$$\forall c \in C. \exists\, const' \in \mathbb{R}\ \forall x \in X\colon P(c|x) = const'. \qquad (4)$$

Proof

(3)⇒(4) is clear with $const' := P(c)$.

(4)⇒(3): Conversely, we show $const' = P(c)$:

$$P(c) = \sum_x P(x) \cdot P(c|x) = \sum_x P(x) \cdot const' = const' \cdot \sum_x P(x) = const'$$

$\forall c \in C\ \exists\, const' \in \mathrm{IR}\ \forall x \in \mathcal{X}$: $P(c|x) = const'$ (4)

$\forall c \in C\ \forall x \in \mathcal{X}$: $P(c|x) = P(c)$. (3)

$\forall c \in C\ \exists\, const \in \mathrm{IN}\ \forall x \in \mathcal{X}$: $|\{k \in \mathcal{K} |\ k(x) = c\}| = const$. (1)

(4) is already quite the same as (1): In general holds

$$\mathrm{P}(c|x) = \mathrm{P}(\{k \mid k(x) = c\}),$$

and if all keys have the same probability,

$$\mathrm{P}(c|x) = |\{k \mid k(x) = c\}| \,/\, |\mathcal{K}|.$$

Then (4) is equivalent (1) with

$$const = const' \bullet |\mathcal{K}|.$$

Symmetric authentication systems (1)

Key distribution:

like for symmetric encryption systems

Simple example (view of attacker)

The outcome of tossing a coin (Head (H) or Tail (T)) shall be sent in an authenticated fashion:

Key		m, MAC			
		H,0	H,1	T,0	T,1
k	00	H		T	
	01	H			T
	10		H	T	
	11		H		T

Security: e.g. attacker wants to send T.

a) blind: get caught with a probability of 0.5

b) seeing: e.g. attacker gets (H,0) $\Rightarrow$ $k \in \{00, 01\}$

still both, (T,0) and (T,1), have a probability of 0.5

Symmetric authentication systems (2)

Definition "Information-theoretical security" with error probability ε:

$\forall x$, MAC (that attacker can see)

$\forall y \neq x$ (that attacker sends instead of x)

$\forall$ MAC' (where attacker chooses the one with the highest probability fitting y)

$$P(k(y) = \text{MAC}' \mid k(x) = \text{MAC}) \leq \varepsilon$$

(probability that MAC' is correct if one only takes the keys k which are still possible under the constraint of (x,MAC) being correct.)

Improvement of the example:

a) 2σ key bits instead of 2: $k = k_1 k_1^* \ldots k_\sigma k_\sigma^*$
MAC = $\text{MAC}_1,\ldots,\text{MAC}_\sigma$; MAC_i calculated using $k_i k_i^*$
$\Rightarrow$ error probability $2^{-\sigma}$

b) l message bits: $x^{(1)}$, $\text{MAC}^{(1)} = \text{MAC}_1^{(1)}, \ldots, \text{MAC}_\sigma^{(1)}$

$\vdots \quad \vdots \quad \vdots$

$x^{(l)}$, $\text{MAC}^{(l)} = \text{MAC}_1^{(l)}, \ldots, \text{MAC}_\sigma^{(l)}$

Symmetric authentication systems (3)

Limits:

σ-bit-MAC $\Rightarrow$ error probability $\geq 2^{-\sigma}$
(guess MAC)

σ-bit-key $\Rightarrow$ error probability $\geq 2^{-\sigma}$
(guess key, calculate MAC)

still clear: for an error probability of $2^{-\sigma}$, a σ-bit-key is too short, because $k(x)$ = MAC eliminates many values of k.

Theorem: for a single/the first message you need 2σ-bit-key

(for succeeding messages σ new key bits suffice, if recipient adequately responds on authentication "errors": attacker learns σ key bits with every message)

Possible at present: $\approx 4\sigma \cdot \log_2(\text{length}(x))$ key bits

(Wegman, Carter)

much shorter as one-time pad

About cryptographically strong systems (1)

Mathematical secrets:

(to decrypt, to sign ...)

p, q, prime numbers

Public part of key-pair:

(to encrypt, to test ...)

$n = p \cdot q$

p, q big, at present $\approx \ell = 500$ up to 2000 bit
(theory : $\ell \to \infty$)

Often: special property

$p \equiv q \equiv 3 \bmod 4$

(the semantics of "$\equiv$ ... mod" is:
$a \equiv b \bmod c$ iff c divides a-b,
putting it another way: dividing a and b by c leaves the same remainder)

About cryptographically strong systems (2)

application: s^2-mod-n-generator,
GMR and many others,
e.g., only well analyzed systems like RSA

(significant alternative: only “discrete logarithm”,
based on number theory, too, similarly well analyzed)

necessary:
1. factoring is difficult
2. to generate p,q is easy
3. operations on the message with n alone, you can only invert using p, q

Factoring

clear: in NP $\Rightarrow$ but difficulty cannot be proved yet

complexity at present

$$L(n) = e^{c \cdot \sqrt[3]{\ln(n) \cdot (\ln \ln(n))^2}}$$, $c \approx 1{,}9$

$$\approx e^{\sqrt[3]{l}}$$ "sub-exponential"

practically up to 155 decimal digits in the year 1999
174 decimal digits in the year 2003
200 decimal digits in the year 2005
232 decimal digits in the year 2010
240 decimal digits in the year 2019 (www.crypto-world.com/FactorRecords.html)
250 decimal digits in the year 2020

(notice :

$\exists$ faster algorithms, e.g., for $2^r \pm 1$, but this doesn't matter)

assumption: factoring is hard

(notice : unacceptable, if attacker could factor, e.g., every 1000th n)

Factoring assumption

$\forall$ PPA $\mathcal{F}$ (probabilistic polynomial algorithm, which tries to factor)

$\forall$ polynomials Q

$\exists L \ \forall \ell \geq L$: (asymptotically holds:)

If p, q are random prime numbers of length ℓ and $n = p \bullet q$:

$$P(\mathcal{F}(n) = (p, q)) \leq \frac{1}{Q(\ell)}$$

(probability that $\mathcal{F}$ truly factors

decreases faster as $\frac{1}{\text{any polynomial}}$.)

trustworthy ??

the best analyzed assumption of all available

Search of prime numbers (1)

1. Are there enough prime numbers ? (important also for factoring assumption)

$$\frac{\pi(x)}{x} \approx \frac{1}{\ln(x)}$$

$\pi(x)$ number of the prime numbers $\leq x$

"prime number theorem"

$\Rightarrow$ up to length l more than every l^{th}.

And $\approx$ every 2nd $\equiv 3 \bmod 4$ "Dirichlet's prime number theorem"

2. Principle of search:

repeat

choose random number p ($\equiv 3 \bmod 4$)

test whether p is prime

until p prime

Search of prime numbers (2)

3. Primality tests:

(notice: trying to factor is much too slow)

probabilistic; “Rabin-Miller”

Little Theorem of Fermat: $a^{p-1} \equiv 1 \bmod p$

special case $p \equiv 3 \bmod 4$:

$$p \text{ prime} \quad \Rightarrow \forall\, a \not\equiv 0 \bmod p: \quad a^{\frac{p-1}{2}} \equiv \pm 1 \pmod{p}$$

$$p \text{ not prime} \quad \Rightarrow \text{for} \le \frac{1}{4} \text{ of } a\text{'s}: \quad a^{\frac{p-1}{2}} \equiv \pm 1 \pmod{p}$$

$\Rightarrow$ test this for m different, independently chosen values of a,

$$\text{error probability} \le \frac{1}{4^m}$$

(doesn’t matter in general)

Calculating with and without *p*,*q* (1)

Z_n : ring of residue classes mod $n \mathrel{\hat{=}} \{0, \ldots , n\text{-}1\}$

- +, -, • fast
- exponentiation "fast" (square & multiply)

example: $7^{26} = 7^{(11010)_2}$; from left

$$\begin{tikzcd}
7^1 \arrow[r, "s"] & 7^{10} \arrow[d, "m"'] & 7^{110} \arrow[r, "s"] & 7^{1100} \arrow[d, "m"'] & 7^{11010} \\
& 7^{11} \arrow[ur, "s"] & & 7^{1101} \arrow[ur, "s"] &
\end{tikzcd}$$

- gcd (greatest common divisor) fast in Z (Euclidean Algorithm)

Calculating with and without *p*,*q* (2)

Z_n^* : multiplicative group

$a \in Z_n^* \Leftrightarrow \gcd(a,n) = 1$

- Inverting is fast (extended Euclidean Algorithm)

  Determine to a,n the values u,v with

  $$a \cdot u + n \cdot v = 1$$

  Then: $u \equiv a^{-1} \bmod n$

example: $3^{-1} \bmod 11$?

$11 = 3 \cdot \underline{3} + 2$

$3 = 1 \cdot \underline{2} + 1$

$1 = 1 \cdot 3 - 1 \cdot 2$

$= 1 \cdot 3 - 1 \cdot (11 - 3 \cdot 3)$

$= -11 + 4 \cdot 3$

$\Rightarrow 3^{-1} \equiv 4 \bmod 11$

Calculating with and without *p*,*q* (3)

Number of elements of Z_n^*

The Euler Φ- Function is defined as

$$\Phi(n) := \left|\{a \in \{0,...,n\text{-}1\} \mid \gcd(a,n)=1\}\right|,$$

whereby for any integer $n \neq 0$ holds: $\gcd(0,n)=|n|$.

It immediately follows from both definitions, that

$$|Z_n^*| = \Phi(n).$$

For $n = p \bullet q$, p,q prime and $p \neq q$ we can easily calculate $\Phi(n)$:

$$\Phi(n) = (p\text{-}1) \bullet (q\text{-}1)$$

gcd ≠ 1 have the numbers 0, then p, $2p$, …, $(q\text{-}1)p$ and q, $2q$, …, $(p\text{-}1)q$, and these $1+(q\text{-}1)+(p\text{-}1) = p+q\text{-}1$ numbers are for $p \neq q$ all different.

Calculating with and without *p*,*q* (4)

Relation between $Z_n \leftrightarrow Z_p, Z_q$:

Chinese Remainder Theorem (CRA)

$$x \equiv y \bmod n \quad \Leftrightarrow \quad x \equiv y \bmod p \;\wedge\; x \equiv y \bmod q$$

since $\Updownarrow$ $\Updownarrow$ $\Updownarrow$

$$n|(x-y) \quad \Leftrightarrow \quad p|(x-y) \;\wedge\; q|(x-y)$$

$n = p \bullet q$, p,q prime, $p \neq q$

$\Rightarrow$ To calculate f(x) mod n, at first you have to calculate mod p, q separately.

y_p := f(x) mod p

y_q := f(x) mod q

Calculating with and without *p*,*q* (5)

Compose ?

extended Euclidean : $u \cdot p \ + \ v \cdot q \ = \ 1$

$$y := (u \cdot p) \cdot y_q + (v \cdot q) \cdot y_p \quad \begin{cases} \equiv y_p \bmod p \\ \equiv y_q \bmod q \end{cases}$$

Since :

$$\underbrace{\begin{array}{l|l|l} & \text{mod } p & \text{mod } q \\ \hline u \cdot p & 0 & 1 \\ \hline v \cdot q & 1 & 0 \\ \hline y & 0 \cdot y_q + 1 \cdot y_p & 1 \cdot y_q + 0 \cdot y_p \\ \hline & \equiv y_p & \equiv y_q \end{array}}_{\text{CRA}}$$

RSA — asymmetric cryptosystem

R. Rivest, A. Shamir, L. Adleman: A Method for obtaining Digital Signatures and Public-Key Cryptosystems; Communications of the ACM 21/2 (Feb. 1978) 120-126.

Key generation

1) Choose two prime numbers p and q at random as well as stochastically independent, with $|p| \approx |q| = l$, $p \neq q$
2) Calculate $n := p \cdot q$
3) Choose c with $3 \leq c < (p-1)(q-1)$ and
4) Calculate d using p, q, c as multiplicative inverse of c mod $\Phi(n)$

   $c \cdot d \equiv 1 \pmod{\Phi(n)}$
5) Publish c and n.

En- / decryption

exponentiation with c respectively d in Z_n

Proposition: $\forall m \in Z_n$ **holds:** $(m^c)^d \equiv m^{c \cdot d} \equiv (m^d)^c \equiv m \pmod{n}$

Proof (1)

$$c \bullet d \equiv 1 \ \big(\text{mod } \Phi(n)\big) \Leftrightarrow$$

$$\exists k \in \mathbb{Z}: c \bullet d - 1 = k \bullet \Phi(n) \Leftrightarrow$$

$$\exists k \in \mathbb{Z}: c \bullet d = k \bullet \Phi(n) + 1$$

Therefore $m^{c \bullet d} \equiv m^{k \bullet \Phi(n) + 1} \quad (\text{mod } n)$

Using the Theorem of Euler/Fermat
$\forall m \in \mathbb{Z}_n^*: m^{\Phi(n)} \equiv 1 \quad (\text{mod } n)$

it follows for all m coprime to p

$$m^{p-1} \equiv 1 \ (\text{mod } p) \qquad \text{[Little Theorem of Fermat]}$$

Because p-1 is a factor of $\Phi(n)$, it holds

$$m^{k \bullet \Phi(n) + 1} \equiv_p m^{k \bullet (p-1)(q-1) + 1} \equiv_p m \bullet \underbrace{(\underbrace{m^{p-1}}_{1})^{k \bullet (q-1)}}_{1} \equiv_p m$$

Proof (2)

Holds, of course, for $m \equiv_p 0$. So we have it for all $m \in Z_p$.

Same argumentation for q gives

$$m^{k \bullet \Phi(n) + 1} \equiv_q m$$

Because congruence holds relating to p as well as q, according

to the CRA, it holds relating to $p \bullet q = n$.

Therefore, for all $m \in Z_n$

$$m^{c \bullet d} \equiv m^{k \bullet \Phi(n) + 1} \equiv m \qquad (\text{mod } n)$$

Attention:

There is (until now ?) **no** proof

RSA is easy to break $\Rightarrow$ to factor is easy

Semantic Security

(Based on slide from Prof. Thorsten Strufe)

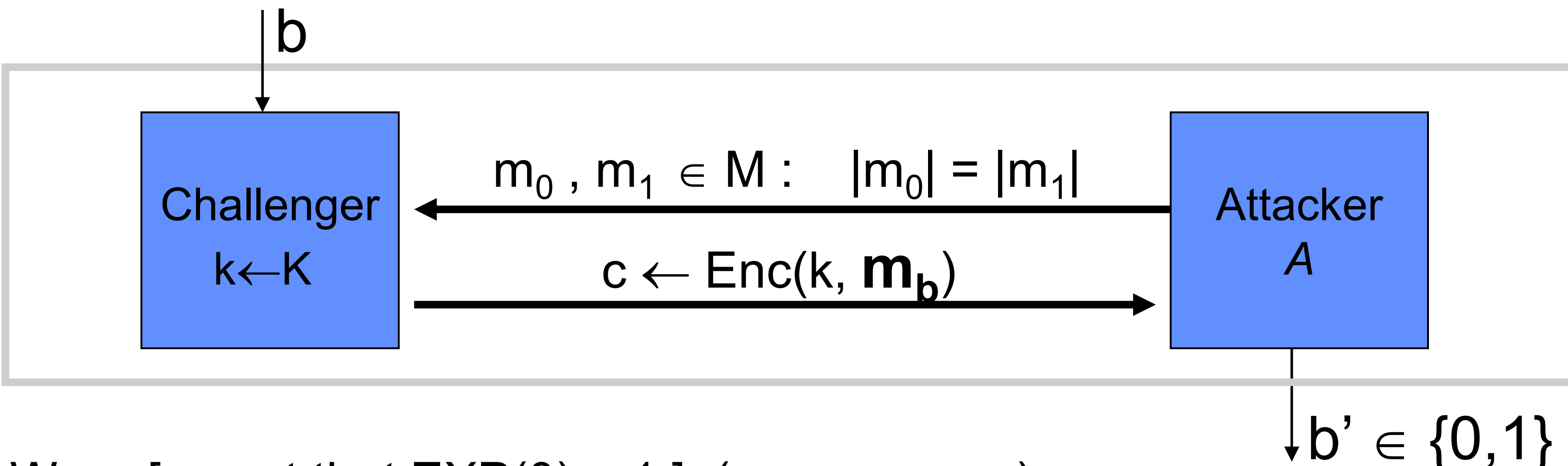
Semantic Security

(Based on slide from Prof. Thorsten Strufe)

Let's play a game:

A challenger flips a coin, and the adversary guesses the outcome

For b=0,1 define experiments EXP(0) and EXP(1) as:



- W_0 := [event that EXP(0) = 1] (wrong guess)
- W_1 := [event that EXP(1) = 1] (correct guess)

- $Advantage_{SS}[A, Enc] \coloneqq |\Pr[W_0] - \Pr[W_1]| \quad \in [0,1]$
- *Enc* is called ***semantically secure*** if for all efficient algorithms *A*, $Advantage_{SS}[A, Enc]$ is negligible (~0)

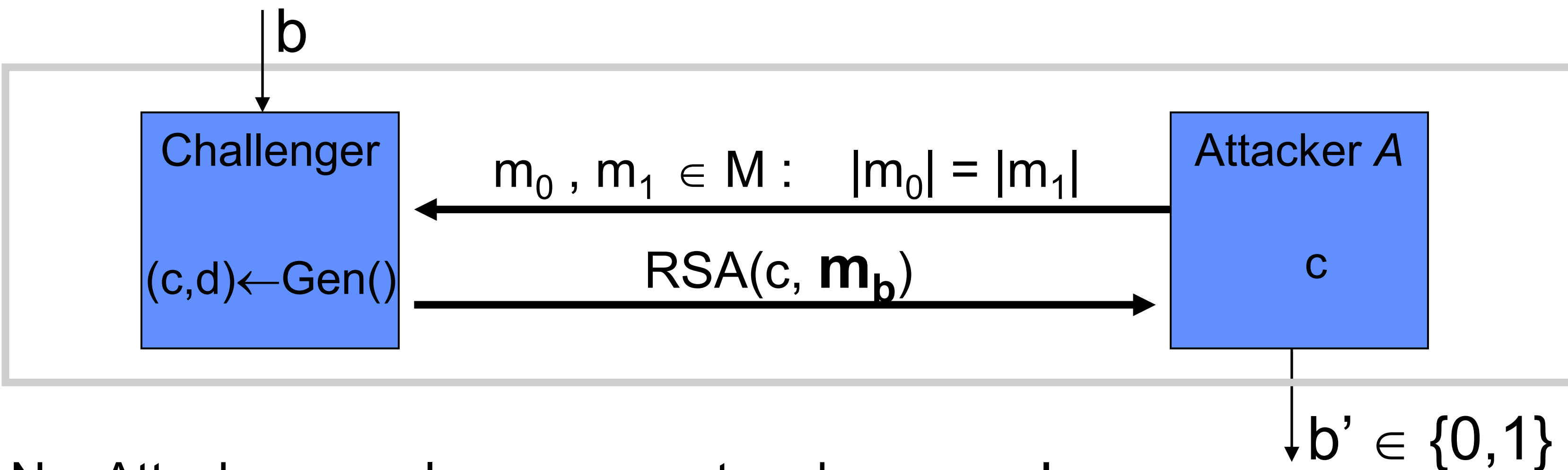
Semantic Security of plain RSA?

(Based on slide from Prof. Thorsten Strufe)

Let's play a game:

A challenger flips a coin, and the adversary guesses the outcome

For b=0,1 define experiments EXP(0) and EXP(1) as:



- No: Attacker can always encrypt and compare!

  ➔ indeterministic encryption necessary!

  add random number…

Attack on encryption with RSA naive

$$(x^c)^d \equiv x$$ Encryption/Decryption

Homomorphic Property of RSA:

$$\mathrm{Enc}(m_1) \cdot \mathrm{Enc}(m_2) = \mathrm{Enc}(m_1 \cdot m_2)$$

$$\boxed{(x \bullet y)^c = x^c \bullet y^c}$$

$$((x \bullet y)^c)^d \equiv x \bullet y$$ Encryption/Decryption

Attack on encryption with RSA naive

$$((x^c))^d \equiv x$$

ciphertext intercepted

$$(x \bullet y)^c = x^c \bullet y^c$$

calculated from y by the attacker

let it decrypt

$$(((x \bullet y)^c))^d \equiv x \bullet y$$

divide by y, get x

➔ Attack should be detectable!

Attack on digital signature with RSA naive

1. Simple version of Davida's attack:

   Given $\quad Sig_1 = m_1^{\,s}$

   $\qquad\qquad Sig_2 = m_2^{\,s}$

   $\Rightarrow \quad Sig := Sig_1 \cdot Sig_2 = (m_1 \cdot m_2)^s$

   New signature generated !

   (Passive attack, *m* not selectable.)

2. Active, desired $Sig = m^s$

   Choose any m_1; $\quad m_2 := m \cdot m_1^{-1}$

   Let m_1, m_2 be signed.

   Further as mentioned above.

3. Active, more skillful (Moore)

   "Blinding" : choose any r ,

   $m_2 := m \cdot r^t$

   $m_2^{\,s} = m^s \cdot r^{t \cdot s} = m^s \cdot r$

$$m_2 \longrightarrow \boxed{\text{sign}} \longrightarrow \overset{\bullet\, r^{-1}}{\rightsquigarrow} \; m^s = Sig$$

Defense against Davida's attacks using a collision-resistant hash function

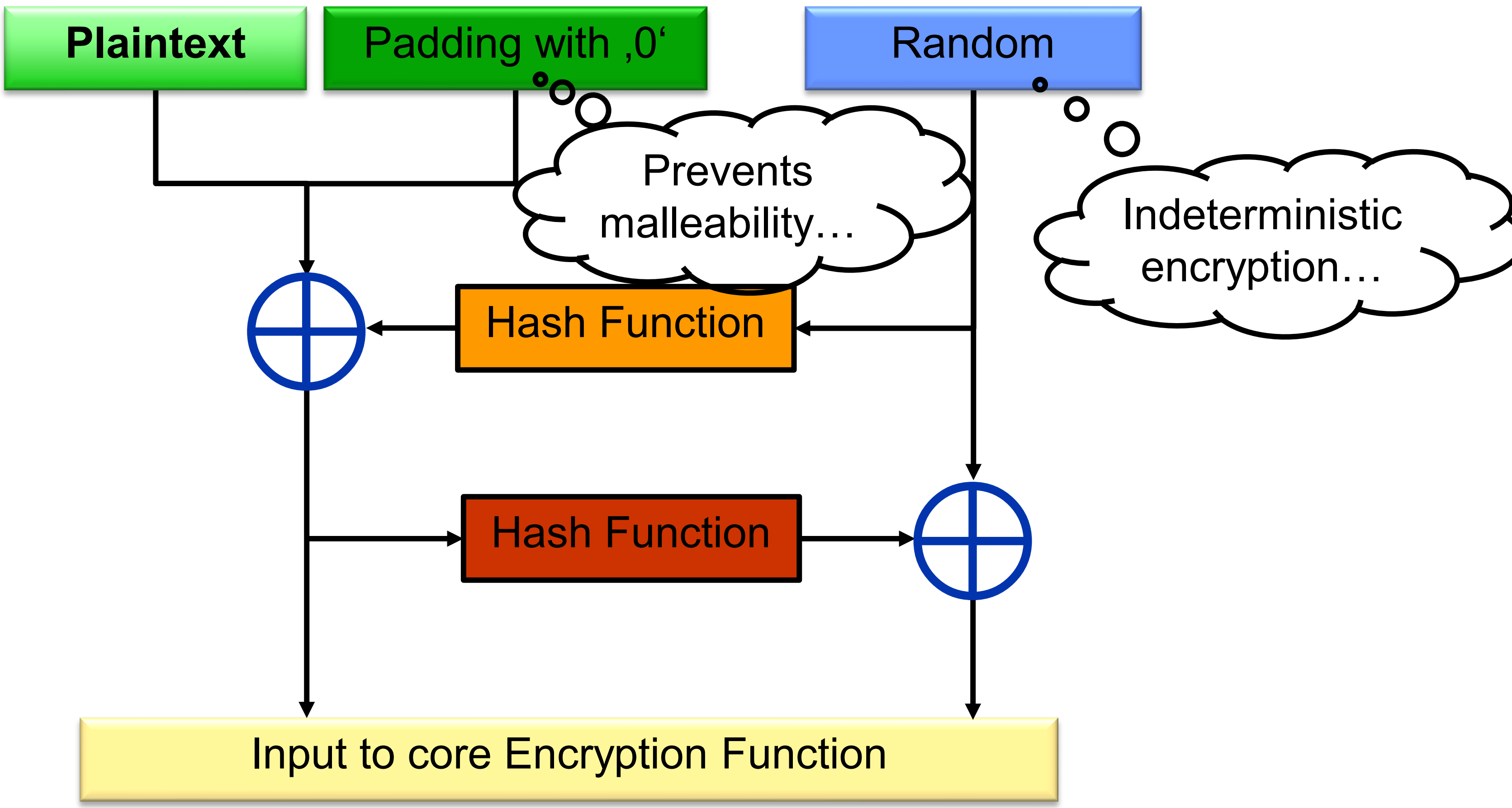
h() : collision-resistant hash function

Before signing, h is applied to the message

signature of m $= \left(\mathrm{h}(m)\right)^s \bmod n$

test if $\mathrm{h}(m) = \left(\left(\mathrm{h}(m)\right)^s\right)^t \bmod n$

Optimal Asymmetric Encryption Padding (OAEP)



Semantic security against adaptive chosen ciphertext attacks

Calculating with and without *p*,*q* (6)

squares and roots

$QR_n := \{ x \in Z_n^* \mid \exists\, y \in Z_n^* : y^2 \equiv x \bmod n \}$

x : "quadratic residue"

y : "root of x"

$-y$ is also a root

$(-1)^2 = 1$

but attention: e.g. mod 8

$1^2 \equiv 1 \quad 3^2 \equiv 1$

$7^2 \equiv 1 \quad 5^2 \equiv 1$

4 roots

QR_n multiplicative group:

$x_1, x_2 \in QR_n \Rightarrow \quad x_1 \bullet x_2 \in QR_n : (y_1 y_2)^2 = y_1^2 y_2^2 = x_1 x_2$

$x_1^{-1} \in QR_n : (y_1^{-1})^2 = (y_1^2)^{-1} = x_1^{-1}$

Calculating with and without *p*,*q* (7)

squares and roots mod *p*, **prime**:

Z_p field

$\Rightarrow$ as usual $\leq$ 2 roots

$x \not\equiv 0,\ p \neq 2:$ 0 or 2 roots

$\Rightarrow |QR_p| = \frac{p-1}{2}$ (square function is $2 \rightarrow 1$)

x	0	1	2	...	$\frac{p-1}{2}$	$-\frac{p-1}{2}$	...	-2	$-1 = p - 1$
x^2	0	1	4	...			...	4	1

Jacobi symbol $\left(\frac{x}{p}\right) := \begin{cases} 1 & \text{if } x \in QR_p \\ -1 & \text{else} \end{cases}$ (for $x \in Z_p^*$)

Calculating with and without *p*,*q* (8)

Continuation squares and roots mod *p*, **prime**:

Euler criterion : $$\left[\frac{x}{p}\right] \equiv x^{\frac{p-1}{2}} \bmod p$$

(i.e. fast algorithm to test whether square)

Proof using little Theorem of Fermat: $x^{p-1} \equiv 1 \mod p$

co-domain ok : $x^{\frac{p-1}{2}} \in \{\pm 1\}$, because $\left(x^{\frac{p-1}{2}}\right)^2 \equiv 1$

x square : $$\left[\frac{x}{p}\right] = 1 \Rightarrow x^{\frac{p-1}{2}} \equiv (y^2)^{\frac{p-1}{2}} \equiv y^{p-1} \equiv 1$$

x nonsquare : The $\frac{p-1}{2}$ solutions of $x^{\frac{p-1}{2}} \equiv 1$ are the squares.

So no nonsquare satisfies the equation.

Therefore: $x^{\frac{p-1}{2}} \equiv -1$.

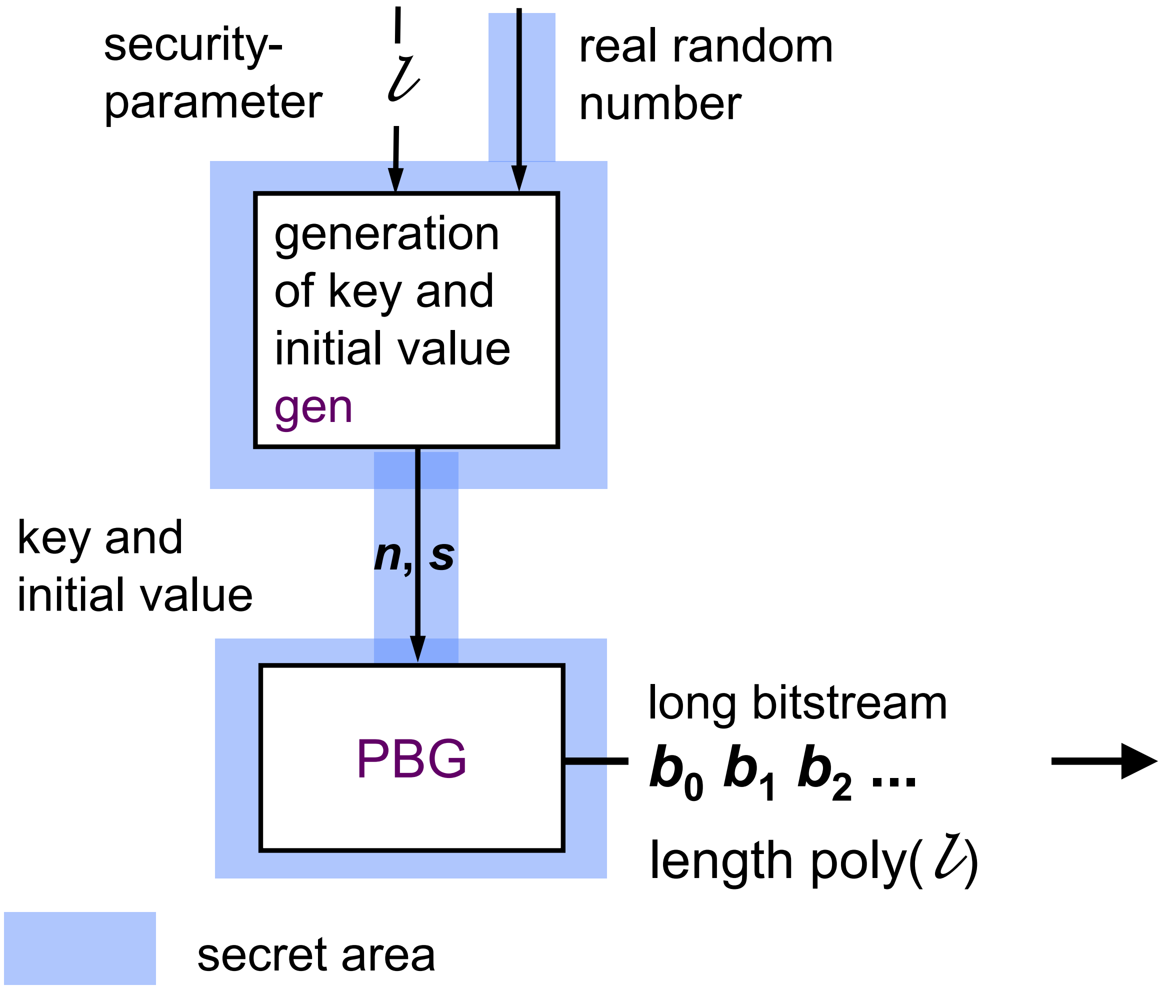
Pseudo-random Bitstream Generator (PBG)

Idea: short initial value (seed) $\rightarrow$ long bit sequence (should look random from a polynomial attacker's point of view)

Pseudo-random Bitstream Generator (PBG)

Idea: short initial value (seed) → long bit sequence (should look random from a polynomial attacker's point of view)

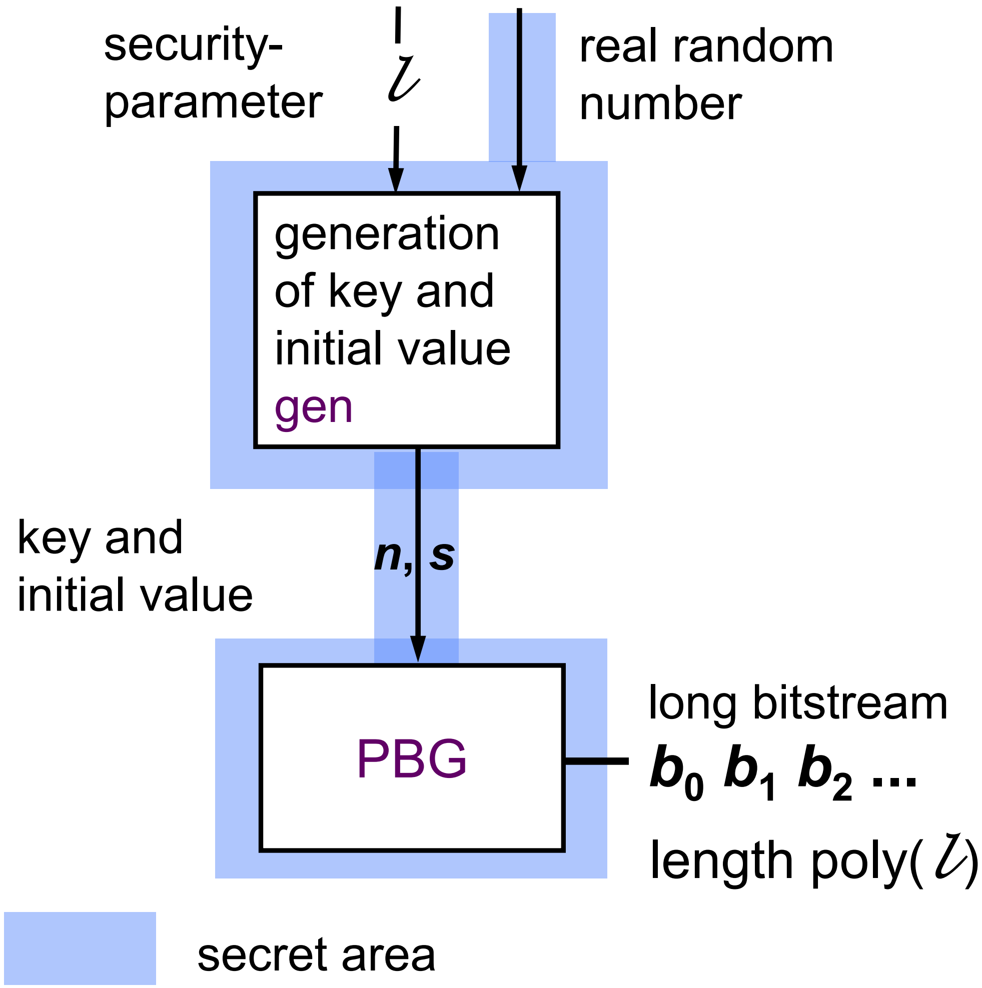
Scheme:



Pseudo-random Bitstream Generator (PBG)

Idea: short initial value (seed) → long bit sequence (should look random from a polynomial attacker's point of view)

Scheme:



Requirements:

- gen and PBG are efficient
- PBG is deterministic

  (⇒ sequence reproducible)
- secure: no probabilistic polynomial test can distinguish PBG-streams from real random streams

s^2-mod-n-generator (Blum/Blum/Shub 1983/86)

Method

- key value: p,q: prime / big / $\equiv 3 \bmod 4$
  $n = p \cdot q$
- initial value (seed): $s \in Z_n^*$
- PBG: $s_0 := s^2 \bmod n$
  $s_{i+1} := s_i^2 \bmod n$ $\quad b_i := s_i \bmod 2$
  ... (last bit)
  ...

Example: $n = 3 \cdot 11 = 33$, $s = 2$

index	0	1	2	3	4
s_i :	4	16	25	31	4
b_i :	0	0	1	1	0

$16^2 \bmod 33$
$= 8 \cdot 32 = 8 \cdot (-1) = 25$

$25^2 = (-8)^2 \equiv 64 \equiv 31$

$31^2 = (-2)^2 = 4$

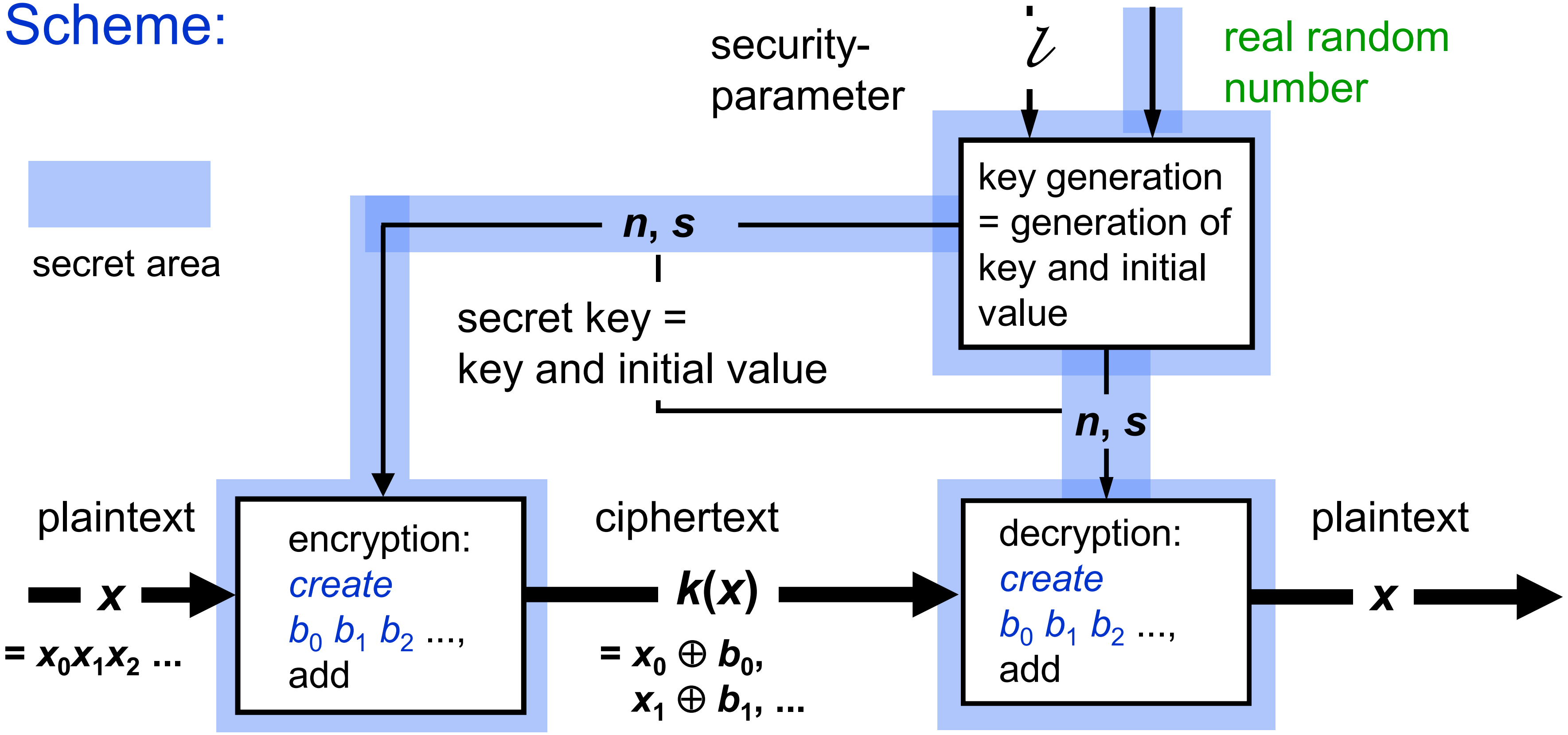
Note: length of period no problem with big numbers

s^2-mod-n-generator as symmetric encryption system

Purpose: application as symmetric encryption system: "Pseudo one-time pad"

Compare: one-time pad: add long real random bit stream with plaintext
Pseudo one-time pad: add long pseudo-random stream with plaintext

Scheme:



s^2-mod-n-generator as sym. encryption system: security

Idea:

If no probabilistic polynomial test can distinguish pseudo-random streams from real random streams, then the pseudo one-time pad is as good as the one-time pad against polynomial attacker.

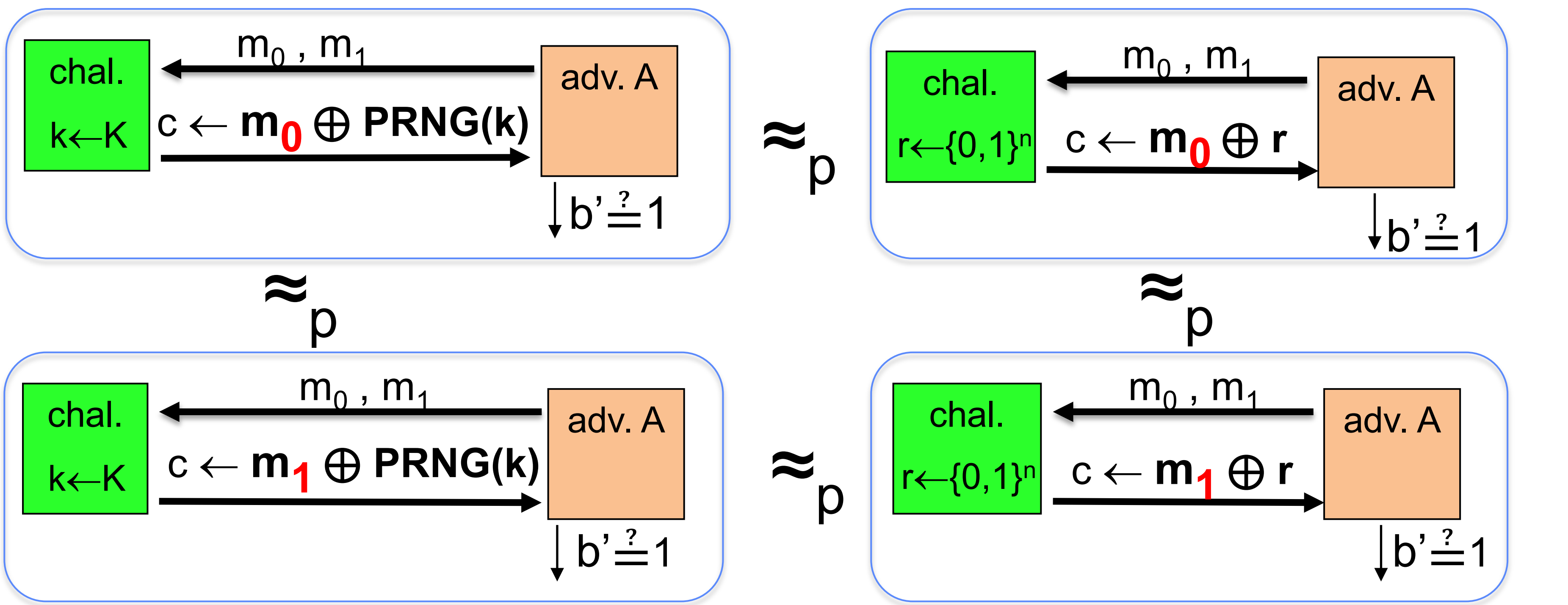
(Else the attack is a test !)

Construction works with any good PBG

Proof of security of pseudo one-time pad: another approach

(Based on slide from Prof. Thorsten Strufe)

- Prerequisite:
  - Unpredictable PRNG, which cannot be distinguished from real randomness
- We known:
  - One-time pad (XOR with truly random bit string) is secure
- Proof intuition:



Calculating with and without *p*,*q* (9)

squares and roots mod $p \equiv 3 \bmod 4$

- extracting roots is easy: given $x \in QR_p$

$$w := x^{\frac{p+1}{4}} \bmod p \text{ is root}$$

proof :

1. $p \equiv 3 \bmod 4 \Rightarrow \frac{p+1}{4} \in \mathbb{N}$

2. $w^2 = x^{\frac{p+1}{2}} = x^{\frac{p-1}{2}+1} = x^{\frac{p-1}{2}} \cdot x = 1 \cdot x$

$\Downarrow$

Euler, $x \in QR_p$

In addition: $w \in QR_p$ (power of $x \in QR_p$) $\rightarrow$ extracting roots iteratively is possible

- $\left(\frac{-1}{p}\right) \equiv (-1)^{\frac{p-1}{2}} = (-1)^{\frac{4r+2}{2}} = (-1)^{2r+1} = -1$

$p = 4r+3$

$\Rightarrow$ $-1 \notin QR_p$

$\Rightarrow$ of the roots $\pm w$: $-w \notin QR_p$ (otherwise $-1 = (-w) \cdot w^{-1} \in QR_p$)

Calculating with and without *p*,*q* (10)

squares and roots mod n using p,q

(usable as secret operations)

- testing whether square is simple $(n = p \cdot q,\ \ p,q$ prime, $\ p \neq q)$

$$x \in QR_n \iff x \in QR_p \land x \in QR_q$$

Chinese Remainder Theorem

proof: "$\Rightarrow$" $x \equiv w^2 \bmod n \Rightarrow x \equiv w^2 \bmod p \land x \equiv w^2 \bmod q$

"$\Leftarrow$" $x \equiv w_p^2 \bmod p \land x \equiv w_q^2 \bmod q$

$w := \mathrm{CRA}(w_p, w_q)$

then $w \equiv w_p \bmod p \land w \equiv w_q \bmod q$

using the Chinese Remainder Theorem for

$w^2 \equiv w_p^2 \equiv x \bmod p \land w^2 \equiv w_q^2 \equiv x \bmod q$

we have

$w^2 \equiv x \bmod n$

Calculating with and without *p*,*q* (11)

Continuation squares und roots mod n using p,q

$x \in QR_n \Rightarrow x$ has exactly 4 roots

(mod p and mod q : $\pm w_p$, $\pm w_q$.

therefore the 4 combinations according to the Chinese Remainder Theorem)

- extracting a root is easy (p, $q \equiv 3 \bmod 4$)

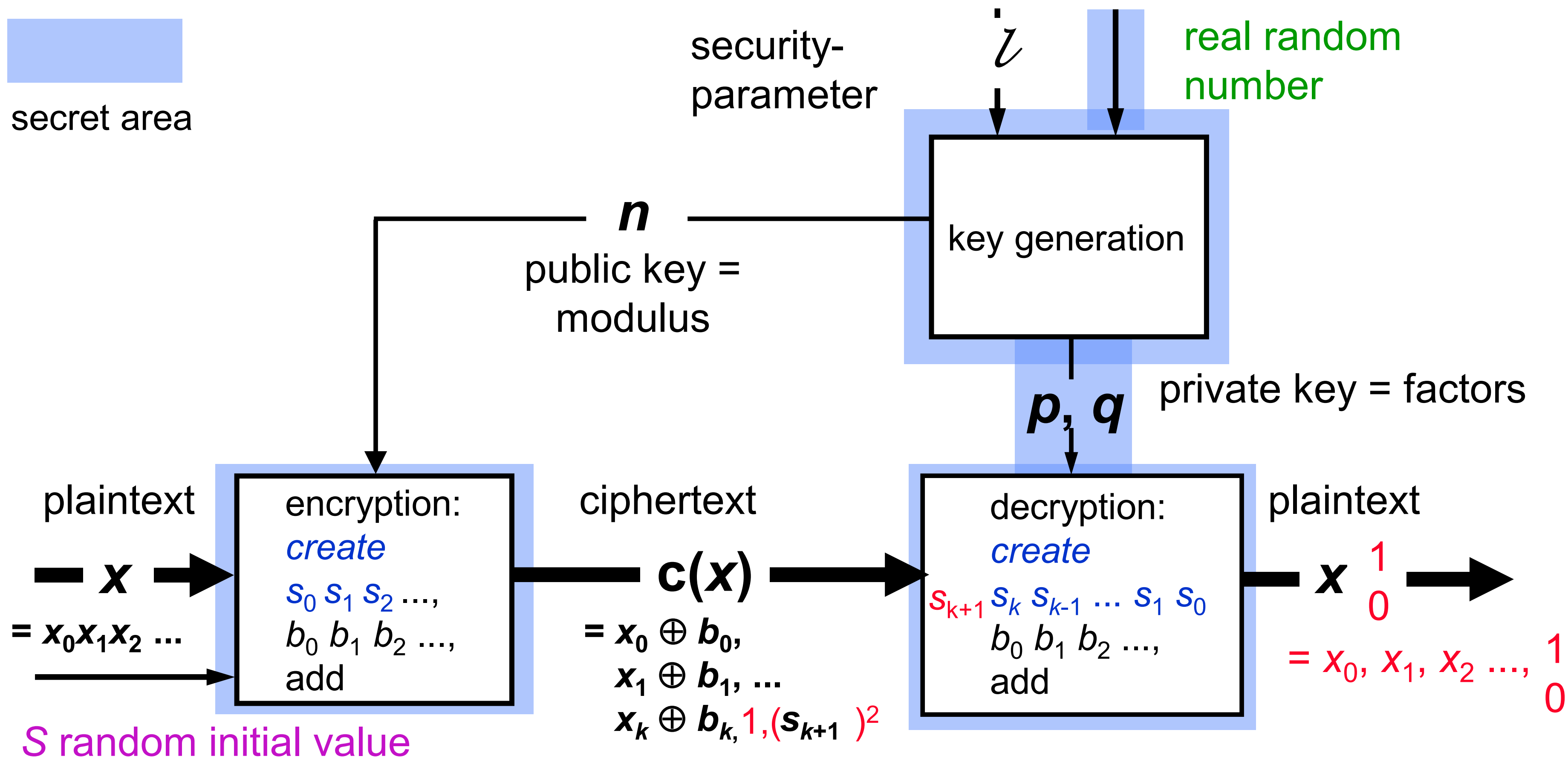
  determine roots w_p, w_q mod p, q

$$w_p := x^{\frac{p+1}{4}} \qquad w_q := x^{\frac{q+1}{4}}$$

combine using CRA

s^2-mod-n-generator as asymmetric encryption system

chosen ciphertext-plaintext attack



Calculating with and without *p*,*q* (12)

Continuation squares und roots mod *n* using *p*,*q*

Jacobi symbol $\left[\dfrac{x}{n}\right] := \left[\dfrac{x}{p}\right] \bullet \left[\dfrac{x}{q}\right]$

So: $$\left[\dfrac{x}{n}\right] = \begin{cases} +1 & \text{if} \quad x \in QR_p \wedge x \in QR_q \vee \\ & \phantom{\text{if}} \quad x \notin QR_p \wedge x \notin QR_q \\ -1 & \text{if} \quad \text{“cross-over”} \end{cases}$$

So : $x \in QR_n \quad \Rightarrow \quad \left[\dfrac{x}{n}\right] = 1$

$\not\Leftarrow$ does not hold

Calculating with and without *p*,*q* (13)

continuation squares und roots mod *n* using *p*,*q*

to determine the Jacobi symbol is easy

e.g. $p \equiv q \equiv 3 \bmod 4$

$$\left(\frac{-1}{n}\right) = \left(\frac{-1}{p}\right) \bullet \left(\frac{-1}{q}\right) = (-1)\bullet(-1) = 1$$

but $-1 \notin QR_n$, because $-1 \notin QR_{p,q}$

Calculating with and without *p*,*q* (14)

squares and roots mod n without p,q

- extracting roots is difficult: provably so difficult as to factor

  a) If someone knows 2 significantly different roots of an x mod n, then he can definitely factor n.

  (i.e. $w_1^2 \equiv w_2^2 \equiv x$, but $w_1 \not\equiv \pm w_2 \Rightarrow n \nmid (w_1 \pm w_2)$)

  proof: $n \mid w_1^2 - w_2^2 \Rightarrow n \mid (w_1+w_2)(w_1-w_2)$

  p in one factor, q in the other

  $\Rightarrow$ gcd(w_1+w_2, n) is p or q

Calculating with and without *p*,*q* (15)

Continuation squares und roots mod *n* without *p*,*q*

b) Sketch of “factoring is difficult $\Rightarrow$ extracting a root is difficult”
proof of “factoring is easy $\Leftarrow$ extracting a root is easy”
So assumption : $\exists\, \mathcal{W} \in$ PPA: algorithm extracting a root
to show : $\exists\, \mathcal{F} \in$ PPA: factoring algorithm

structure

program $\mathcal{F}$
subprogram $\mathcal{W}$
[black box]
begin
...
call $\mathcal{W}$
...
call $\mathcal{W}$
...
end.

polynomially often

Calculating with and without *p*,*q* (16)

to b)

$\mathcal{F}$: input n

repeat forever

choose $w \in Z_n^*$ at random, set $x := w^2$

$w' := \mathcal{W}(n,x)$

test whether $w' \not\equiv \pm w$, if so factor according to a) break

- to determine the Jacobi symbol is easy

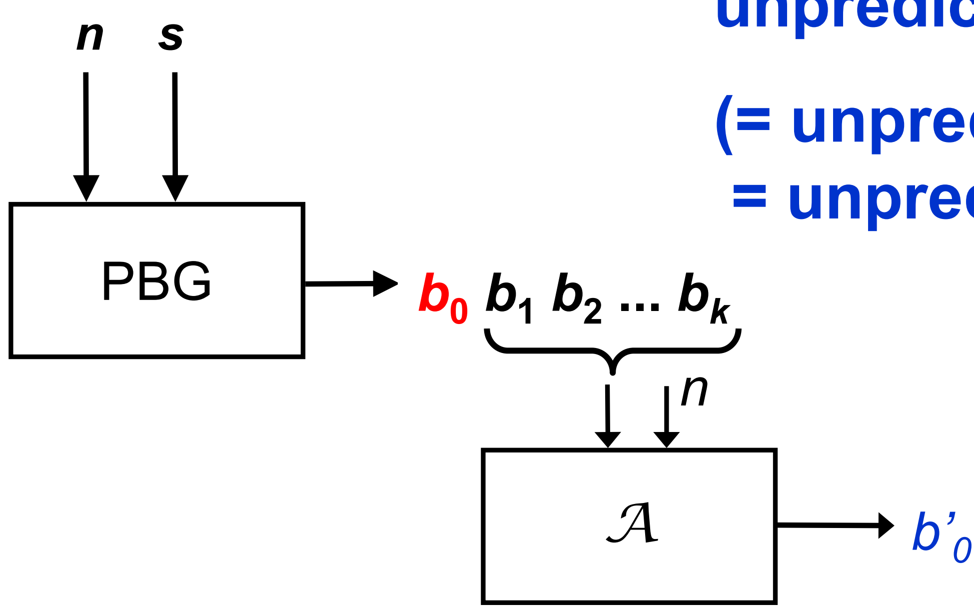
(if p and q unknown: use quadratic law of reciprocity)

but note : If $\left(\frac{x}{n}\right) = 1$, determine whether $x \in QR_n$ is difficult

(i.e. it does not work essentially better than to guess)

QRA = quadratic residuosity assumption

Security of the s^2-mod-n-generator (1)



unpredictability to the left will do

(= unpredictability to right
= unpredictability of middle)

(see L. Blum, M. Blum, M. Shub 1986)

s^2-mod-n-generator is cryptographically strong: $\Leftrightarrow$

$\forall\ \mathcal{A} \in$ PPA { predictor for b_0 }

$\forall$ constants δ, $0 < \delta < 1$ { frequency of "bad" n }

$\forall\ t \in$ N : { degree of the polynomial }

if l (= $|n|$) sufficiently big it holds: for all keys n except of at most a δ-fraction

$$P(b_0=\mathcal{A}(n,b_1b_2...b_k)=b'_0 \mid s \in Z_n^* \text{ random}) \leftarrow \frac{1}{2} + \frac{1}{l^t}$$

Security of the s^2-mod-n-generator (2)

Proof: Contradiction to QRA in 2 steps

Assumption: s^2-mod-n-generator is weak, i.e. there is a predictor $\mathcal{P}$, which guesses b_0 with ε-advantage given $b_1\ b_2\ b_3$ ...

Step 1: Transform $\mathcal{P}$ in $\mathcal{P}^*$, which to a given s_1 of QR_n guesses the last bit of s_0 with ε-advantage.

Given s_1.
Generate $b_1\ b_2\ b_3$ ... with s^2-mod-n-generator, apply $\mathcal{P}$ to that stream.
$\mathcal{P}$ guesses b_0 with ε-advantage. That is exactly the result of $\mathcal{P}^*$.

Step 2: Construct using $\mathcal{P}^*$ a method $\mathcal{R}$, that guesses with ε-advantage, whether a given s^* with Jacobi symbol +1 is a square.

Given s^*. Set $s_1 := (s^*)^2$.
Apply $\mathcal{P}^*$ to s_1. $\mathcal{P}^*$ guesses the last bit of s_0 with ε-advantage, where s^* and s_0 are roots of s_1; $s_0 \in QR_n$.
Therefore $s^* \in QR_n \Leftrightarrow s^* = s_0$

Security of the s^2-mod-n-generator (3)

The last bit b^* of s^* and the guessed b_0 of s_0 suffice to guess correctly, because

1) if $s^* = s_0$, then $b^* = b_0$
2) to show: if $s^* \neq s_0$, then $b^* \neq b_0$

   if $s^* \neq s_0$ because of the same Jacobi symbols, it holds

   $$s^* \equiv -s_0 \mod n$$

   therefore $s^* = n - s_0$ in Z

   n is odd, therefore s^* and s_0 have different last bits

The constructed $\mathcal{R}$ is in contradiction to QRA.

Notes:

1) You can take O(log($\mathcal{l}$)) random bits in place of (last) 1 bit per squaring.
2) There is a more difficult proof that s^2-mod-n-generator is secure under the factoring assumption.

RSA: Faster calculation of the secret operation

$$\underbrace{\qquad\qquad\qquad}_{y^d \equiv w \bmod n}$$

once and for all:

$$\boxed{\begin{array}{l} d_p := c^{-1} \bmod p\text{-}1 \\ d_q := c^{-1} \bmod q\text{-}1 \end{array}} \begin{array}{l} \Rightarrow (y^{d_p})^c \equiv y \bmod p \\ \Rightarrow (y^{d_q})^c \equiv y \bmod q \end{array}$$

every time:

$$\boxed{\text{set } w := \text{CRA}(y^{d_p}, y^{d_q})}$$

proof:

$$\Rightarrow w^c \equiv \begin{cases} (y^{d_p})^c \equiv y \bmod p \\ (y^{d_q})^c \equiv y \bmod q \end{cases}$$

$$\Rightarrow w^c \equiv y \quad \bmod n$$

How much faster ?

complexity exponentiation: $\approx \ell^3$

complexity 2 exponentiations of half the length: $\approx 2 \bullet \left(\frac{\ell}{2}\right)^3 = \frac{\ell^3}{4}$

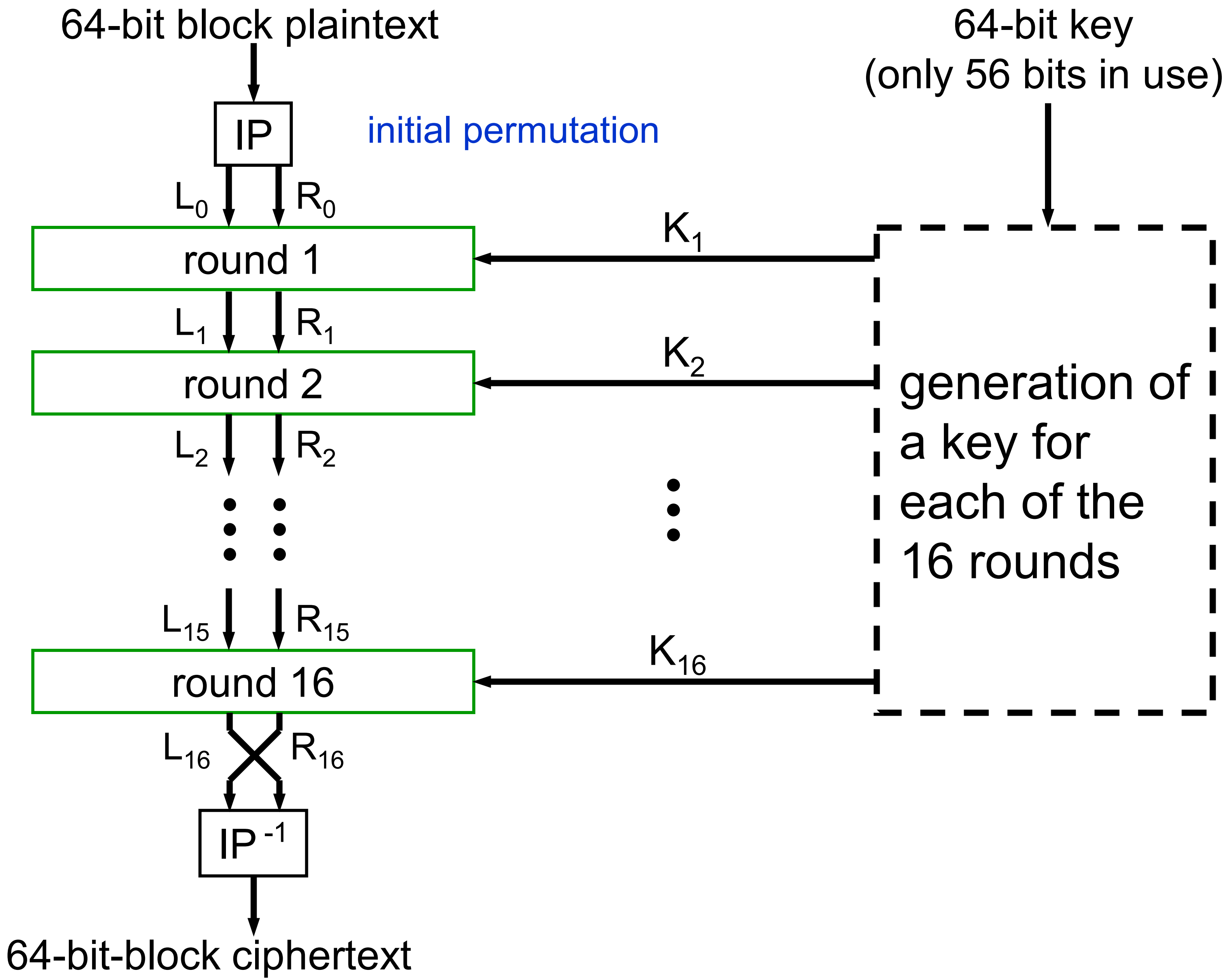
complexity CRA: 2 multiplications $\approx 2 \bullet \ell^2$

1 addition $\approx \ell$

irrelevant

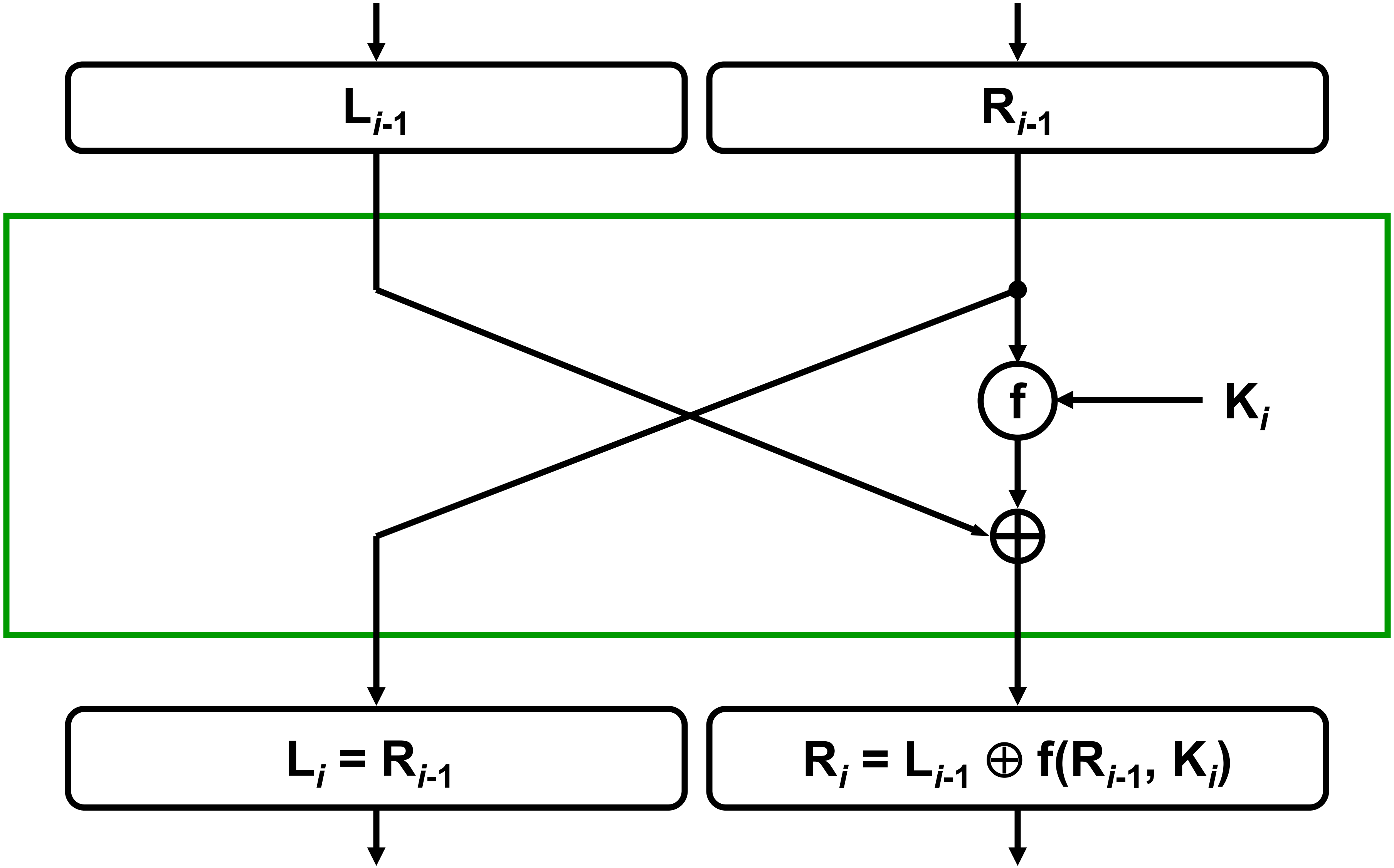
So: $\approx$ Factor 4

Symmetric Cryptosystem DES (IBM, 1975)

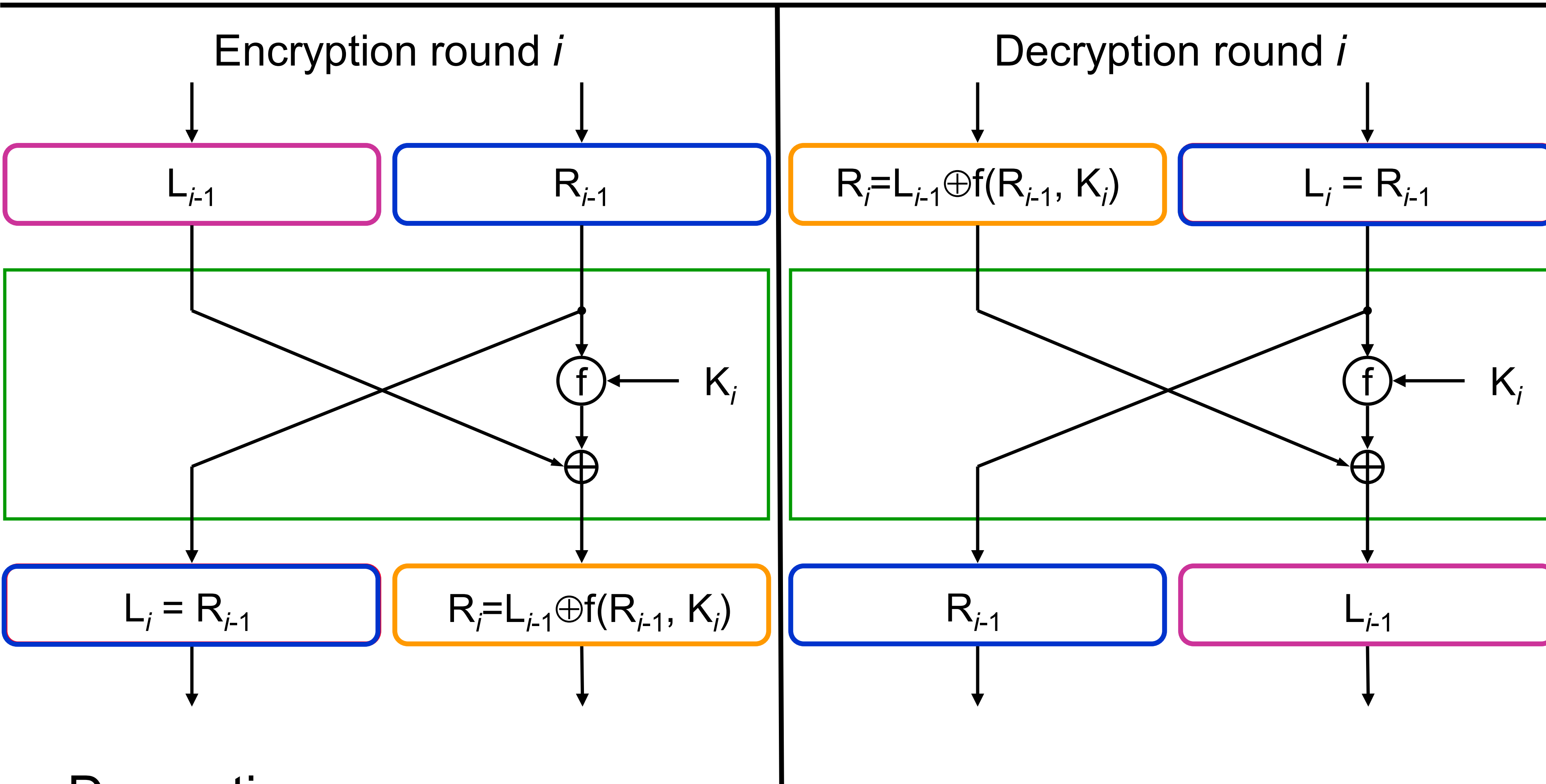


One round

Feistel ciphers self-inverse!



Why does decryption work?



Decryption

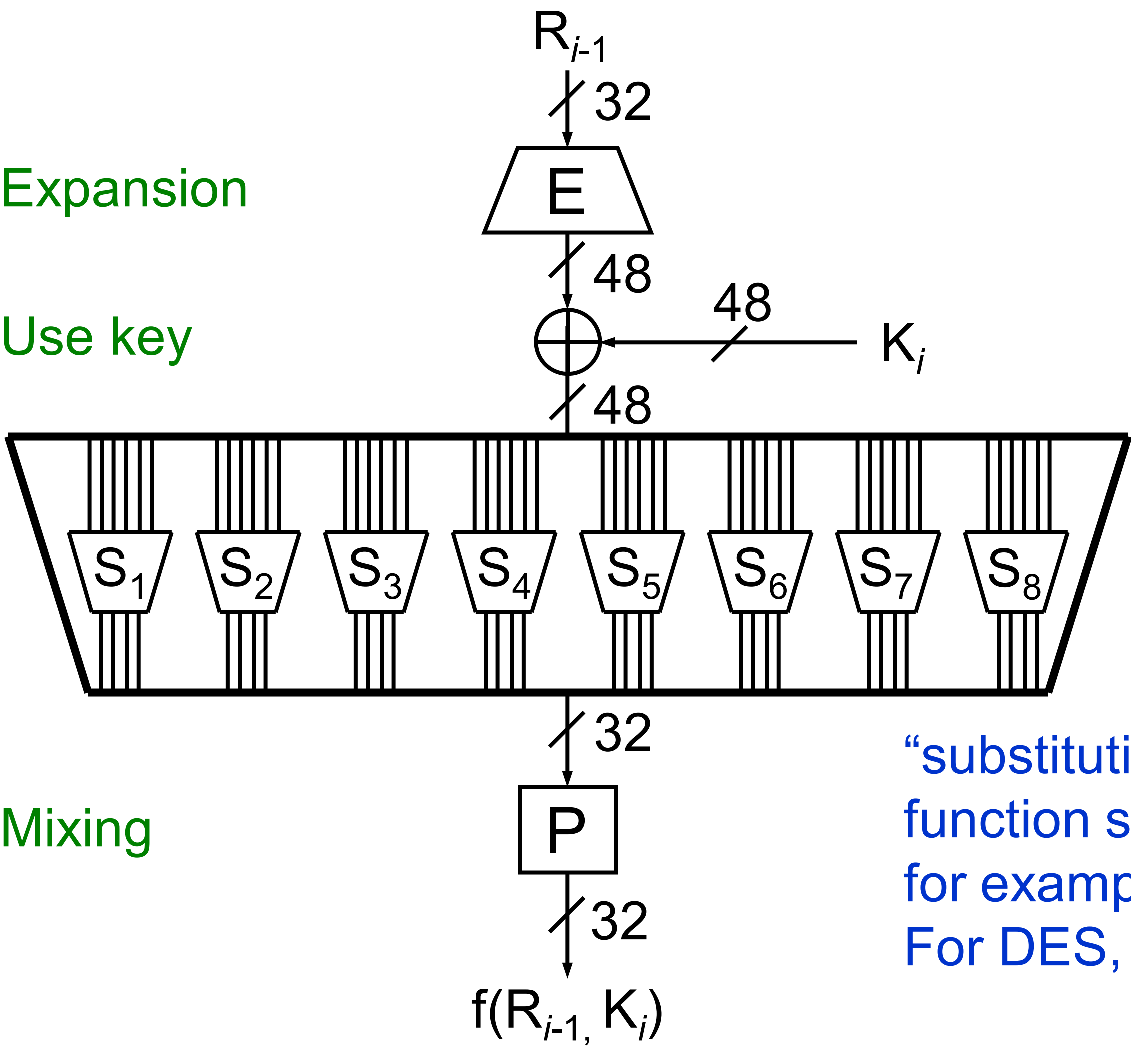
→ trivial

→ $\boxed{L_{i-1} \oplus f(R_{i-1}, K_i)} \oplus f(\boxed{L_i}, K_i) =$

$L_{i-1} \oplus f(R_{i-1}, K_i) \oplus f(\boxed{R_{i-1}}, K_i) = \boxed{L_{i-1}}$

replace L_i by R_{i-1}

Encryption function f



Expansion

Use key

Make f (and DES) non-linear (permutations and $\oplus$ are linear)

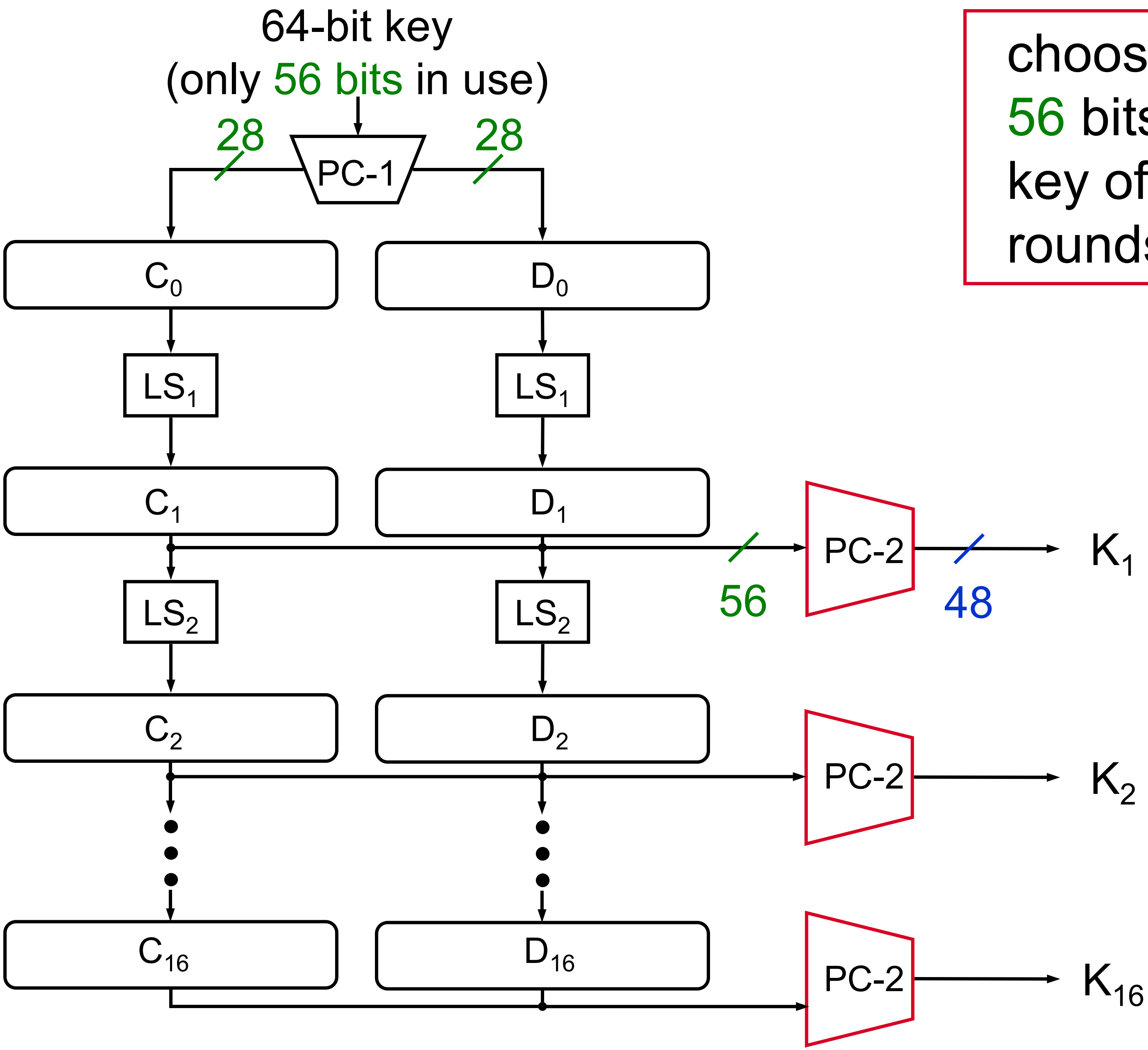
Mixing

“substitution box” S can implement any function $s : \{0,1\}^6 \to \{0,1\}^4$, for example as table.
For DES, the functions are fixed.

Terms

- Substitution-permutation networks
- Confusion - diffusion

Generation of a key for each of the 16 rounds

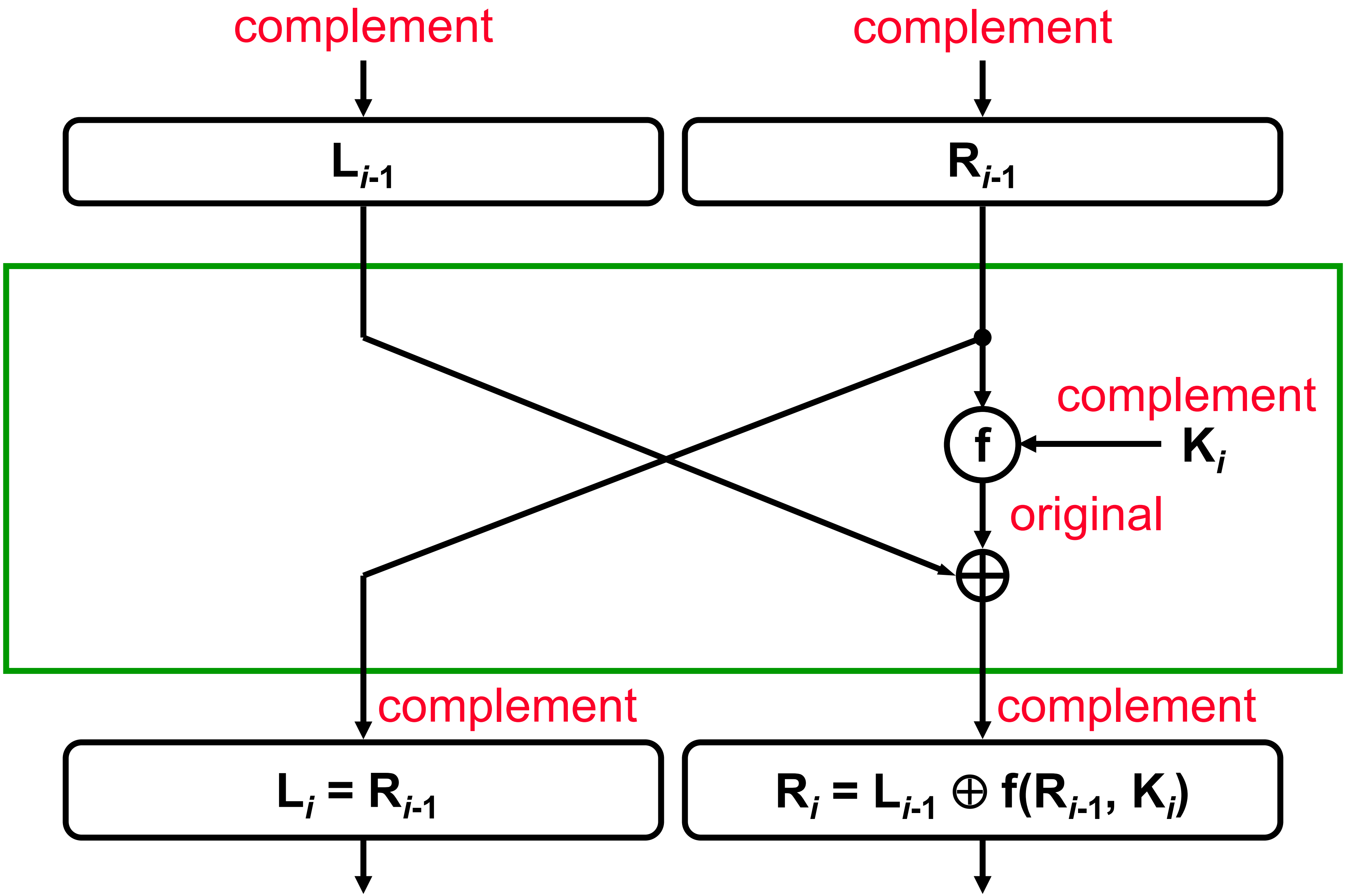


choose 48 of the 56 bits for each key of the 16 rounds

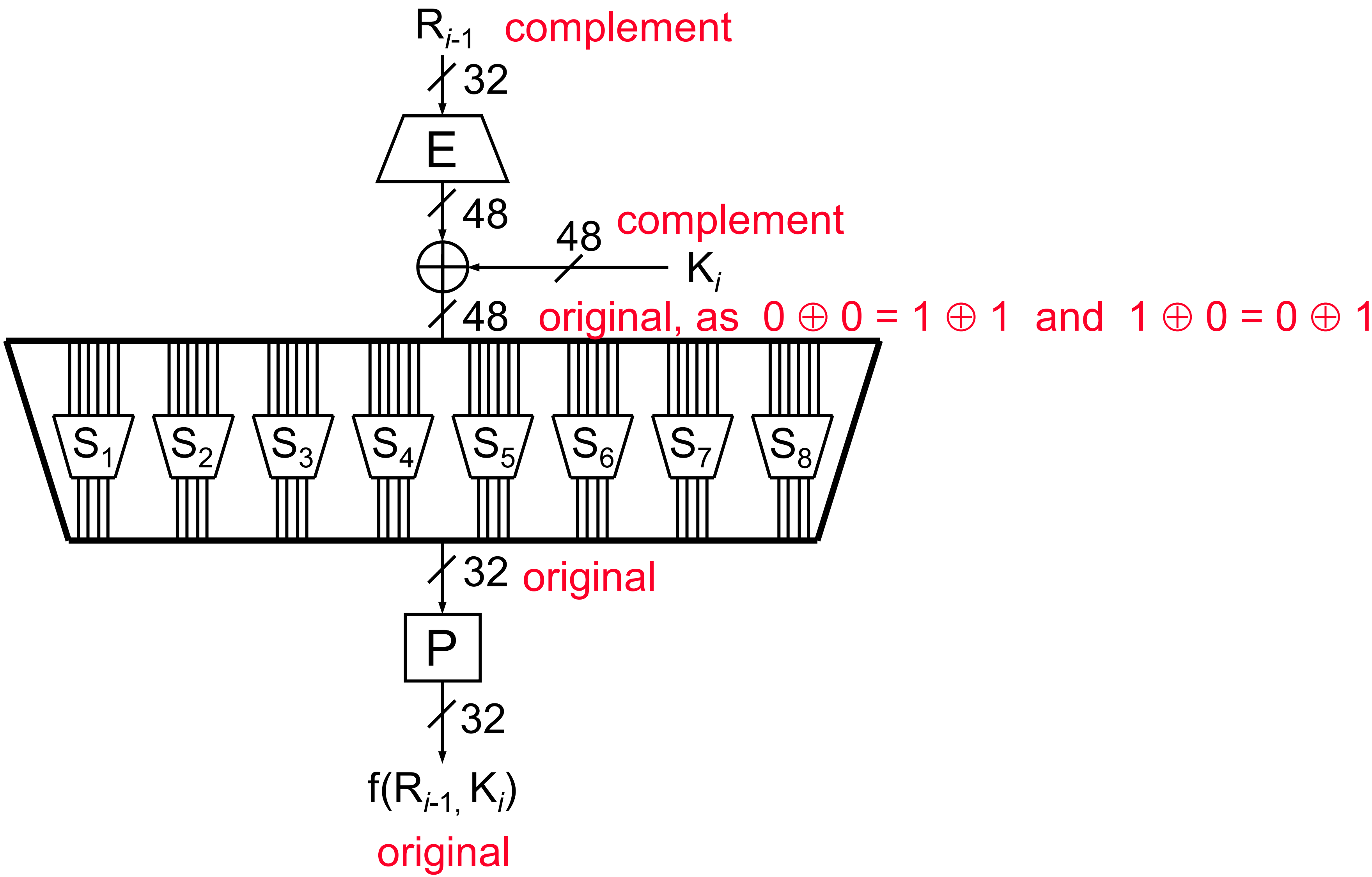
The complementation property of DES

$$\mathrm{DES}(\bar{k}, \bar{x}) = \overline{\mathrm{DES}(k, x)}$$

One round


complement
complement
L_{i-1}
R_{i-1}
complement
f
K_i
original
complement
complement
$L_i = R_{i-1}$
$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$

Encryption function f



From DES to 3DES

Goal:

Strengthen DES by increasing key length

Let $E : K \times M \longrightarrow M$ be a block cipher (DES)

Define $\mathbf{3E}: K^3 \times M \longrightarrow M$ as

$$\mathbf{3E}\big((k_1,k_2,k_3), m\big) = E(k_1, D(k_2, E(k_3,m)))$$

For 3DES: key-size = 3×56 = 168 bits. 3×slower than DES.

Why not E(E(E(m)))? … What if: $k_1 = k_2 = k_3$?

Simple attack feasible in time $\approx 2^{118}$

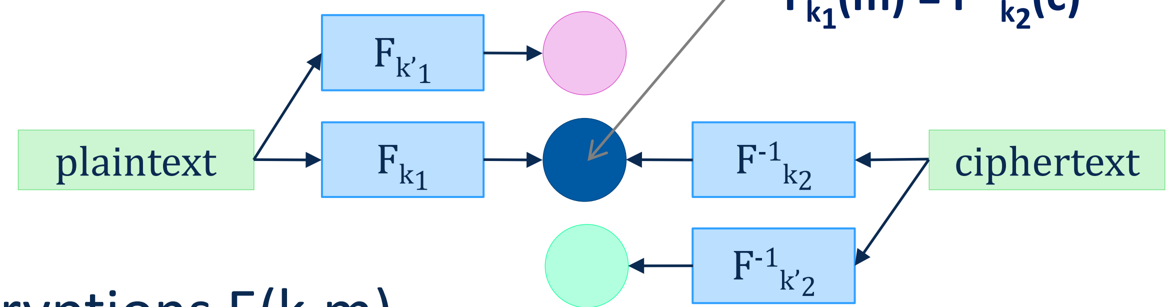
Meet-in-the-middle attack (no double DES?)

Define $2E\big((k_1,k_2), m\big) = E\big(k_1, E(k_2, m)\big)$

Idea: test if $E(m) = D(c)$

if $c = 2E_k(m)$ then

$F_{k_1}(m) = F^{-1}_{k_2}(c)$



Step 1: build table of encryptions $E(k,m)$

Step 2: for all $k \in \{0,1\}^{56}$ do:

test if $D(k, c)$ is in 2nd column.

k^0 = 00…00	$E(k^0, M)$
k^1 = 00…01	$E(k^1, M)$
k^2 = 00…10	$E(k^2, M)$
⋮	⋮
k^N = 11…11	$E(k^N, M)$

2^{56} entries

k^0 = 00…00	$E(k^0, M)$
k^1 = 00…01	$E(k^1, M)$
k^2 = 00…10	$E(k^2, M)$
⋮	⋮
k^N = 11…11	$E(k^N, M)$

Complexity of Meet-in-the-Middle

$$m \rightarrow E(\mathbf{k}_2,\cdot) \rightarrow E(\mathbf{k}_1,\cdot) \rightarrow c$$

Time = $2^{56}\log(2^{56}) + 2^{56}\log(2^{56}) < 2^{63} \ll 2^{112}$, space $\approx 2^{56}$

Same attack on 3DES: Time = 2^{118} , space $\approx 2^{56}$

$$m \rightarrow E(\mathbf{k}_3,\cdot) \rightarrow E(\mathbf{k}_2,\cdot) \rightarrow E(\mathbf{k}_1,\cdot) \rightarrow c$$

Cipher

Stream cipher

synchronous

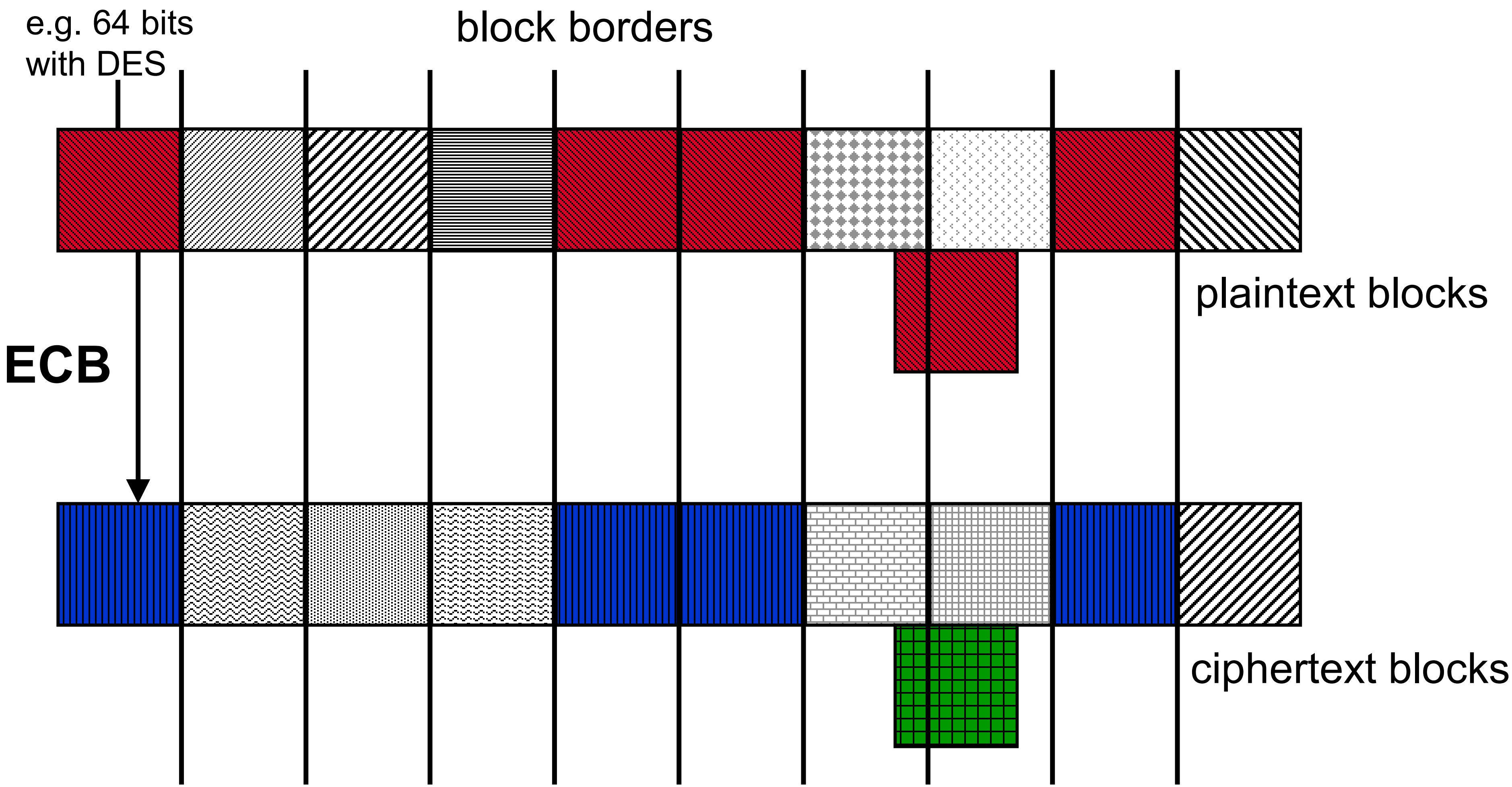
self synchronizing

Block cipher

Modes of operation:

Simplest:	ECB (<u>e</u>lectronic <u>c</u>ode<u>b</u>ook)
	each block separately
But:	concealment: block patterns identifiable
	authentication: blocks permutable

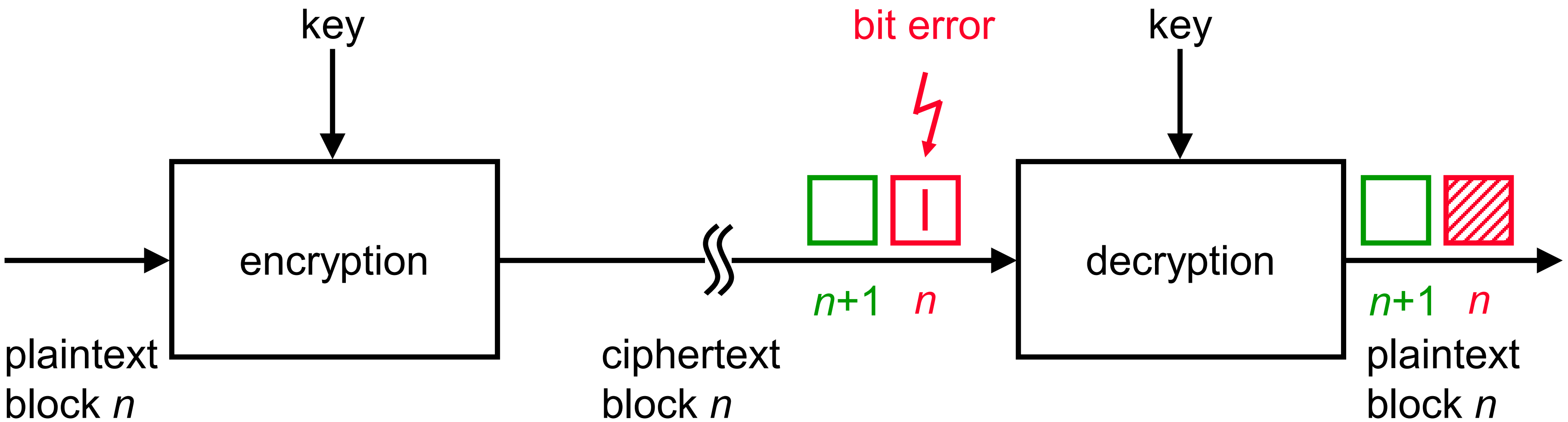
Main problem of ECB



same plaintext blocks $\xrightarrow{\text{ECB}}$ same ciphertext blocks

Telefax example (→ compression is helpful)

Electronic Codebook (ECB)



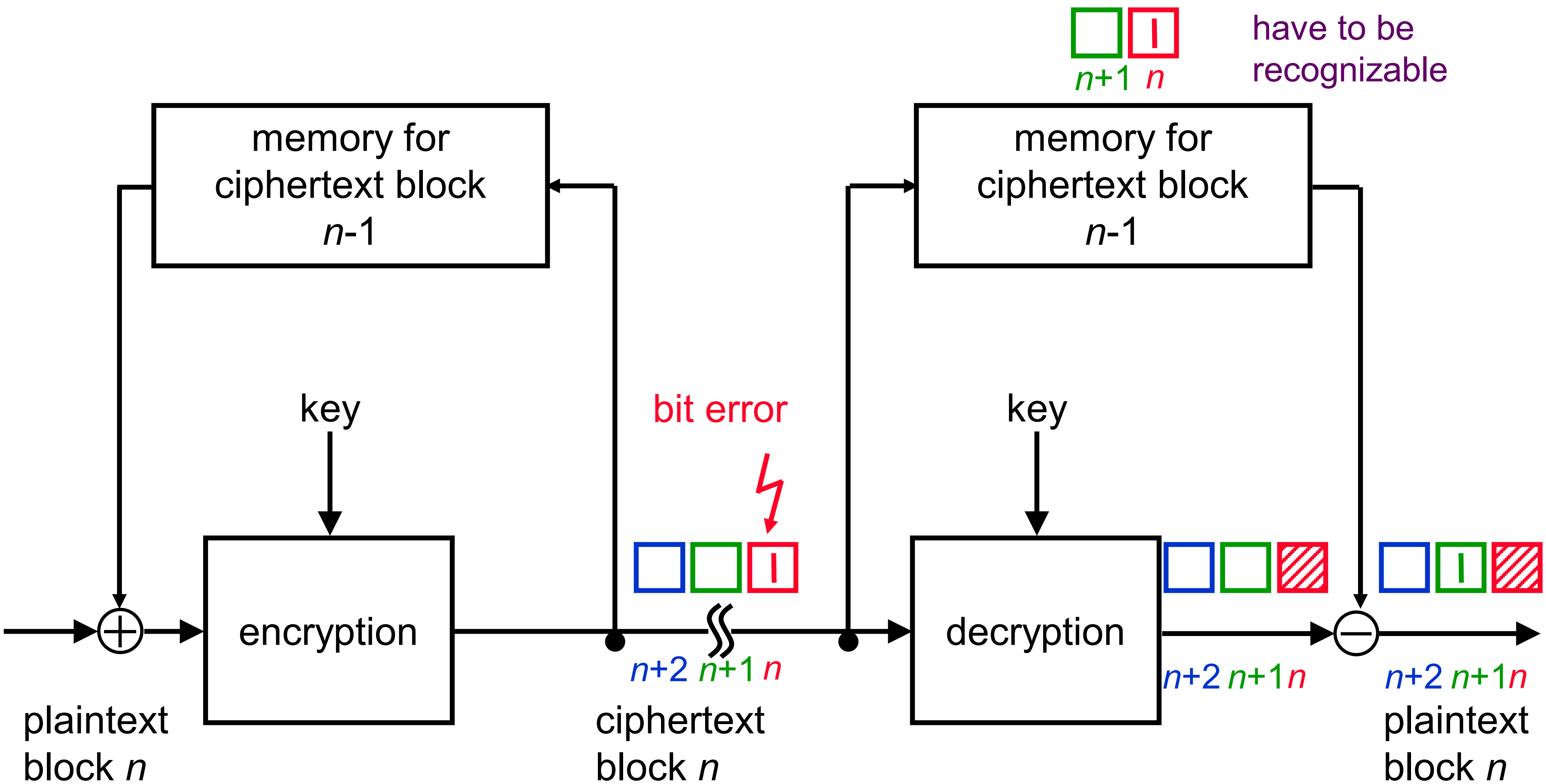
Cipher Block Chaining (CBC)

All lines transmit as many characters as a block comprises

⊕ Addition mod appropriately chosen modulus

⊖ Subtraction mod appropriately chosen modulus

If error on the line: Resynchronization after 2 blocks, but block borders have to be recognizable



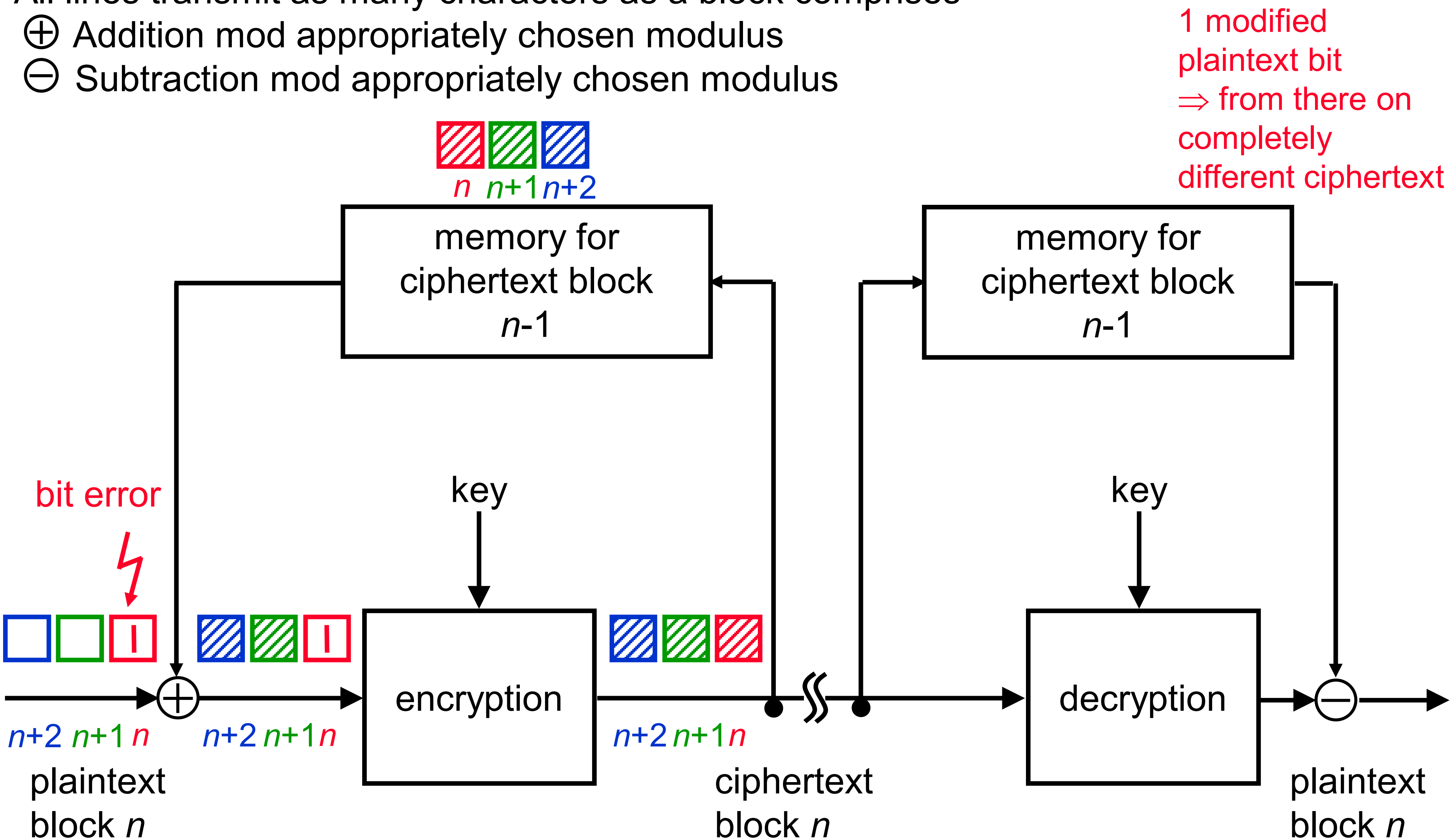
self synchronizing

Cipher Block Chaining (CBC) (2)

All lines transmit as many characters as a block comprises

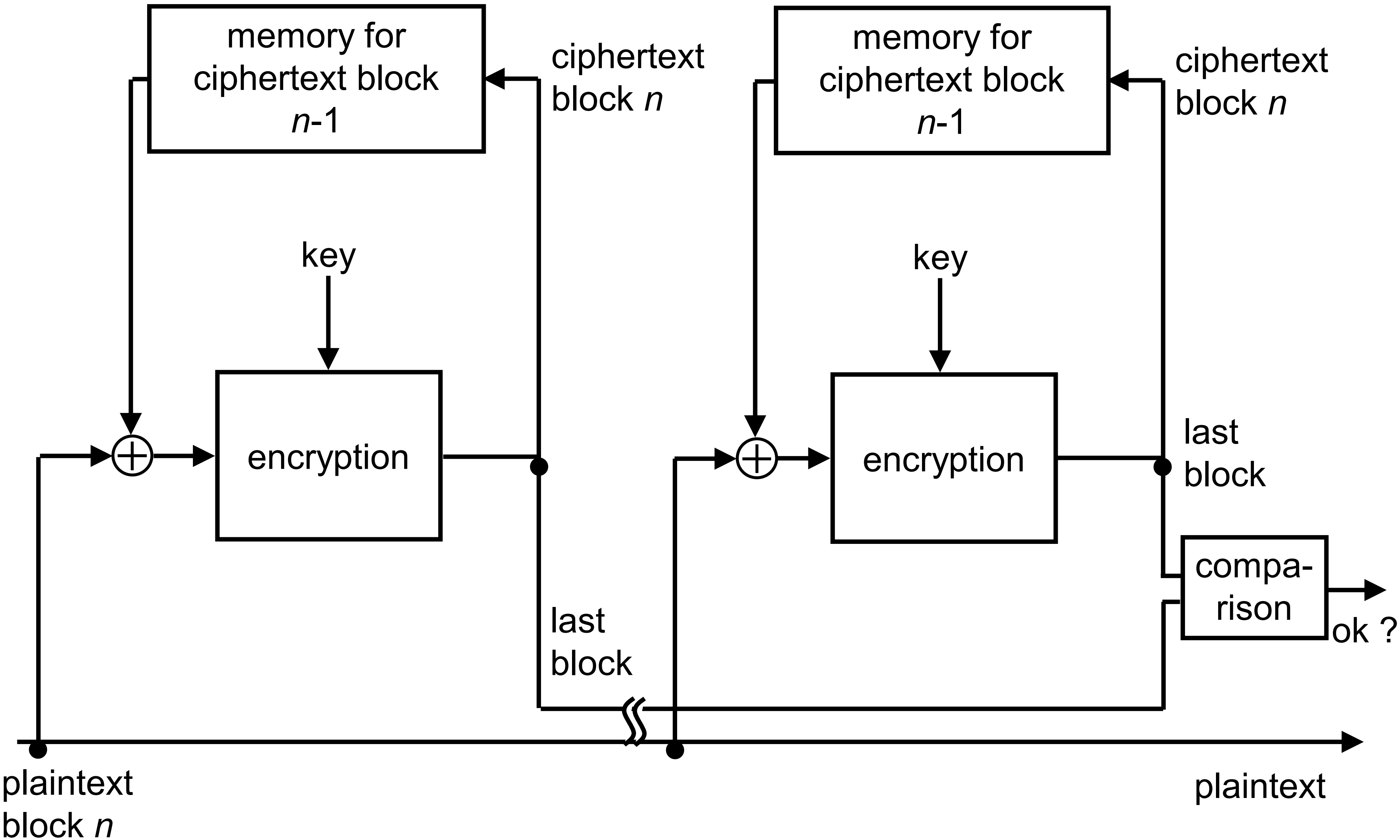
⊕ Addition mod appropriately chosen modulus

⊖ Subtraction mod appropriately chosen modulus

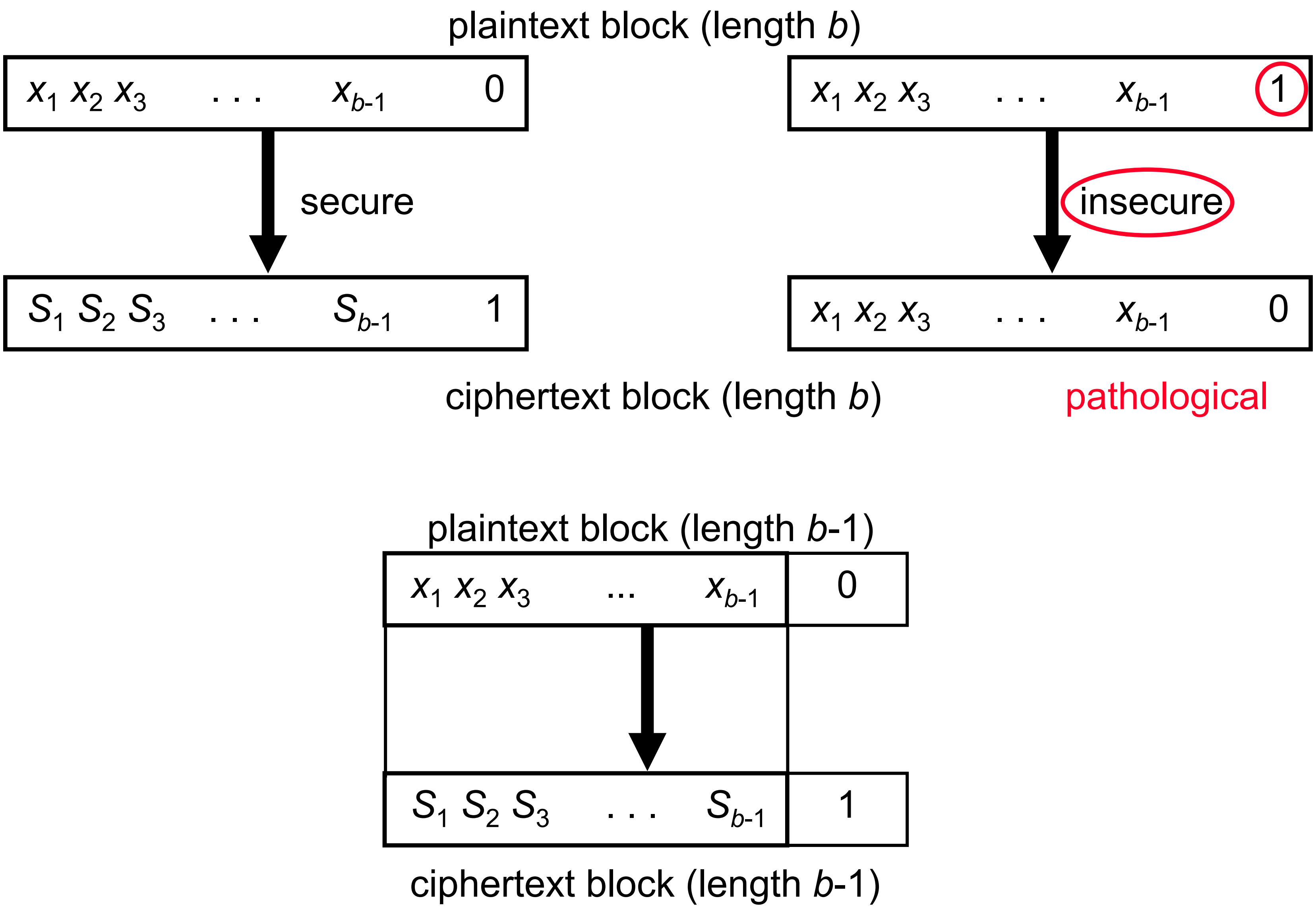


useable for authentication ⇒ use last block as MAC

CBC for authentication



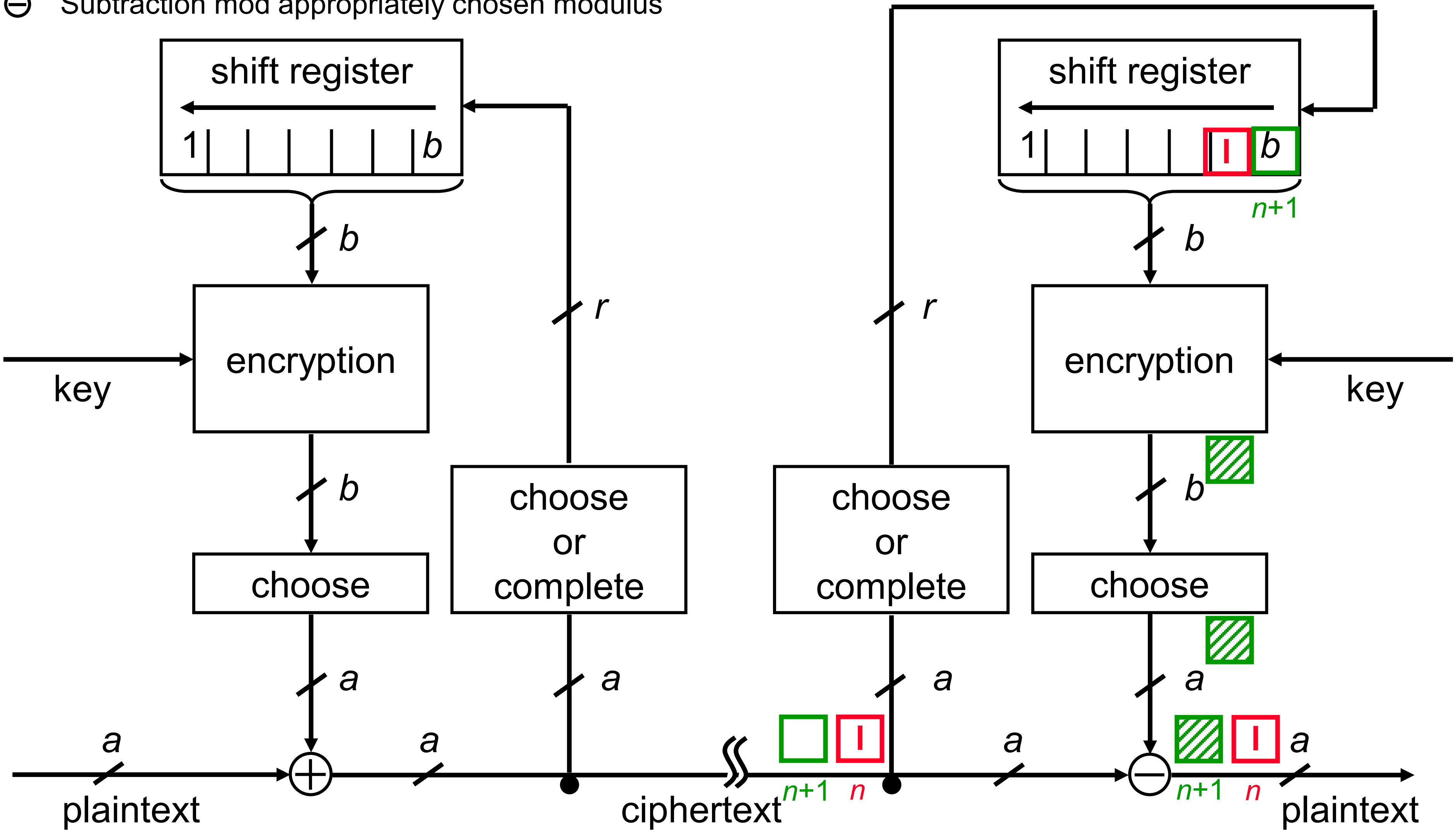
Pathological Block cipher



Cipher FeedBack (CFB)

b	Block length
a	Length of the output unit, $a \le b$
r	Length of the feedback unit, $r \le b$
$\oplus$	Addition mod appropriately chosen modulus
$\ominus$	Subtraction mod appropriately chosen modulus

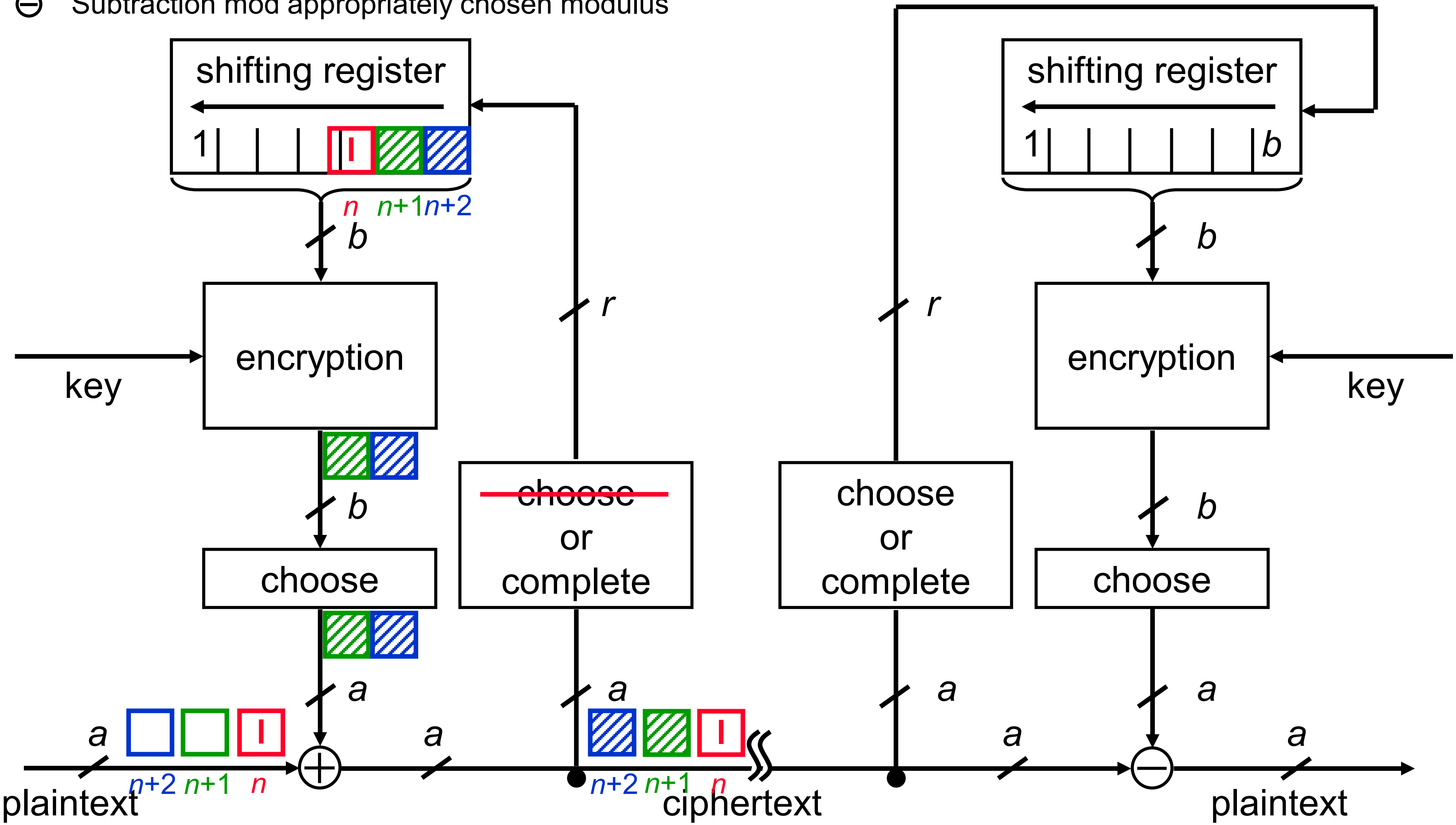
symmetric;
self synchronizing



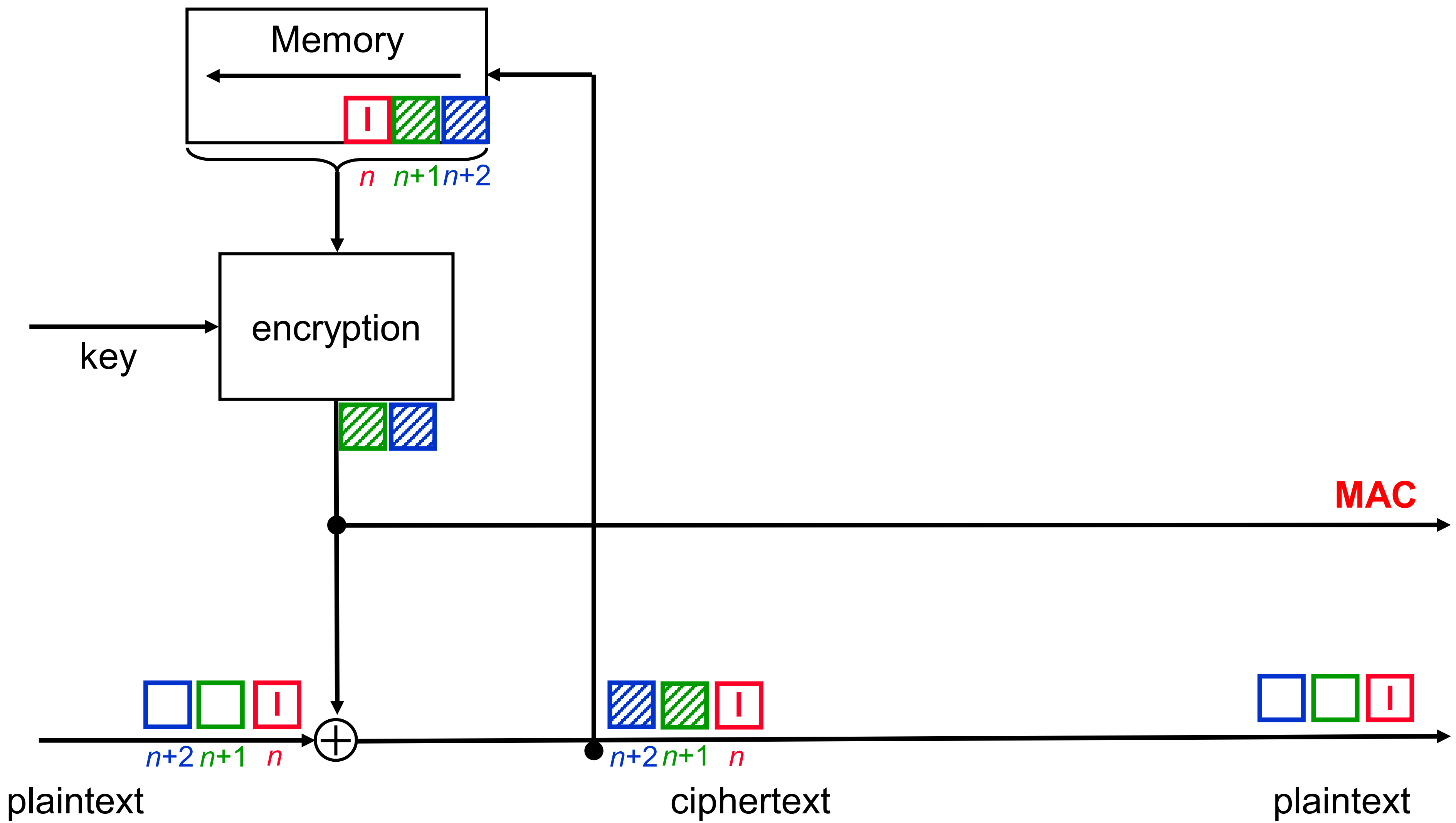
Cipher FeedBack (CFB) (2)

b	Block length
a	Length of the output unit, $a \le b$
r	Length of the feedback unit, $r \le b$
$\oplus$	Addition mod appropriately chosen modulus
$\ominus$	Subtraction mod appropriately chosen modulus

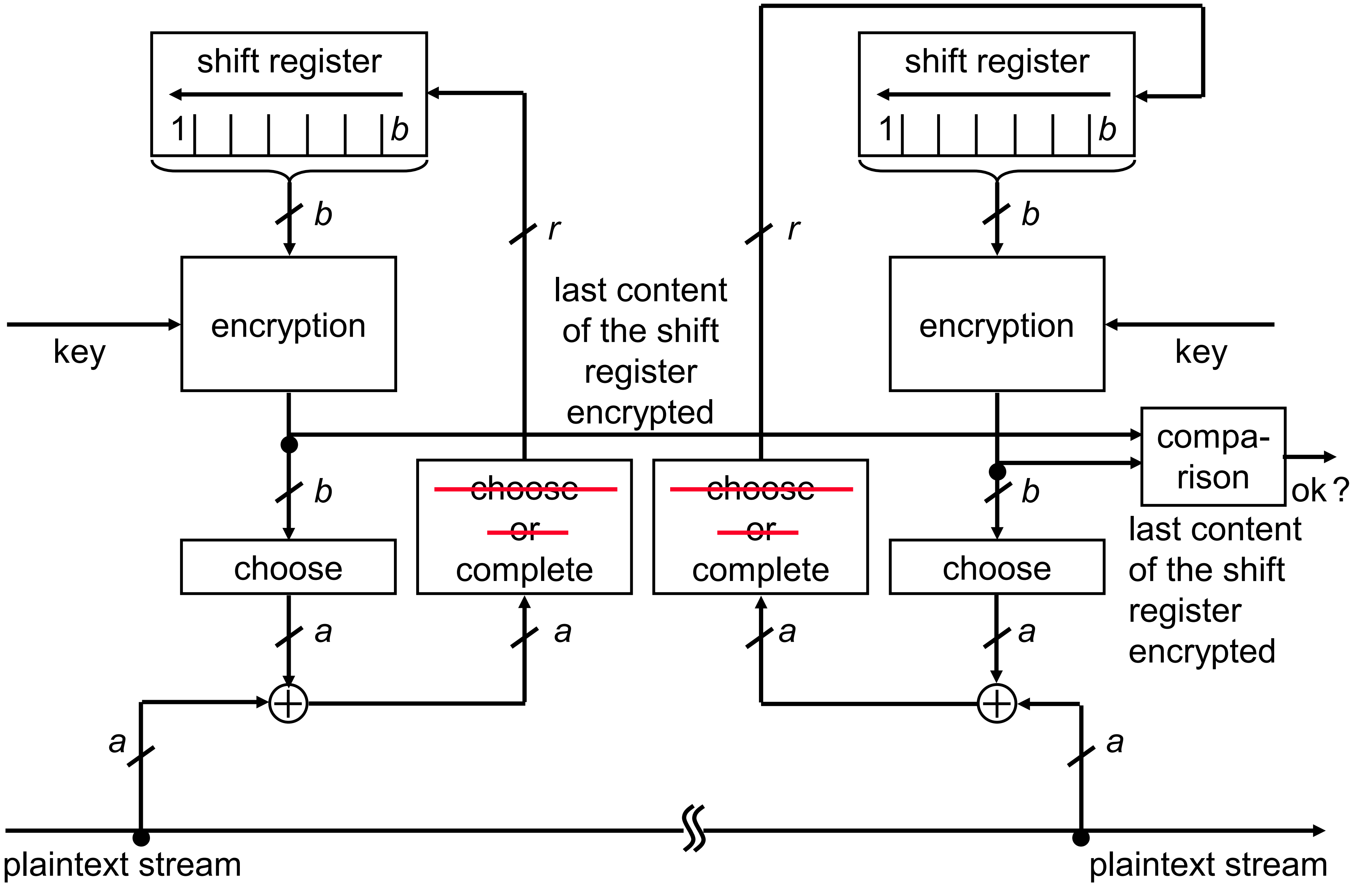
symmetric;
self synchronizing



CFB for Authentication



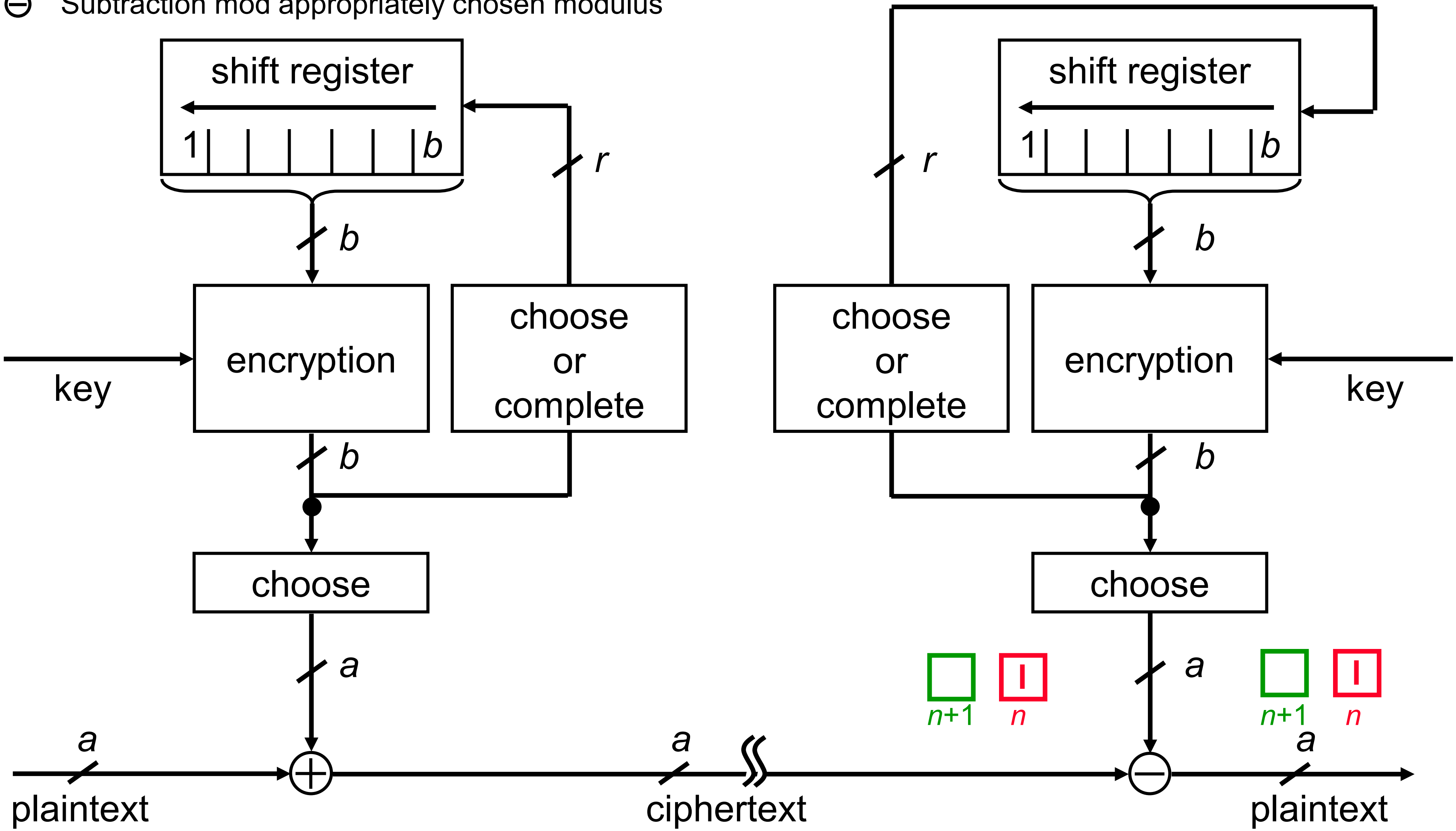
CFB for authentication



Output FeedBack (OFB)

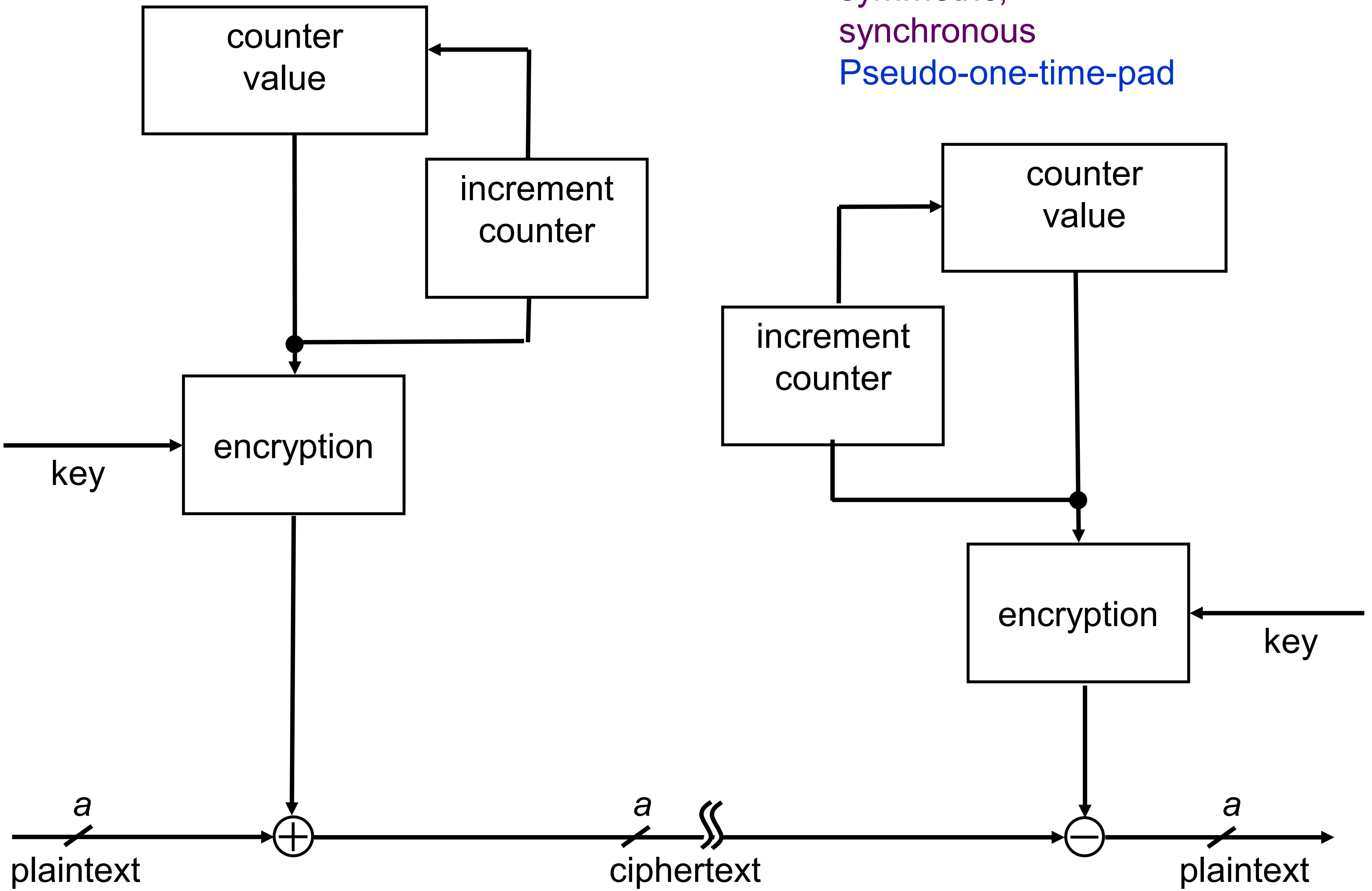
b Block length
a Length of the output unit, $a \leq b$
r Length of the feedback unit, $r \leq b$
$\oplus$ Addition mod appropriately chosen modulus
$\ominus$ Subtraction mod appropriately chosen modulus

symmetric;
synchronous
Pseudo-one-time-pad



Counter Mode (CTR)

symmetric;
synchronous
Pseudo-one-time-pad



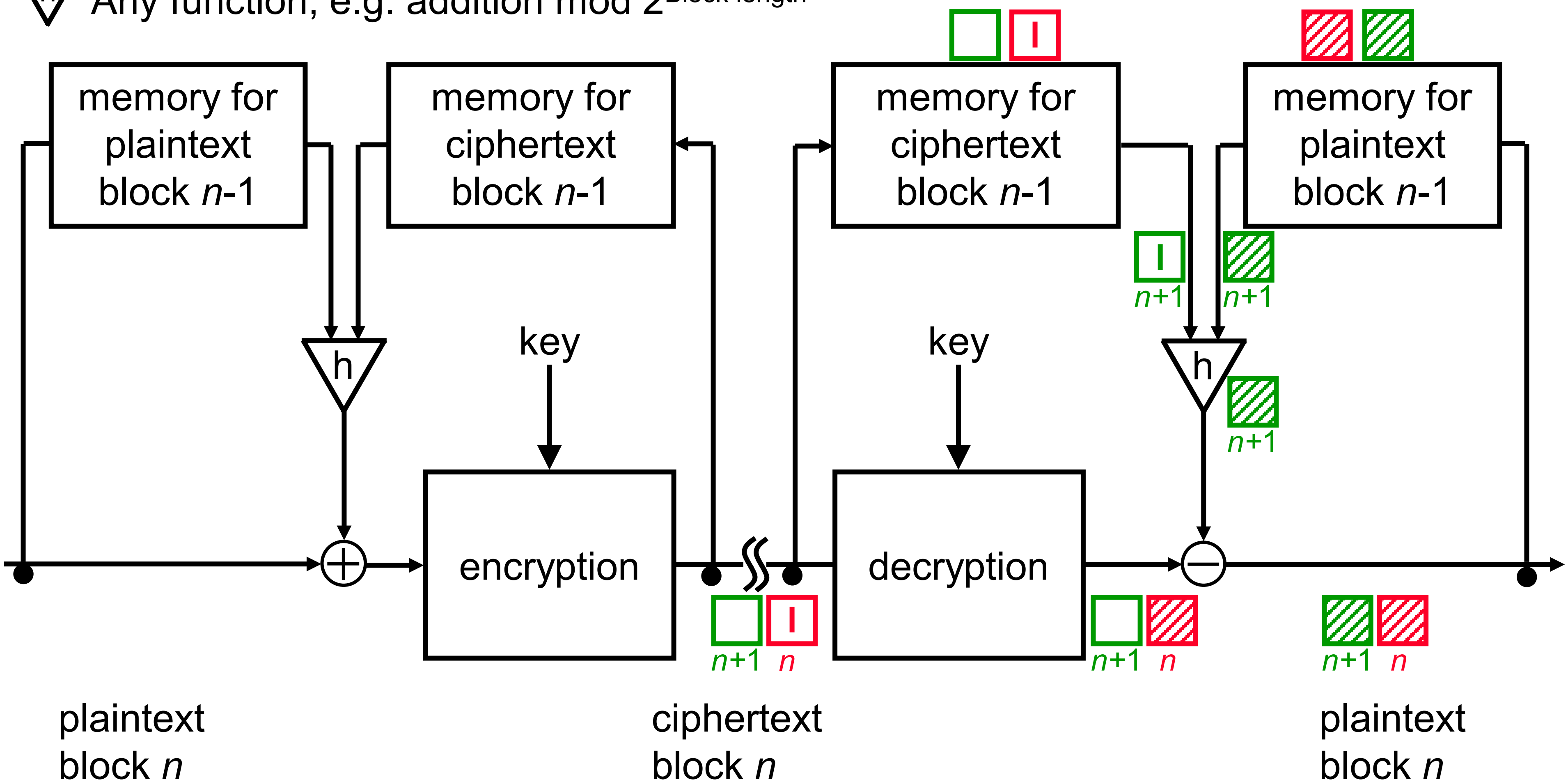
Plain Cipher Block Chaining (PCBC)

All lines transmit as many characters as a block comprises

⊕ Addition mod appropriately chosen modulus, e.g. 2

⊖ Subtraction mod appropriately chosen modulus, e.g. 2

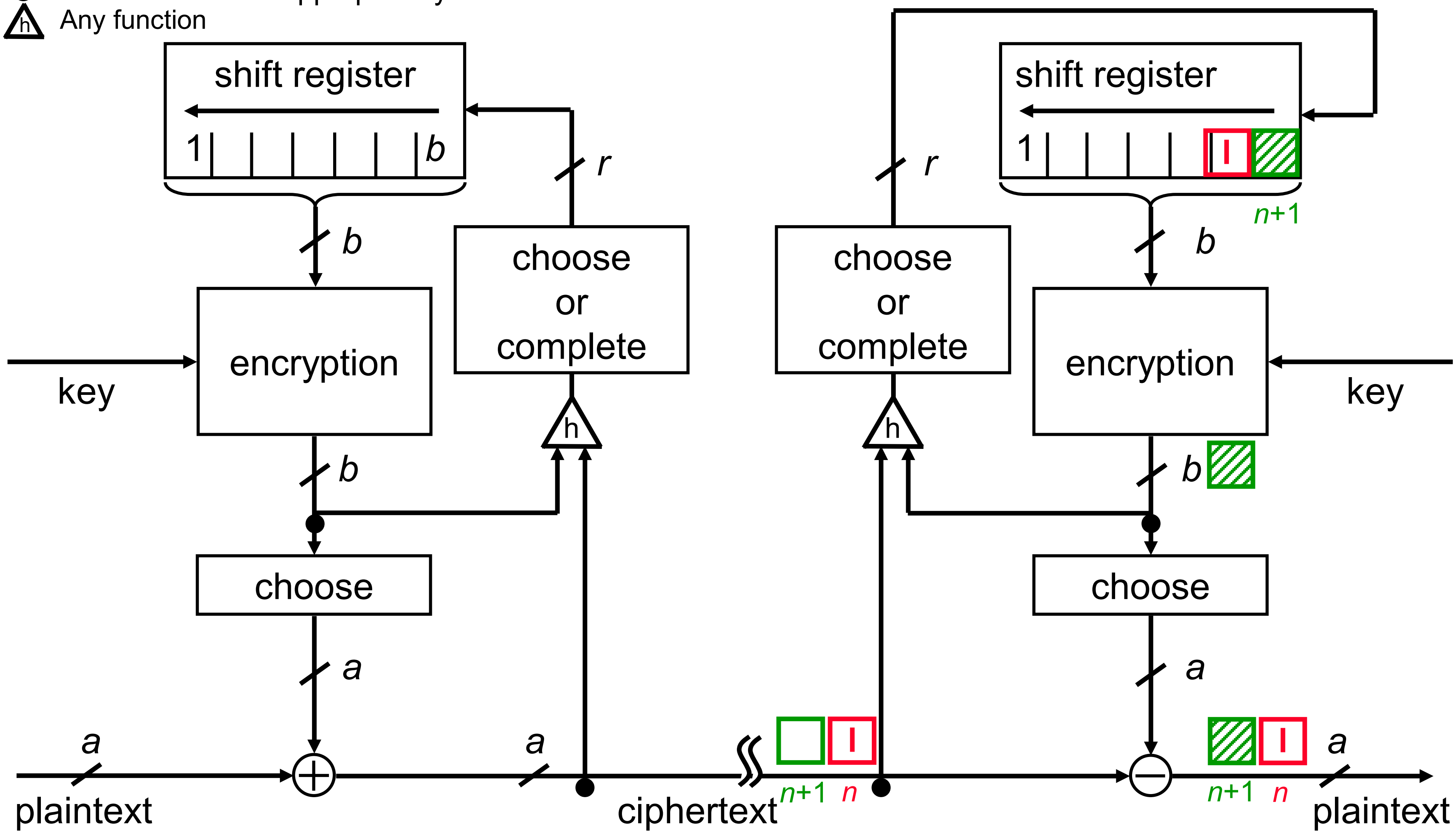
h Any function, e.g. addition mod $2^{\text{Block length}}$



Output Cipher FeedBack (OCFB)

b Block length
a Length of the output unit, $a \le b$
r Length of the feedback unit, $r \le b$
$\oplus$ Addition mod appropriately chosen modulus
$\ominus$ Subtraction mod appropriately chosen modulus
h Any function

symmetric;
synchronous



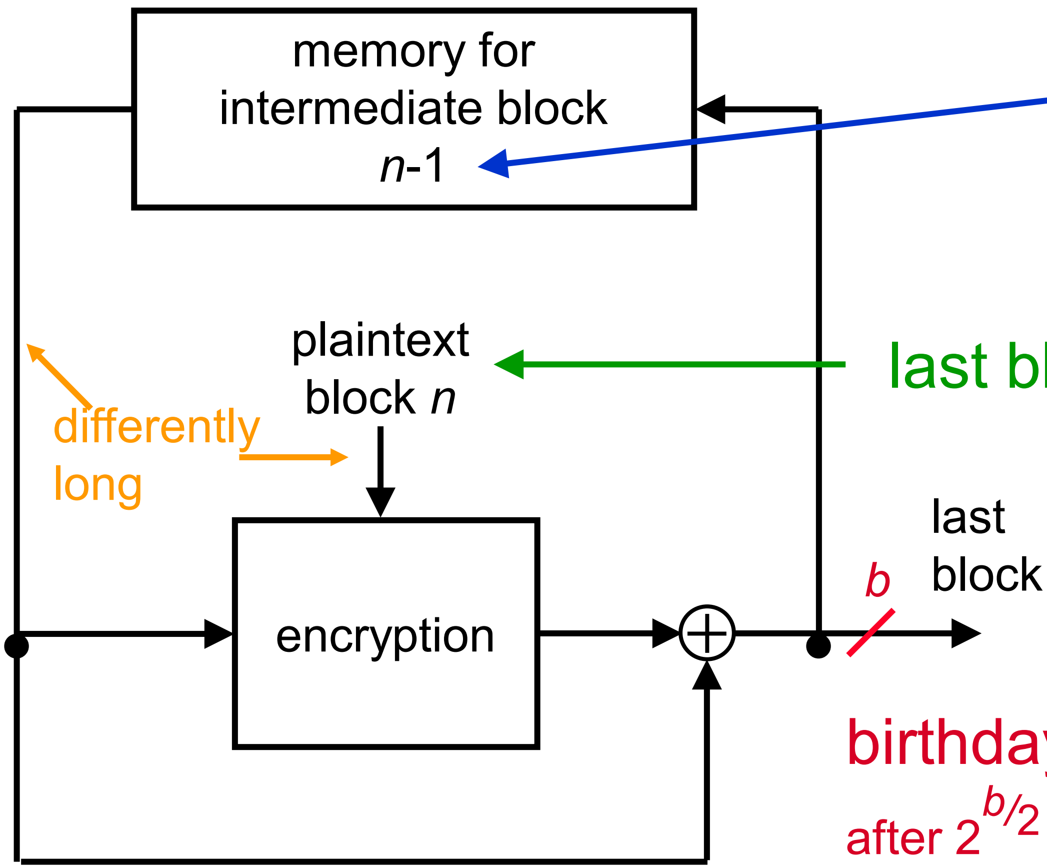
Properties of the operation modes

	ECB	CBC	PCBC	CFB	OFB	OCFB
Utilization of indeterministic block cipher	+ possible			- impossible		
Use of an asymmetric block cipher results in	+ asymmetric stream cipher			- symmetric stream cipher		
Length of the units of encryption	- determined by block length of the block cipher			+ user-defined		
Error extension	only within the block (assuming the borders of blocks are preserved)	2 blocks (assuming the borders of blocks are preserved)	potentially unlimited	$1 + \lceil b/r \rceil$ blocks, if error placed rightmost, else possibly one block less	none as long as no bits are lost or added	potentially unlimited
Qualified also for authentication?	yes, if redundancy within every block	yes, if deterministic block cipher	yes, even concealment in the same pass	yes, if deterministic block cipher	yes, if adequate redundancy	yes, even concealment in the same pass

Collision-resistant hash function using determ. block cipher

efficient ! — nearly any

cryptographically strong — no, but well analyzed



initial value is fixed!
(else trivial collisions: intermediate blocks and truncated plaintexts)

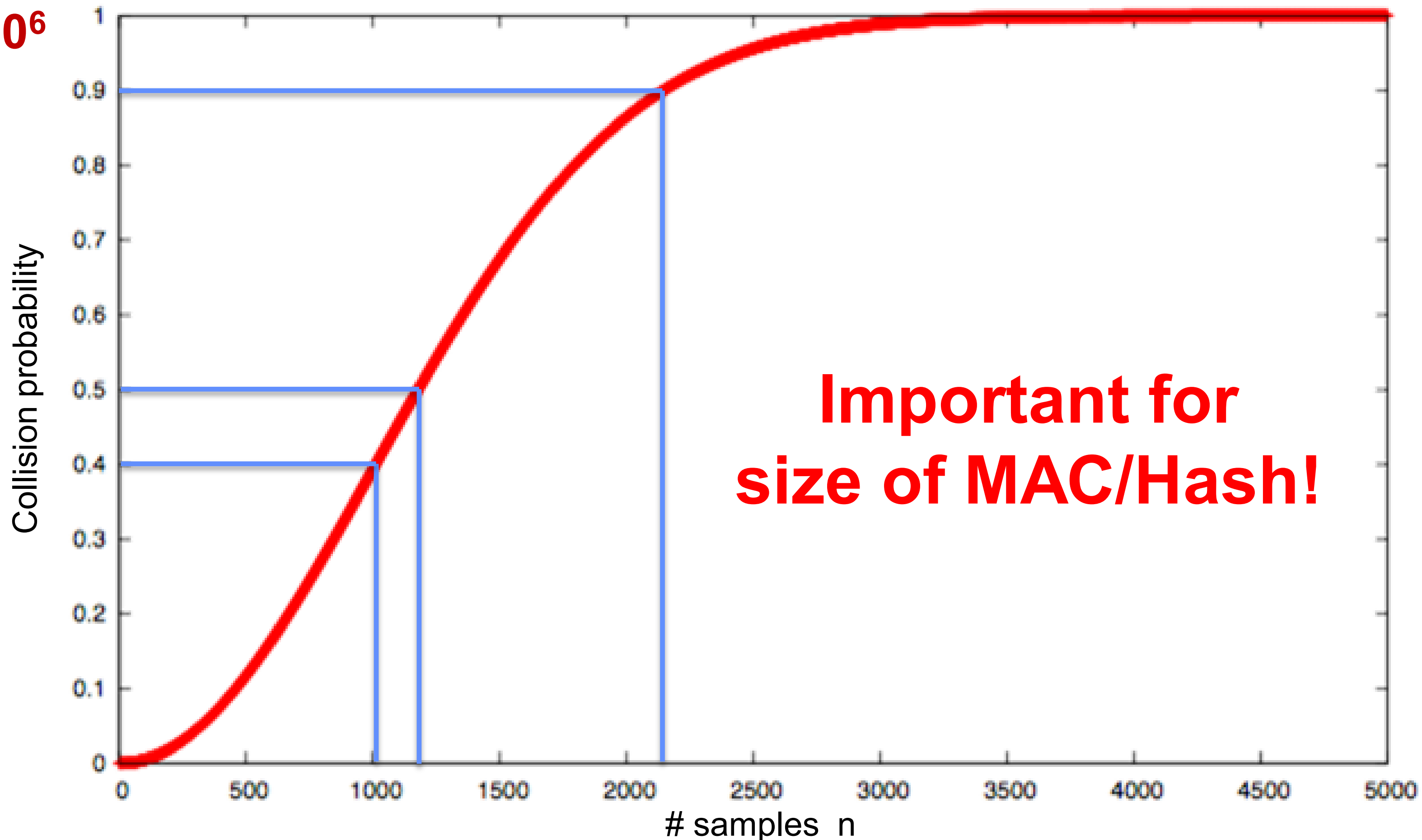
last block contains length in bit

birthday paradox
after $2^{b/2}$ tests collision

The Birthday Paradox

- Let $r_1, \ldots, r_n \in \{1,\ldots,B\}$ be random integers, chosen *independent and identically distributed* (iid).
- if **n**= 1.2 · **B**$^{1/2}$ then $\Pr[\,\exists i\neq j:\ r_i = r_j\,] \geq ½$

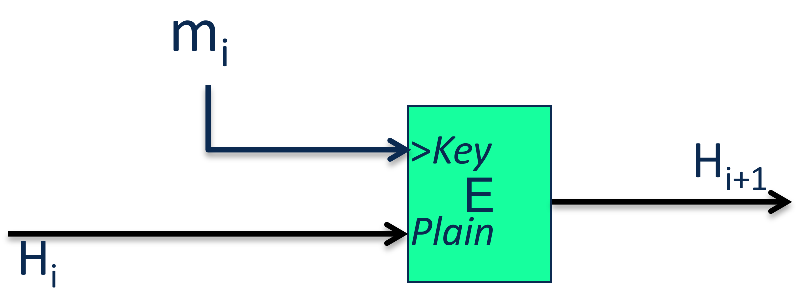
B=10⁶



An insecure attempt…

$E: K \times \{0,1\}^n \longrightarrow \{0,1\}^n$ a block cipher.

Construct cascade, for compression encrypt message blocks:



What‘s wrong with that?

$H_{i+1} = E(m_i, H_i)$

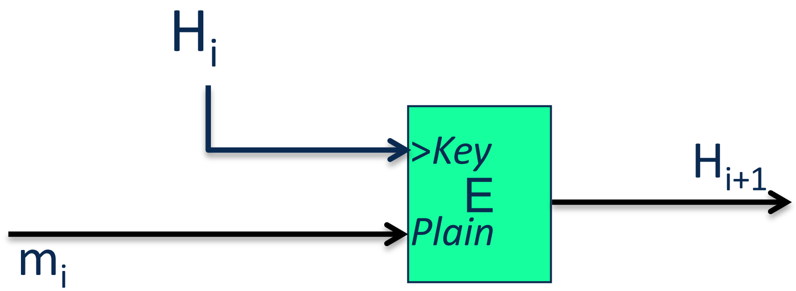
Can you find a collision on this compression function?

$H_{i+1} = E(m', D(m', H_{i+1}))$

An insecure attempt…

$E: K\times \{0,1\}^n \longrightarrow \{0,1\}^n$ a block cipher.

Construct cascade, for compression encrypt message blocks:



Message $m=m_1|m_2 \rightarrow H_0=IV \rightarrow H_1=E(IV,m_1) \rightarrow H_2=E(H_1,m_2)=h$

Manipulated Message: $m'=m'_1|m'_2= m'_1\ |D(E(IV, m'_1), H_2)$

$\rightarrow H_0=IV \rightarrow H_1=E(IV,m'_1) \rightarrow H_2=E(H_1, m'_2)=$

$E(H_1,D(E(IV, m'_1), H_2))=$

$E(H_1,D(H_1 \qquad , H_2))= H_2 =h$

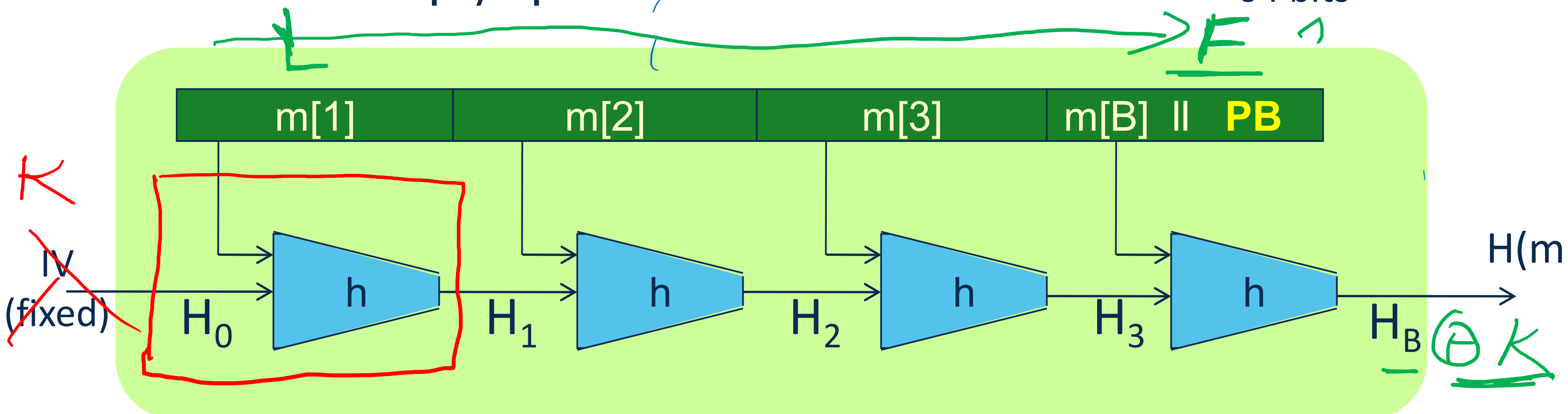
The Merkle-Damgård construction

From *short* message blocks to *arbitrarily long* messages…

Given: compression function $h: \{0,1\}^{2s} \longrightarrow \{0,1\}^{s}$ and

Input $m \in \{0,1\}^{*}$ of length L and PB: = | 1000…0 ll L | ($\underbrace{\text{L}}_{\text{64 bits}}$)

Construct H of $B = \lceil L/s \rceil$ iterations of h:



If h is a fixed length CRHF, then H is an ~~arbitrary length~~ CRHF

m, MAC(m)

Proof: either $\underbrace{m=m'}_{\text{no collision}}$, or $\underbrace{H_{B-i}(m[B-i])=H_{B-i}(m'[B-i])}_{\text{collision on } h}$

m', MAC(m') → valid

Diffie-Hellman key agreement (1)

practically important:	patent exhausted before that of RSA $\rightarrow$ used in PGP from Version 5 on
theoretically important:	steganography using public keys

based on difficulty to calculate **discrete logarithms**

Given a prime number $\boldsymbol{p}$ and $\boldsymbol{g}$ a generator of $Z_{\boldsymbol{p}}^*$

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

$\boldsymbol{x}$ is the **discrete logarithm** of $\boldsymbol{h}$ to basis $\boldsymbol{g}$ modulo $\boldsymbol{p}$:

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

discrete logarithm assumption

Discrete logarithm assumption

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

$\forall$ polynomials Q

$\exists L \ \forall \ l \geq L$: (asymptotically holds)

If p is a random prime of length l

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

x is chosen randomly in Z_p^*

and $g^x = h \mod p$

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

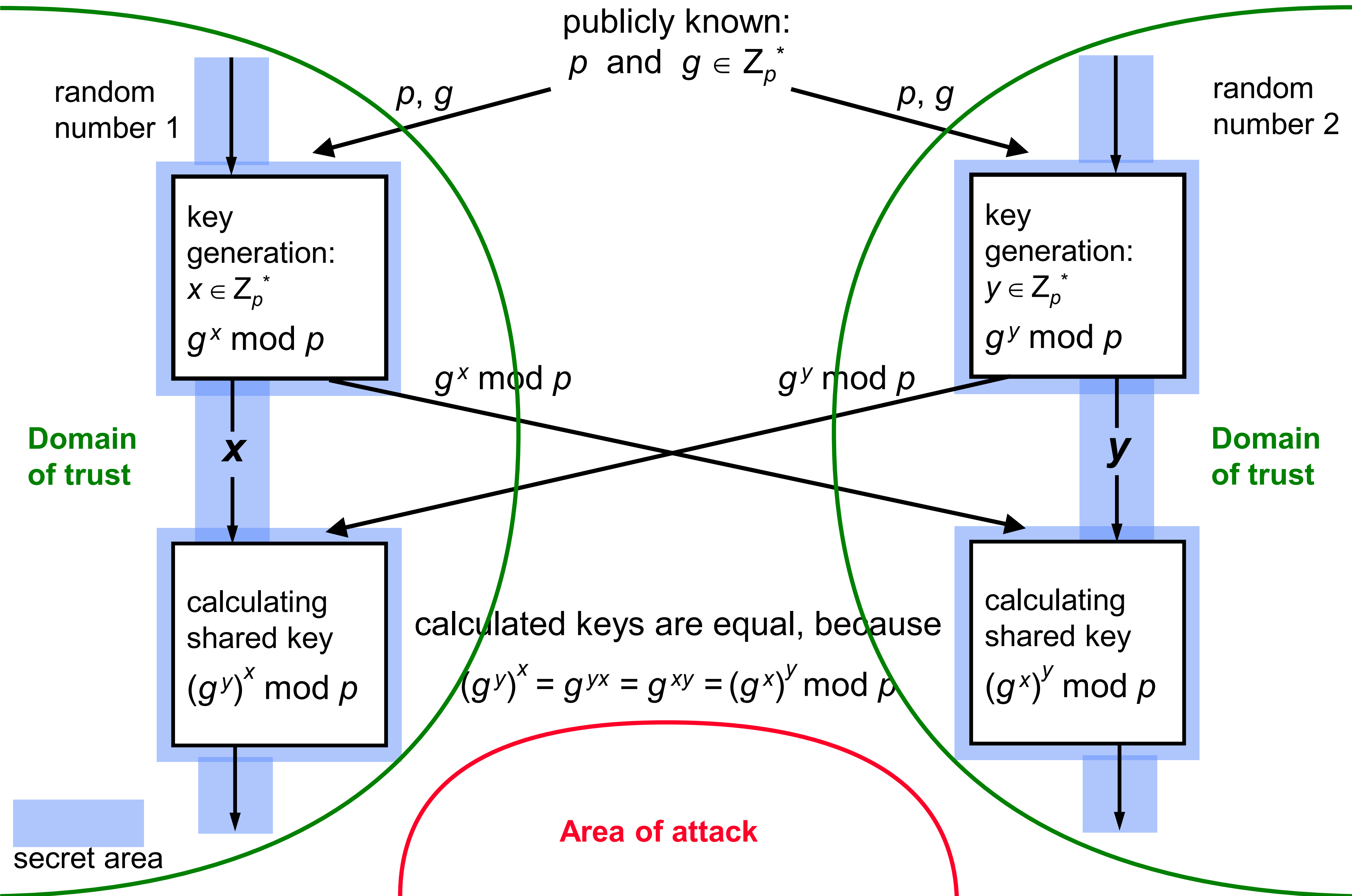
(probability that $\mathcal{DL}$ really calculates the discrete logarithm,

decreases faster than $\dfrac{1}{\text{any polynomial}}$)

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.

ElGamal: PKC from Diffie-Hellman

Fix a finite cyclic group G (e.g G = $(Z_p)^*$) of order *n*

Fix a generator *g* in G (i.e. G = $\{1, g, g^2, g^3, \dots, g^{n-1}\}$)

Alice

choose random **a** in {1,…,n}

$A = g^a$

Treat as a public key

Bob

choose random **b** in {1,…,n}

compute $g^{ab} = A^b$,

derive symmetric key *k* ,

ct = [$B = g^b$, encrypt message *m* with *k*]

To decrypt:
compute $g^{ab} = B^a$,
derive *k*, and decrypt

The ElGamal System (modern)

G: finite cyclic group of order n

(E_s, D_s) : symmetric auth. encryption defined over (K,M,C)

H: G x G $\longrightarrow$ K a hash function

Construct a pub-key encryption system (Gen, E, D):

Key generation Gen:

- choose random generator g in G and random a in Z_n
- output sk = a , pk = (g, h=g^a)

E(pk=(g,h), m) :

$b \longleftarrow Z_n$, $u \longleftarrow g^b$, $v \longleftarrow h^b = g^{a \cdot b}$

$k \longleftarrow H(u,v)$, $c \longleftarrow E_s(k, m)$

output (u, c)

D(sk=a, (u,c)) :

$v \longleftarrow u^a = g^{b \cdot a}$

$k \longleftarrow H(u,v)$, $m \longleftarrow D_s(k, c)$

output m

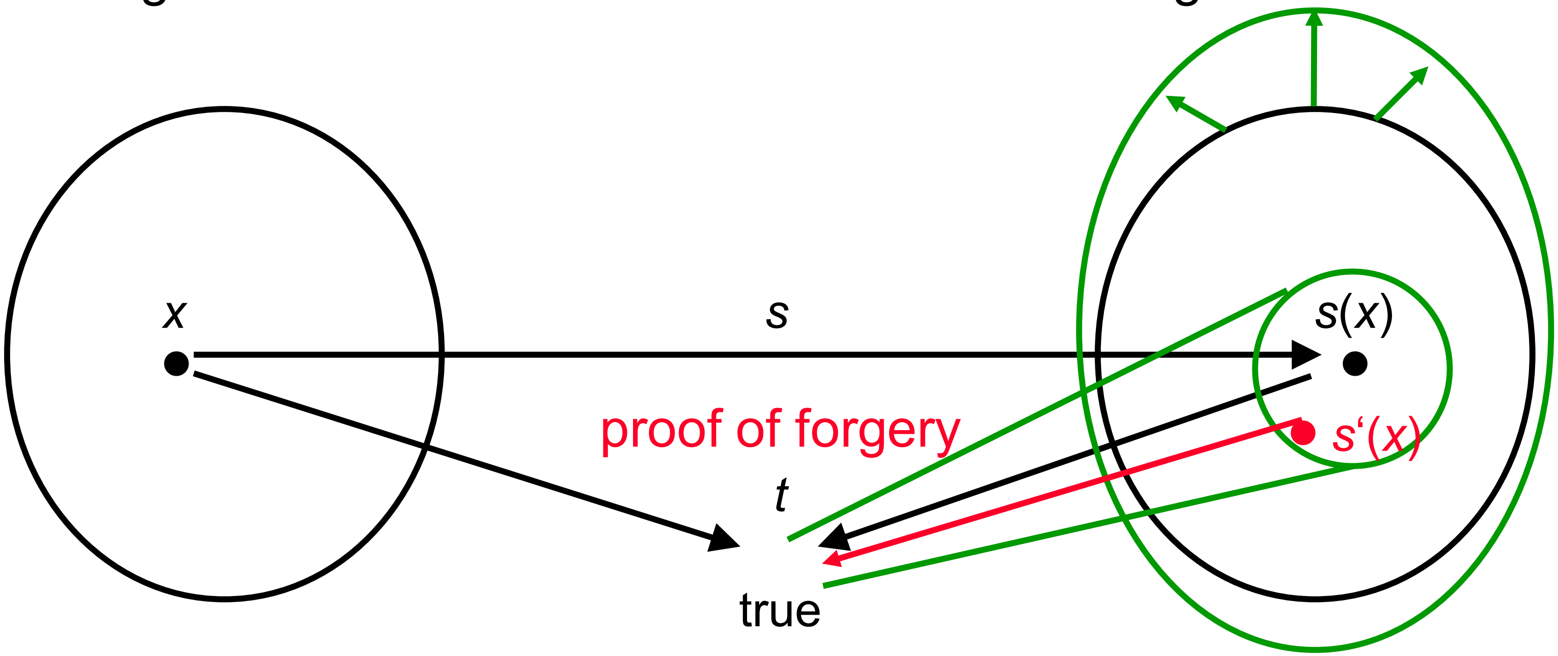
Digital signature system

Security is asymmetric, too

usually: unconditionally secure for recipient
only cryptographically secure for signer

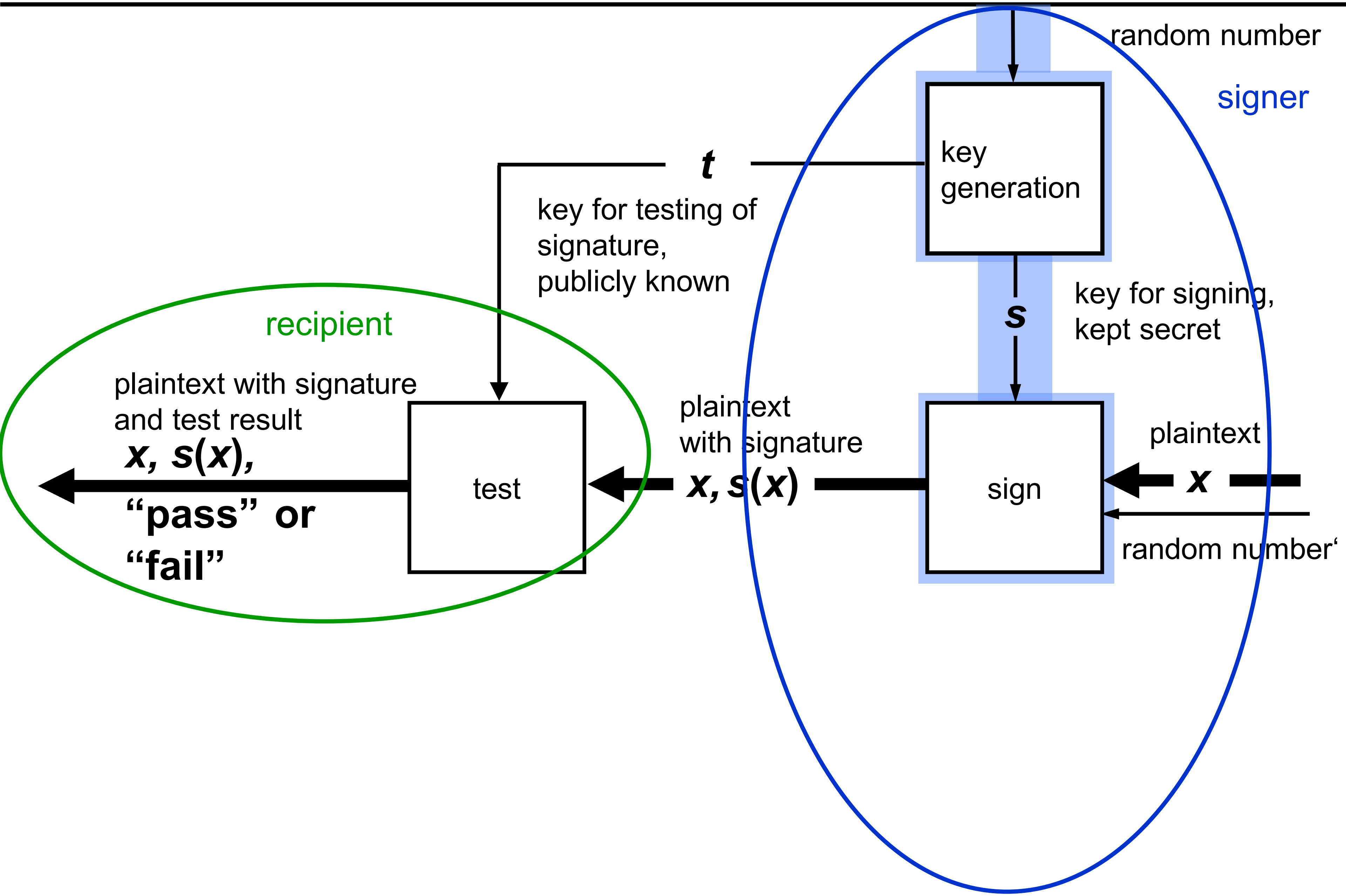
new: signer is absolutely secure against breaking his signatures
provable only cryptographically secure for recipient

message domain signature domain

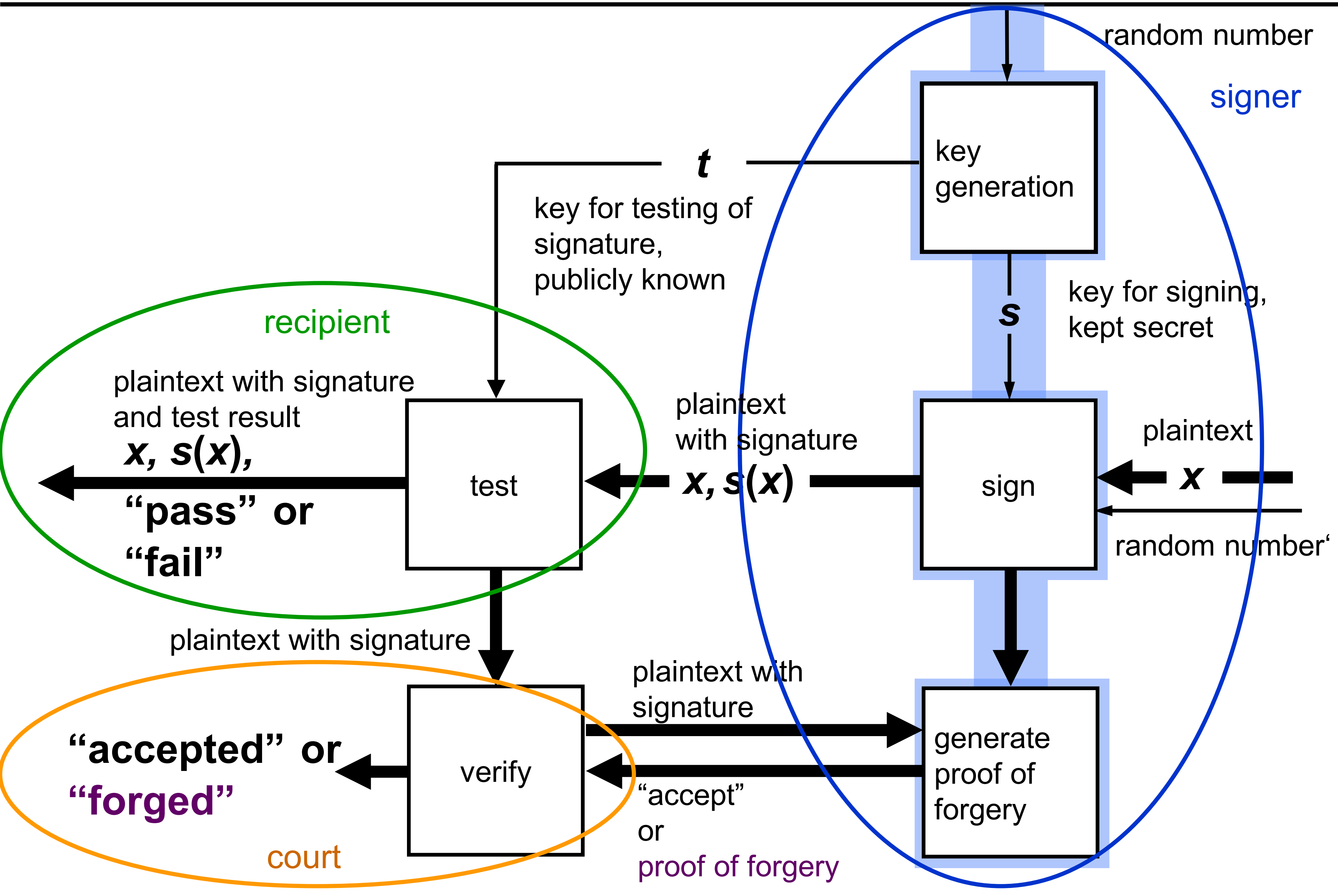


distribution of risks if signature is forged:
1. recipient
2. insurance or system operator
3. signer

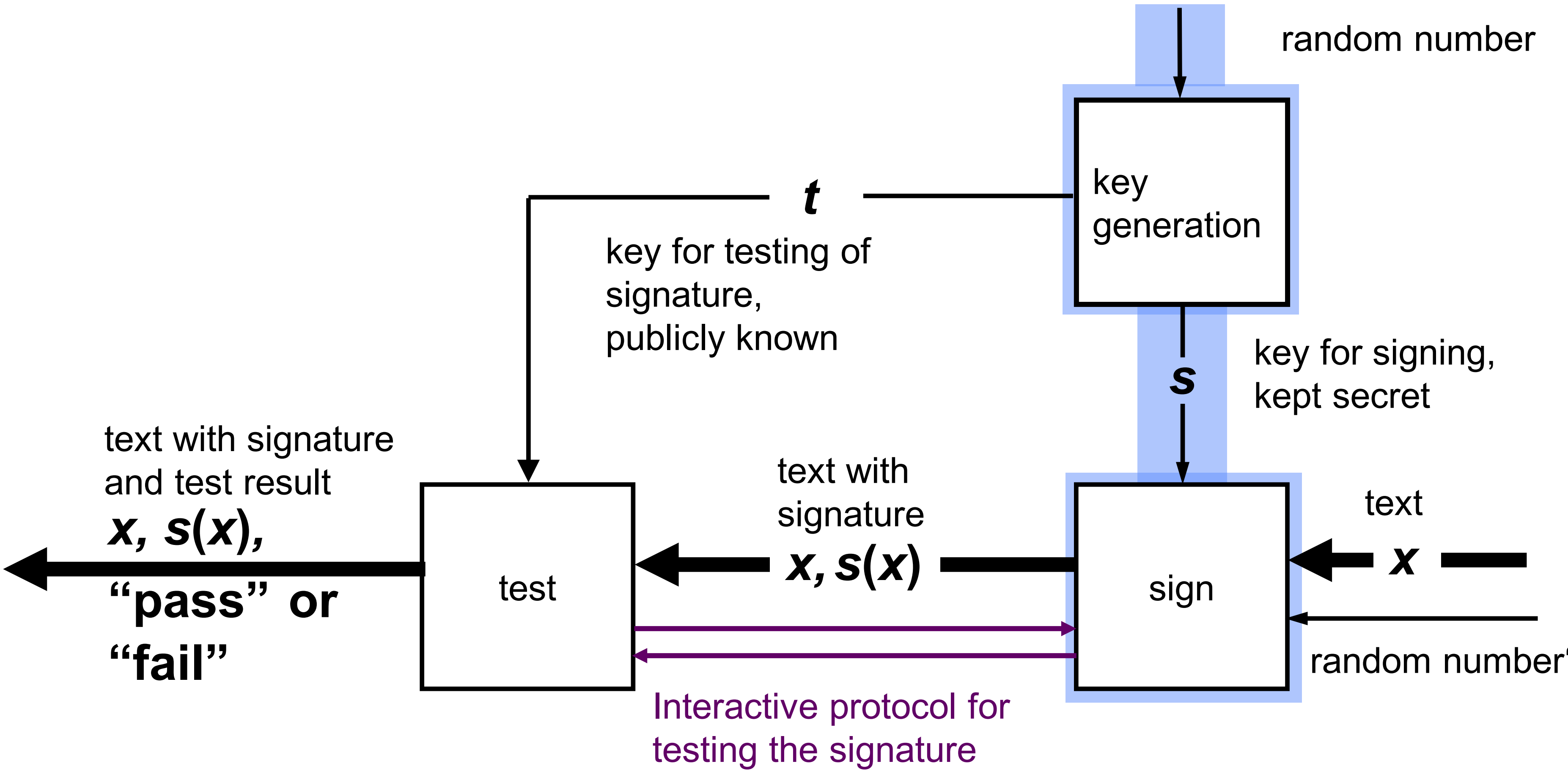
Fail-stop signature system



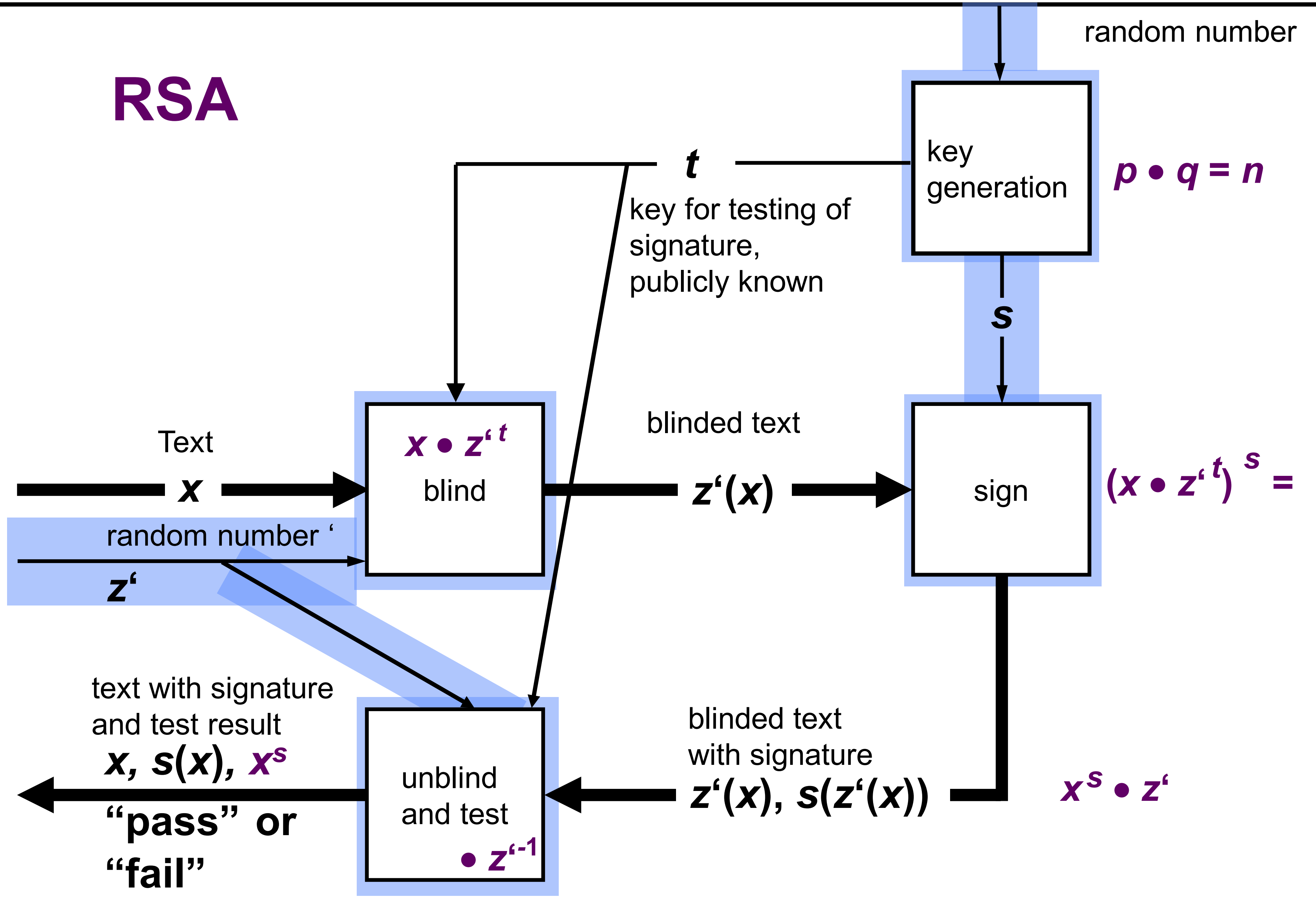
Fail-stop signature system



Undeniable signatures



Signature system for blindly providing of signatures



Threshold scheme for secret sharing (1)

Threshold scheme:

Secret S

n parts

k parts: efficient reconstruction of S

k-1 parts: no information about S

Implementation: polynomial interpolation (Shamir, 1979)

Decomposition of the secret:

Let secret S be an element of Z_p, p being a prime number.

Polynomial $q(x)$ of degree k-1:

Choose $a_1, a_2, \ldots, a_{k-1}$ randomly in Z_p

$\mathbf{q(x) := S + a_1x + a_2x^2 + \ldots + a_{k-1}x^{k-1}}$

n parts $(i, q(i))$ with $1 \le i \le n$, where $n < p$.

Threshold scheme (2)

Reconstruction of the secret:

Lagrange polynomial interpolation

k parts $(x_j, q(x_j))$ $(j = 1 \dots k)$:

$$q(x) = \sum_{j=1}^{k} q(x_j) \prod_{m=1,\, m \neq j}^{k} \frac{(x - x_m)}{(x_j - x_m)} \mod p$$

The secret S is $q(0)$.

Sketch of proof:

1. k-1 parts $(j, q(j))$ deliver no information about S, because for each value of S there is still exactly one polynomial of degree k-1.
2. correct degree k-1; delivers for any argument x_j the value $q(x_j)$ (because product delivers on insertion of x_j for x the value 1 and on insertion of all other x_i for x the value 0).

Threshold scheme (3)

Polynomial interpolation is Homomorphism w.r.t. addition

Addition of the parts $\Rightarrow$ Addition of the secrets

Share refreshing

1.) Choose random polynomial q' for $S' = 0$
2.) Distribute the n parts $(i, q'(i))$
3.) Everyone adds his "new" part to his "old" part
$\rightarrow$ "new" random polynomial $q+q'$ with "old" secret S

- Repeat this, so that anyone chooses the random polynomial once
- Use *verifiable secret sharing*, so that anyone can test that polynomials are generated correctly.