Security in Computer Networks

Multilateral Security in Distributed and by Distributed Systems

Transparencies for the Lecture:

Security and Cryptography I
(and the beginning of Security and Cryptography II)

Andreas Pfitzmann
Technische Universität Dresden, Faculty of Computer Science, D-01062 Dresden
Nöthnitzer Str. 46, Room 3071
Phone: +49 351 463-38277, e-mail: pfitza@inf.tu-dresden.de, http://dud.inf.tu-dresden.de/

Field of Specialization: Security and Privacy

Lectures	*Staff*	*SWS*
Security and Cryptography I, II		
Introduction to Data Security	Pfitzmann	1/1
Cryptography	Pfitzmann	2/2
Data Security by Distributed Systems	Pfitzmann	1/1
Data Security and Data Protection		
– National and International	Lazarek	2
Cryptography and -analysis	Franz	2
Channel Coding	Schönfeld	2/2
Steganography and Multimedia Forensics	Franz	2/1
Data Security and Cryptography	Clauß	/4
Privacy Enhancing Technologies …	Clauß, Köpsell	/2
Computers and Society	Pfitzmann	2
Seminar: Privacy and Security	Pfitzmann et.al.	2

Areas of Teaching and Research

- Multilateral security, in particular security by distributed systems
- Privacy Enhancing Technologies (PETs)
- Cryptography
- Steganography
- Multimedia-Forensics
- Information- and coding theory

- Anonymous access to the web (project: AN.ON, JAP)
- Identity management (projects: PRIME, PrimeLife, FIDIS)
- SSONET and succeeding activities
- Steganography (project: CRYSTAL)

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
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.

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: **cryptography** and **steganography**
- 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**

Table of Contents (1)

Table of Contents (2)

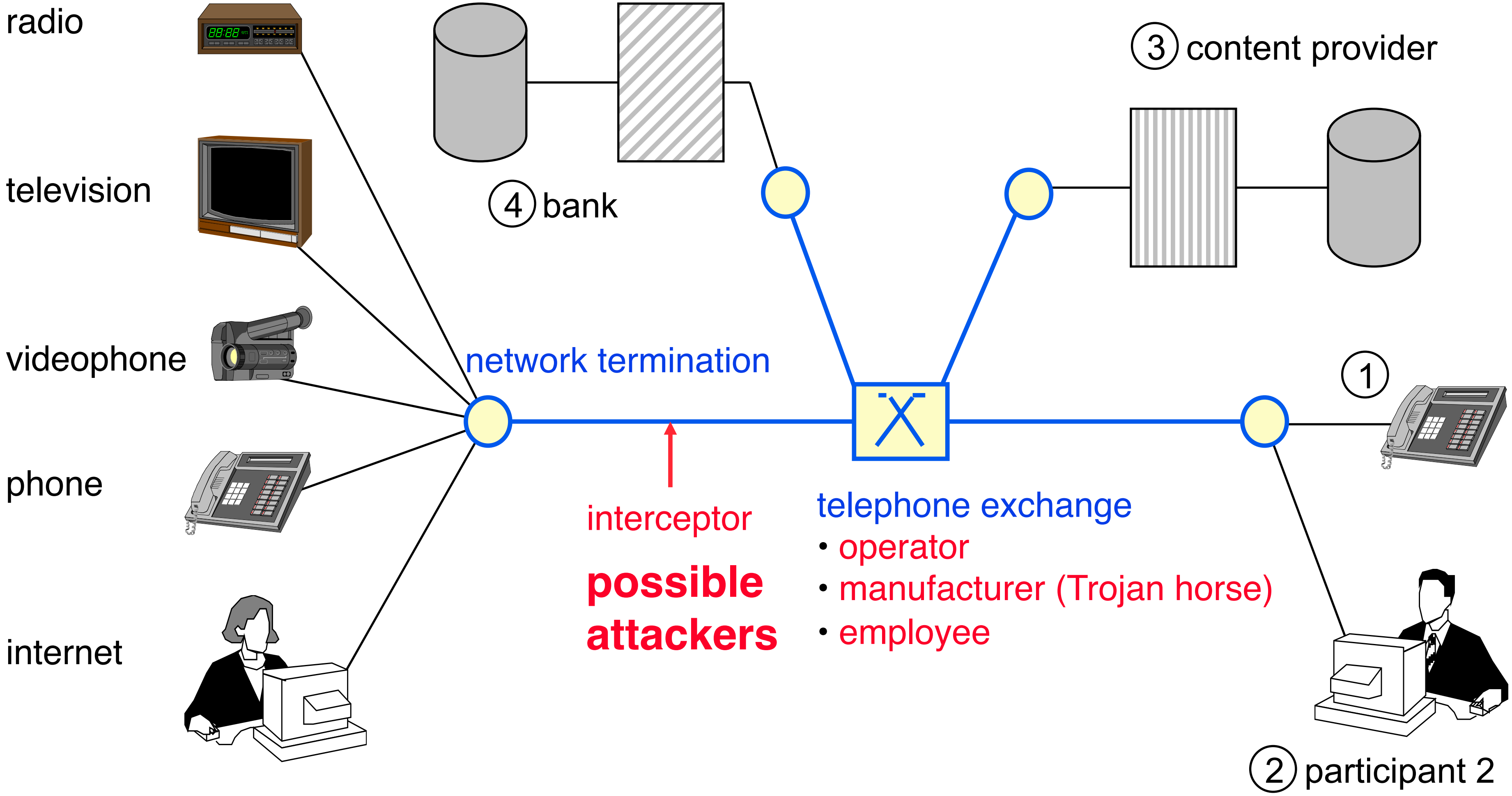
3 Cryptographic basics

4 Communication networks providing data protection guarantees

5 Digital payment systems and credentials as generalization

6 Summary and outlook

Part of a Computer Network



radio

television

videophone

phone

internet

④ bank

③ content provider

network termination

①

interceptor

possible attackers

telephone exchange

- operator
- manufacturer (Trojan horse)
- employee

② participant 2

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

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

Important Terms

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

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

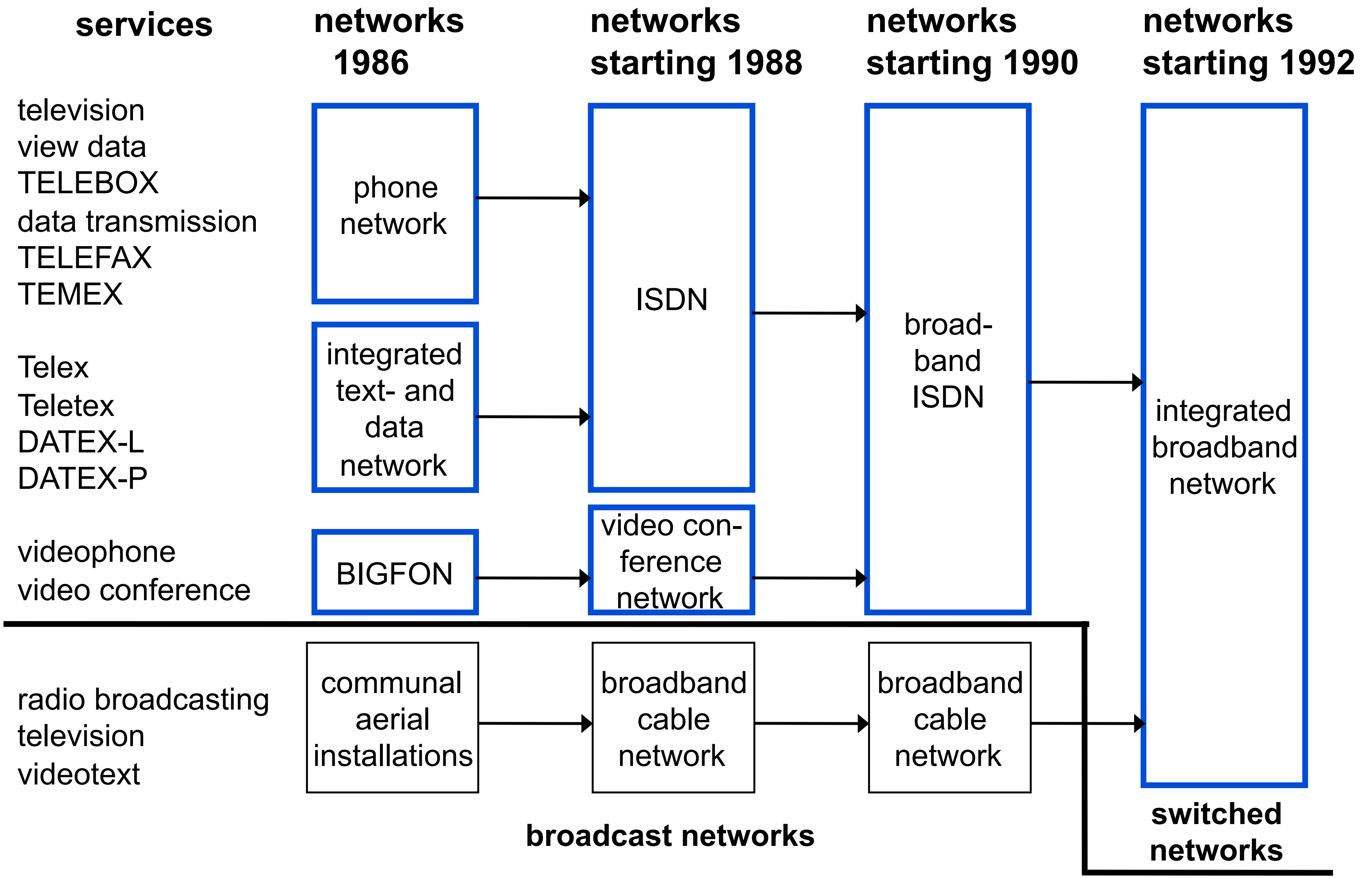
distributed system
spatial
control and implementation structure

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

service integrated system

digital system

Development of the fixed communication networks of the German Post



services

networks 1986

networks starting 1988

networks starting 1990

networks starting 1992

television
view data
TELEBOX
data transmission
TELEFAX
TEMEX

Telex
Teletex
DATEX-L
DATEX-P

videophone
video conference

radio broadcasting
television
videotext

phone network

integrated text- and data network

BIGFON

ISDN

video conference network

broadband ISDN

integrated broadband network

communal aerial installations

broadband cable network

broadband cable network

broadcast networks

switched networks

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

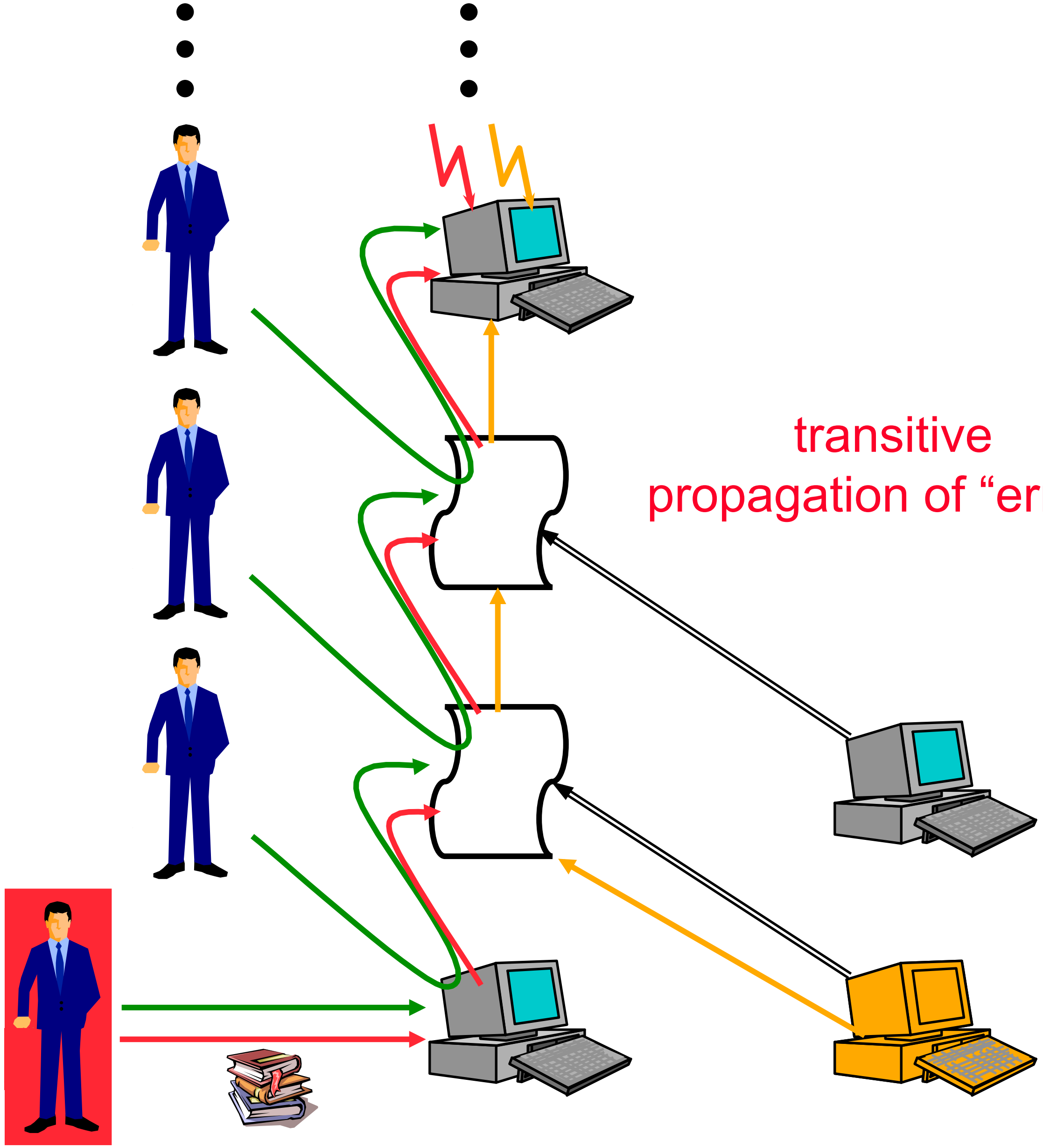
Information and resources are accessible where and
when the authorized user needs them.

- subsume: data, programs, hardware structure

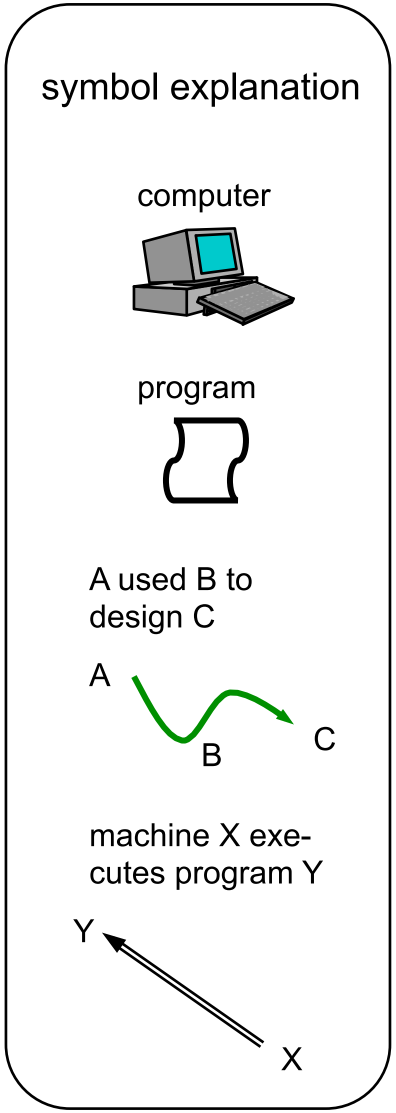
- it has to be clear, who is authorized to do what in which situation

- it can only refer to the inside of a system

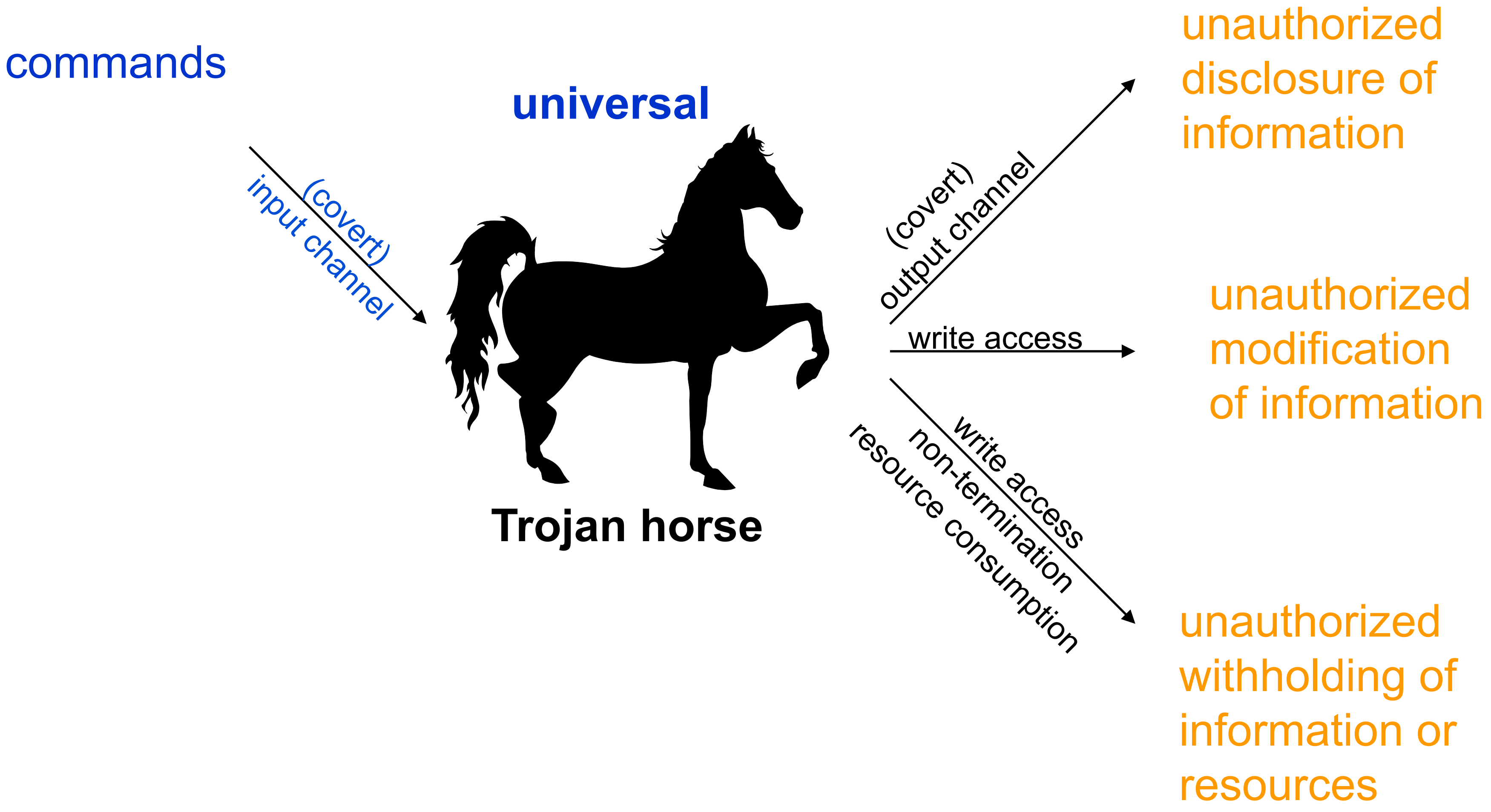
Transitive propagation of errors and attacks



symbol explanation



universal Trojan horse



commands

(covert)
input channel

universal

Trojan horse

(covert)
output channel

write access

write access
non-termination
resource consumption

unauthorized disclosure of information

unauthorized modification of information

unauthorized withholding of information or resources

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)
- 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

unobservability, anonymity, unlinkability:

avoid the ability to gather “unnecessary data”

Considered maximal strength of the attacker

attacker model

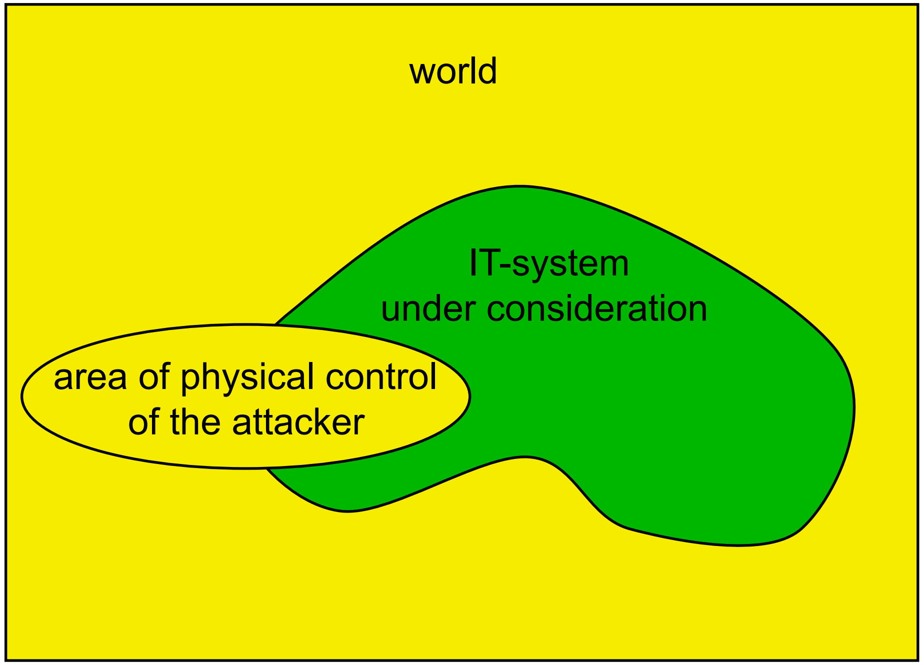
It‘s not possible to protect against an omnipotent attacker.

- 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

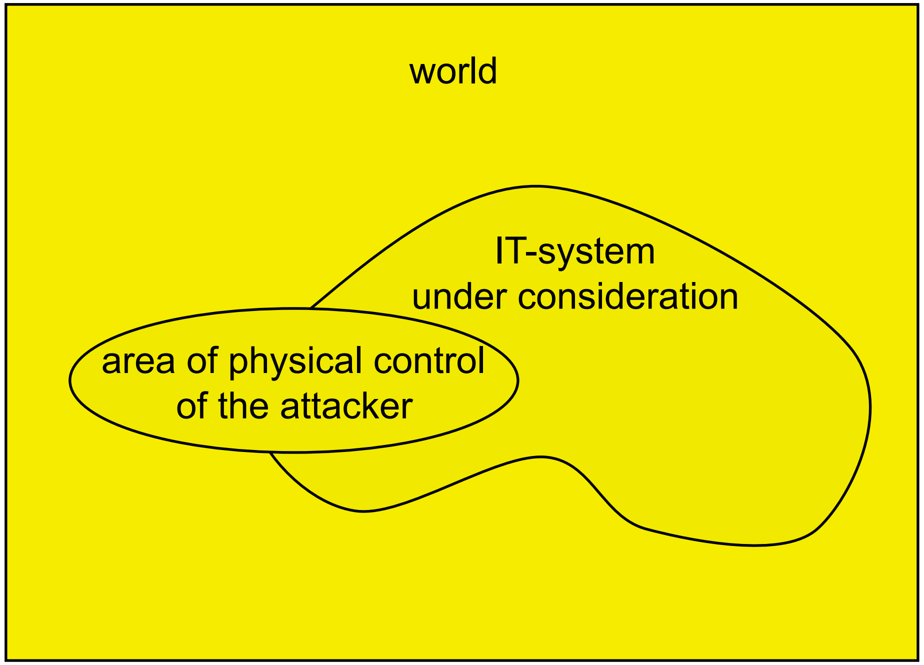
money

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

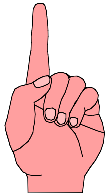
	• detect forgery	**authentication system(s)**
• **time**	• recipient can prove transmission	**sign messages**
	• sender can prove transmission	**receipt**
	• ensure payment for service	**during service by digital payment systems**

availability

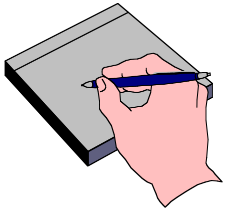
• enable communication	**diverse networks;** **fair sharing of resources**

Multilateral security

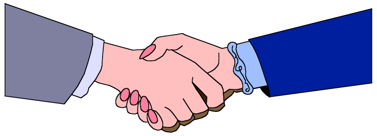
- Each party has its particular protection goals.



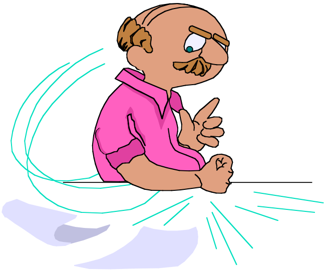
- Each party can formulate its protection goals.



- Security conflicts are recognized and compromises negotiated.



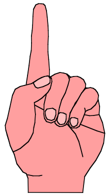
- Each party can enforce its protection goals within the agreed compromise.



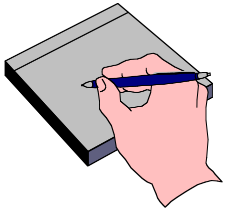
Security with minimal assumptions about others

Multilateral security (2nd version)

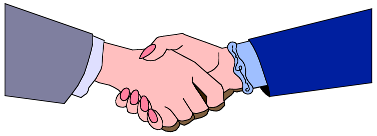
- Each party has its particular goals.



- Each party can formulate its protection goals.



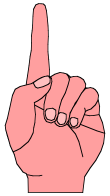
- Security conflicts are recognized and compromises negotiated.



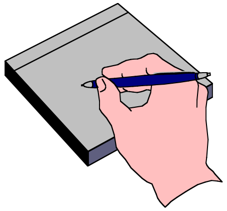
- Each party can enforce its protection goals within the agreed compromise.

Security with minimal assumptions about others

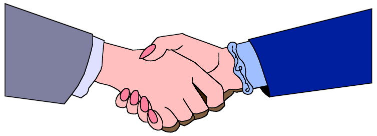
Multilateral security (3rd version)



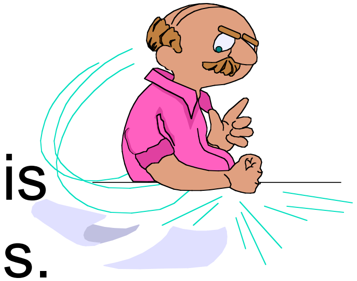
- Each party has its particular goals.



- 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

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.

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

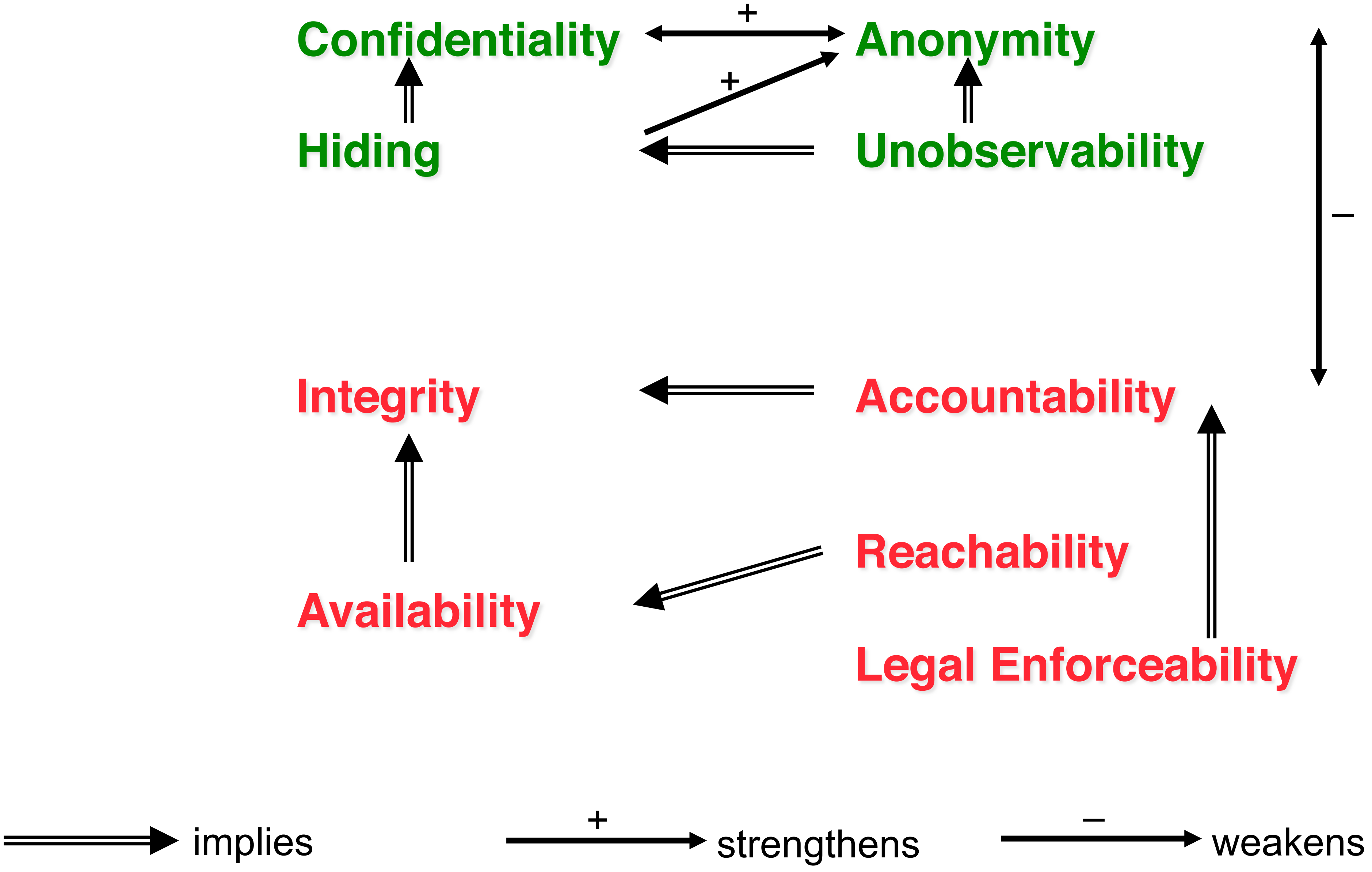
Accountability ensures that sender and recipients of information cannot successfully deny having sent or received the information. This means that communication takes place in a provable way.

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

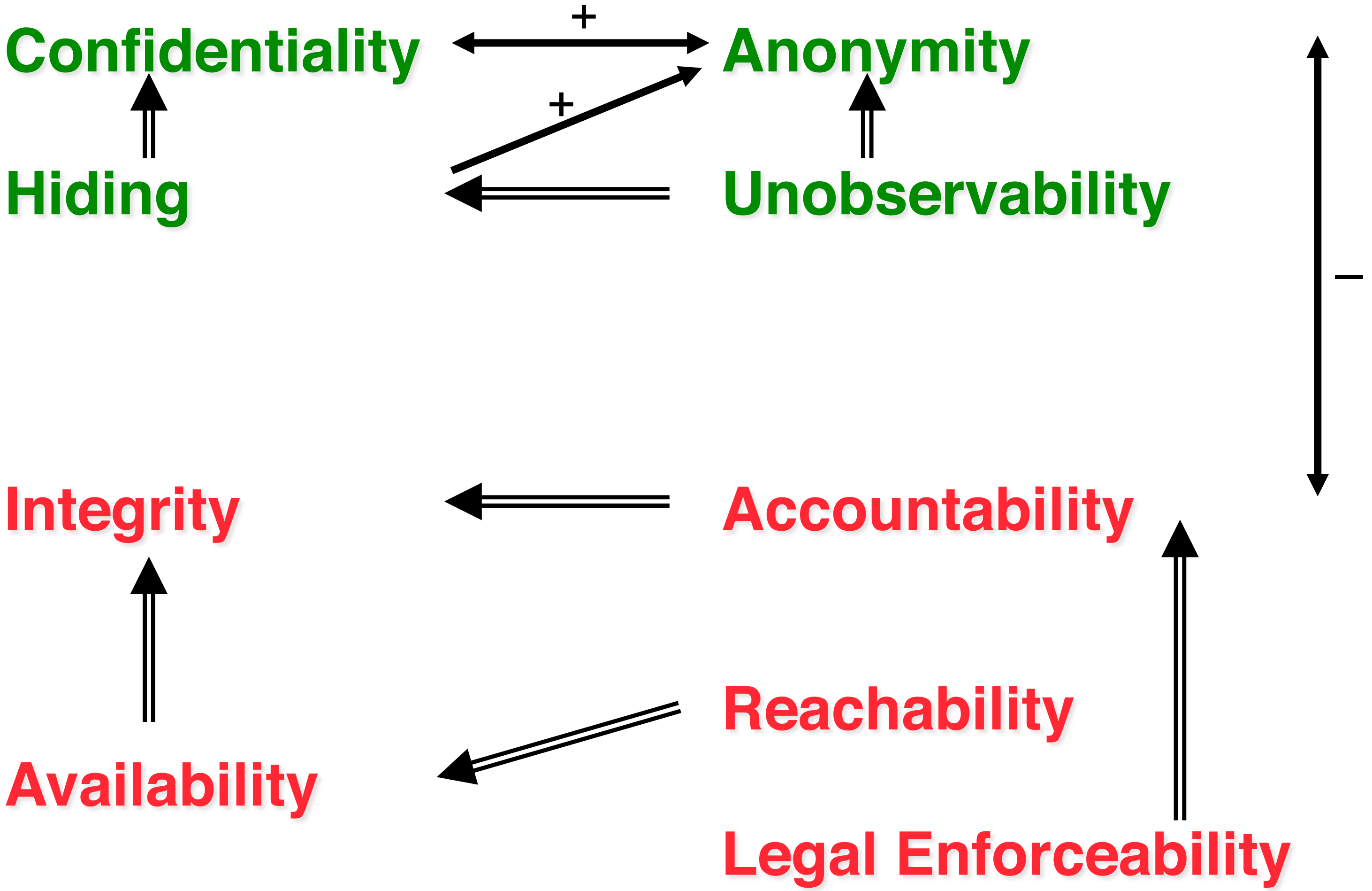
Reachability ensures that a peer entity (user, machine, etc.) either can or cannot be contacted depending on user interests.

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

Correlations between protection goals



Correlations between protection goals



Confidentiality

Anonymity

Hiding

Unobservability

Integrity

Accountability

Availability

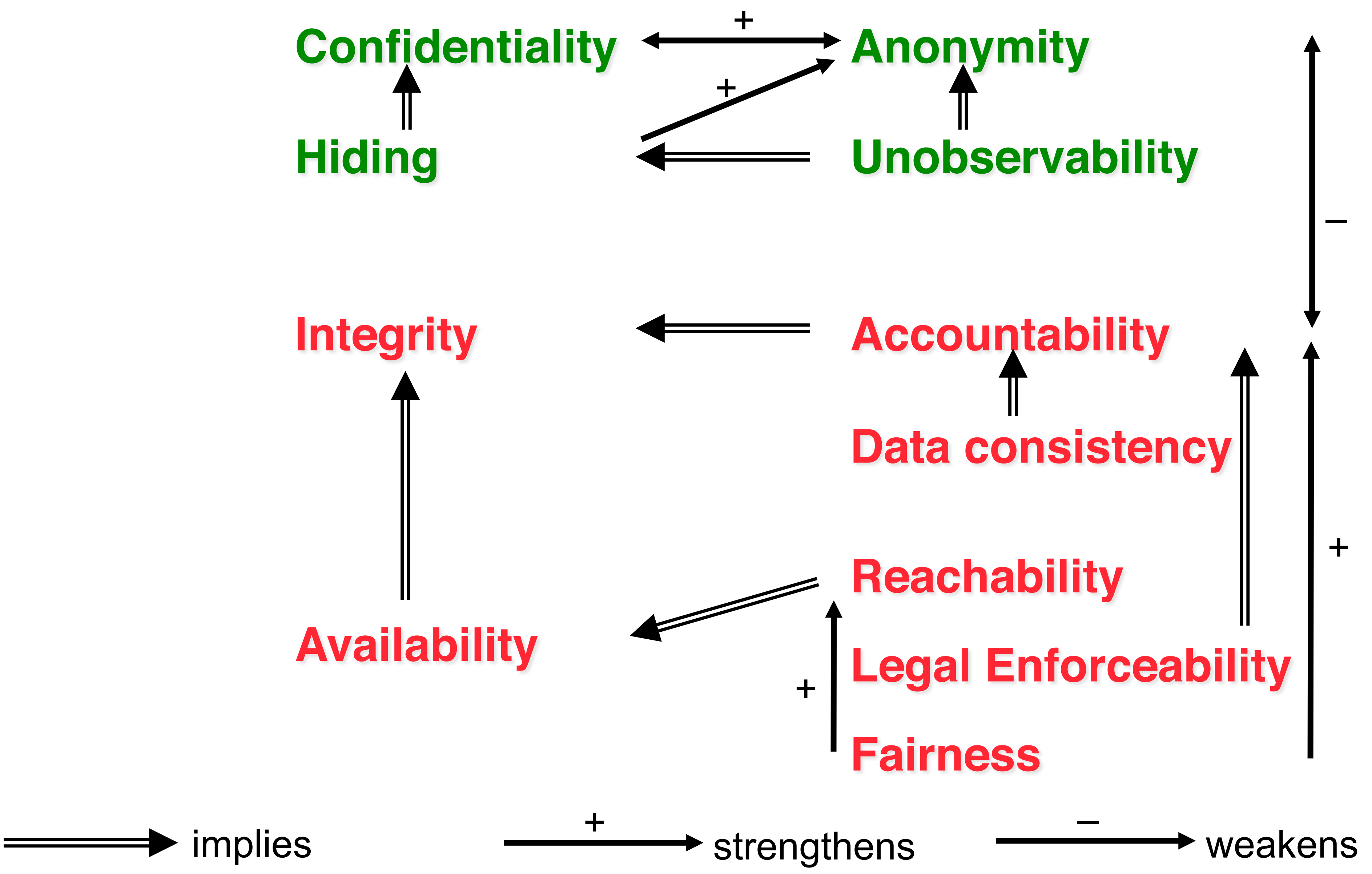
Reachability

Legal Enforceability

Transitive closure to be added

⇒ implies $\xrightarrow{+}$ strengthens $\xrightarrow{-}$ weakens

Correlations between protection goals, two added



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**

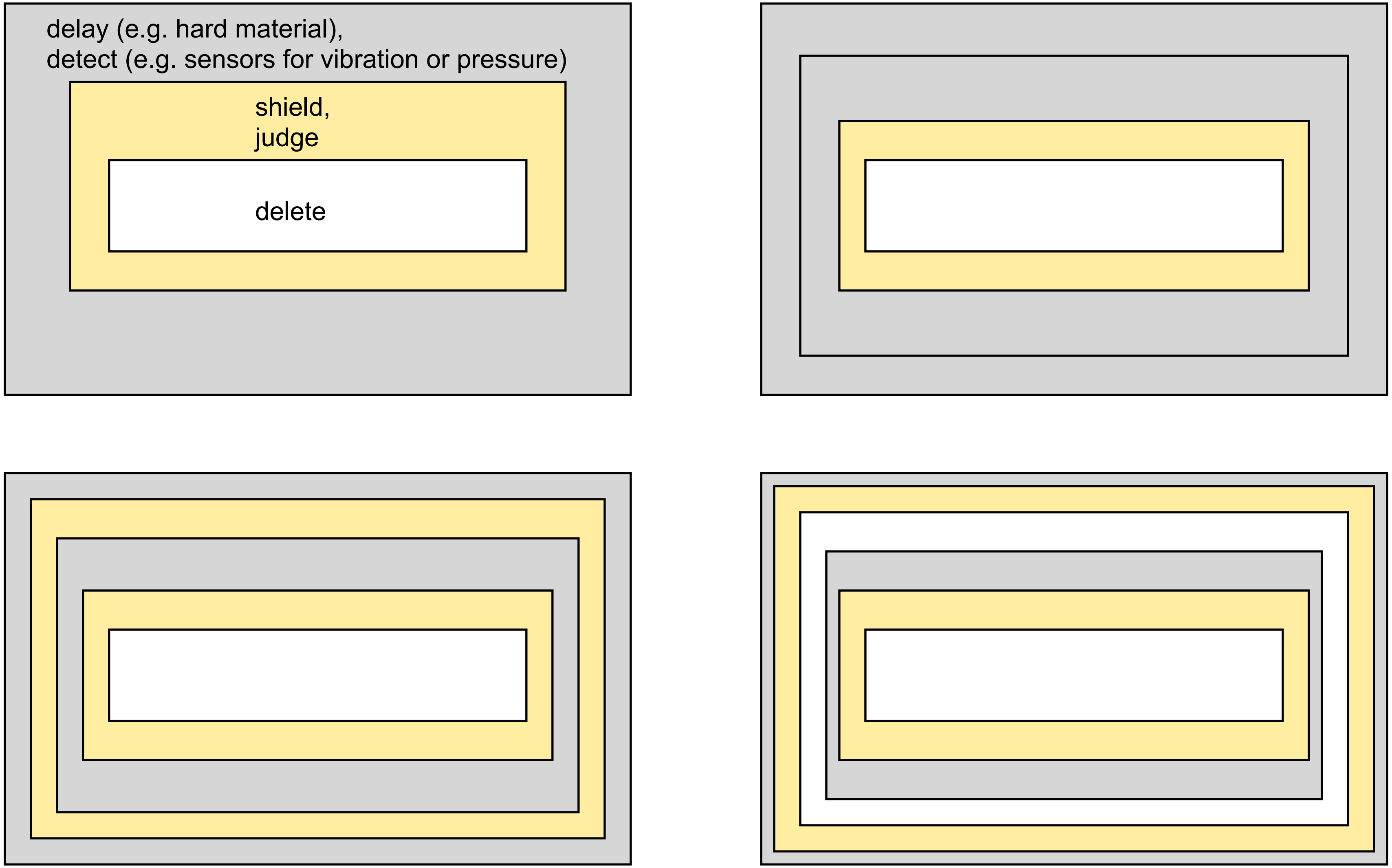
… hope the attacker cannot be at many places at the same time.

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



delay (e.g. hard material),
detect (e.g. sensors for vibration or pressure)

shield,
judge

delete

Tamper-resistant casings

Interference: detect
judge

Attack: delay
delete data (etc.)

Possibility: several layers, shielding

Problem: validation ... credibility

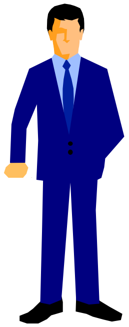
Negative example: smart cards

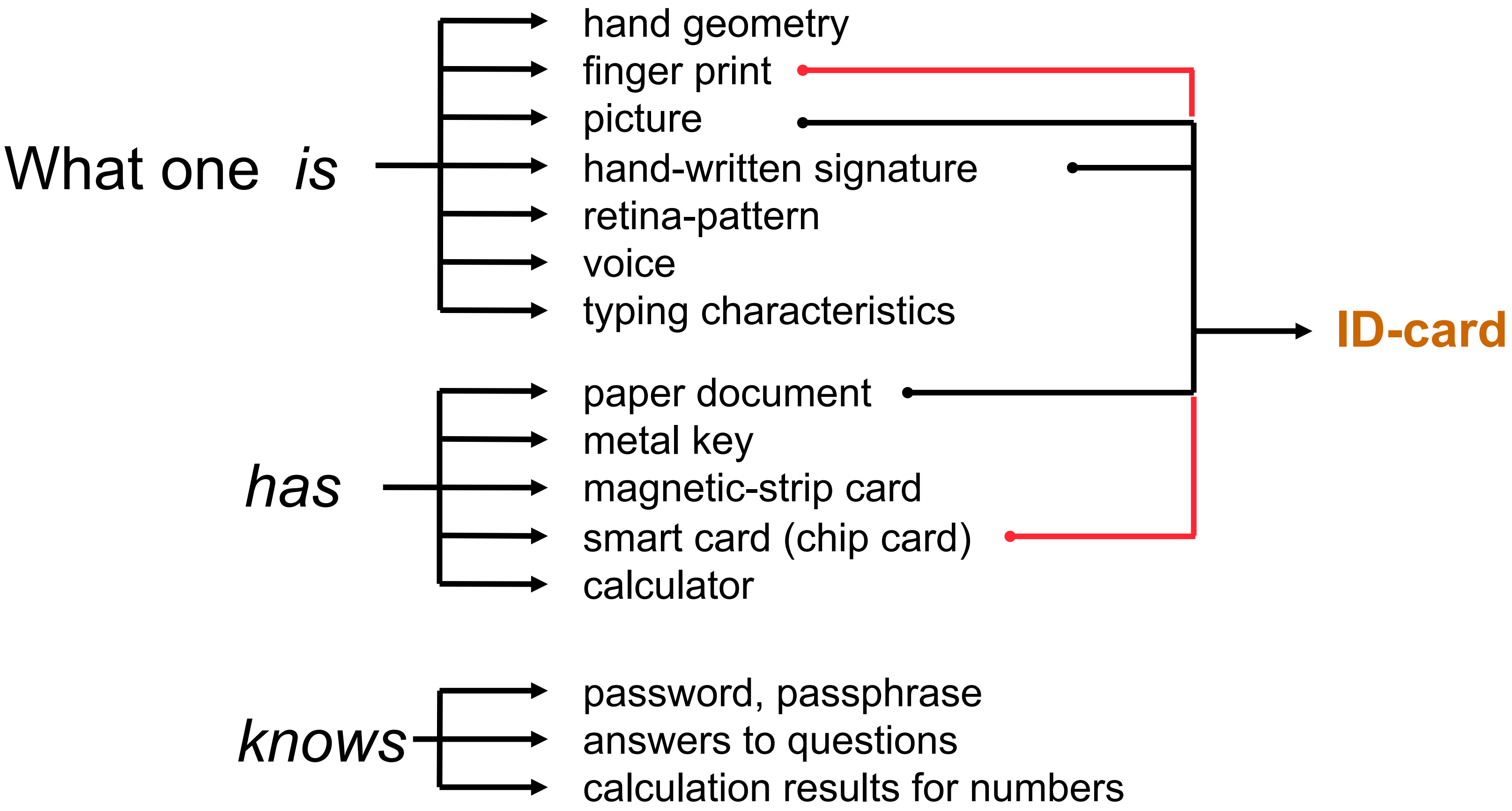
- no detection (battery missing etc.)
- shielding difficult (card is thin and flexible)
- no deletion of data intended, even when power supplied

Golden rule

Correspondence between organizational and IT structures

Identification of human beings by IT-systems





What one *is*

- hand geometry
- finger print
- picture
- hand-written signature
- retina-pattern
- voice
- typing characteristics

has

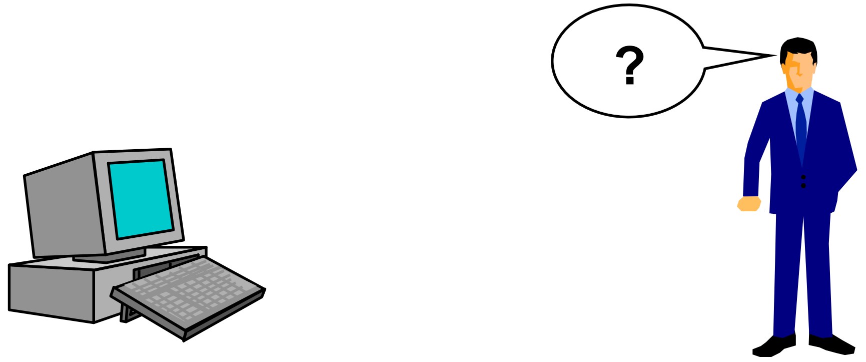
- paper document
- metal key
- magnetic-strip card
- smart card (chip card)
- calculator

knows

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

ID-card

Identification of IT-systems by human beings



What it *is*

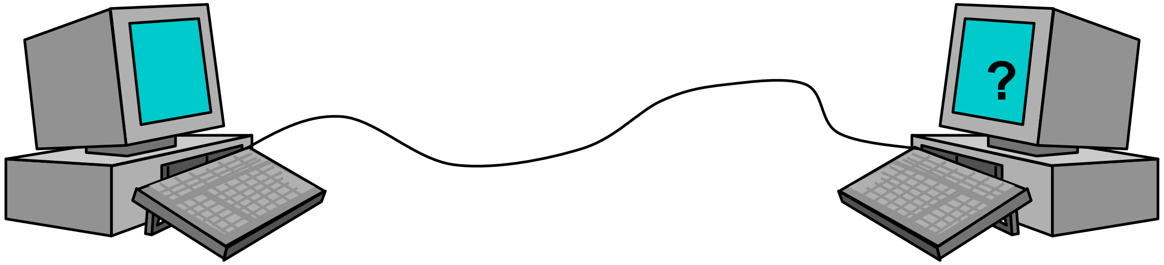
- casing
- seal, hologram
- pollution

knows

- password
- answers to questions
- calculation results for numbers

Where it *stands*

Identification of IT-systems by IT-systems



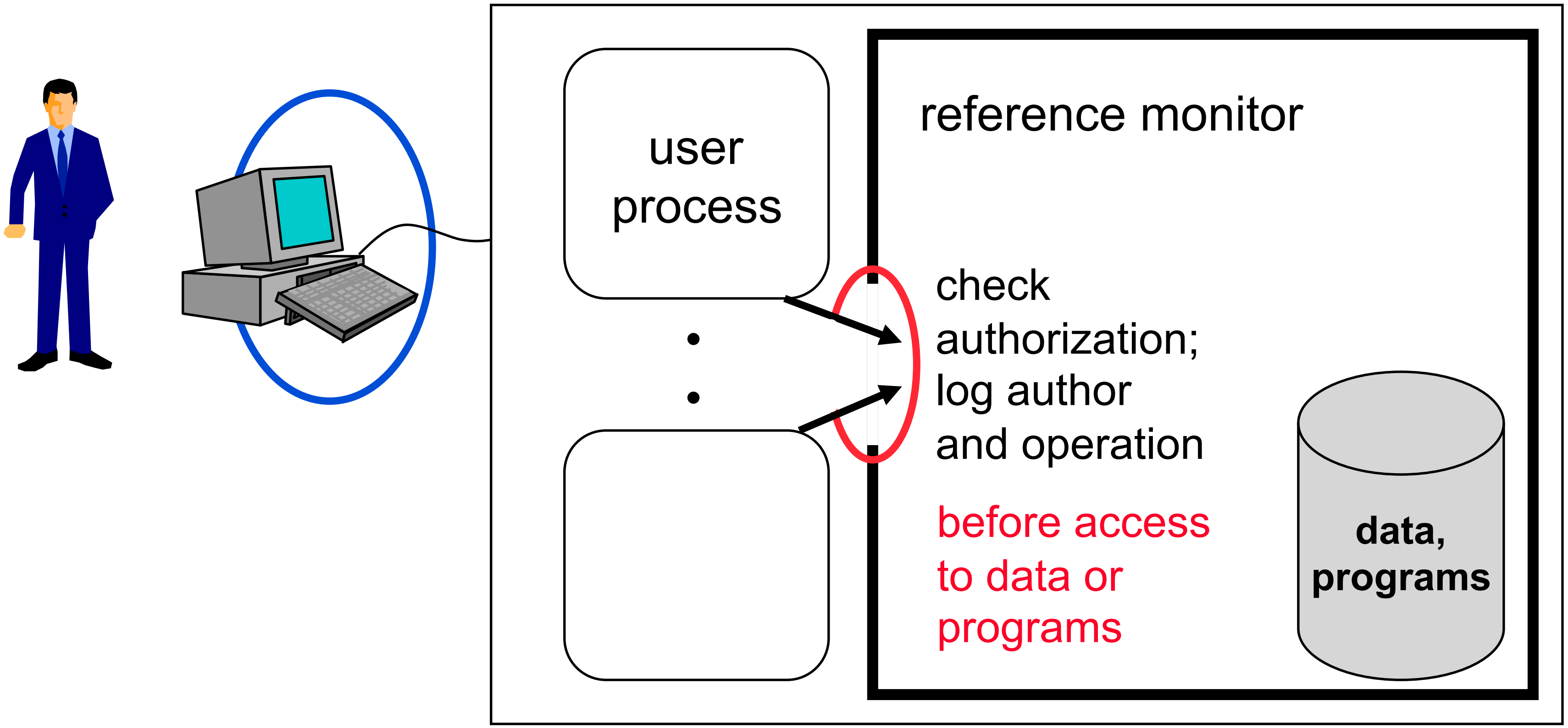
What it *knows*

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

Wiring *from where*

Admission and access control

Admission control communicate with authorized partners only



user process

reference monitor

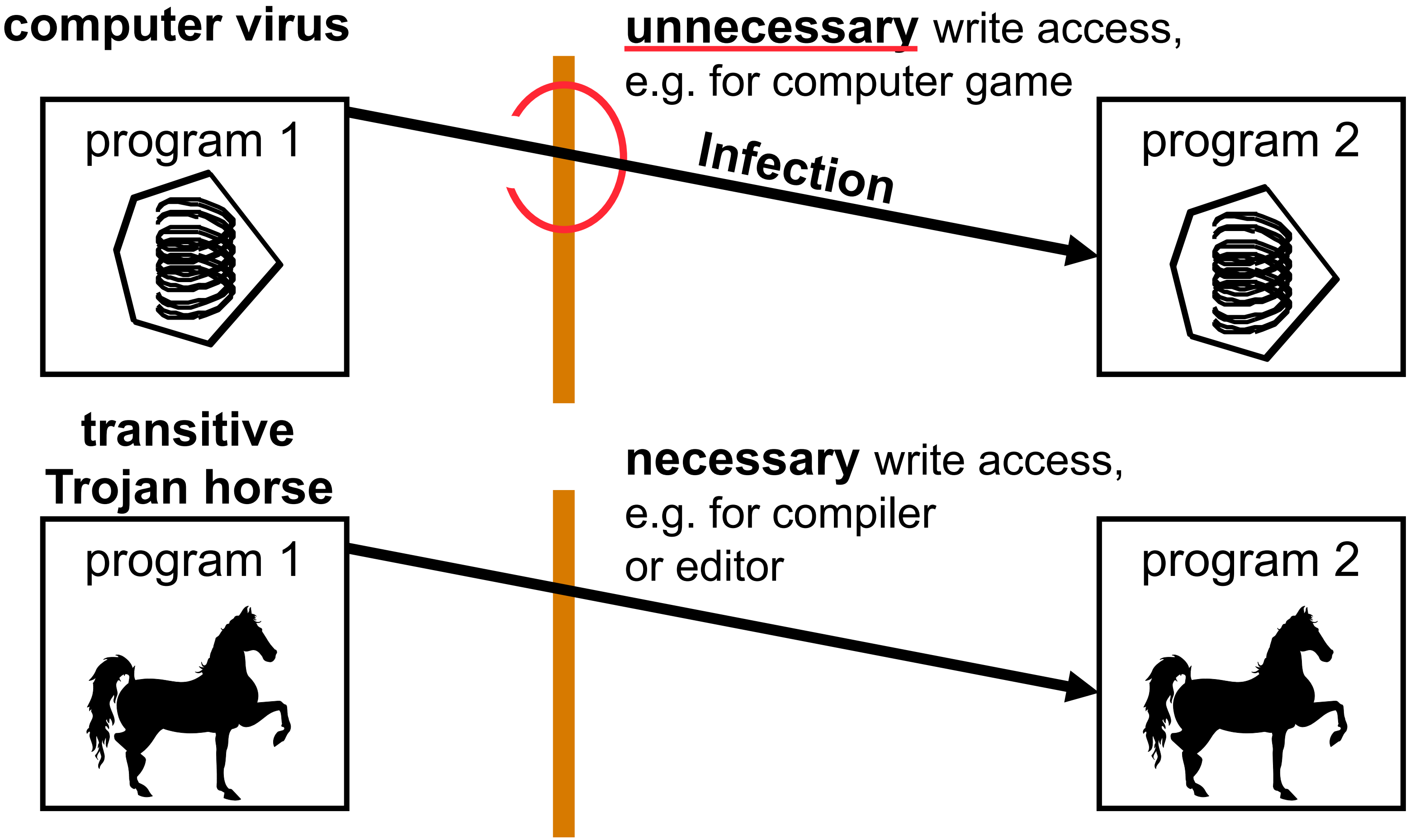
check authorization; log author and operation

before access to data or programs

data, programs

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

Golden Rule

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

Distributed System

Aspects of distribution

physical distribution
distributed control and implementation structure

distributed system:

no entity has a global view on the system

Security in distributed systems

Trustworthy terminals

Trustworthy	only to user
	to others as well

Ability to communicate

Availability by	redundancy and diversity

Cryptography

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

Availability

Infrastructure with the least possible complexity of design

Connection to completely diverse networks

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

Avoid bottlenecks of diversity

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

Basics of Cryptology

Achievable protection goals:

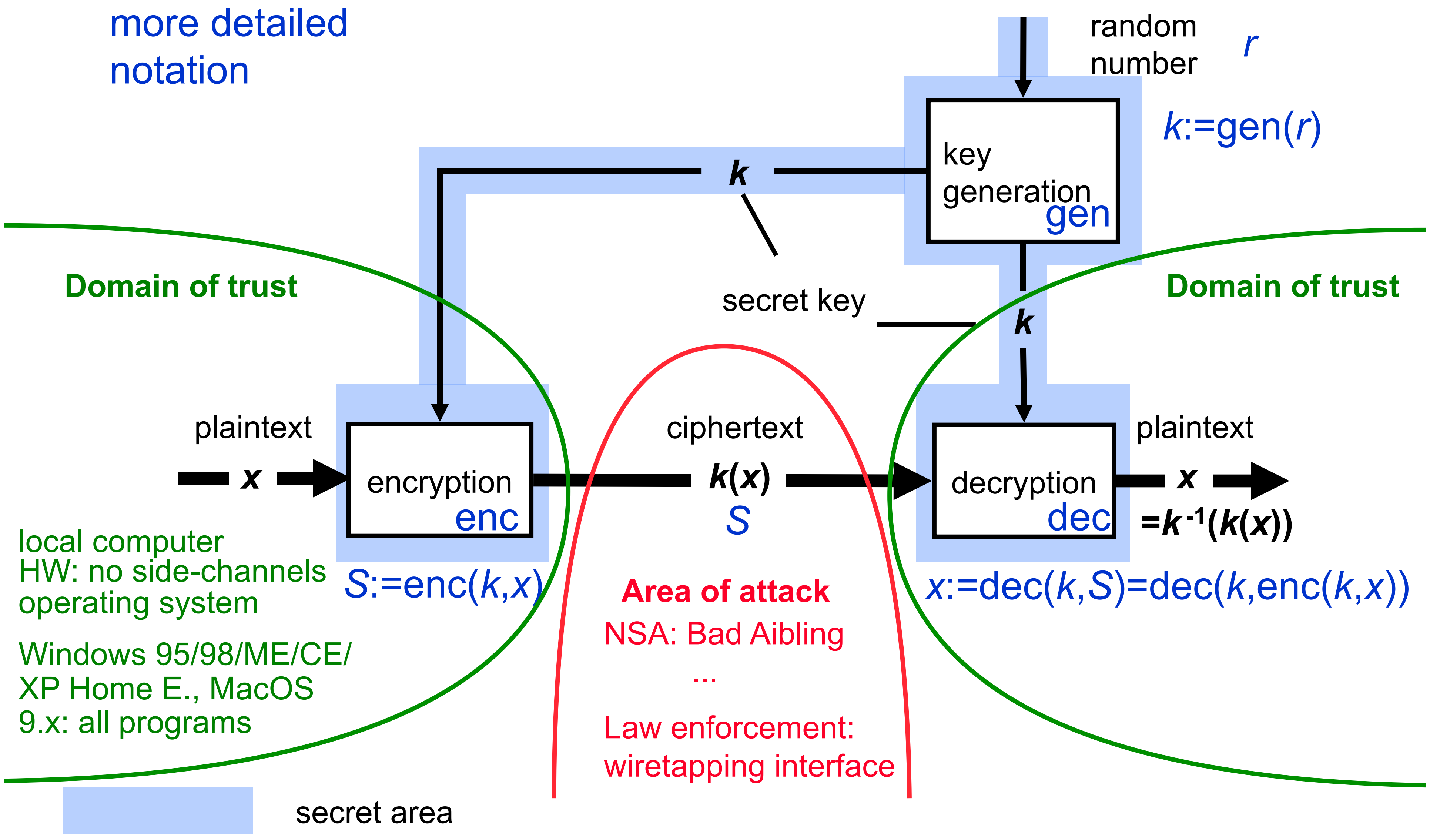
confidentiality, called **concealment**

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

Unachievable by cryptography:

availability – at least not against strong attackers

Symmetric encryption system



more detailed notation

random number r

key generation gen

k:=gen(r)

$\boldsymbol{k}$

secret key

$\boldsymbol{k}$

Domain of trust

Domain of trust

plaintext

$\boldsymbol{x}$

encryption enc

ciphertext

$\boldsymbol{k(x)}$

S

decryption dec

plaintext

$\boldsymbol{x}$

$\boldsymbol{=k^{-1}(k(x))}$

local computer
HW: no side-channels
operating system

Windows 95/98/ME/CE/
XP Home E., MacOS
9.x: all programs

S:=enc(k,x)

Area of attack

NSA: Bad Aibling

...

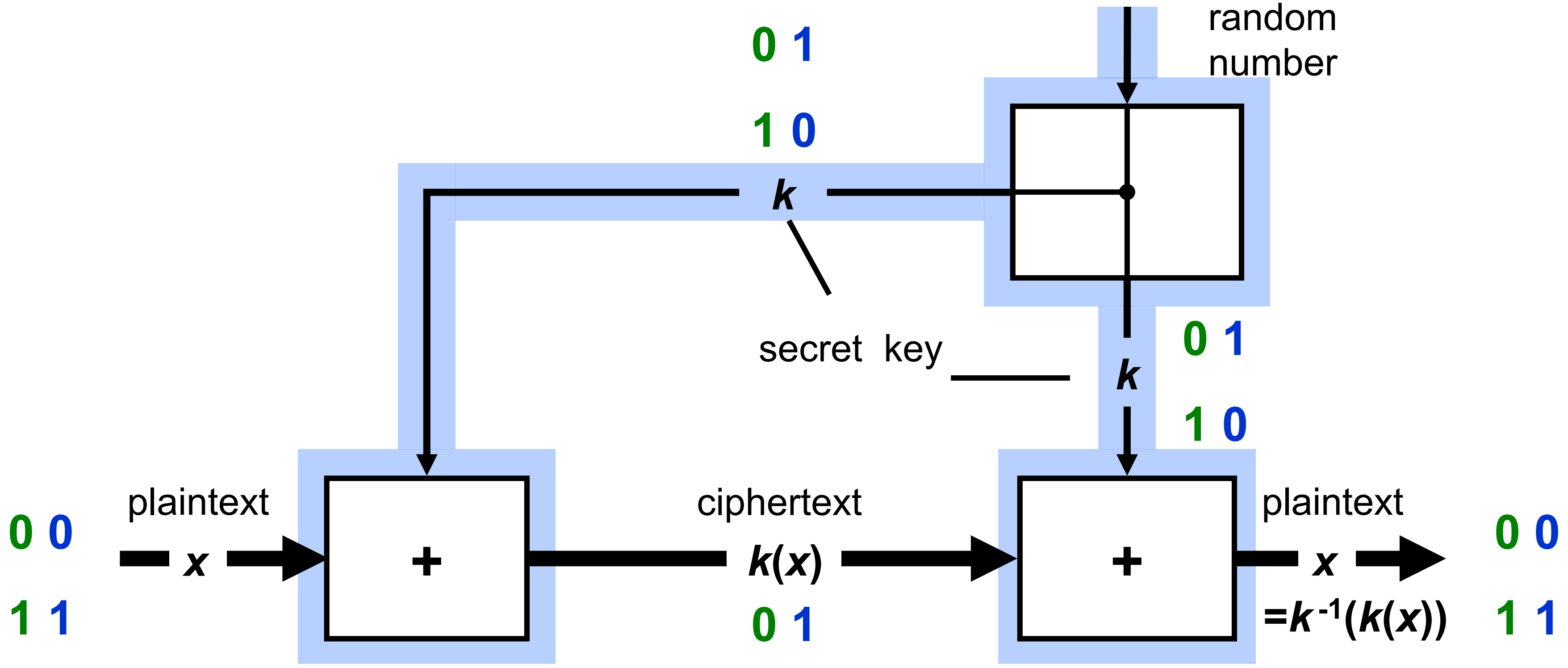
Law enforcement:
wiretapping interface

x:=dec(k,S)=dec(k,enc(k,x))

🟦 secret area

Opaque box with lock; 2 identical keys

Example: Vernam cipher (=one-time pad)

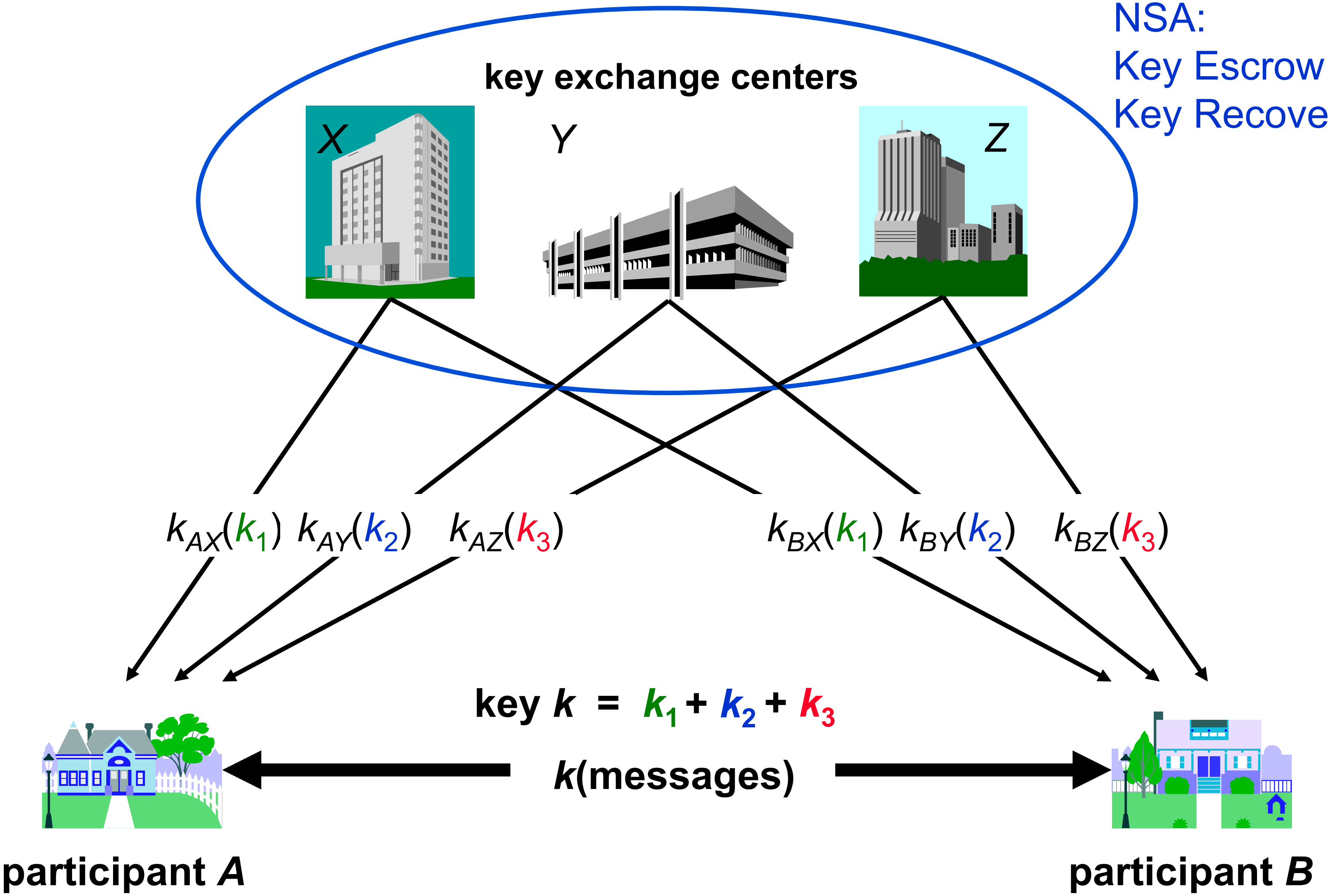


🟦 secret area

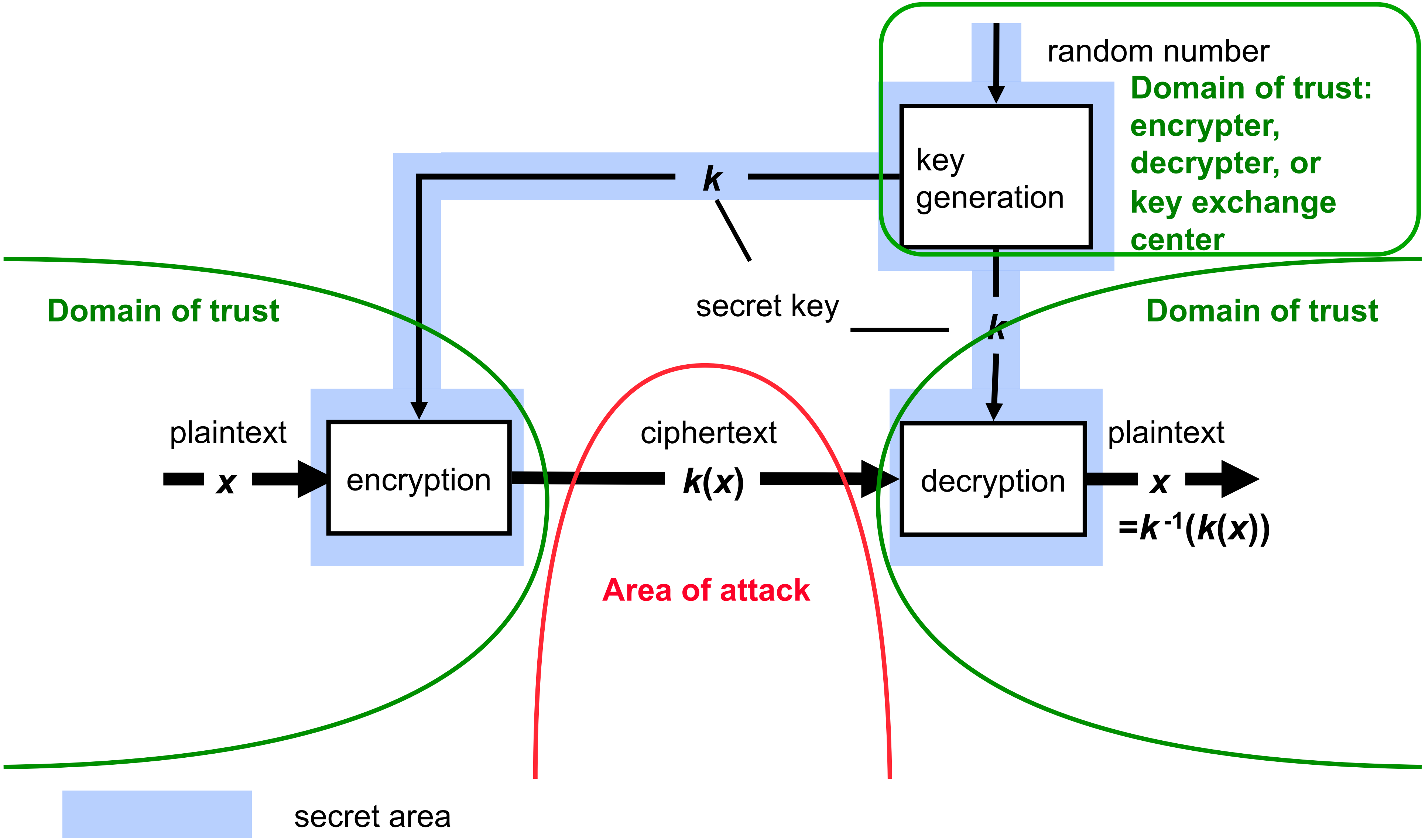
Opaque box with lock; 2 identical keys

Key exchange using symmetric encryption systems

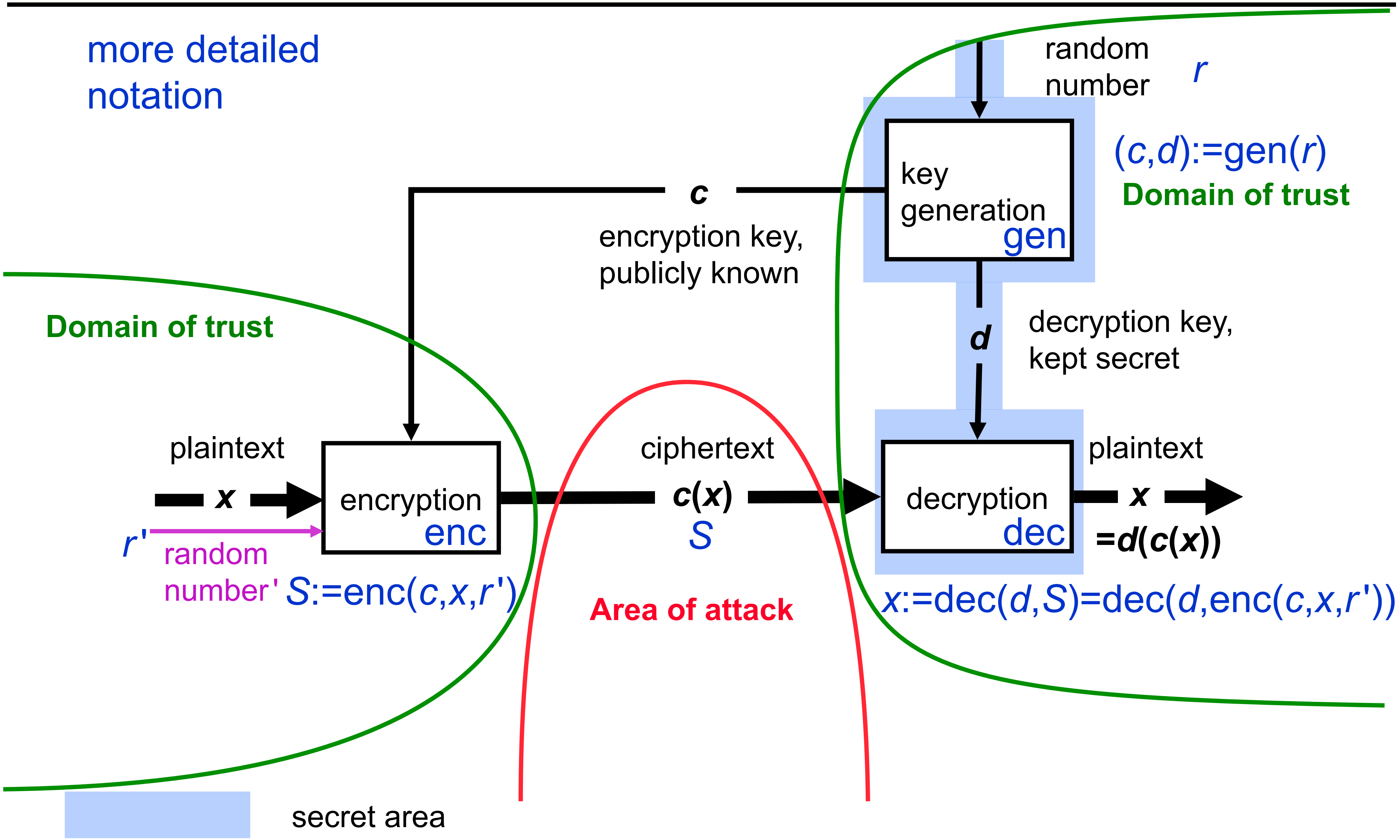
NSA:
Key Escrow
Key Recovery


key exchange centers
X
Y
Z
$k_{AX}(k_1)$ $k_{AY}(k_2)$ $k_{AZ}(k_3)$
$k_{BX}(k_1)$ $k_{BY}(k_2)$ $k_{BZ}(k_3)$
key k = $k_1 + k_2 + k_3$
k(messages)
participant A
participant B

Sym. encryption system: Domain of trust key generation

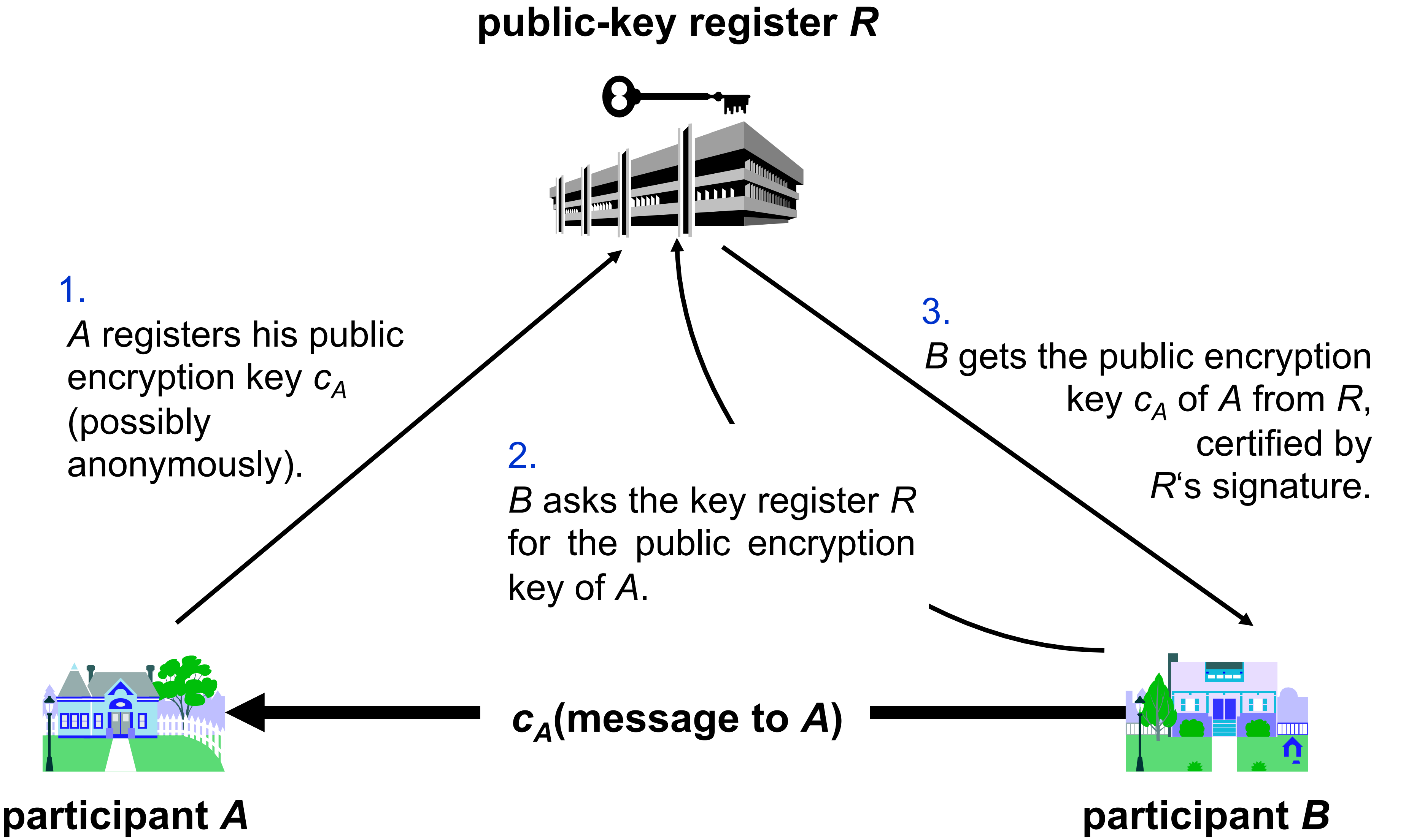


Asymmetric encryption system



Opaque box with spring lock; 1 key

Key distribution using asymmetric encryption systems



public-key register *R*

1\.
A registers his public encryption key c_A (possibly anonymously).

2\.
B asks the key register *R* for the public encryption key of *A*.

3\.
B gets the public encryption key c_A of *A* from *R*, certified by *R*'s signature.

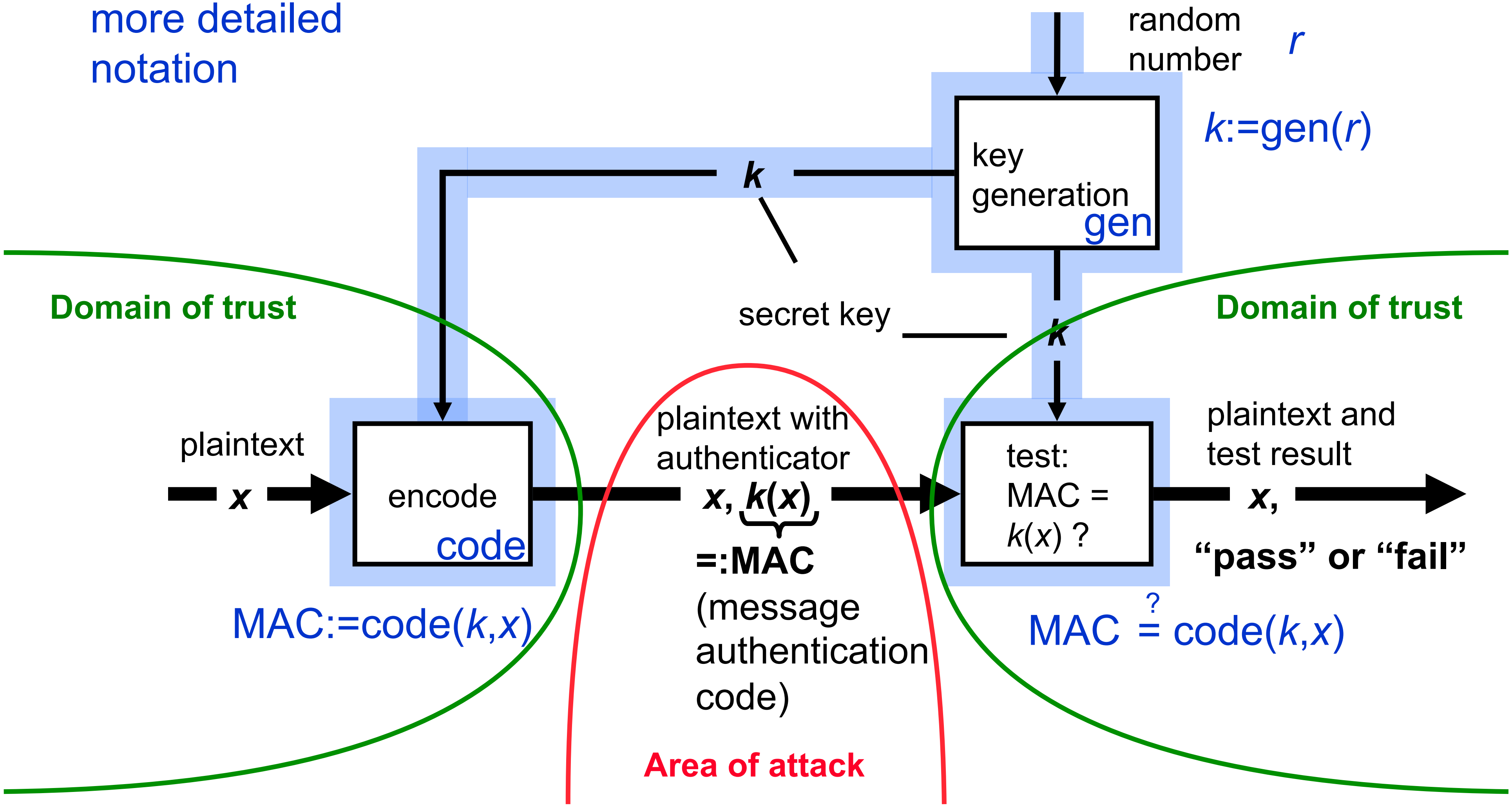
c_A(message to *A*)

participant *A*

participant *B*

Symmetric authentication system

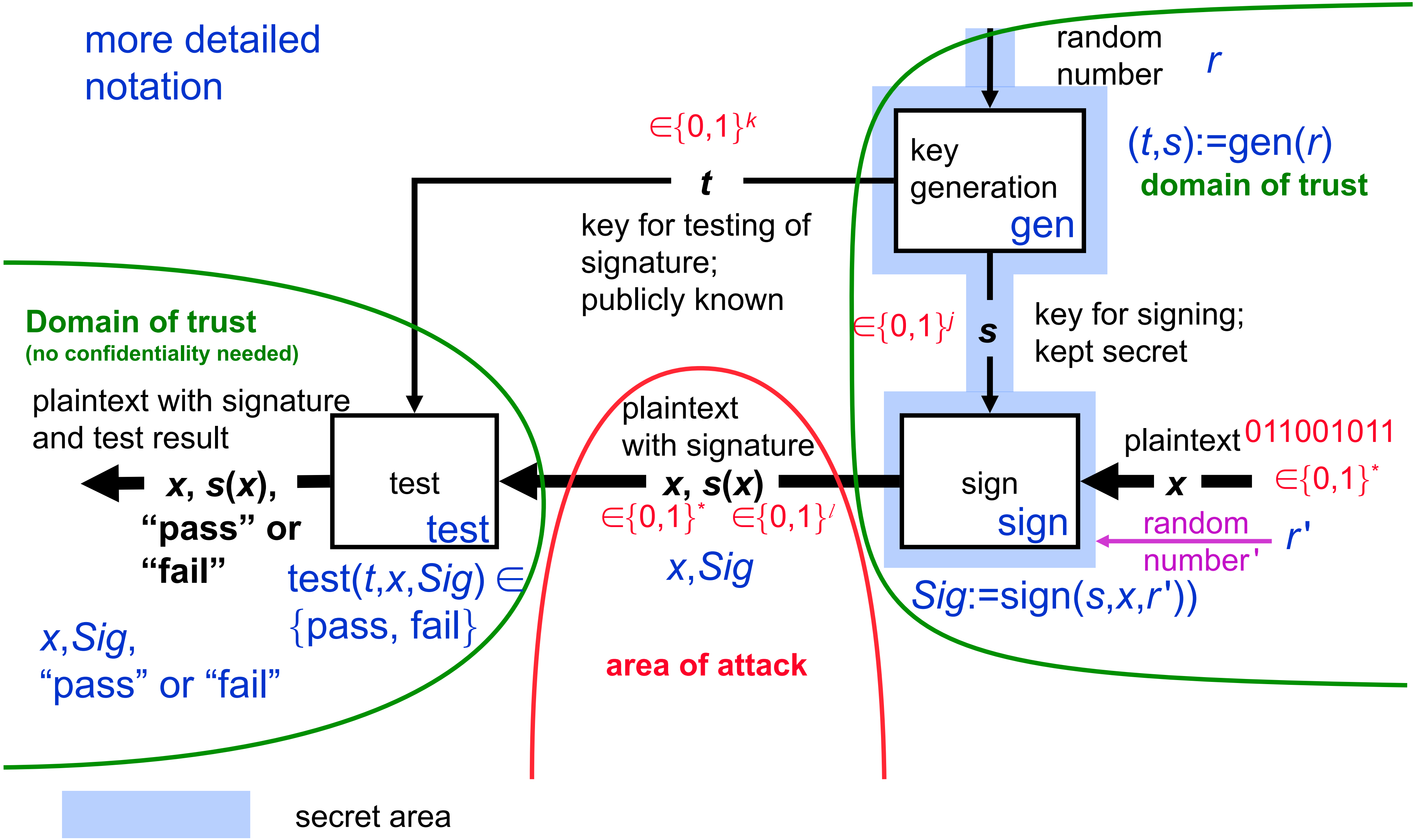
more detailed notation



🟦 secret area

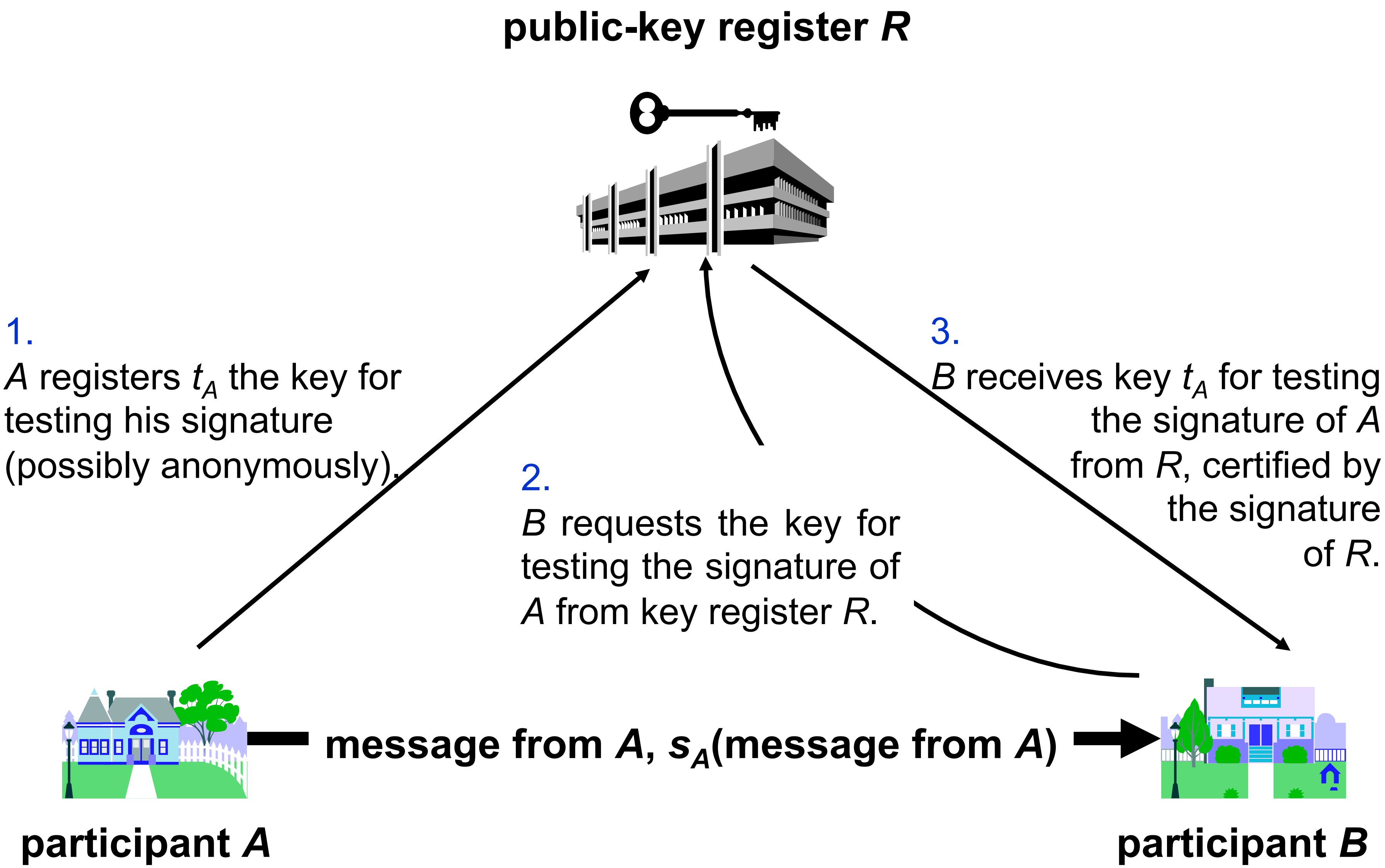
Show-case with lock; 2 identical keys

Digital signature system



Show-case with lock; 1 key

Key distribution using digital signature systems



public-key register *R*

1.
A registers t_A the key for testing his signature (possibly anonymously).

2.
B requests the key for testing the signature of *A* from key register *R*.

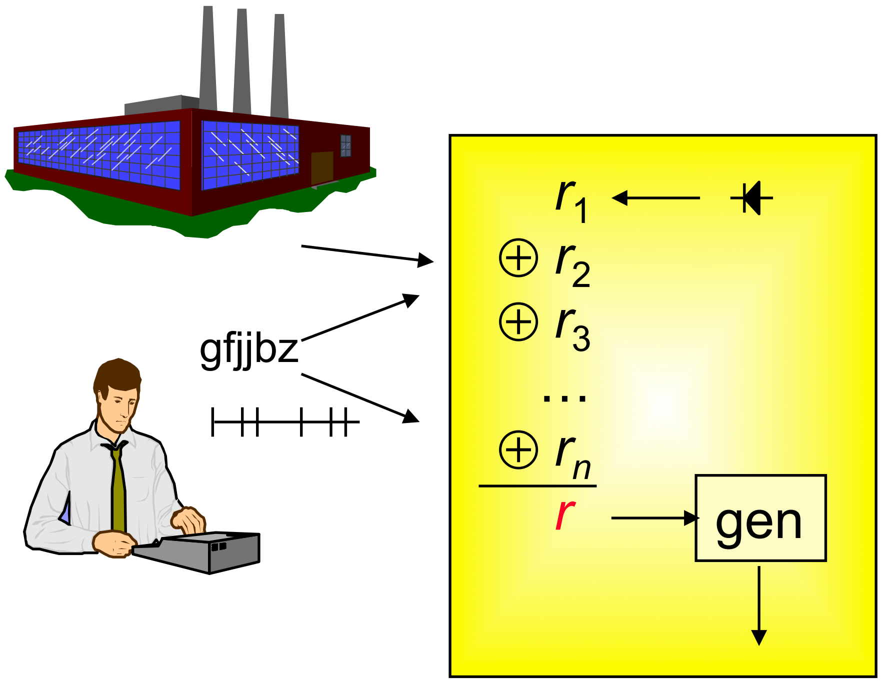
3.
B receives key t_A for testing the signature of *A* from *R*, certified by the signature of *R*.

message from *A*, s_A(message from *A*)

participant *A*

participant *B*

Key generation



gfjjbz

r_1

$\oplus\ r_2$

$\oplus\ r_3$

…

$\oplus\ r_n$

r

gen

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.

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

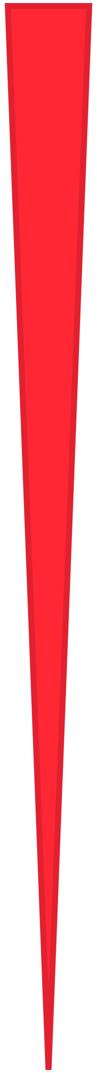
asymmetric systems	*n* per system
symmetric systems	$n \bullet (n\text{-}1)$

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)

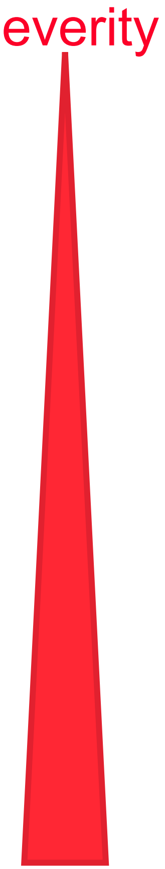
c) individual messages,

e.g. especially for authentication systems

c1) one selected message (selective break)

c2) any message (existential break)

Types of attack



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 $\underbrace{l \rightarrow \infty}_{\text{slow}}$, $\underbrace{\text{then probability of breaking} \rightarrow 0}_{\text{fast}}$.
   - hope:

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!

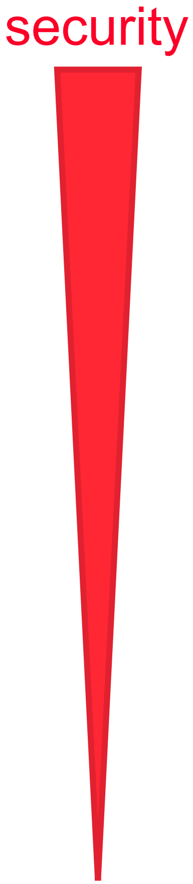
5) What if assumption turns out to be wrong?

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



1. attacker assumed to be computationally unrestricted

2. cryptographically strong

3. well analyzed

4. somewhat analyzed

5. kept secret

Overview of cryptographic systems

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

Hybrid cryptosystems (1)

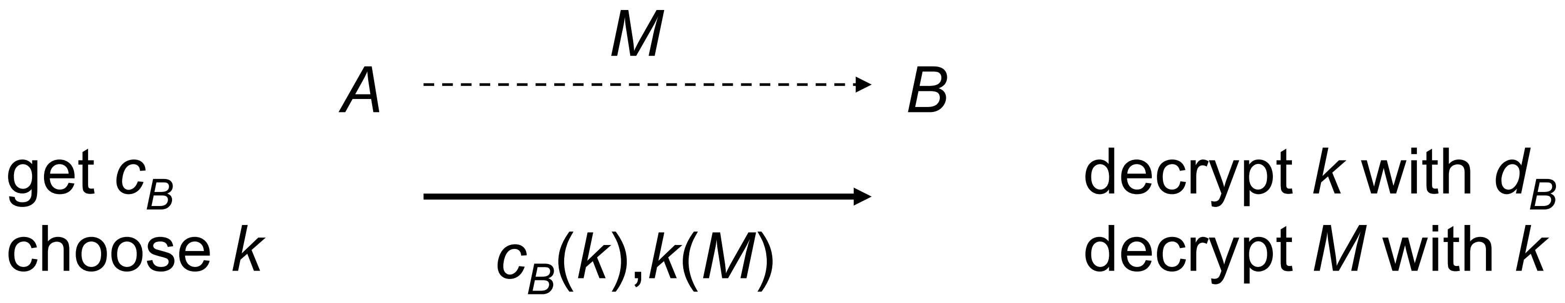
Combine:

- 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:



$A \xrightarrow{M} B$

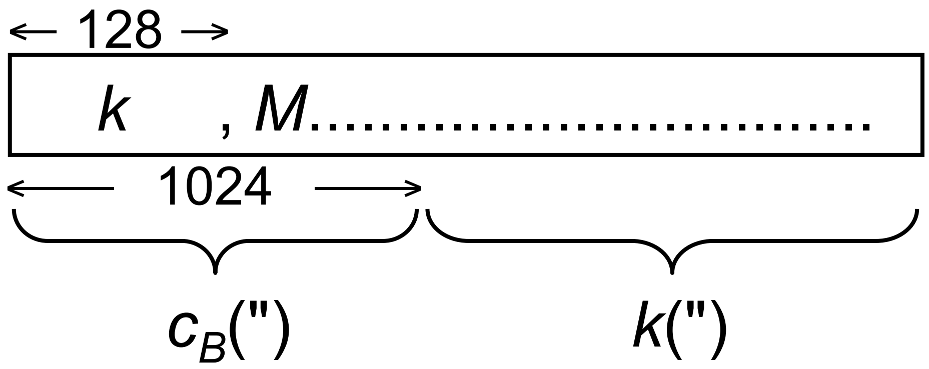
get c_B
choose k

$c_B(k),k(M)$

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

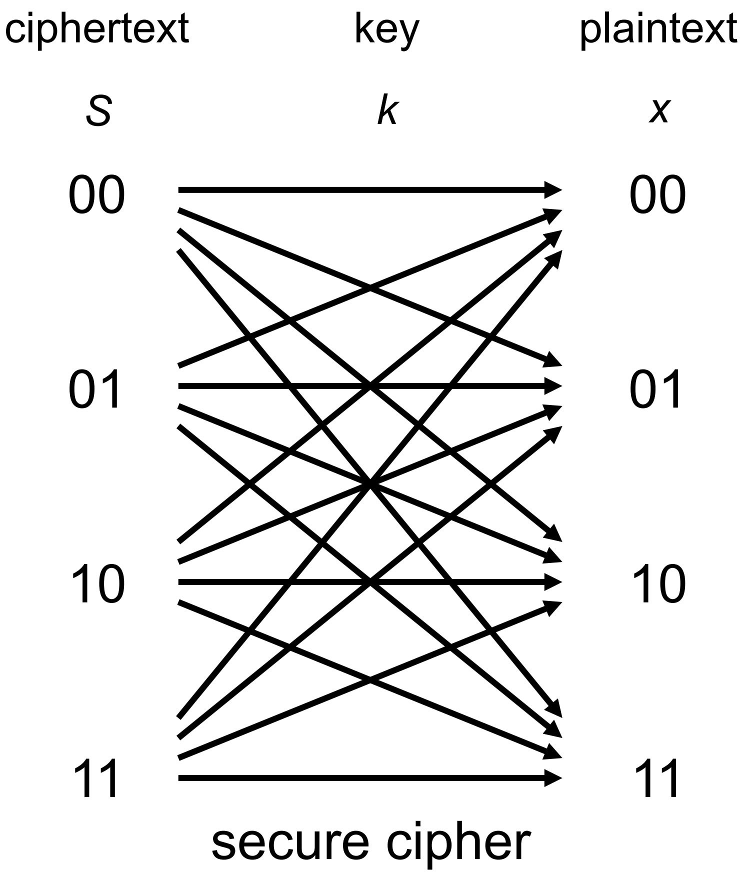
get c_B
choose k

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

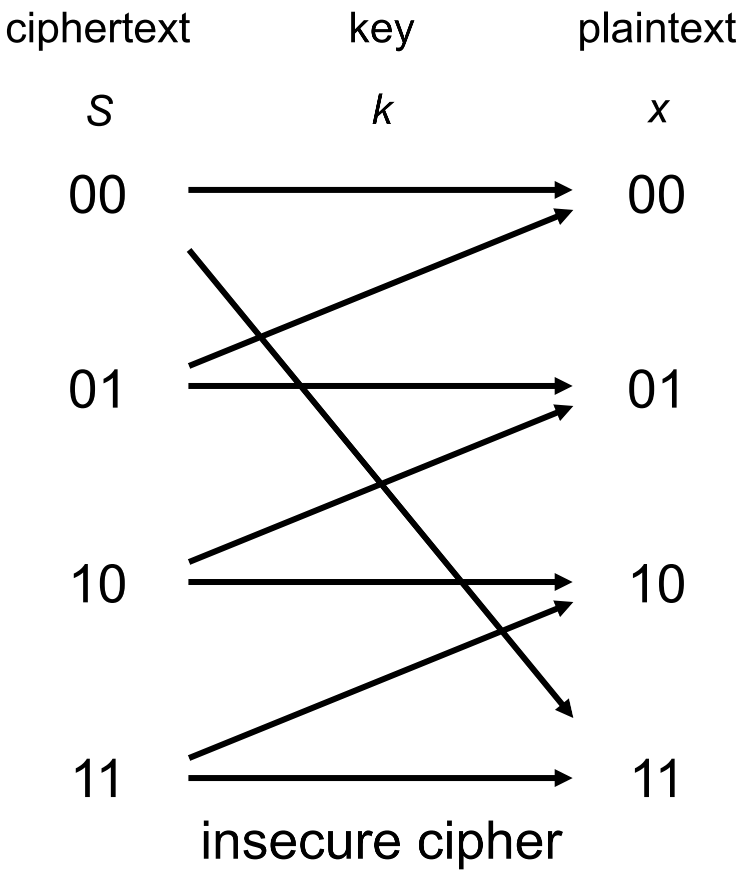
get t_A
decrypt $c_B(B,k,s_A(B,k))$
test B,k with t_A
test M with k

Information-theoretically secure encryption (1)

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



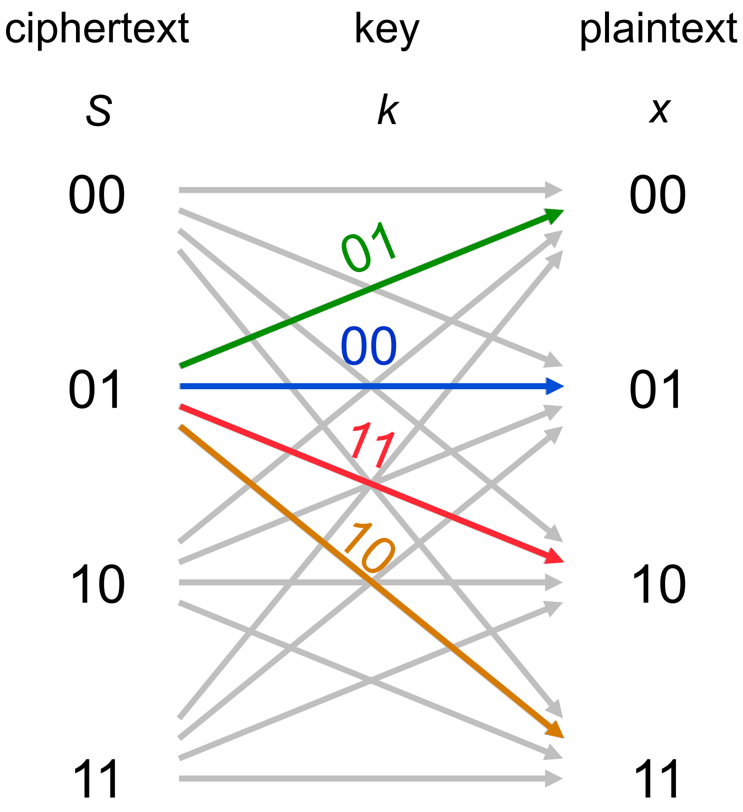
secure cipher



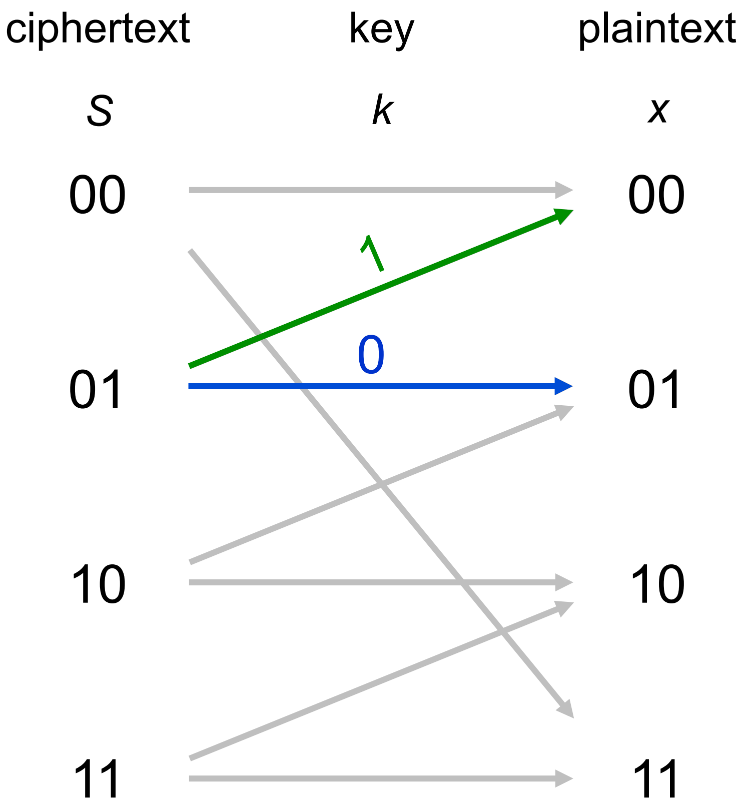
insecure cipher

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

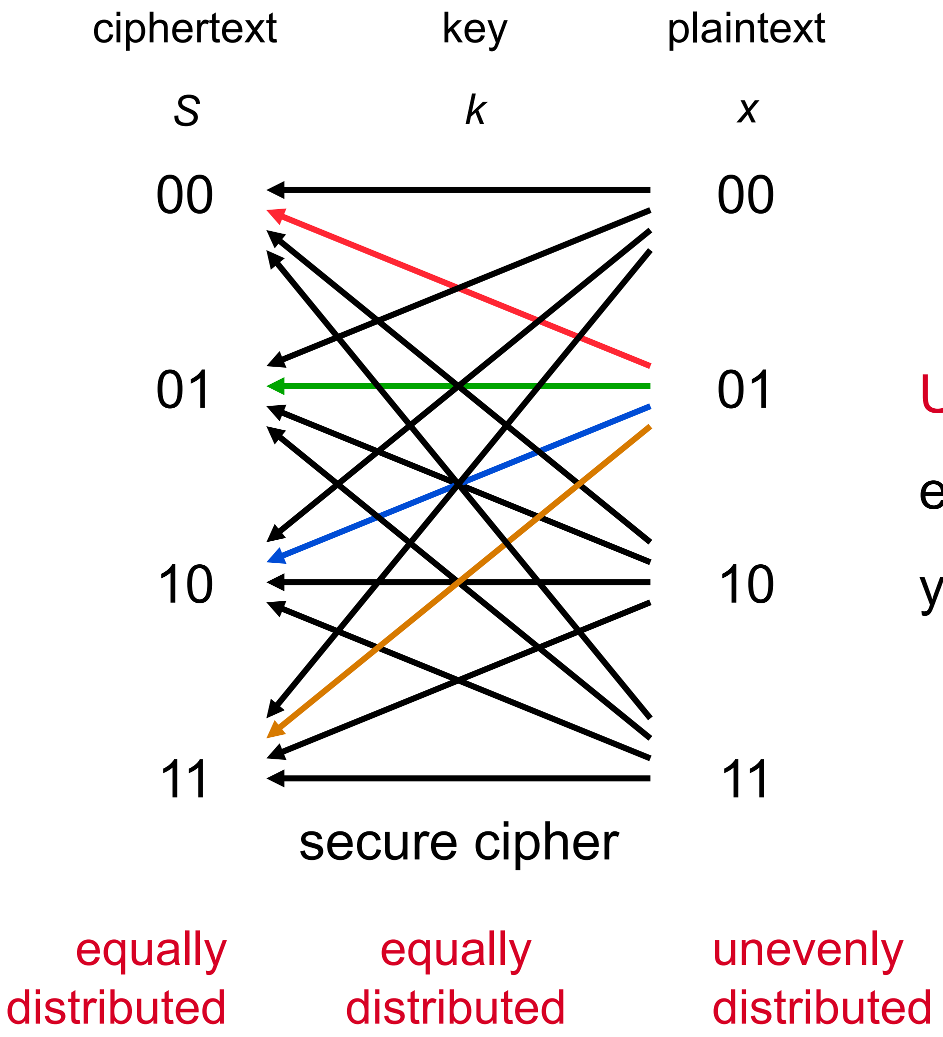
$\oplus$ k = 10 11 01 00

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?



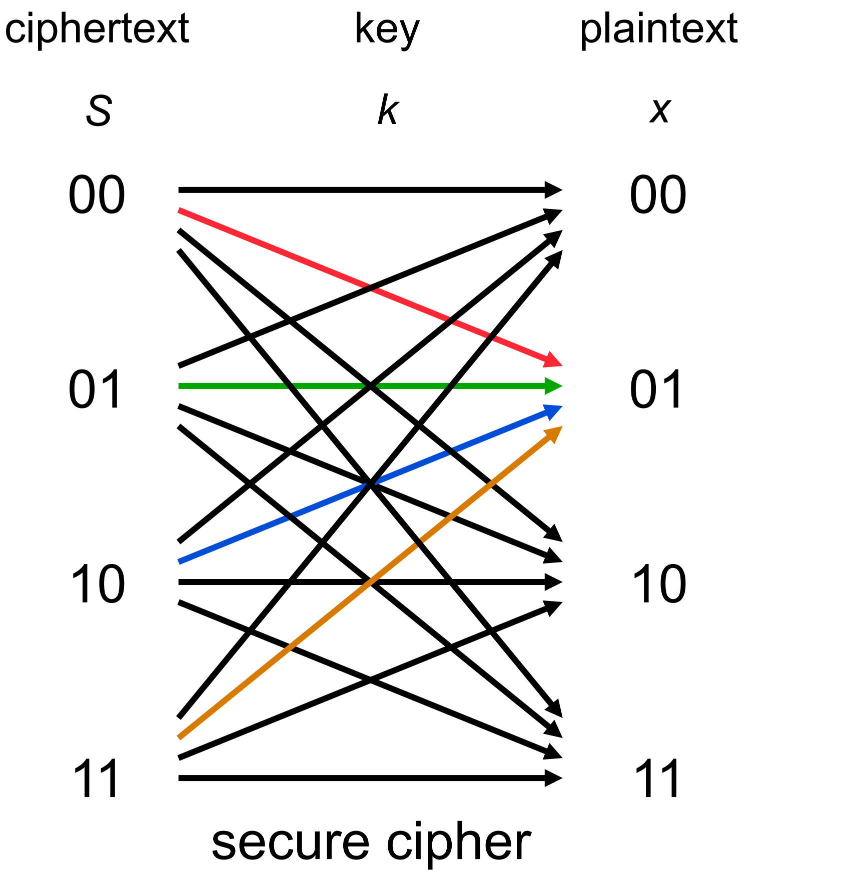
Unevenly distributed plaintexts

enciphered with equally distributed keys

yield equally distributed ciphertexts.

Information-theoretically secure encryption (4)

Different probability distributions – how do they fit?



ciphertext key plaintext

S *k* *x*

00 00

01 01

10 10

11 11

secure cipher

equally distributed

equally distributed, 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

$$S_i \ := \ x_i + k_i$$

It can be decrypted with

$$x_i \ := \ S_i - k_i.$$

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
$\mathcal{S}$ is the set of ciphertexts, which appear at least once.

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

$|\mathcal{K}| \geq |\mathcal{S}|$ 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.

Preparation: Definition for information-theoretical security

How would you define

information-theoretical security

for encryption?

Write down at least

2 definitions

and argue for them!

Definition for information-theoretical security

1. Definition for information-theoretical security

(all keys are chosen with the same probability)

$$\forall S \in \mathcal{S}\ \exists\, const \in \mathrm{IN}\ \forall x \in \mathcal{X}\colon |\{k \in \mathcal{K} | \ k(x) = S\}| = const. \qquad (1)$$

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

2. Definition

$$\forall S \in \mathcal{S}\ \forall x \in \mathcal{X}\colon W(x|S) = W(x). \qquad (2)$$

Both definitions are equivalent (if W(x) > 0):

According to Bayes: $\boxed{W(x \mid S) = \frac{W(x) \bullet W(S \mid x)}{W(S)}}$

Therefore, (2) is equivalent to

$$\forall S \in \mathcal{S}\ \forall x \in \mathcal{X}\colon W(S|x) = W(S). \qquad (3)$$

We show that this is equivalent to

$$\forall S \in \mathcal{S}\ \exists\, const' \in \mathrm{IR}\ \forall x \in \mathcal{X}\colon W(S|x) = const'. \qquad (4)$$

Proof

(3)⇒(4) is clear with $const' := W(S)$.

Conversely, we show $const' = W(S)$:

$$\begin{aligned} W(S) &= \sum_x W(x) \bullet W(S|x) \\ &= \sum_x W(x) \bullet const' \\ &= const' \bullet \sum_x W(x) \\ &= const'. \end{aligned}$$

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

$$\mathrm{W}(S|x) = \mathrm{W}(\{k \mid k(x) = S\}),$$

and if all keys have the same probability,

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

Then (4) is equivalent (1) with

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

Another definition for information-theoretical security

Sometimes, students come up with the following definition:

$$\forall S \in \mathcal{S} \quad \forall x \in \mathcal{X}\colon \; W(S) = W(S|x).$$

This is *not* equivalent, but a slight modification is:

3. Definition

$$\forall S \in \mathcal{S} \quad \forall x \in \mathcal{X} \text{ with } W(x)>0\colon \; W(S) = W(S|x).$$

Definitions 2. and 3. are equivalent:

Remember Bayes:

$$\boxed{W(x \mid S) = \frac{W(x) \bullet W(S \mid x)}{W(S)}}$$

$W(x|S) \;=\; W(x)$ <==> (Bayes)

$\dfrac{W(x) \bullet W(S \mid x)}{W(S)} \;=\; W(x)$ <==> (if $W(x) \neq 0$, we can divide by $W(x)$)

$W(S|x) \;=\; W(S)$

$W(S|x)$ as proposed by some students assumes that x may be sent, i.e. $W(x)>0$.

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:

		x, 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)

$$W(k(y) = \mathrm{MAC'} \mid k(x) = \mathrm{MAC}\ ) \leq \varepsilon$$

(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 = $\mathrm{MAC}_1,\ldots,\mathrm{MAC}_\sigma$; MAC_i calculated using $k_i k_i^*$
$\Rightarrow$ error probability $2^{-\sigma}$

b) l message bits:

$$\begin{array}{llll} x^{(1)}, & \mathrm{MAC}^{(1)} = \mathrm{MAC}_1^{(1)}, & \ldots, & \mathrm{MAC}_\sigma^{(1)} \\ \vdots & \vdots & & \vdots \\ x^{(l)}, & \mathrm{MAC}^{(l)} = \mathrm{MAC}_1^{(l)}, & \ldots, & \mathrm{MAC}_\sigma^{(l)} \end{array}$$

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: you need 2σ-bit-key
(for succeeding messages σ bits suffice, if recipient adequately responds on authentication “errors”)

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

(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 ...)

$n = p \bullet q$

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

Often: special property

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

(the semantics of "≡ ... 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 ⇒ but difficulty cannot be proved yet

complexity at present

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

, $c \approx 1{,}9$

"sub-exponential"

$$\approx e^{\sqrt[3]{l}}$$

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 (www.crypto-world.com/FactorRecords.html)

(notice :

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

assumption: factoring is hard

(notice : If an attacker could factor, e.g., every 1000th *n*, this would be unacceptable.)

Factoring assumption

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

$\forall$ polynomials $\mathcal{Q}$

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

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

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

(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 lth.

And ≈ every 2nd ≡ 3 mod 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”

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

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

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

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

error probability $\leq \dfrac{1}{4^m}$

(doesn’t matter in general)

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

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

- +, -, •   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

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

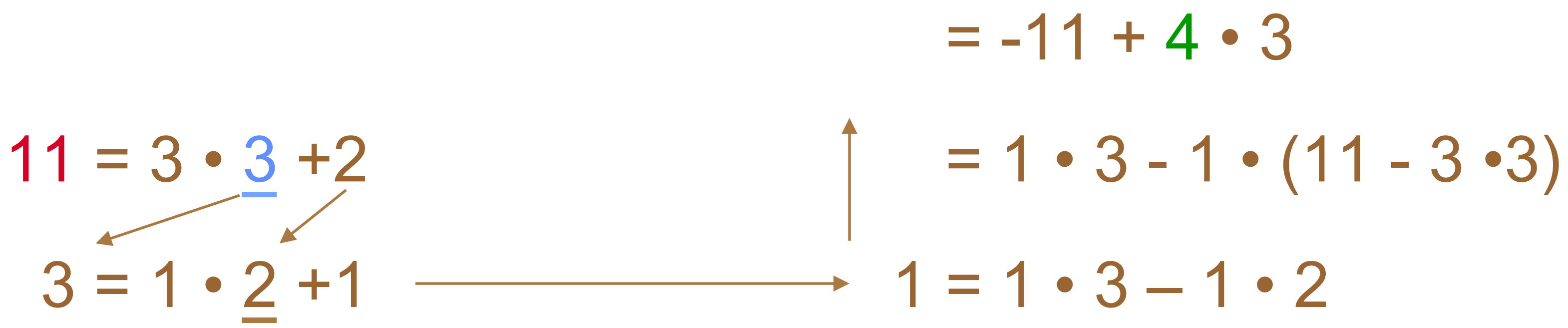
- Inverting is fast (extended Euclidean Algorithm)

  Determine to a,n the values u,v with

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

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

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



$= -11 + 4 \cdot 3$

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

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

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

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

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

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

Number of elements of Z_n^*

The Euler Φ- Function is defined as

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

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)

$$\begin{tikzcd}
x \equiv y \bmod n \arrow[d, Leftrightarrow, "\text{since}"'] & \Leftrightarrow & x \equiv y \bmod p \arrow[d, Leftrightarrow] & \wedge & x \equiv y \bmod q \arrow[d, Leftrightarrow] \\
n|(x-y) & \Leftrightarrow & p|(x-y) & \wedge & q|(x-y)
\end{tikzcd}$$

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

⇒ 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 : $u \bullet p \ + \ v \bullet q \ = \ 1$

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

Since :

	mod p	mod q
$u \bullet p$	0	1
$v \bullet q$	1	0
y	$0 \bullet y_q + 1 \bullet y_p$	$1 \bullet y_q + 0 \bullet y_p$
	$\equiv \mathrm{y}_p$	$\equiv \mathrm{y}_q$

$$\underbrace{\hphantom{xxxxxxxxxxxxxxxxxxxx}}_{\text{CRA}}$$

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 $\qquad (-1)^2 = 1$

but attention: e.g. mod 8

$$\left.\begin{array}{ll} 1^2 \equiv 1 & 3^2 \equiv 1 \\ 7^2 \equiv 1 & 5^2 \equiv 1 \end{array}\right\} \begin{array}{l} 4 \\ \text{roots} \end{array}$$

QR_n multiplicative group:

$$x_1, x_2 \in QR_n \;\Rightarrow\; \begin{array}{ll} x_1 \bullet x_2 \in QR_n & : (y_1 y_2)^2 = y_1^2 y_2^2 = x_1 x_2 \\ x_1^{-1} \in QR_n & : (y_1^{-1})^2 = (y_1^2)^{-1} = x_1^{-1} \end{array}$$

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 → 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 $(x^{\frac{p-1}{2}})^2 \equiv 1$

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

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

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

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

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

proof :

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

2. $$w^2 = x^{\frac{p+1}{2}} = x^{\frac{p-1}{2}+1} = \underset{\substack{\Downarrow \\ \text{Euler, } x \in QR_p}}{x^{\frac{p-1}{2}}} \bullet x = 1 \bullet x$$

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

- $$\left(\frac{-1}{p}\right) \equiv (-1)^{\frac{p-1}{2}} \underset{\substack{\uparrow \\ p\,=\,4r+3}}{=} (-1)^{\frac{4r+2}{2}} = (-1)^{2r+1} = -1$$

⇒ $-1 \notin QR_p$

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

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

squares and roots mod n using p,q

(usable as secret operations)

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

$$x \in \mathrm{QR}_n \quad \Leftrightarrow \quad x \in \mathrm{QR}_p \ \wedge \ x \in \mathrm{QR}_q$$

Chinese Remainder Theorem

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

"$\Leftarrow$" $x \equiv w_p{}^2 \bmod p \ \wedge \ x \equiv w_q{}^2 \bmod q$

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

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

using the Chinese Remainder Theorem for

$w^2 \equiv w_p{}^2 \equiv x \bmod p \ \wedge \ w^2 \equiv w_q{}^2 \equiv x \bmod q$

we have

$w^2 \equiv x \bmod n$

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

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

$x \in \mathrm{QR}_n \Rightarrow x$ 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

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

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

Jacobi symbol $$\left(\frac{x}{n}\right) := \left(\frac{x}{p}\right) \bullet \left(\frac{x}{q}\right)$$

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

So : $x \in QR_n \quad \Rightarrow \quad \left(\frac{x}{n}\right) = 1$

$\not\Leftarrow$ 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 $\notin \mathrm{QR}_n$, because $\notin \mathrm{QR}_{p,q}$

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

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 ⇒ extracting a root is difficult”
proof of “factoring is easy ⇐ extracting a root is easy”
So assumption : ∃ $\mathcal{W} \in$ PPA: algorithm extracting a root
to show : ∃ $\mathcal{F} \in$ PPA: factoring algorithm

structure

```
program F
        subprogram W
        [black box]
begin
...
call W   ⎫
...      ⎬  polynomially often
call W   ⎪
...      ⎭
end.
```

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

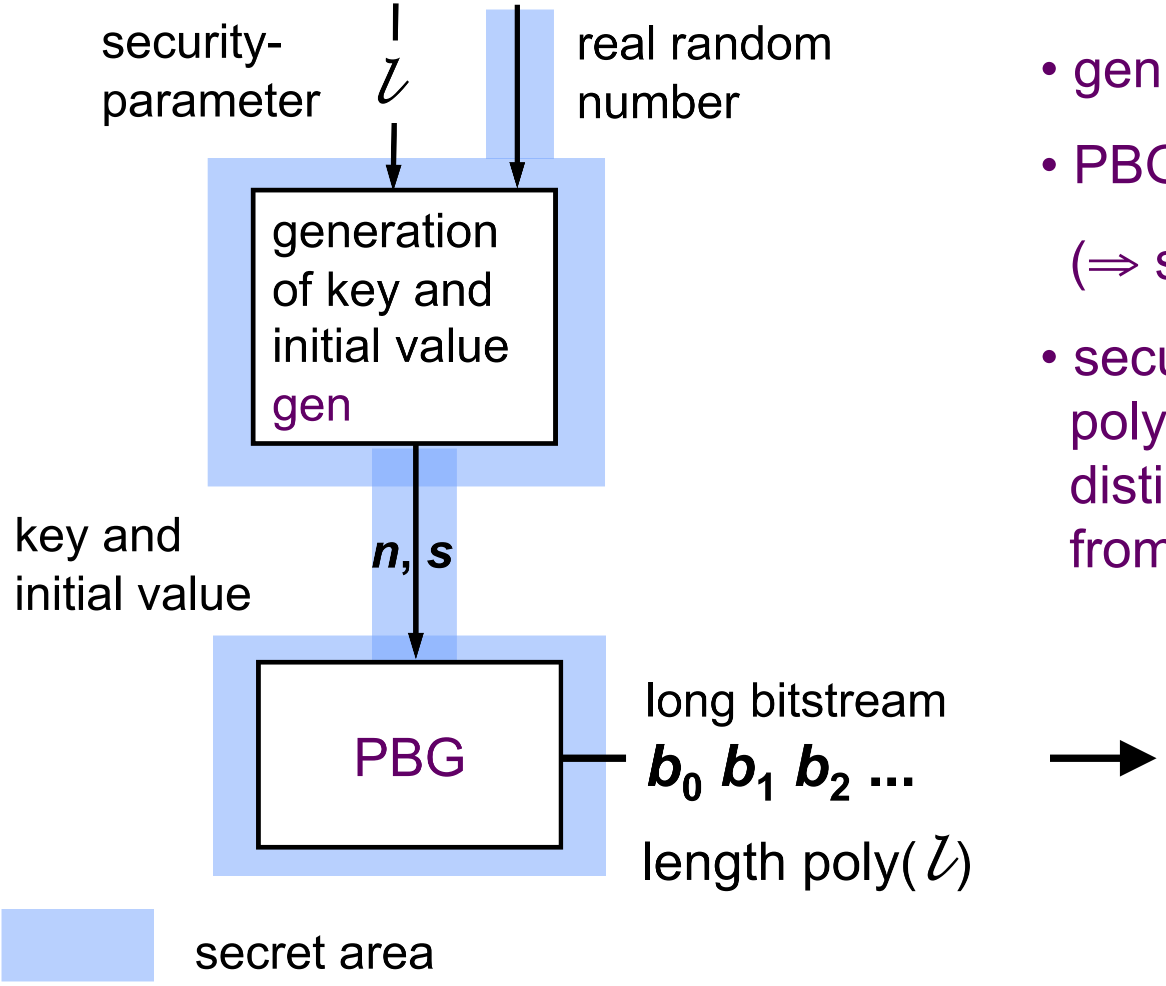
(i.e. it does not work essentially better than to guess)

QRA = quadratic residuosity assumption

The s^2-mod-n-Pseudo-random Bitstream Generator (PBG)

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

Scheme:



security-parameter ℓ

real random number

generation of key and initial value gen

key and initial value

$\boldsymbol{n, s}$

PBG

long bitstream $\boldsymbol{b_0\ b_1\ b_2 \ldots}$

length poly(ℓ)

🟦 secret area

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

Method

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

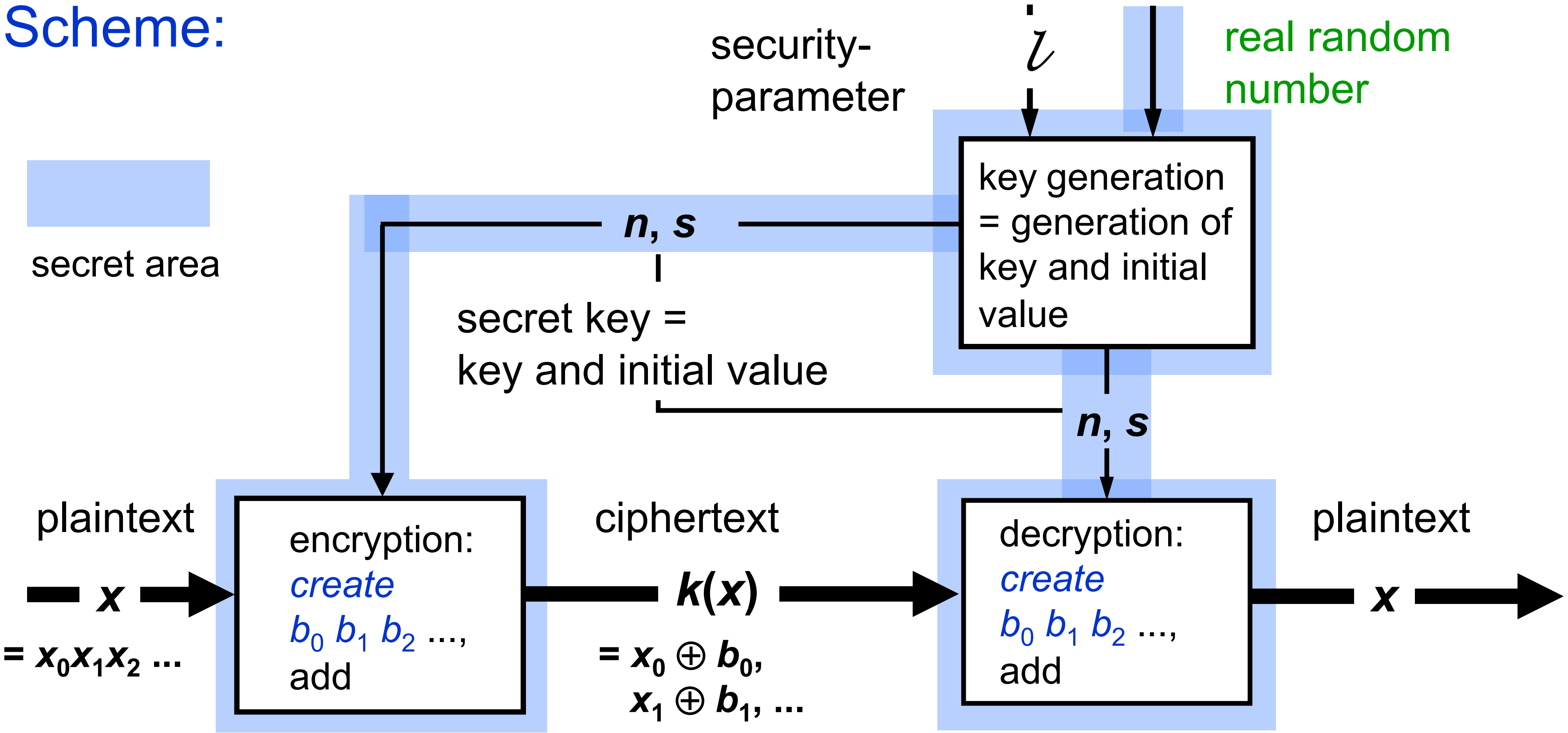
(Blum / Blum / Shub 1983 / 86)

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:



security-parameter
l
real random number
key generation = generation of key and initial value
n, s
secret key = key and initial value
n, s
secret area
plaintext
x
= $x_0x_1x_2$ ...
encryption: create b_0 b_1 b_2 ..., add
ciphertext
k(x)
= $x_0 \oplus b_0$, $x_1 \oplus b_1$, ...
decryption: create b_0 b_1 b_2 ..., add
plaintext
x

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

Idea:

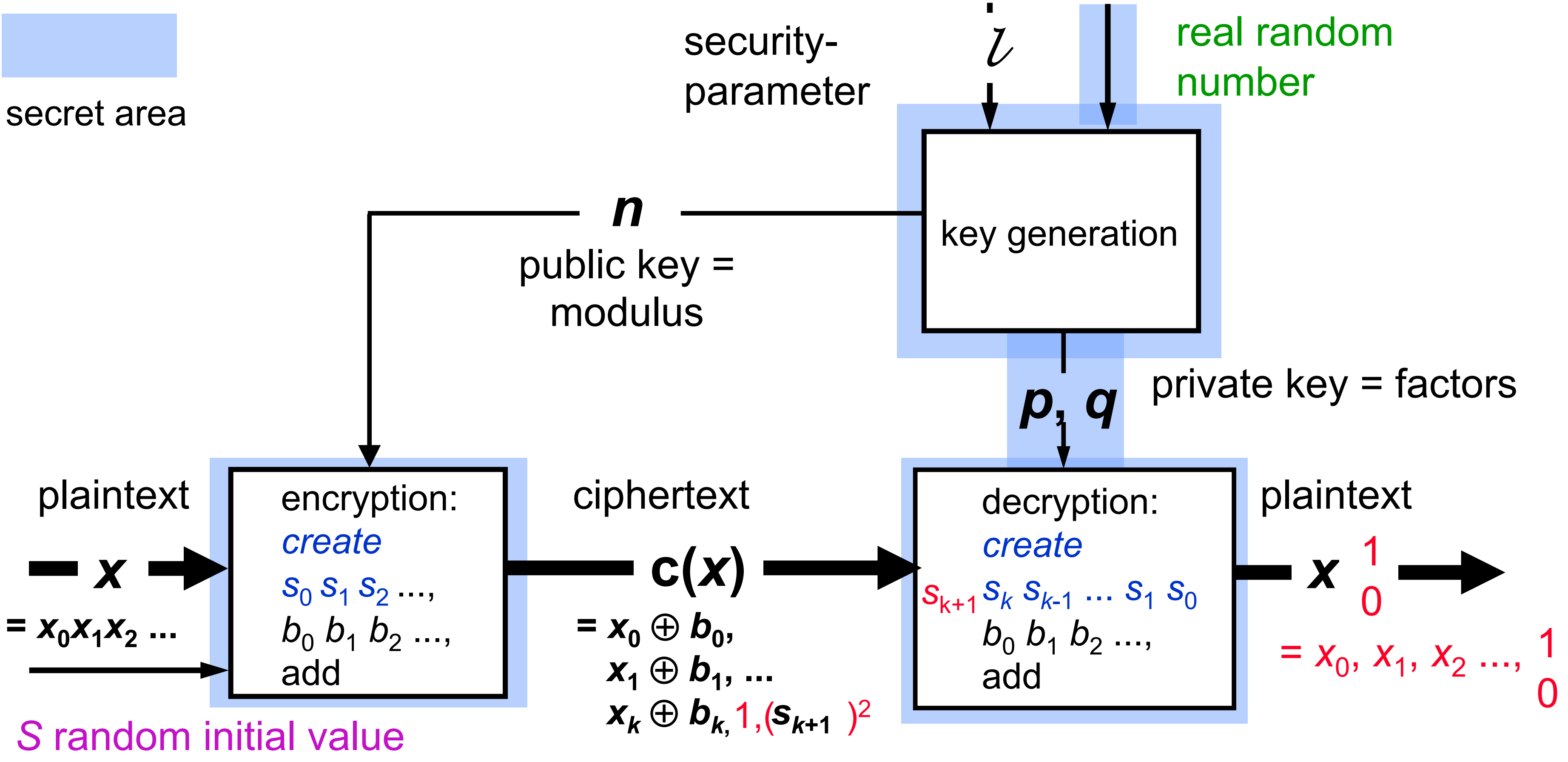
If no probabilistic polynomial test can distinguish pseudo-random streams from real random streams, then the pseudo one-time pad is as good as the one-time pad against polynomial attacker.

(Else the attacker is a test !)

Construction works with any good PBG

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

chosen ciphertext-plaintext attack



secret area

security-parameter

l

real random number

key generation

n

public key = modulus

p, q

private key = factors

plaintext

x

$= x_0 x_1 x_2 \dots$

S random initial value

encryption: *create* $s_0\ s_1\ s_2 \dots,$ $b_0\ b_1\ b_2 \dots,$ add

ciphertext

$c(x)$

$= x_0 \oplus b_0,$
$x_1 \oplus b_1, \dots$
$x_k \oplus b_k, 1, (s_{k+1})^2$

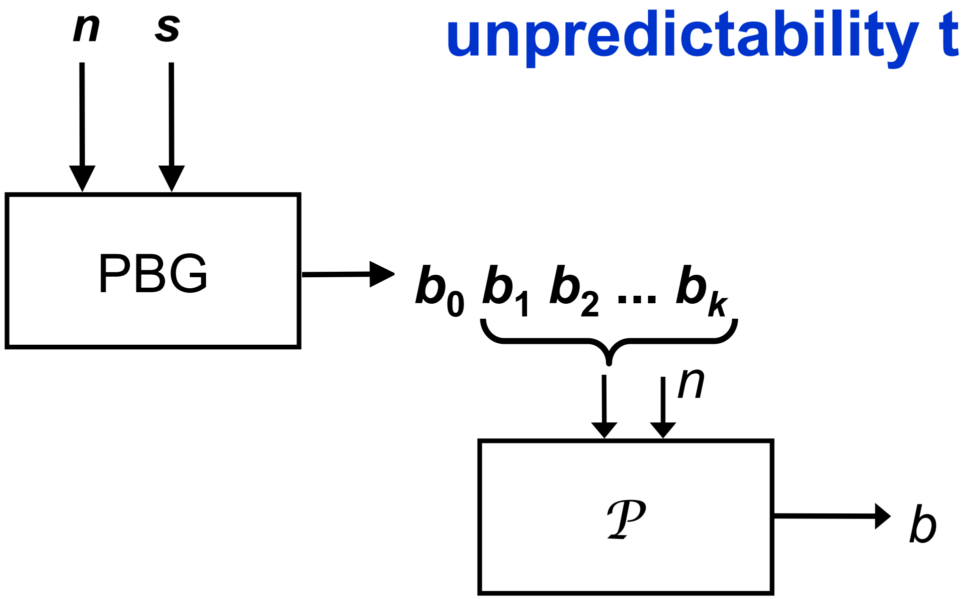
decryption: *create* $s_{k+1}\ s_k\ s_{k-1} \dots s_1\ s_0$ $b_0\ b_1\ b_2 \dots,$ add

plaintext

$x\ \begin{matrix}1\\0\end{matrix}$

$= x_0, x_1, x_2 \dots, \begin{matrix}1\\0\end{matrix}$

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



unpredictability to the left will do

s^2-mod-n-generator is cryptographically strong: ⇔

$\forall\ \mathcal{P} \in$ PPA	{ predictor for b_0 }
$\forall$ constants δ, $0 < \delta < 1$	{ frequency of the “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

$$W(b_0 = \mathcal{P}(n, b_1 b_2 \ldots b_k) \mid s \in Z_n^* \text{ random}) < \frac{1}{2} + \frac{1}{\ell^t}$$

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(l)) in place of 1 bit per squaring.
2) There is a more difficult proof that s^2-mod-n-generator is secure under the factoring assumption.

Security of PBGs more precisely (1)

Requirements for a PBG:

“strongest” requirement: PBG passes *each* probabilistic Test T with polynomial running time.

pass = streams of the PBG cannot be distinguished from real random bit stream with significant probability by any probabilistic test with polynomial running time.

probabilistic test with polynomial running time = probabilistic polynomial-time restricted algorithm that assigns to each input of $\{0,1\}^*$ a real number of the interval $[0,1]$. (value depends in general on the sequence of the random decisions.)

Let α_m be the average (with respect to an even distribution) value, that T assigns to a random m-bit-string.

Security of PBGs more precisely (2)

PBG passes $\mathcal{T}$ iff

For all $t > 0$, for sufficiently big l the average (over all initial values of length l), that $\mathcal{T}$ assigns to the poly(l)-bit-stream generated by the PBG, is in $\alpha_{\text{poly}(l)} \pm 1/l^t$

To this “strongest” requirement, the following 3 are equivalent (but easier to prove):

For each generated finite initial bit string, of which any (the rightmost, leftmost) bit is missing, each polynomial-time algorithm $\mathcal{P}$ (predictor) can “only guess” the missing bit.

Idea of proof: From each of these 3 requirements follows the “strongest”

easy: construct test from predictor

hard: construct predictor from test

Security of PBGs more precisely (3)

Proof (indirect): Construct predictor $\mathcal{P}$ from the test $\mathcal{T}$.

For a t>0 and infinitely many ℓ the average (over all initial values of length ℓ), that $\mathcal{T}$ assigns to the generated poly(ℓ)-bit-string of the PBG is (e.g. above) $\alpha_{\text{poly}(\ell)} \pm 1/\ell^t$. Input to $\mathcal{T}$ a bit string of 2 parts: $j+k=\text{poly}(\ell)$

real random

A={r_1 ... r_j r_{j+1} b_1 ... b_k} are assigned a value closer to $\alpha_{\text{poly}(\ell)}$

B={r_1 ... r_j $\underline{b_0\ b_1\ \dots\ b_k}$} are assigned a value more distant to $\alpha_{\text{poly}(\ell)}$,

generated by PBG e.g. higher

Predictor for bit string b_1 ... b_k constructed as follows:

$\mathcal{T}$ on input {r_1 ... r_j 0 b_1 ... b_k} estimate α^0

$\mathcal{T}$ on input {r_1 ... r_j 1 b_1 ... b_k} estimate α^1

Guess b_0 = 0 with probability of 1/2 + 1/2 (α^0- α^1)

(more precisely: L. Blum, M. Blum, M. Shub: A simple unpredictable Pseudo-Random Number Generator; SIAM J. Comput. 15/2 (May 1986) page 375f)

Summary of PBG and motivation of GMR

Reminder:

s^2-mod-n-generator is secure against passive attackers for arbitrary distributions of messages

→ reason for arrow: random number’ in picture asymmetric encryption systems

→ memorize term: probabilistic encryption

Terms:

one-way function

one-way permutation

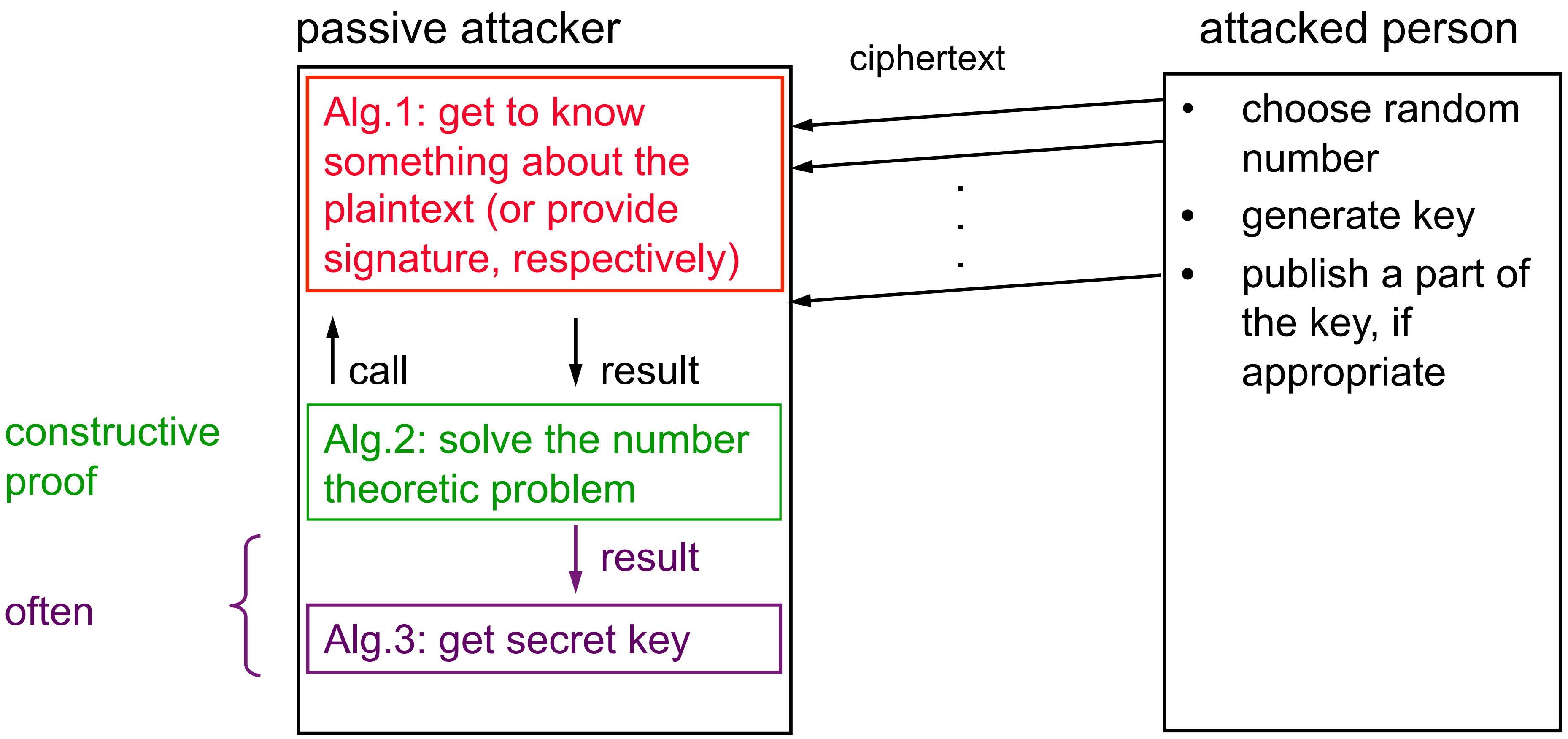
one-way = nearly nowhere practically invertible

variant: invertible with secret (trap door)

Motivation:

active attack on s^2-mod-n-generator as asymmetric encryption system

Scheme of security proofs (1)



passive attacker

Alg.1: get to know something about the plaintext (or provide signature, respectively)

call

result

Alg.2: solve the number theoretic problem

result

Alg.3: get secret key

constructive proof

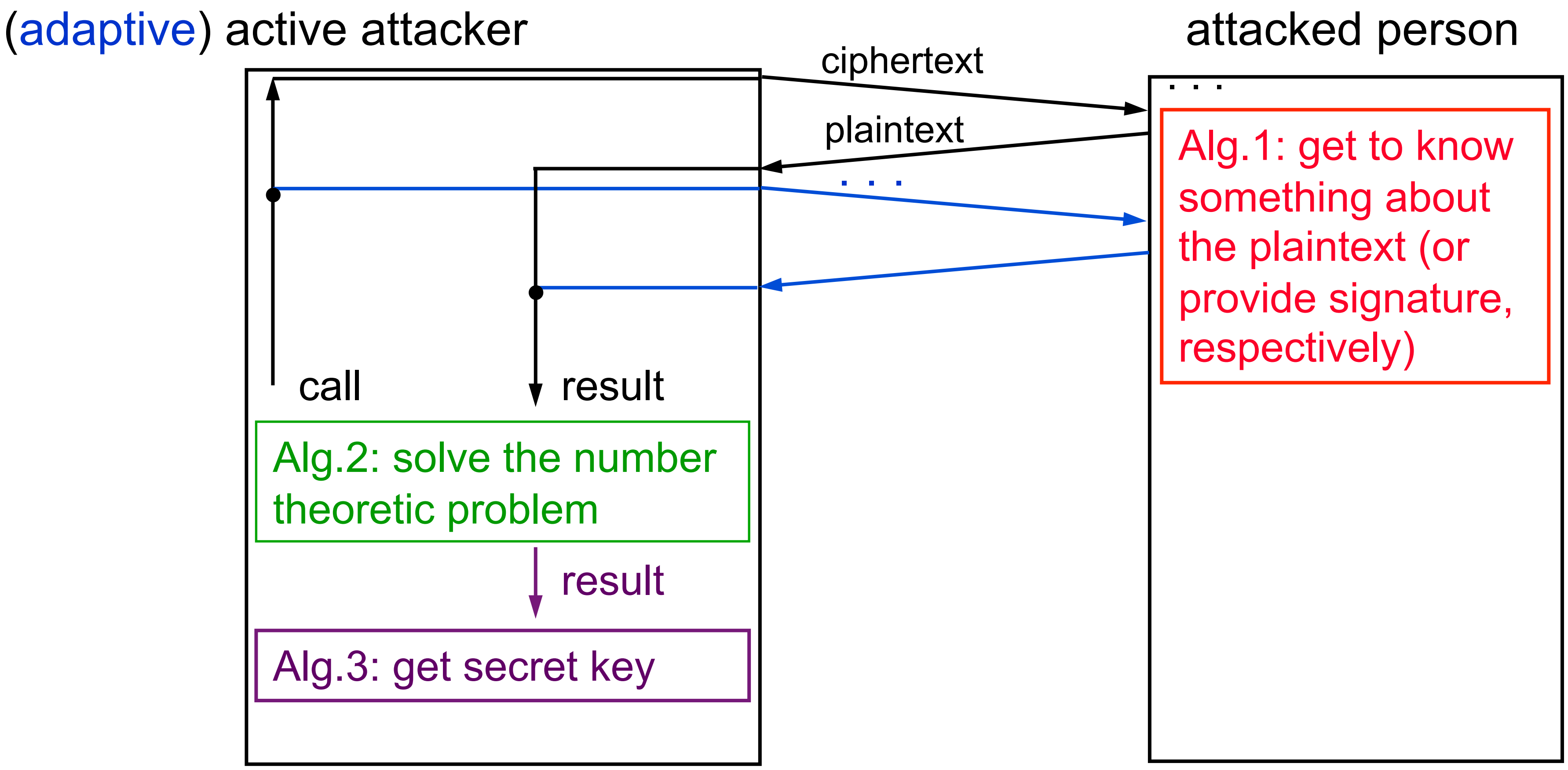
often

ciphertext

attacked person

- choose random number
- generate key
- publish a part of the key, if appropriate

Scheme of security proofs (2)


(adaptive) active attacker
attacked person
ciphertext
plaintext
. . .
. . .
call
result
Alg.1: get to know something about the plaintext (or provide signature, respectively)
Alg.2: solve the number theoretic problem
result
Alg.3: get secret key

Seemingly, there are no provably secure cryptosystems against adaptive active attacks.

A constructive security proof seems to be a game with fire.

Why fallacy ?

attacker

Alg.1: uniform for any key

Alg.2: has to demand uniformity

attacked person

Alg.1: non uniform: only own key

GMR – signature system

Shafi Goldwasser, Silvio Micali, Ronald Rivest:
A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks;
SIAM J. Comput. 17/2 (April 1988) 281 – 308

Main ideas

1) Map a randomly chosen reference $\mathcal{R}$, which is only used once.
2) Out of a set of collision-resistant permutations (which are invertible using a secret) assign to any message *m* one permutation.

$$\begin{tikzcd}
\mathcal{R} \arrow[r, shift left, "\mathcal{F}^{-1}_{n,m}\ (\mathcal{R})"] & \mathrm{Sig}^{\mathcal{R}}_{m} \arrow[l, shift left, "\mathcal{F}_{n,m}\ (\mathrm{Sig}^{\mathcal{R}}_{m})"]
\end{tikzcd}$$

GMR – signature system (1)

Consequence

“variation of *m*” (active attack) now means also a “variation of $\mathcal{R}$” – a randomly chosen reference, that is unknown to the attacker when he chooses *m*.

Problems

1) securing the originality of the randomly chosen reference
2) construction of the collision-resistant permutations (which are invertible only using the secret) which depend on the messages

Solution of problem 2

Idea Choose 2 collision-resistant permutations f_0, f_1 (which are invertible only using the secret) and compose $\mathcal{F}_{n,m}$ by f_0, f_1.
{for simplicity, we will write f_0 instead of $f_{n,0}$ and f_1 instead of $f_{n,1}$}

Def. Two permutations f_0,f_1 are called collision-resistant iff it is difficult to find any x,y,z with $f_0(x) = f_1(y) = z$

Note Proposition: collision-resistant ⇒ one-way
Proof (indir.): If f_i isn't one-way: 1) choose x; 2) $f_{1-i}(x) = z$; 3) $f_i^{-1}(z) = y$

$$\begin{tikzcd} & z & \\ x \arrow[ur, "f_0"] & & y \arrow[ul, "f_1"'] \end{tikzcd}$$

$$\begin{tikzcd} & z \arrow[dr, "3)\ f_i^{-1}"] & \\ 1)\ x \arrow[ur, "2)\ f_{1-i}"] & & y \end{tikzcd}$$

GMR – signature system (2)

Construction:

For $m = b_0 b_1 \ldots b_k$ $(b_0,\ldots,b_k \in \{0,1\})$ let

$$\mathcal{F}_{n,m} := f_{b_0} \circ f_{b_1} \circ \ldots \circ f_{b_k}$$

$$\mathcal{F}^{-1}_{n,m} := f^{-1}_{b_k} \circ \ldots \circ f^{-1}_{b_1} \circ f^{-1}_{b_0}$$

Signing: $\mathcal{R} \xrightarrow{f^{-1}_{b_0}} f^{-1}_{b_0}(\mathcal{R}) \xrightarrow{f^{-1}_{b_1}} \ldots \xrightarrow{f^{-1}_{b_k}} f^{-1}_{b_k}(\ldots(f^{-1}_{b_0}(\mathcal{R}))\ldots) =: \mathrm{Sig}^{\mathcal{R}}_{m}$

Testing: $\mathrm{Sig}^{\mathcal{R}}_{m} \xrightarrow{f_{b_k}} f_{b_k}(\mathrm{Sig}^{\mathcal{R}}_{m}) \xrightarrow{f_{b_{k-1}}} \ldots \xrightarrow{f_{b_0}} f_{b_0}(\ldots(f_{b_k}(\mathrm{Sig}^{\mathcal{R}}_{m}))\ldots) = \mathcal{R}$?

Example:

$$\mathrm{Sig}^{\mathcal{R}}_{\mathbf{1110}} \bullet \xrightarrow{f_0} \bullet \xrightarrow{f_1} \bullet \xrightarrow{f_1} \bullet \xrightarrow{f_1} \bullet \mathcal{R}$$

GMR – signature system (3)

Problem: intermediate results of the tests are valid signatures for the start section of the message *m*

Idea: coding the message prefix free

Def. A mapping <•>: M → M is called prefix free

iff $\forall\, m_1,m_2 \in \mathrm{M}: \forall\, b \in \{0,1\}^+: <m_1>b \neq <m_2>$

<•> injective

Example for a prefix free mapping

0 → 00 ; 1 → 11 ; end identifier 10

Prefix-free encoding should be efficient to calculate both ways.

To factor is difficult (1)

Theorem: If factoring is difficult, then collision-resistant permutation pairs exist

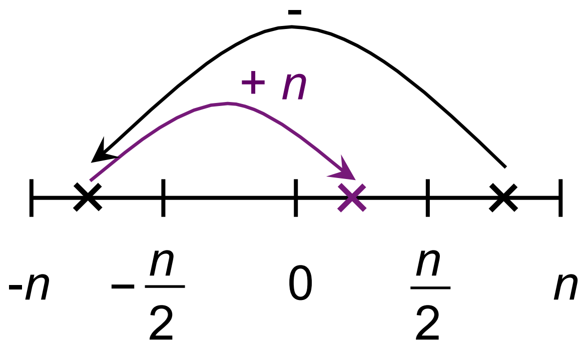
Proof: secret: $p \bullet q = n$; $p \equiv_8 3$ und $q \equiv_8 7$ (Blum numbers)

it holds: $\left(\frac{-1}{n}\right) = 1$ $\quad -1 \notin QR_n$

$\left(\frac{2}{n}\right) = -1$

$$f_0(x) := \begin{cases} x^2 \bmod n \text{ , if } < \frac{n}{2} \\ -x^2 \bmod n \text{ , else} \end{cases}$$

$$f_1(x) := \begin{cases} (2x)^2 \bmod n\text{, if } < \frac{n}{2} \\ -(2x)^2 \bmod n\text{, else} \end{cases}$$



Domain : $\{x \in Z_n{}^* \mid \left(\frac{x}{n}\right) = 1 \text{ , } 0 < x < \frac{n}{2} \}$

To factor is difficult (2)

to show : 1) Permutation = one-to-one mapping with co-domain = domain

2) To calculate the inverse is easy using p,q

3) If there is a fast collision finding algorithm, then there is a fast algorithm to factor.

$-1 \notin QR_n$

$x^2 \equiv_n -(2y)^2$ cannot hold, since $(2y)^2 \in QR_n$.

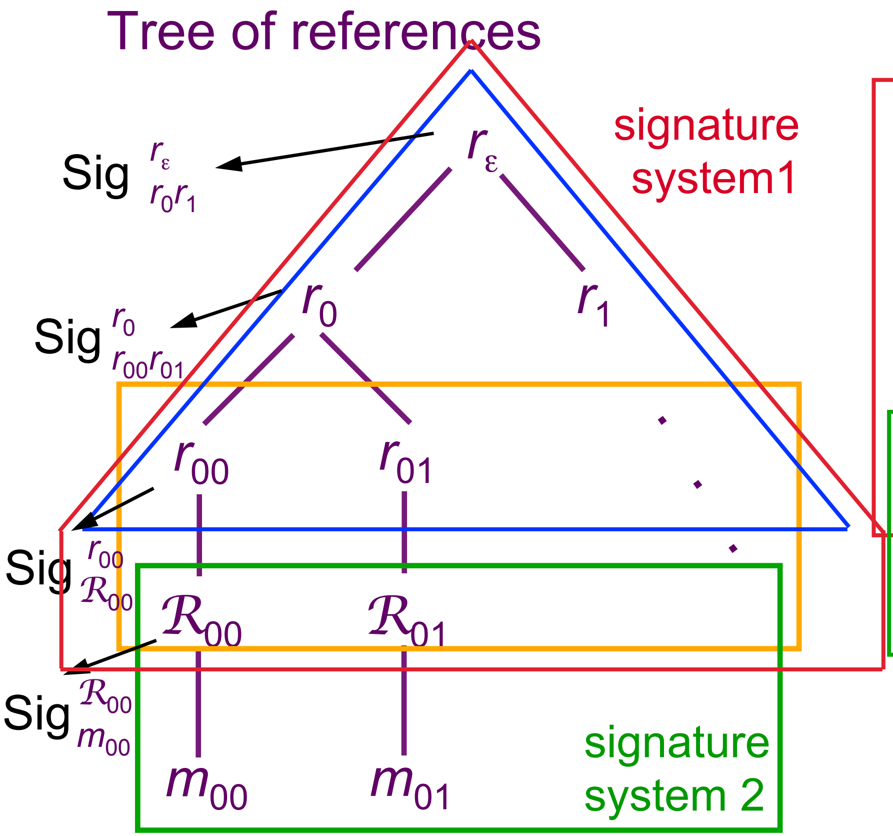
Therefore $x^2 \equiv_n (2y)^2 \Rightarrow (x+2y)(x-2y) \equiv_n 0$.

Because $\left(\frac{x}{n}\right) = 1$ and $\left(\frac{\pm 2y}{n}\right) = -1$ it follows that

$x \not\equiv_n \pm 2y$

Therefore gcd ($x \pm 2y, n$) provides a non-trivial factor of n, i.e. p or q.

Solution of problem 1 (1)


Tree of references
$\text{Sig}^{r_\varepsilon}_{r_0 r_1}$
r_ε
signature system1
$\text{Sig}^{r_0}_{r_{00} r_{01}}$
r_0
r_1
r_{00}
r_{01}
$\text{Sig}^{r_{00}}_{\mathcal{R}_{00}}$
$\mathcal{R}_{00}$
$\mathcal{R}_{01}$
$\text{Sig}^{\mathcal{R}_{00}}_{m_{00}}$
m_{00}
m_{01}
signature system 2

generate (≈ sign)

$$\text{Sig}^{r_j}_{r_{j0} r_{j1}} = \mathcal{F}^{-1}_{n,<r_{j0} r_{j1}>}(r_j)$$

$$\text{Sig}^{r_i}_{\mathcal{R}_i} = \mathcal{F}^{-1}_{n,<\mathcal{R}_i>}(r_i)$$

signature system 1 no active attack

reference $\mathcal{R}_i$

$$\text{Sig}^{\mathcal{R}_i}_{m_i} = \mathcal{F}^{-1}_{n',<m_i>}(\mathcal{R}_i)$$

probabilistic signature system 2

test

$$\mathcal{F}_{n,<r_{j0} r_{j1}>}(\text{Sig}^{r_j}_{r_{j0} r_{j1}}) = r_j \,?$$

$$\mathcal{F}_{n,<\mathcal{R}_i>}(\text{Sig}^{r_i}_{\mathcal{R}_i}) = r_i \,?$$

$$\mathcal{F}_{n',<m_i>}(\text{Sig}^{\mathcal{R}_i}_{m_i}) = \mathcal{R}_i \,?$$

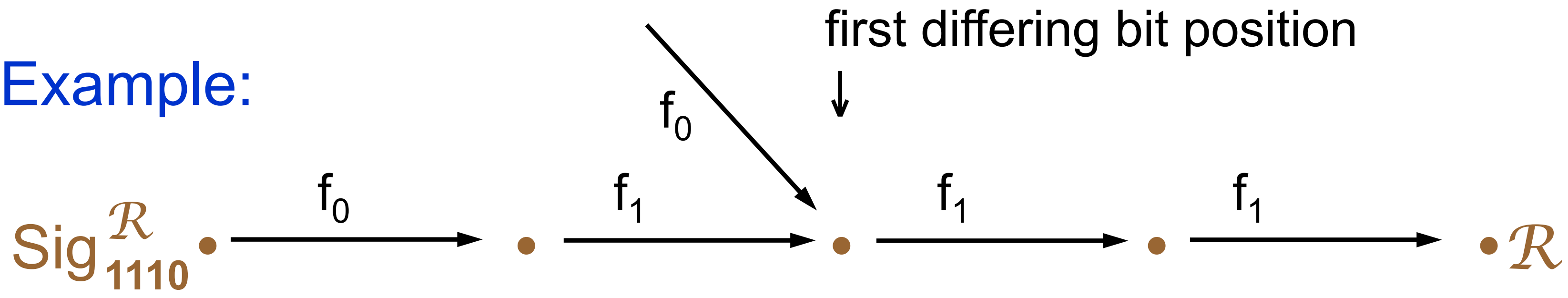
The attacker gets to know $\mathcal{R}_i$ only after choosing m_i.

Solution of problem 1 (2)

Proposition If the permutation pairs are collision resistant, then the adaptive active attacker can’t sign any message with GMR.

Proof A forged signature leads either to a collision in the tree of references (contradiction) or to an additional legal signature. So the attacker has inverted the collision-resistant permutation. With this ability he could generate collisions (contradiction).

Example:


first differing bit position
f_0
$\mathrm{Sig}^{\mathcal{R}}_{1110}$
f_0
f_1
f_1
f_1
$\mathcal{R}$

Note

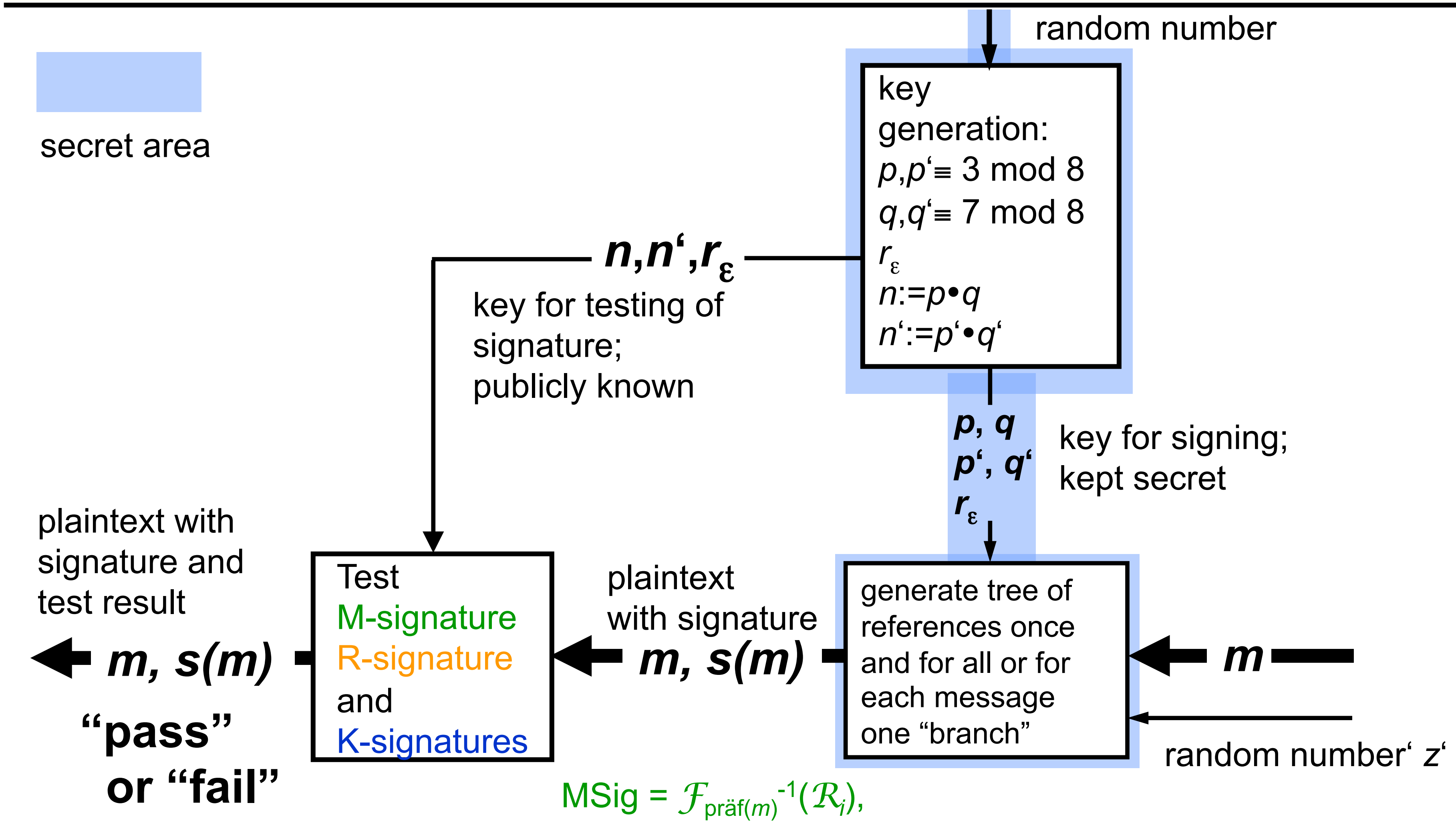
In the proof you dispose the “Oracle” (the attacked person) by showing that the attacker can generate „half“ the tree from the bottom or (exclusive) “half” the tree from the top with the same probability distribution as the attacked person.

Lesson:

randomly chosen references each used only once (compare one-time-pad) make adaptive active attacks ineffective

→ arrow explained (random number z') in figure signature system

GMR signature system



MSig = $\mathcal{F}_{\text{präf}(m)}^{-1}(\mathcal{R}_i)$,

RSig = $\mathcal{F}_{\text{präf}(\mathcal{R}_i)}^{-1}(r_i)$,

KSig = $\mathcal{F}_{\text{präf}(r_i|\bullet)}^{-1}(r_{i-1})$, …
$\mathcal{F}_{\text{präf}(r_i|r_1)}^{-1}(r_\varepsilon)$

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 $|p| \approx |q| = \ell$, $p \neq q$
2) Calculate $n := p \bullet q$
3) Choose c with $3 \leq c < (p\text{-}1)(q\text{-}1)$ and $\gcd\left(c, \underbrace{(p\text{-}1)(q\text{-}1)}_{\Phi(n)}\right) = 1$
4) Calculate d using p, q, c as multiplicative inverse of c mod $\Phi(n)$

   $c \bullet d \equiv 1 \ \left(\text{mod } \Phi(n)\right)$
5) Publish c and n.

En- / decryption

exponentiation with c respectively d in Z_n

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

Proof (1)

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

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

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

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

Using the Theorem of Fermat
$\forall m \in Z_n^*: m^{\Phi(n)} \equiv 1 \quad (\mathrm{mod}\ n)$

it follows for all m coprime to p

$$m^{p\text{-}1} \equiv 1 \quad (\mathrm{mod}\ p)$$

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

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

Because congruence holds relating to p as well as q, according to the CRA, it holds relating to $p \cdot q = n$.

Therefore, for all $m \in Z_n$

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

Attention:

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

RSA is easy to break ⇒ to factor is easy

Naive insecure use of RSA

RSA as asymmetric encryption system

Code message (if necessary in several pieces) as number $m < n$

Encryption of m: $m^c \bmod n$

Decryption of m^c: $(m^c)^d \bmod n = m$

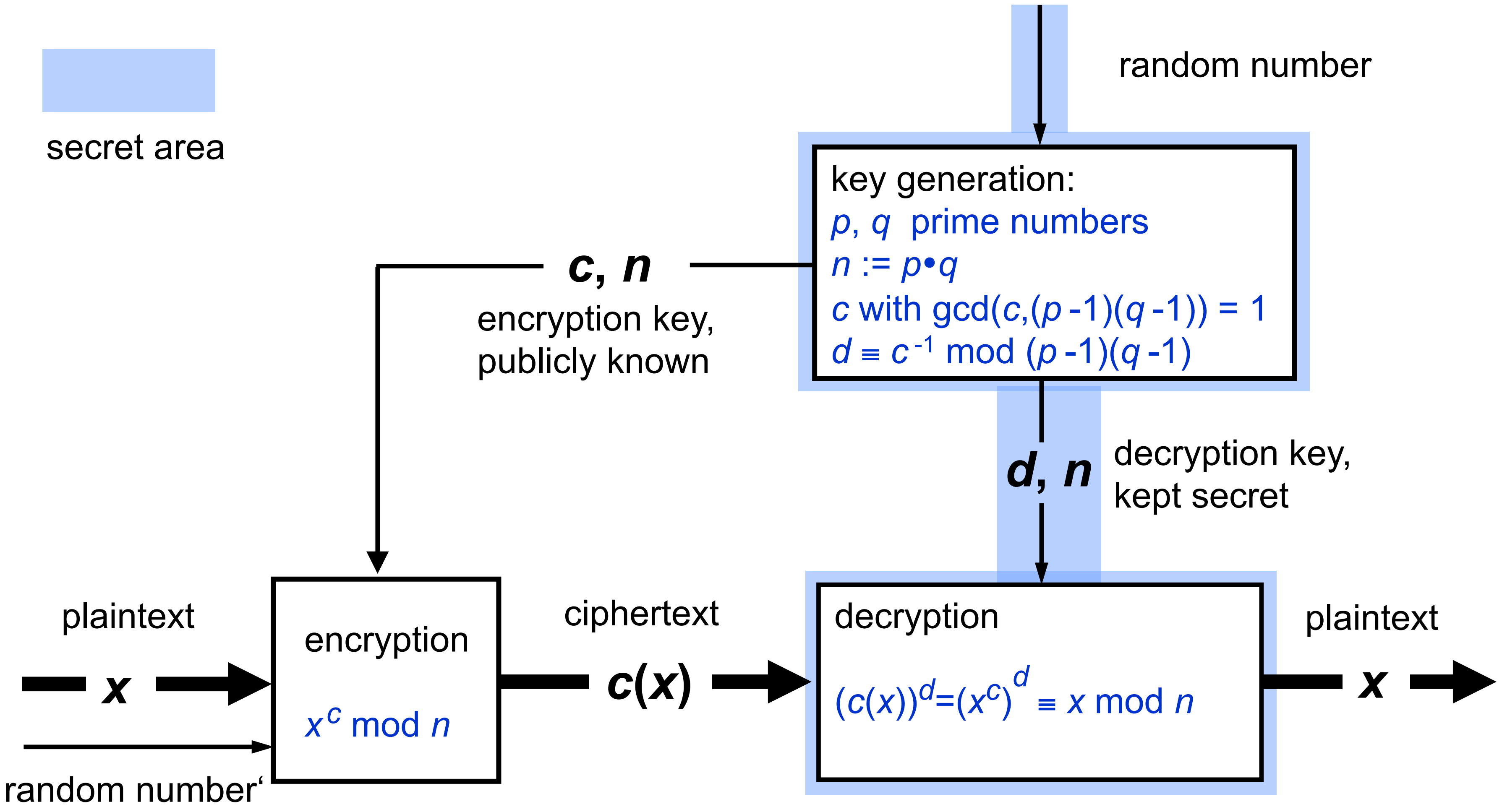
RSA as digital signature system

Renaming: $c \rightarrow t$, $d \rightarrow s$

Signing of m: $m^s \bmod n$

Testing of m, m^s: $(m^s)^t \bmod n = m$?

RSA as asymmetric encryption system: naive



secret area

random number

key generation:
p, q prime numbers
$n := p \bullet q$
c with $\gcd(c,(p-1)(q-1)) = 1$
$d \equiv c^{-1} \bmod (p-1)(q-1)$

c, n
encryption key,
publicly known

d, n
decryption key,
kept secret

plaintext
x
random number‘

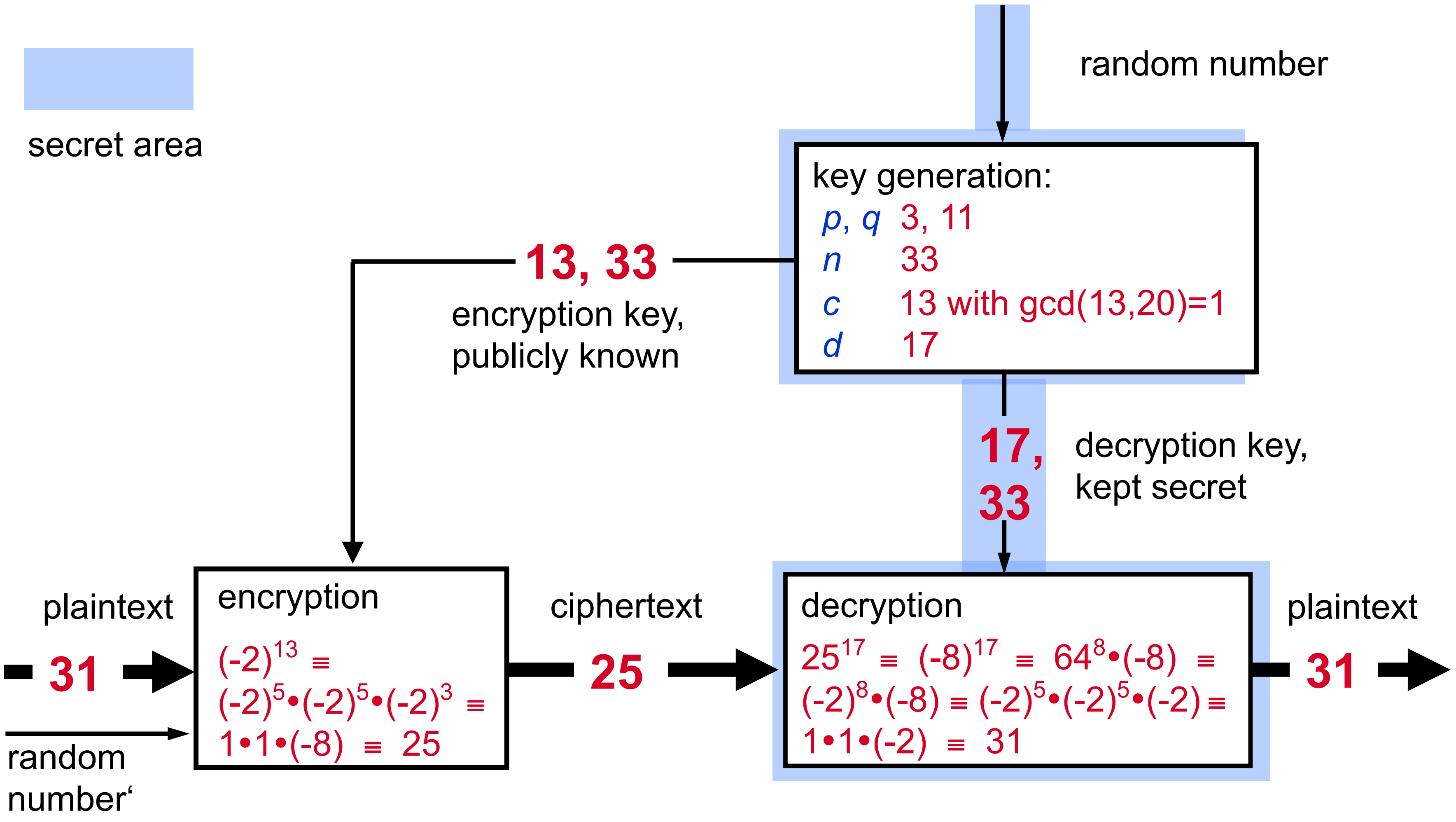
encryption
$x^c \bmod n$

ciphertext
$c(x)$

decryption
$(c(x))^d = (x^c)^d \equiv x \bmod n$

plaintext
x

RSA as asymmetric encryption system: example



secret area

random number

key generation:

p, *q*	3, 11
n	33
c	13 with gcd(13,20)=1
d	17

13, 33

encryption key, publicly known

17, 33

decryption key, kept secret

plaintext

31

random number‘

encryption

$(-2)^{13} \equiv$
$(-2)^5 \bullet (-2)^5 \bullet (-2)^3 \equiv$
$1 \bullet 1 \bullet (-8) \equiv 25$

ciphertext

25

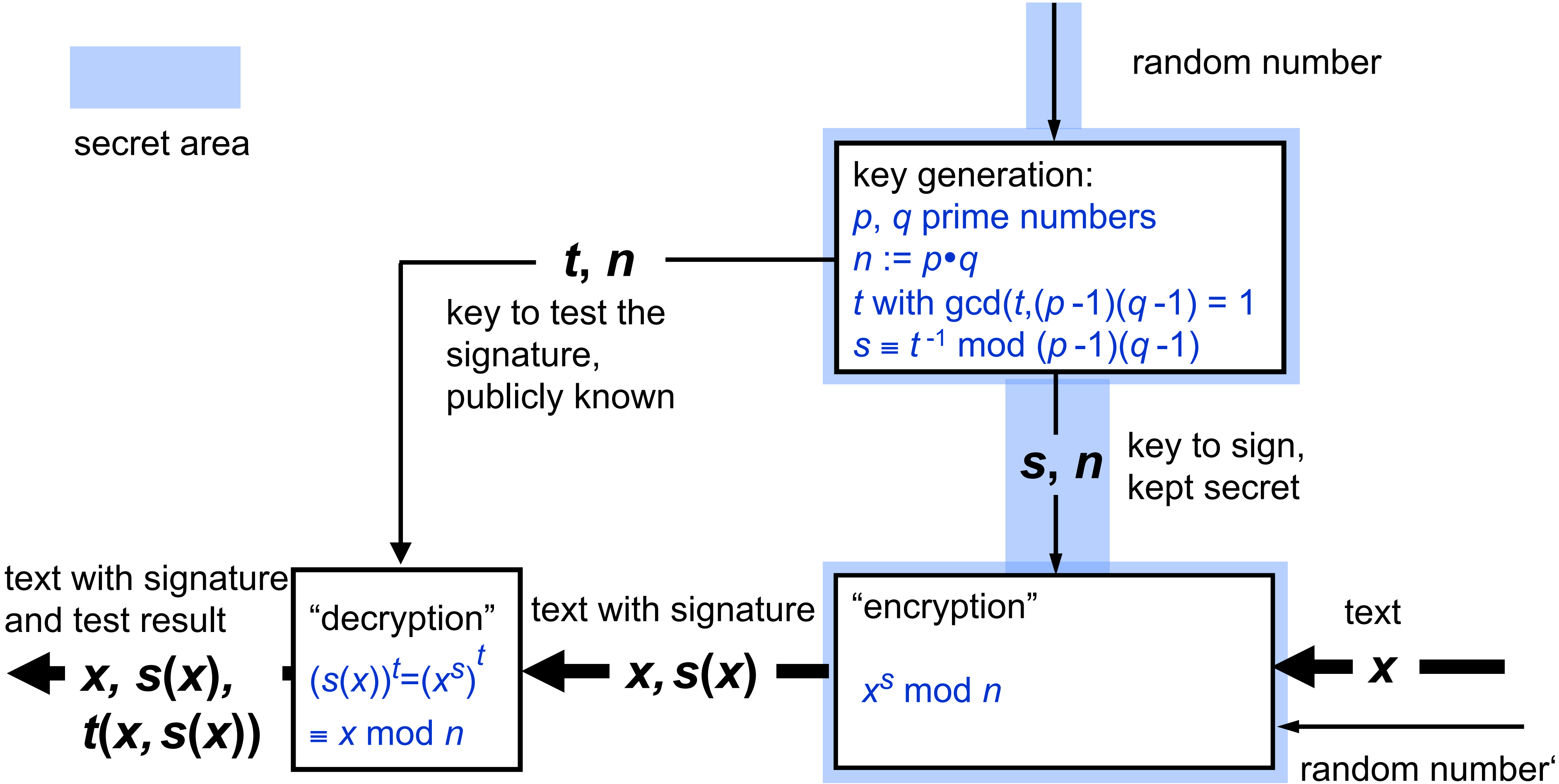
decryption

$25^{17} \equiv (-8)^{17} \equiv 64^8 \bullet (-8) \equiv$
$(-2)^8 \bullet (-8) \equiv (-2)^5 \bullet (-2)^5 \bullet (-2) \equiv$
$1 \bullet 1 \bullet (-2) \equiv 31$

plaintext

31

RSA as digital signature system: naive



secret area

random number

key generation:
p, q prime numbers
$n := p \bullet q$
t with $\gcd(t,(p-1)(q-1)) = 1$
$s \equiv t^{-1} \bmod (p-1)(q-1)$

t, n

key to test the signature, publicly known

s, n

key to sign, kept secret

text

x

random number‘

“encryption”

$x^s \bmod n$

text with signature

$x, s(x)$

“decryption”

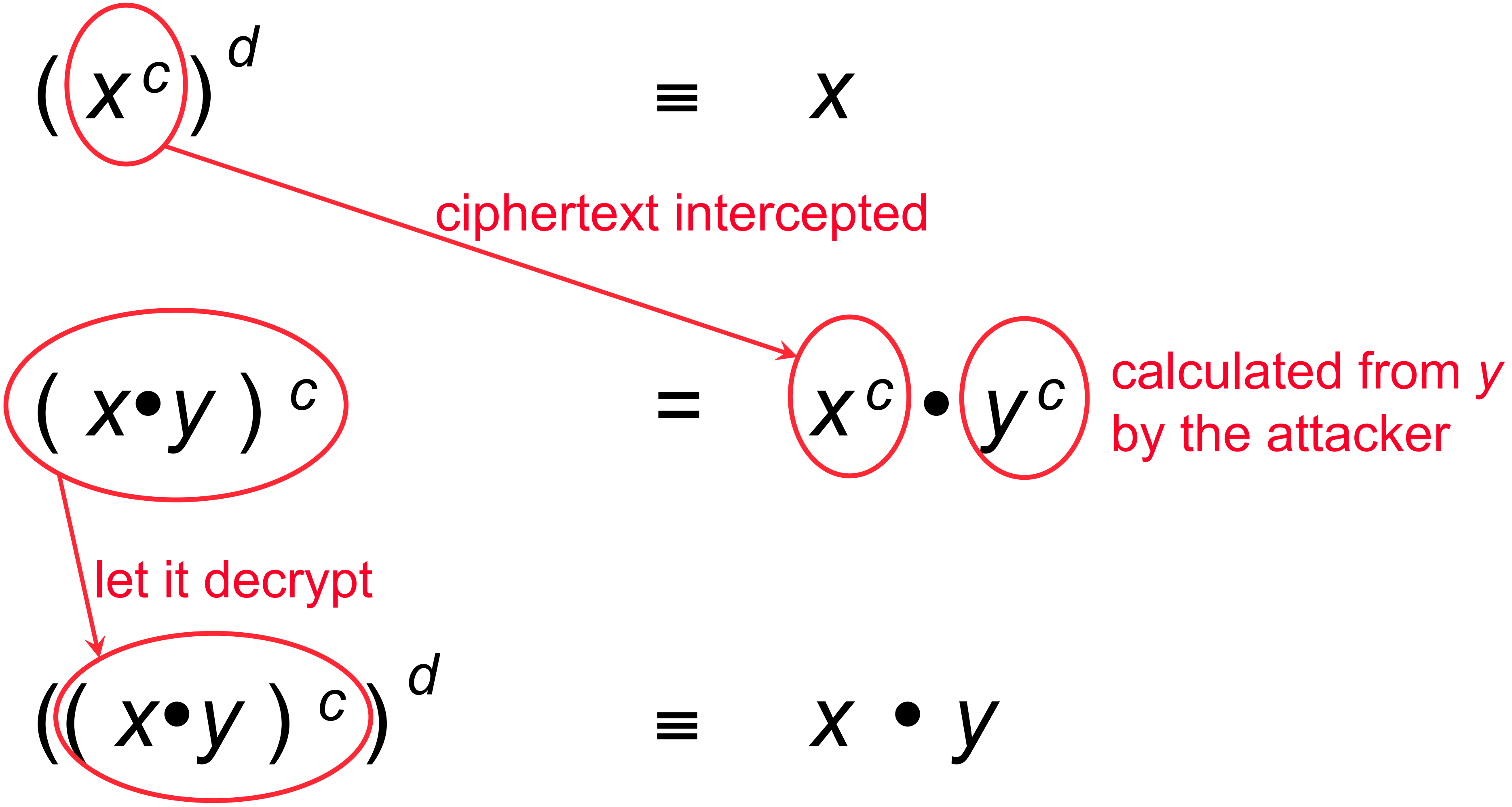
$(s(x))^t = (x^s)^t$

$\equiv x \bmod n$

text with signature and test result

$x, s(x), t(x, s(x))$

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 on digital signature with RSA naive

$$(x^s)^t \equiv x$$

message wanted

$$(x^s \bullet y)^t \equiv x \bullet y^t$$

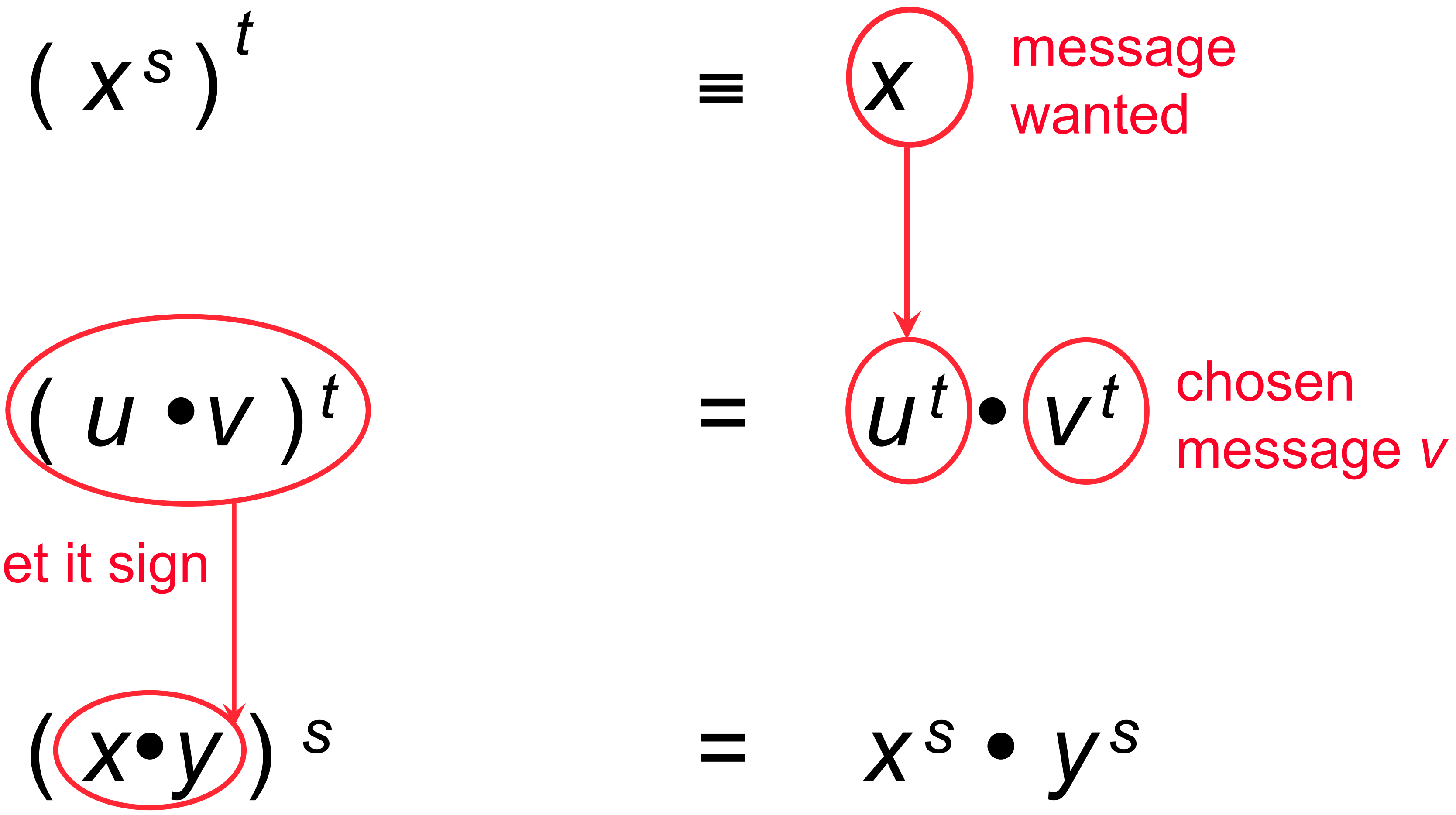
chosen message *y*

let it sign

$$((x^s \bullet y)^t)^s \equiv x^s \bullet y$$

divide by *y*, get x^s

Attack on digital signature with RSA: alternative presentation



$(x^s)^t \equiv x$ message wanted

$(u \bullet v)^t = u^t \bullet v^t$ chosen message *v*

let it sign

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

$= x^s \bullet v$

divide by *v*, get x^s

Transition to Davida’s attacks

simple version of Davida’s attack: (against RSA as signature system)

1. Given $Sig_1 = m_1^s$

   $Sig_2 = m_2^s$

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

   New signature generated !

   (Passive attack, *m* not selectable.)

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

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

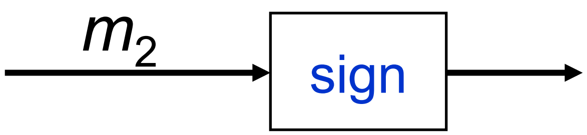
   Let m_1, m_2 be signed.

   Further as mentioned above.

3. Active, more skillful (Moore) {see next transparency}

   “Blinding” : choose any r ,

   $m_2 := m \bullet r^t$



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

$\overset{\bullet\, r^{-1}}{\rightsquigarrow} \quad m^s = Sig$

Active Attack of Davida against RSA

1.) asymmetric encryption system:

Decryption of the chosen message m^c

Attacker chooses random number r, $0 < r < n$
generates $r^c \bmod n$; this is uniformly distributed in $[1, n\text{-}1]$
lets the attacked person decrypt $r^c \bullet m^c \equiv:_n prod$

Attacked person generates $prod^d \bmod n$

Attacker knows that $prod^d \equiv_n (r^c \bullet m^c)^d \equiv_n r^{c \bullet d} \bullet m^{c \bullet d} \equiv_n r \bullet m$
divides $prod^d$ by r and thereby gets m.

If this doesn't work: Factor n.

2.) digital signature system:

Signing of the chosen message m.

Attacker chooses random number r, $0 < r < n$
generate $r^t \bmod n$; this is uniformly distributed in $[1, n\text{-}1]$
lets the attacked person sign $r^t \bullet m \equiv:_n prod$

Attacked person generates $prod^s \bmod n$

Attacker knows that $prod^s \equiv_n (r^t \bullet m)^s \equiv_n r^{t \bullet s} \bullet m^s \equiv_n r \bullet m^s$
divides $prod^s$ by r and thereby gets m^s.

If this doesn't work: Factor n.

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

h() : collision-resistant hash function

1.) asymmetric encryption system

plaintext messages have to fulfill redundancy predicate

m, redundancy ⇒ test if $h(m)$ = redundancy

2.) digital signature system

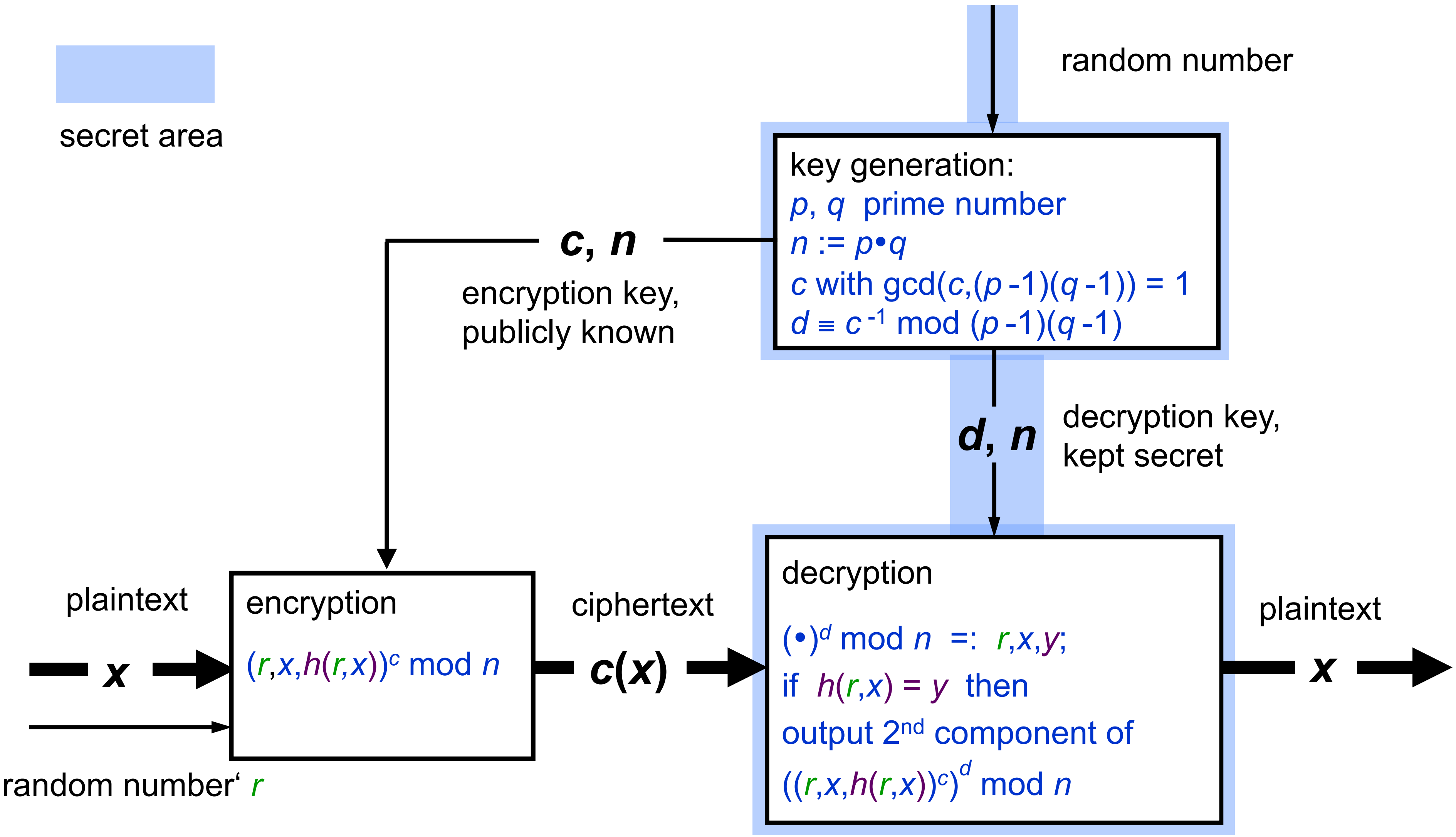
Before signing, h is applied to the message

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

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

Attention: There is no proof of security (so far?)

RSA as asymmetric encryption system



secret area

random number

key generation:
p, q prime number
$n := p \bullet q$
c with $\gcd(c,(p-1)(q-1)) = 1$
$d \equiv c^{-1} \bmod (p-1)(q-1)$

c*, *n

encryption key,
publicly known

d*, *n

decryption key,
kept secret

plaintext

x

encryption

$(r,x,h(r,x))^c \bmod n$

random number‘ r

ciphertext

***c*(*x*)**

decryption

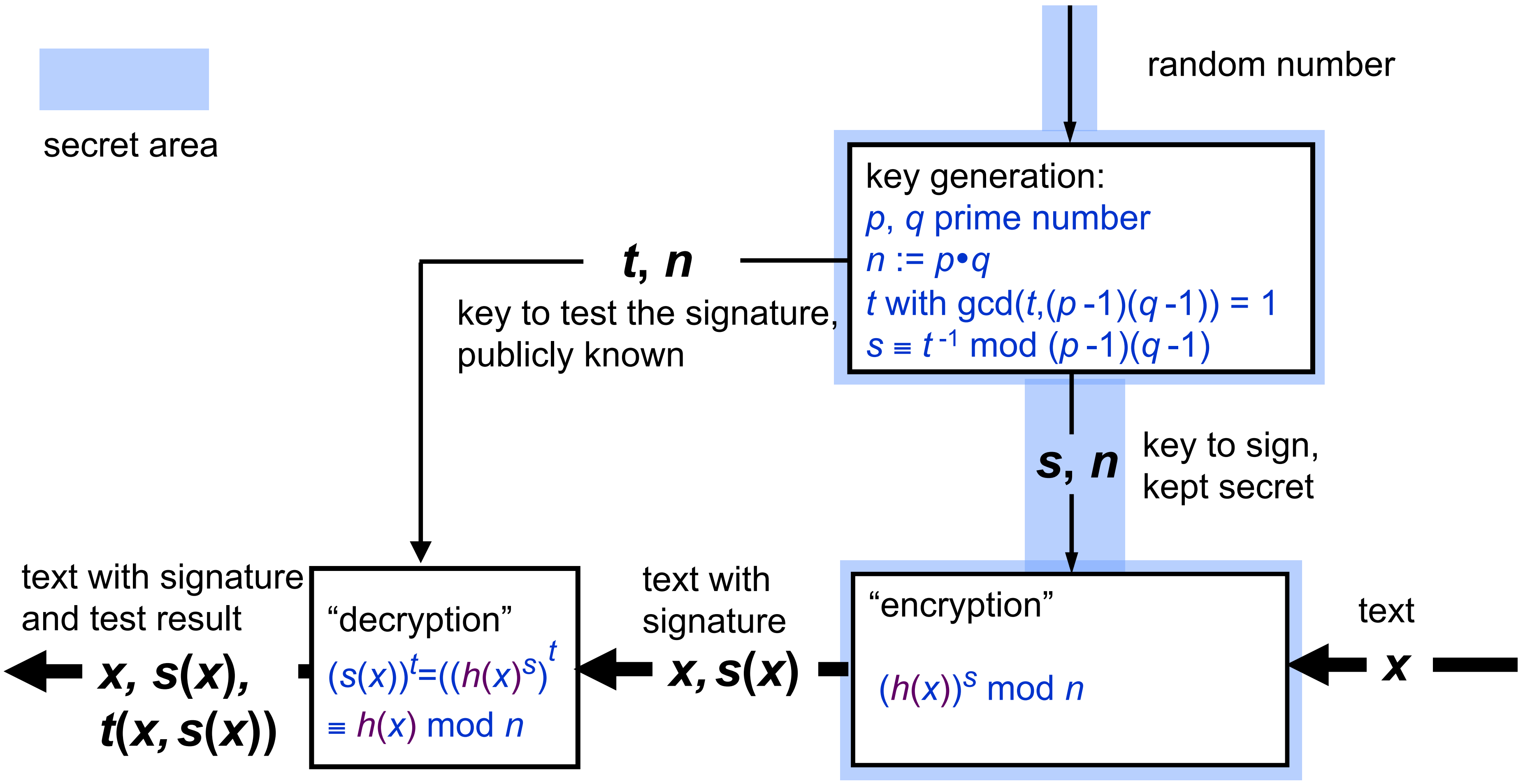
$(\bullet)^d \bmod n \ =: \ r,x,y;$
if $h(r,x) = y$ then
output 2nd component of
$((r,x,h(r,x))^c)^d \bmod n$

plaintext

x

collision-resistant hash function *h*
- globally known -

RSA as digital signature system



secret area

random number

key generation:
p, q prime number
$n := p \bullet q$
t with $\gcd(t,(p-1)(q-1)) = 1$
$s \equiv t^{-1} \bmod (p-1)(q-1)$

t, n

key to test the signature, publicly known

s, n

key to sign, kept secret

text

x

“encryption”

$(h(x))^s \bmod n$

text with signature

$x, s(x)$

“decryption”

$(s(x))^t = ((h(x))^s)^t$
$\equiv h(x) \bmod n$

text with signature and test result

$x, s(x),$
$t(x, s(x))$

collision-resistant hash function h
- globally known -

Faster calculation of the secret operation

$y^d \equiv w$

mod *p*, *q* separately:

once and for all:

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

every time:

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

proof:

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

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

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 $\approx \ell$

So: $\approx$ Factor 4

irrelevant

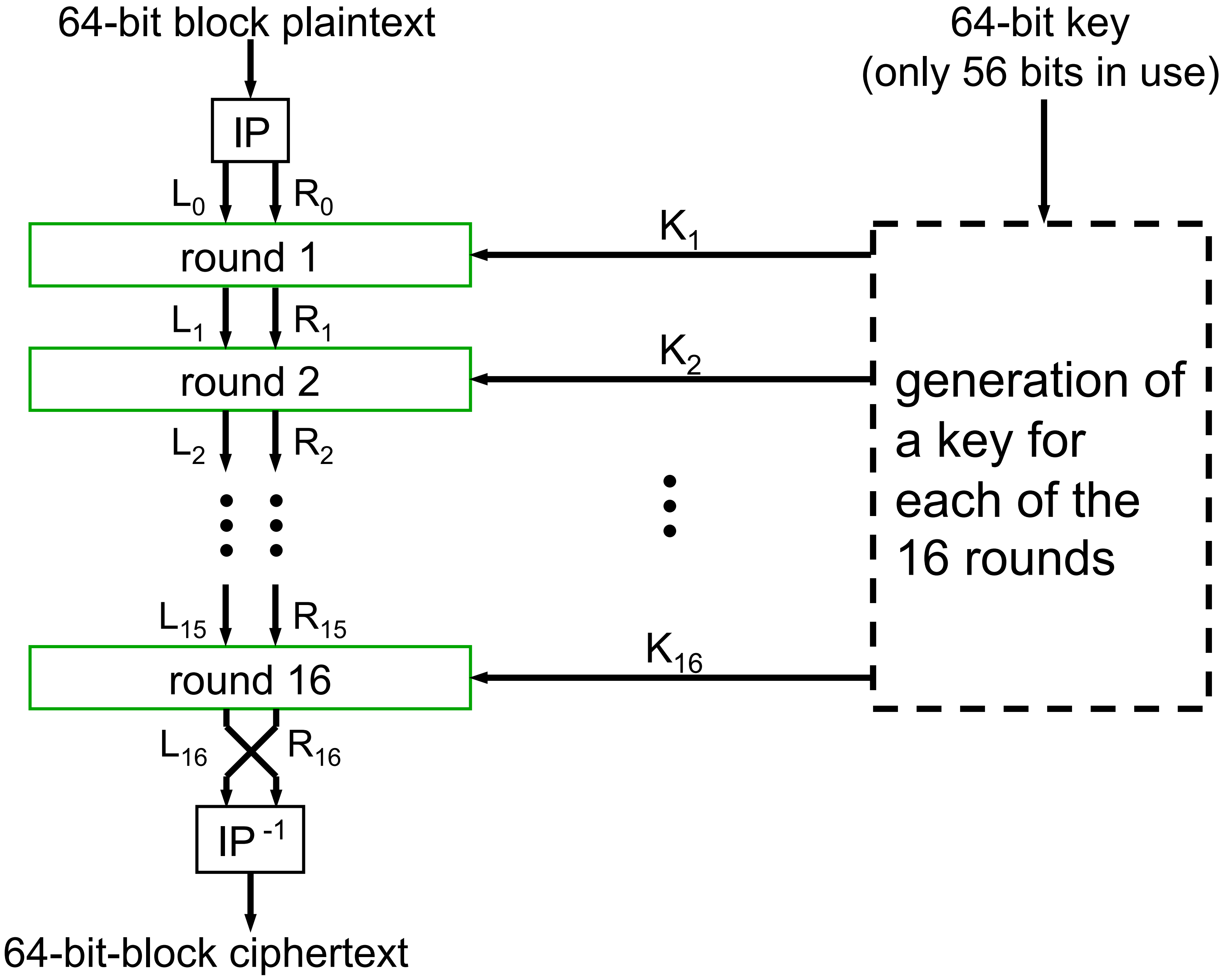
c^{th} roots are unique

Shown : each $y \in \mathrm{Z}_n$ has c^{th} root

$\Rightarrow$ Function $w \rightarrow w^c$ surjective

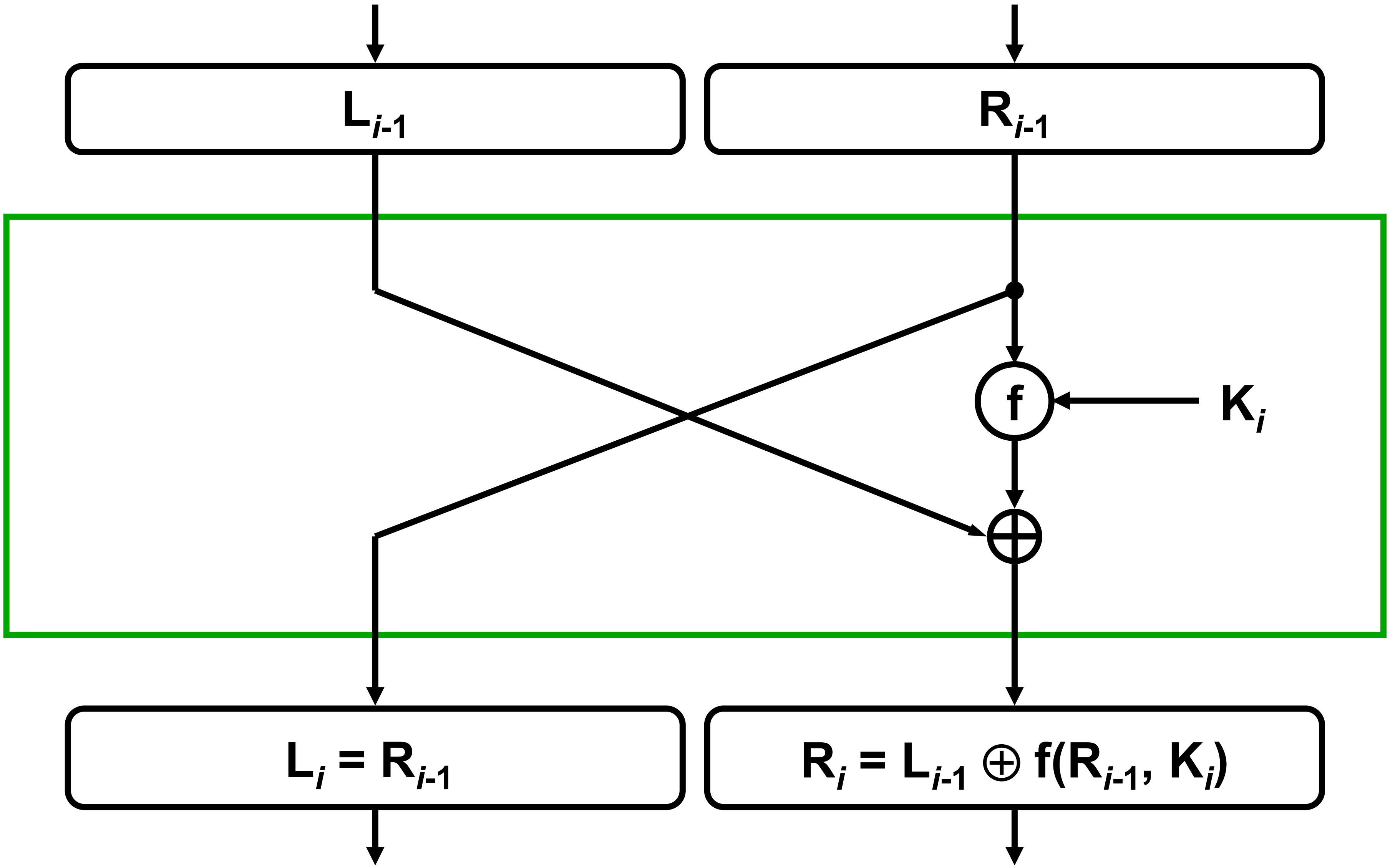
$\Rightarrow$ As well injective.

Symmetric Cryptosystem DES

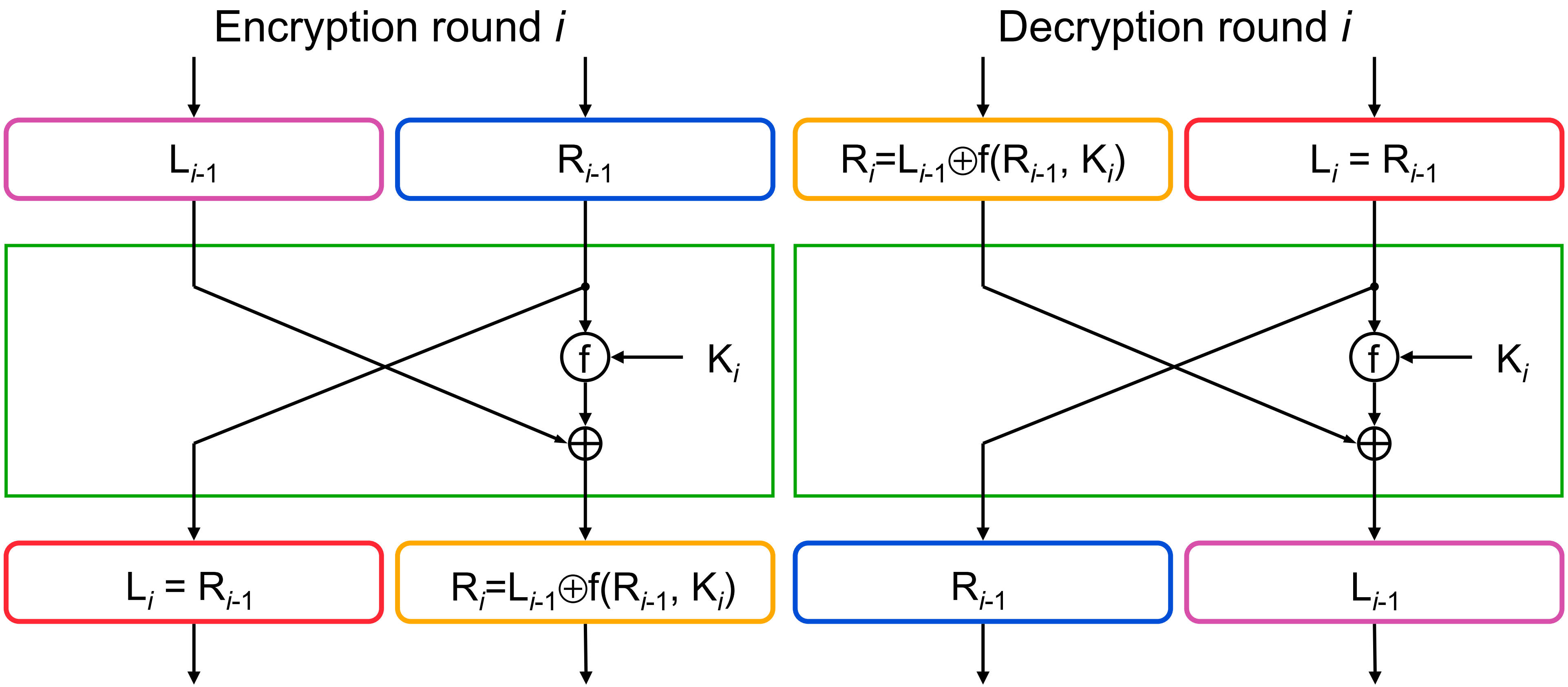


One round

Feistel ciphers



Why does decryption work?



Decryption

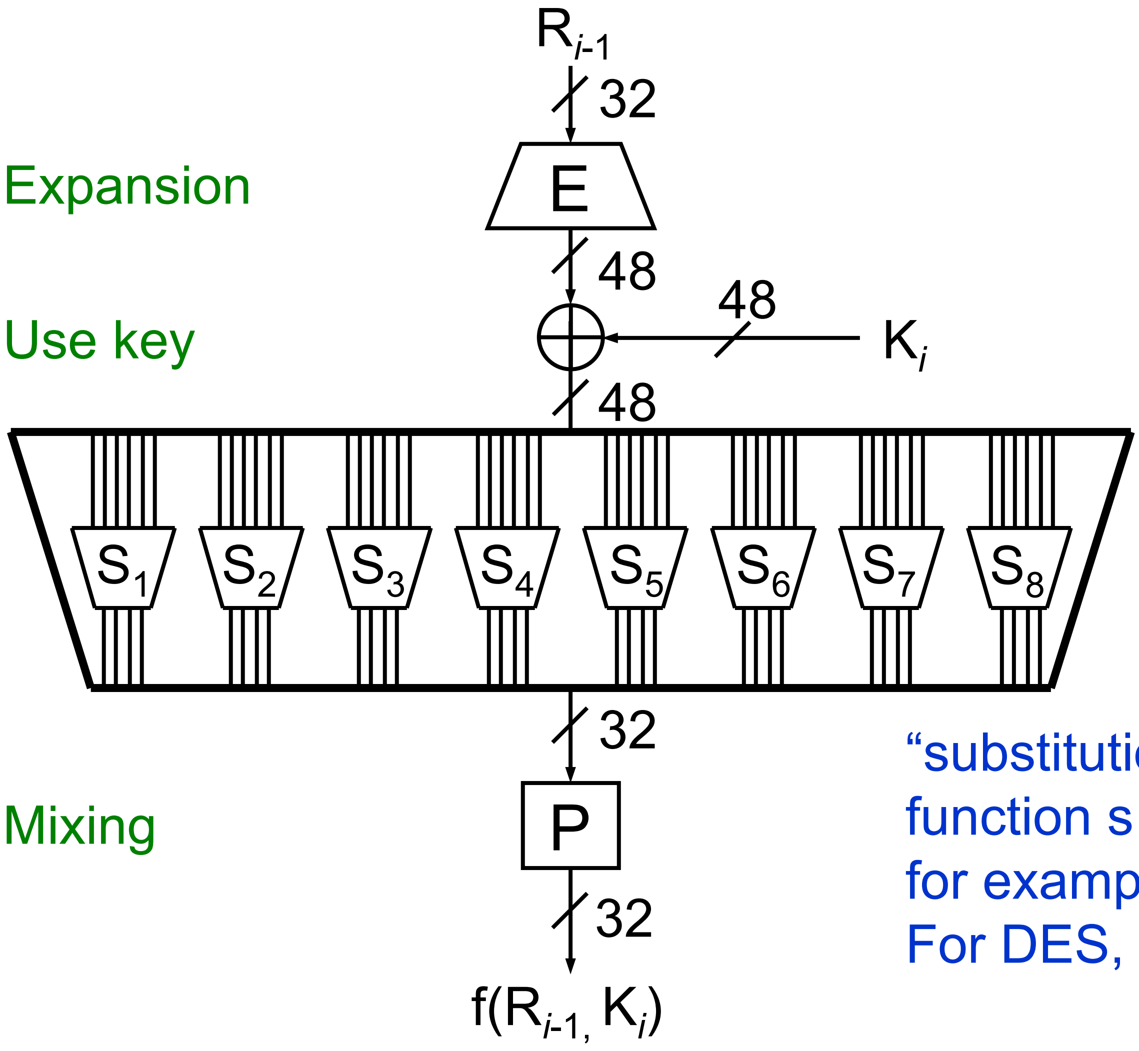
🟥 → 🟦 trivial

replace R_{i-1} by L_i

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

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

Encryption function f



Expansion

Use key

Mixing

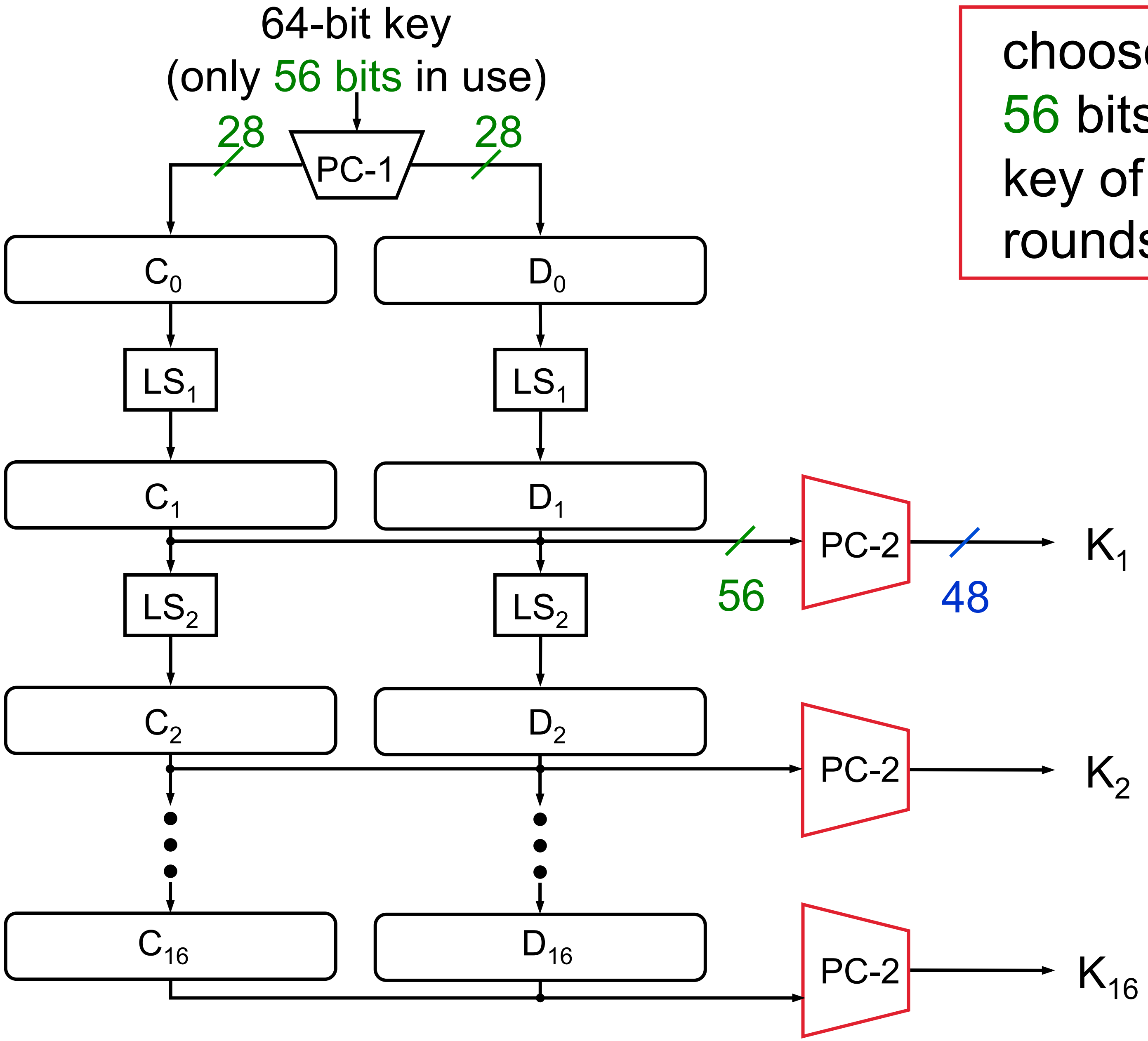
Make f (and DES) non-linear (permutations and ⊕ are linear)

“substitution box” S can implement any function s : $\{0,1\}^6 \rightarrow \{0,1\}^4$, 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

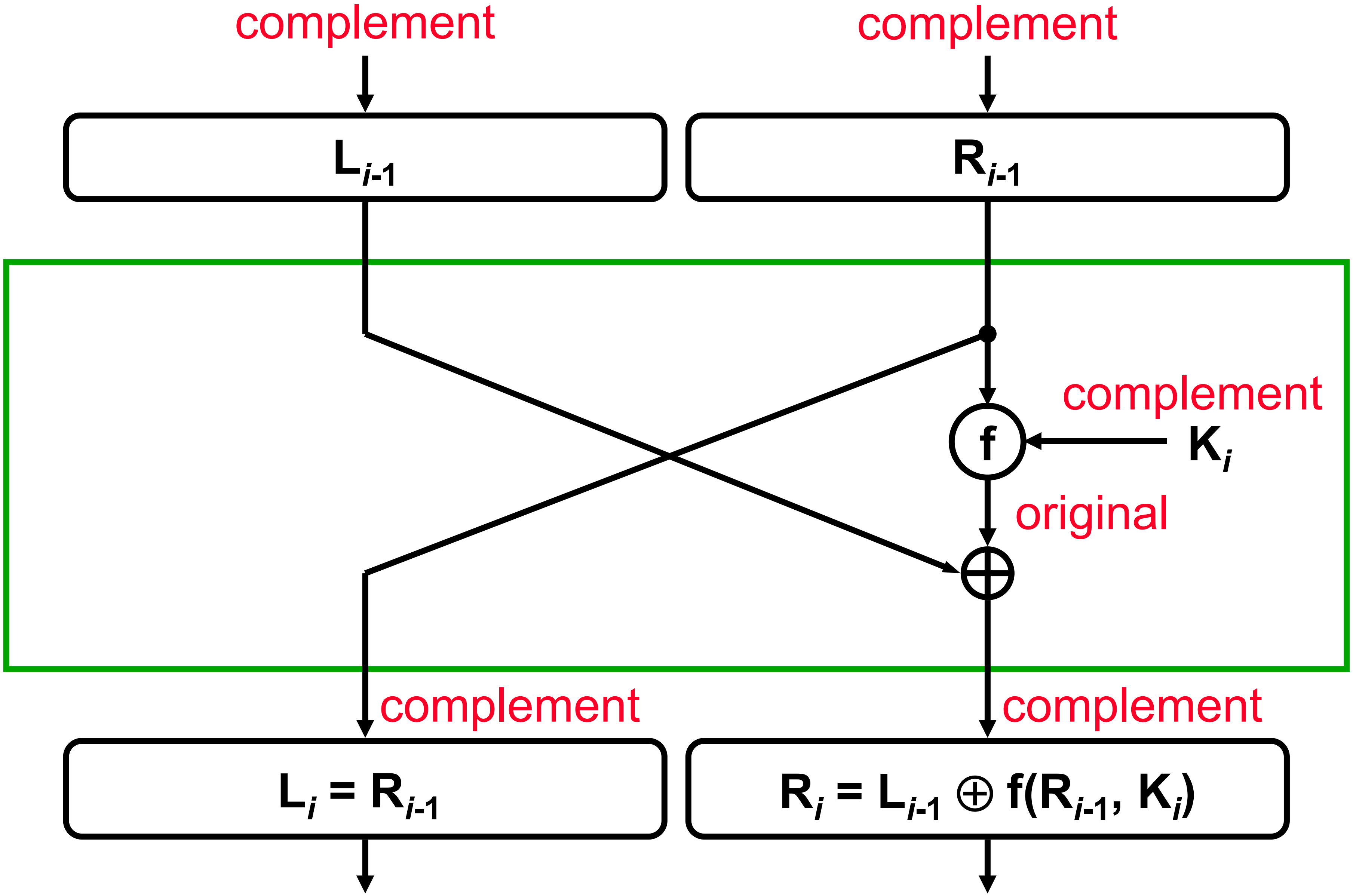


choose 48 of the 56 bits for each key of the 16 rounds

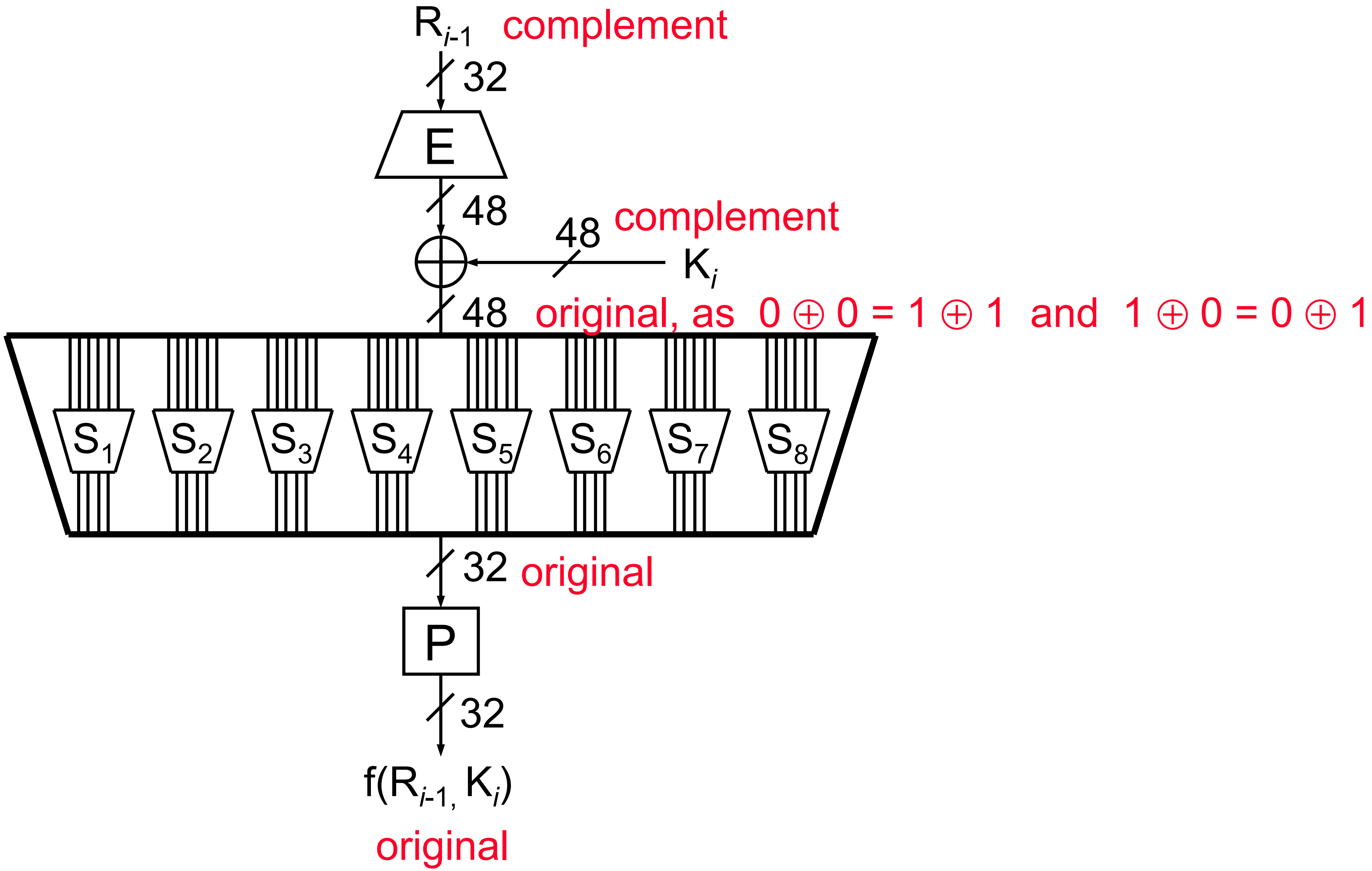
The complementation property of DES

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

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



Generalization of DES

1.) 56 $\Rightarrow$ 16 • 48 = 768 key bits

2.) variable substitution boxes

3.) variable permutations

4.) variable expansion permutation

5.) variable number of rounds

Cipher

Stream cipher

synchronous

self synchronizing

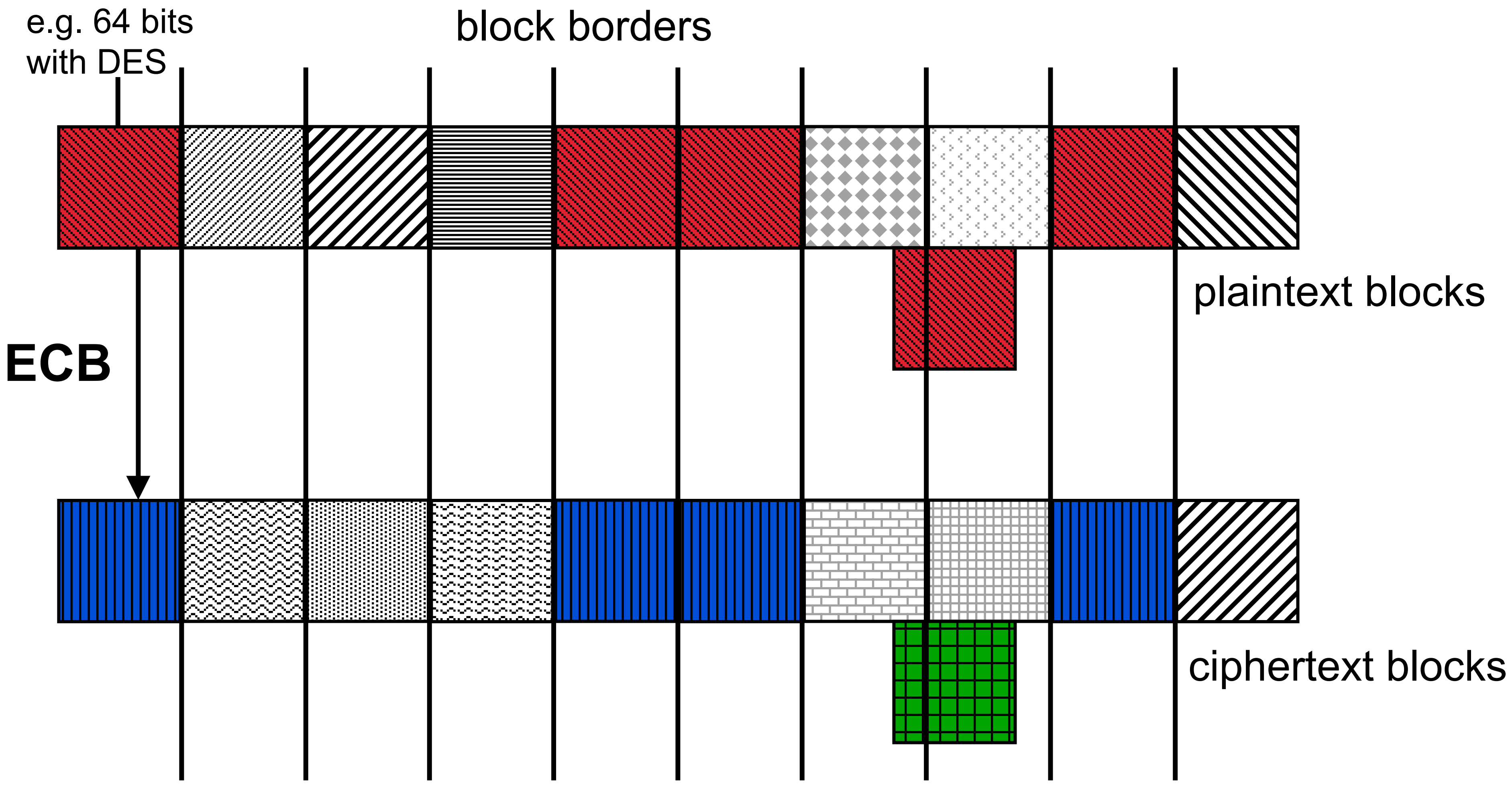
Block cipher

Modes of operation:

Simplest: ECB (electronic codebook)
each block separately

But: concealment: block patterns identifiable
authentication: blocks permutable

Main problem of ECB



e.g. 64 bits
with DES

block borders

ECB

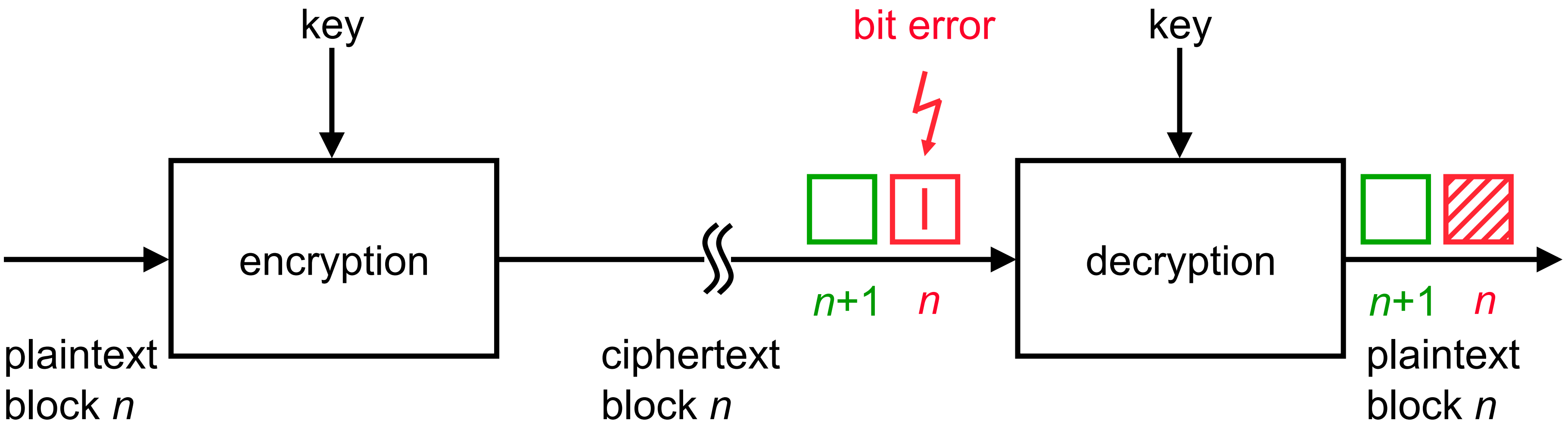
plaintext blocks

ciphertext blocks

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

Telefax example (→ compression is helpful)

Electronic Codebook (ECB)



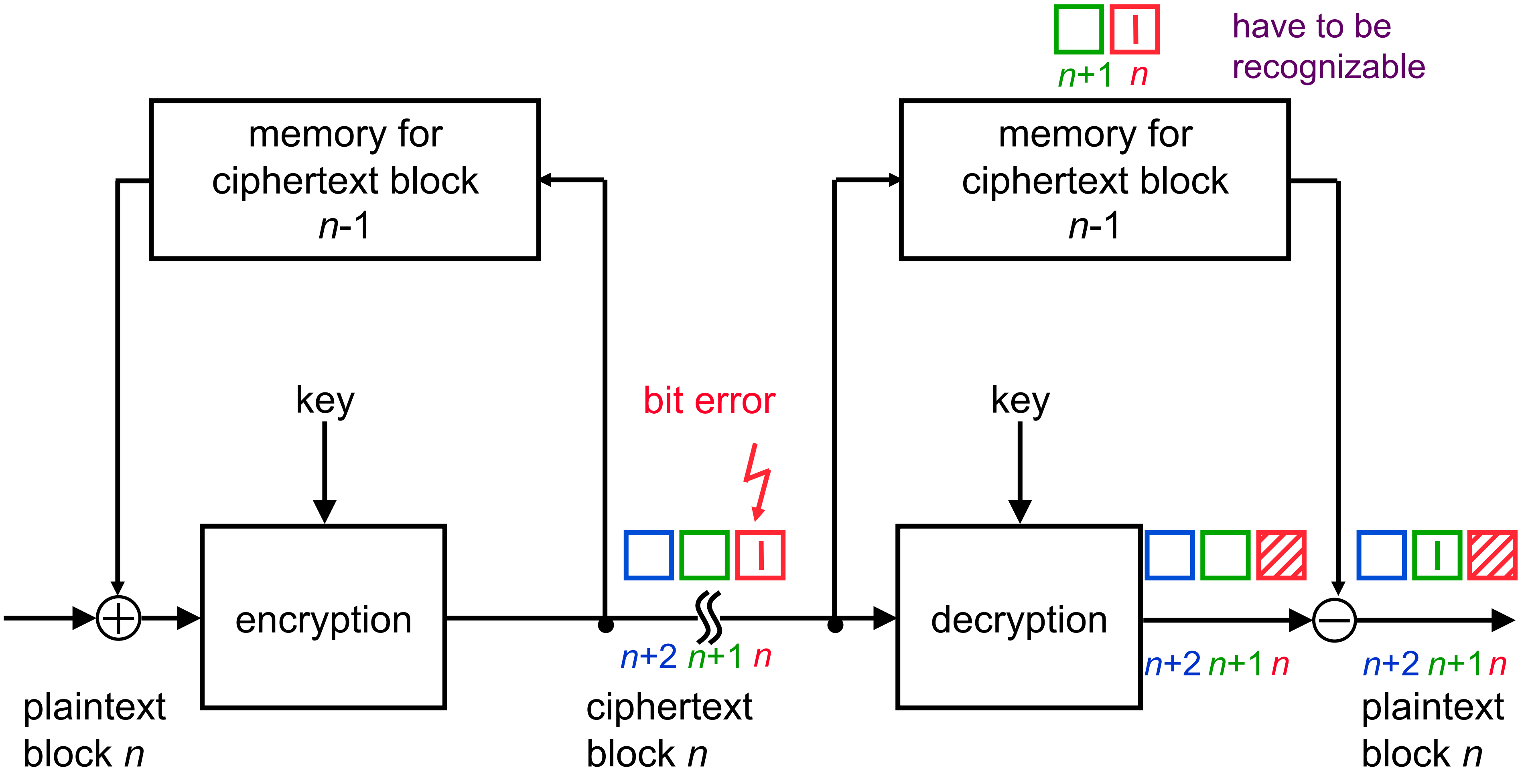
Cipher Block Chaining (CBC)

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



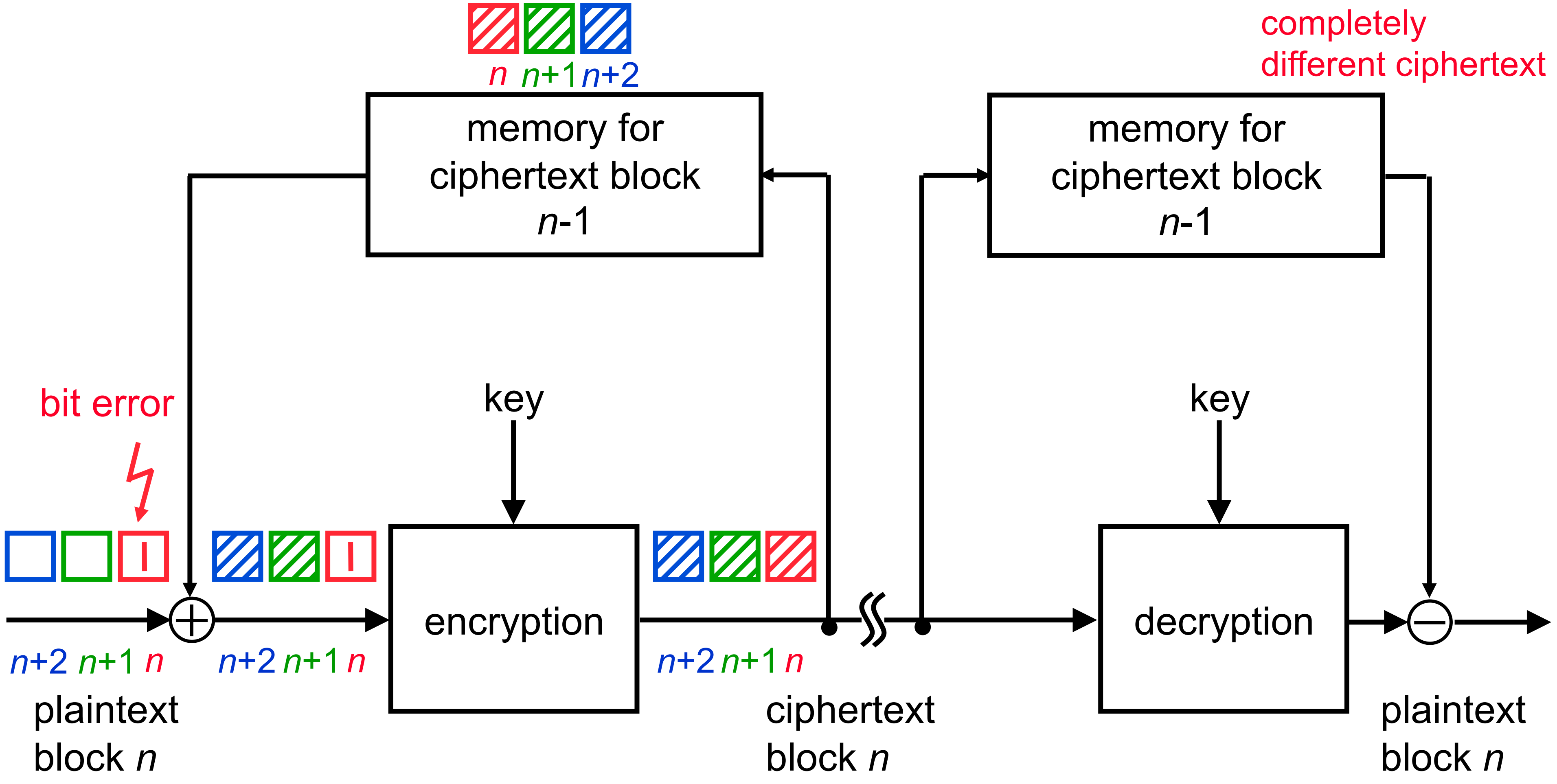
self synchronizing

Cipher Block Chaining (CBC) (2)

All lines transmit as many characters as a block comprises

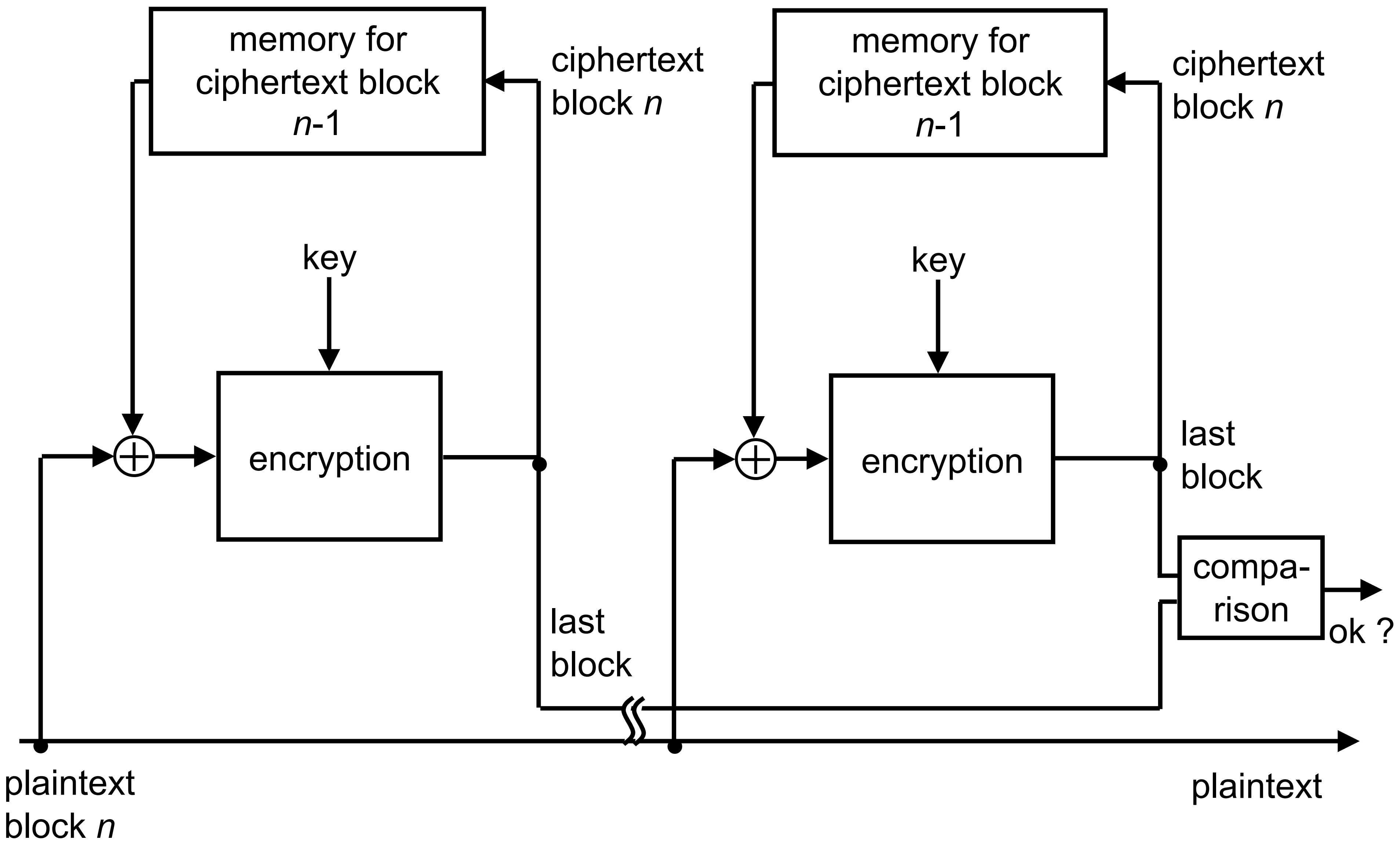
⊕ Addition mod appropriately chosen modulus

⊖ Subtraction mod appropriately chosen modulus

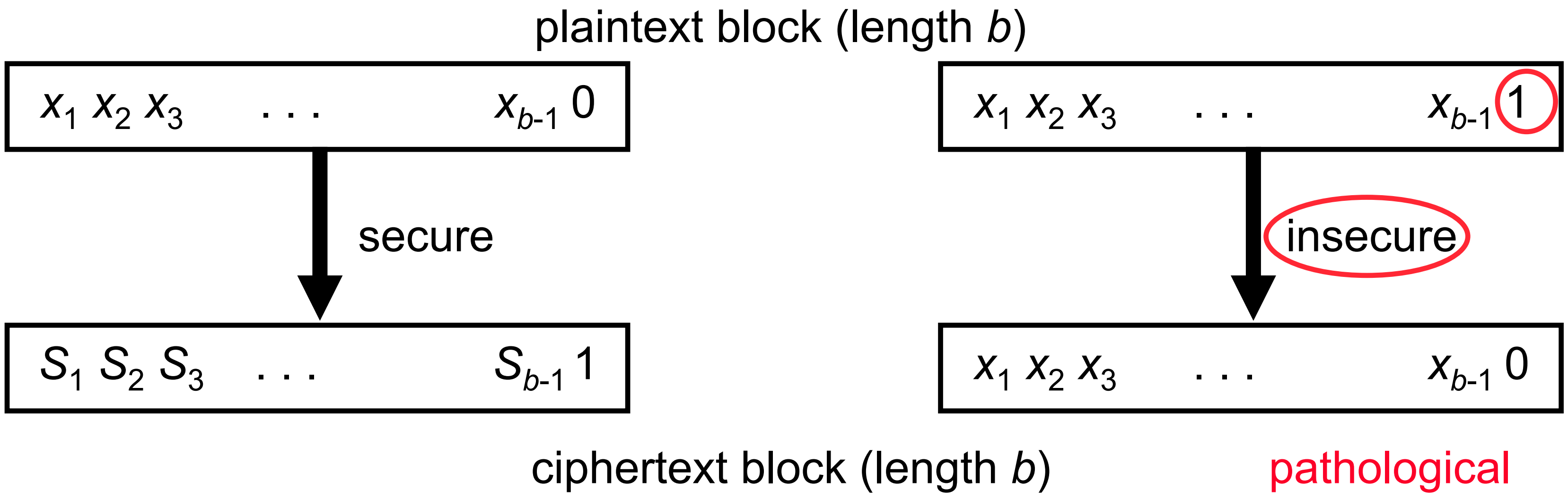


useable for authentication ⇒ use last block as MAC

CBC for authentication



Pathological Block cipher



plaintext block (length b)

$x_1\ x_2\ x_3\ \ldots\ x_{b-1}\ 0$

secure

$S_1\ S_2\ S_3\ \ldots\ S_{b-1}\ 1$

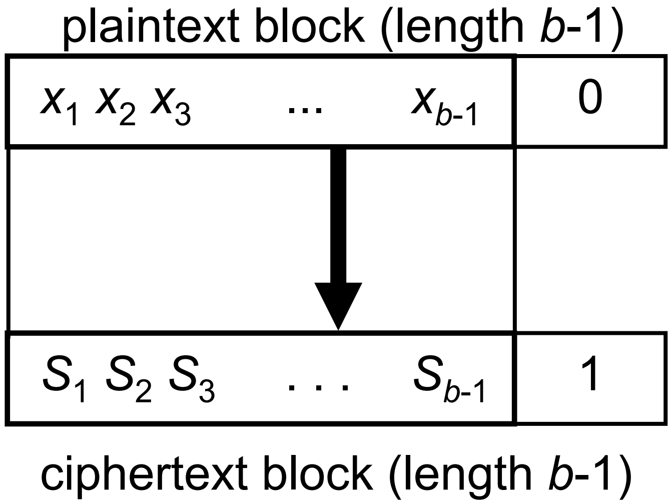
$x_1\ x_2\ x_3\ \ldots\ x_{b-1}\ 1$

insecure

$x_1\ x_2\ x_3\ \ldots\ x_{b-1}\ 0$

ciphertext block (length b)

pathological



plaintext block (length b-1)

$x_1\ x_2\ x_3\ \ldots\ x_{b-1}$ | 0

$S_1\ S_2\ S_3\ \ldots\ S_{b-1}$ | 1

ciphertext block (length b-1)

Cipher FeedBack (CFB)

b Block length

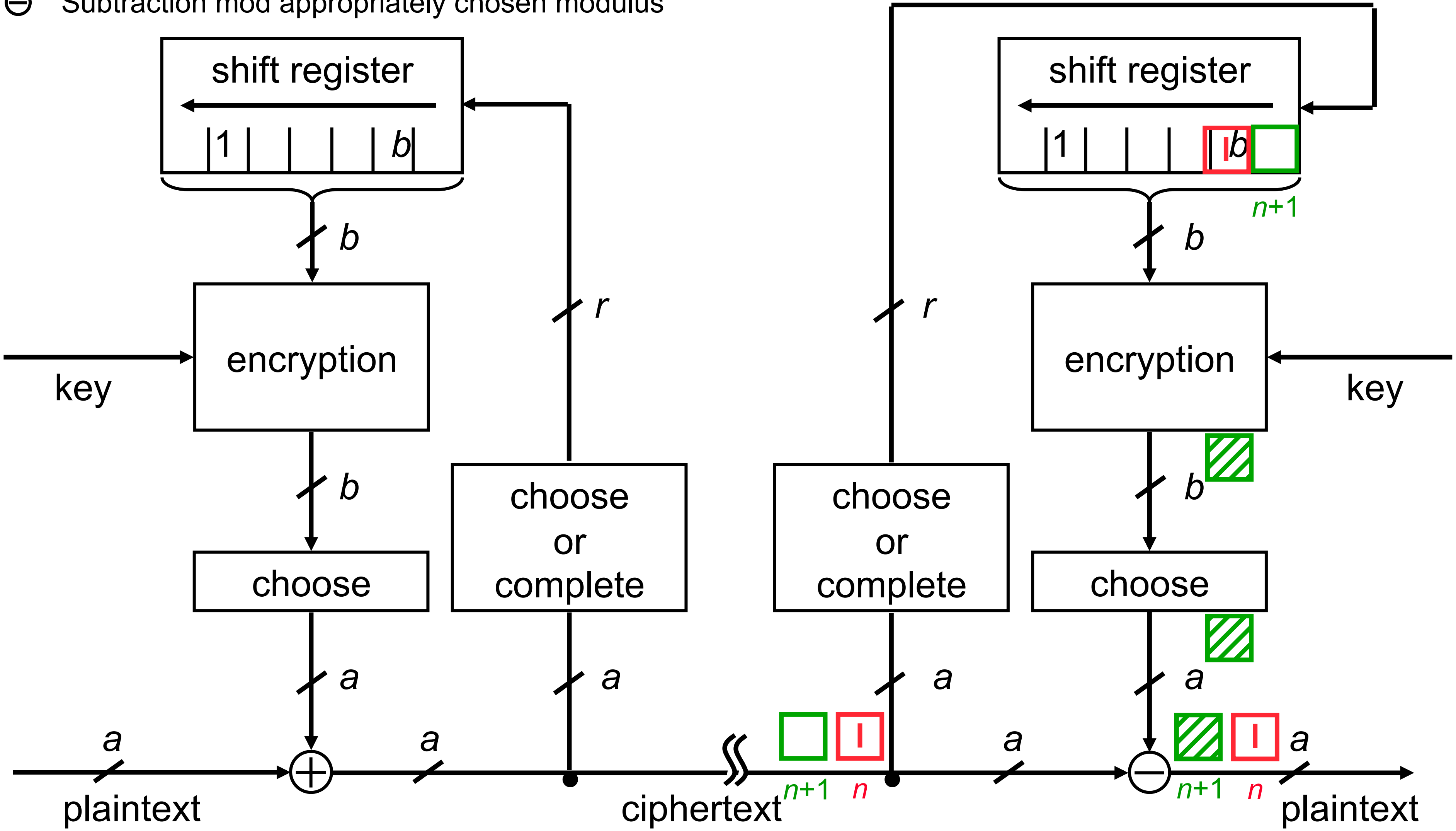
a Length of the output unit, $a \le b$

r Length of the feedback unit, $r \le b$

⊕ Addition mod appropriately chosen modulus

⊖ Subtraction mod appropriately chosen modulus

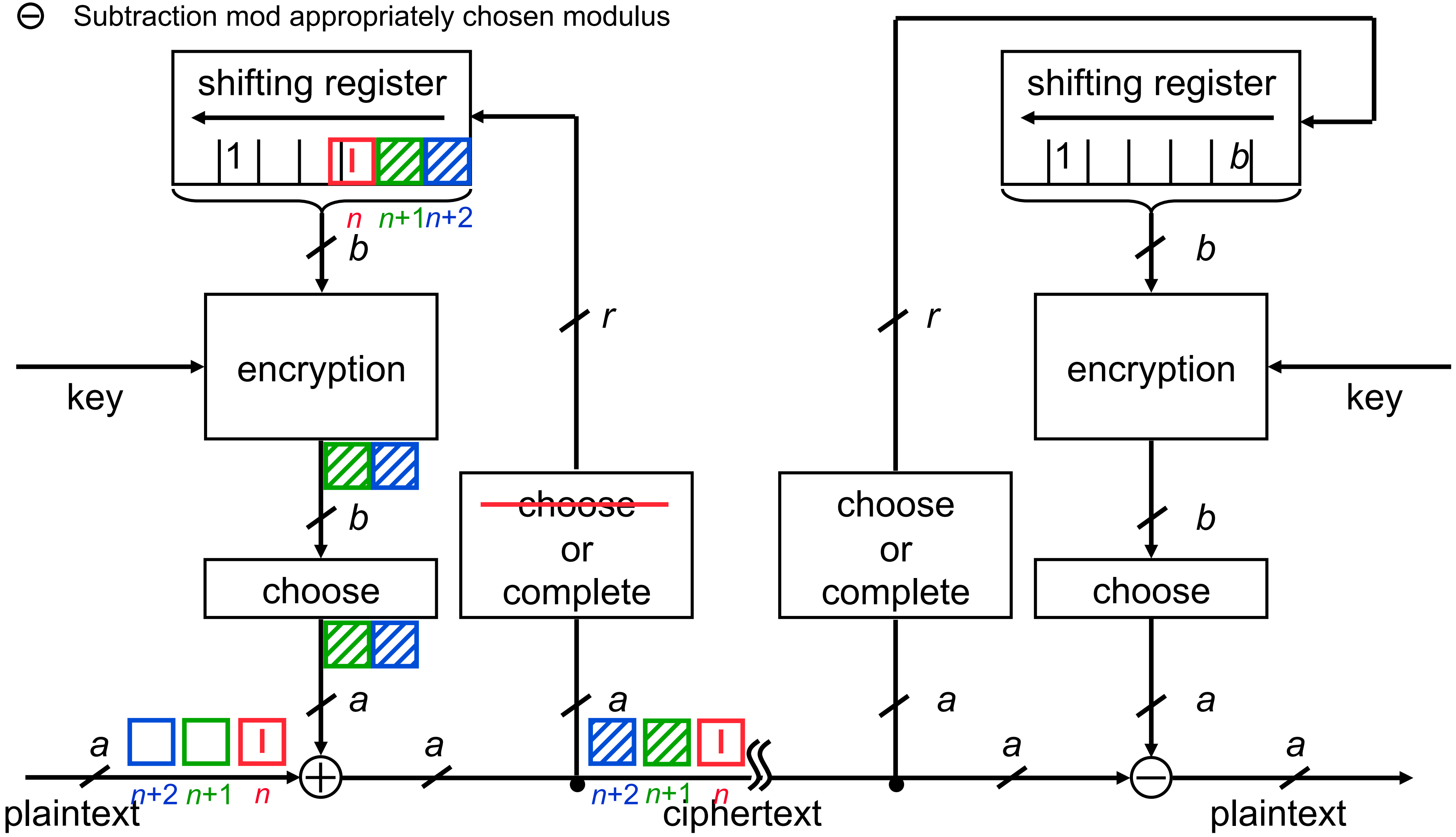
symmetric;
self synchronizing



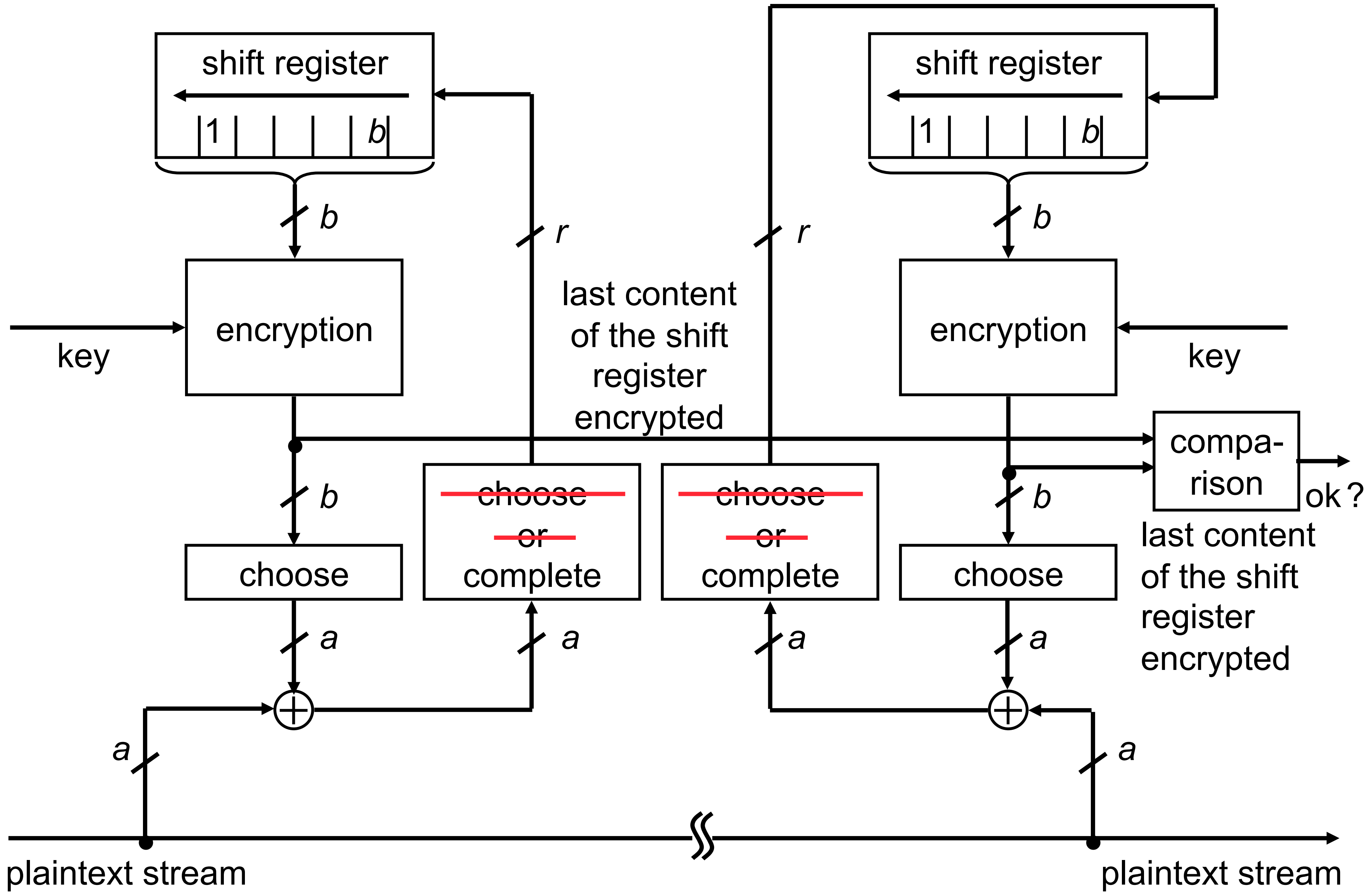
Cipher FeedBack (CFB) (2)

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;
self synchronizing



CFB for authentication



Output FeedBack (OFB)

b Block length

a Length of the output unit, $a \leq b$

r Length of the feedback unit, $r \leq b$

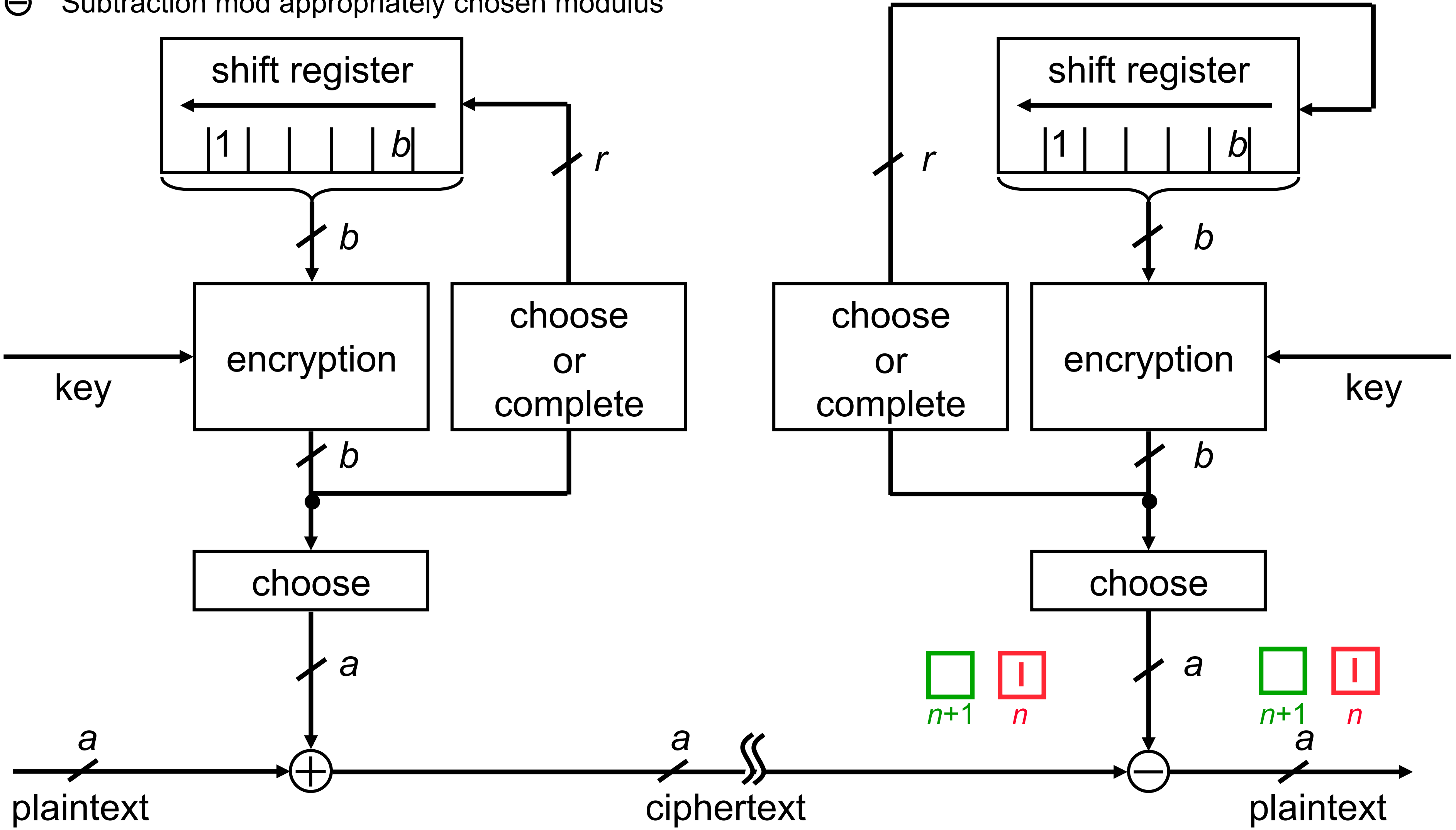
$\oplus$ Addition mod appropriately chosen modulus

$\ominus$ Subtraction mod appropriately chosen modulus

symmetric;

synchronous

Pseudo-one-time-pad



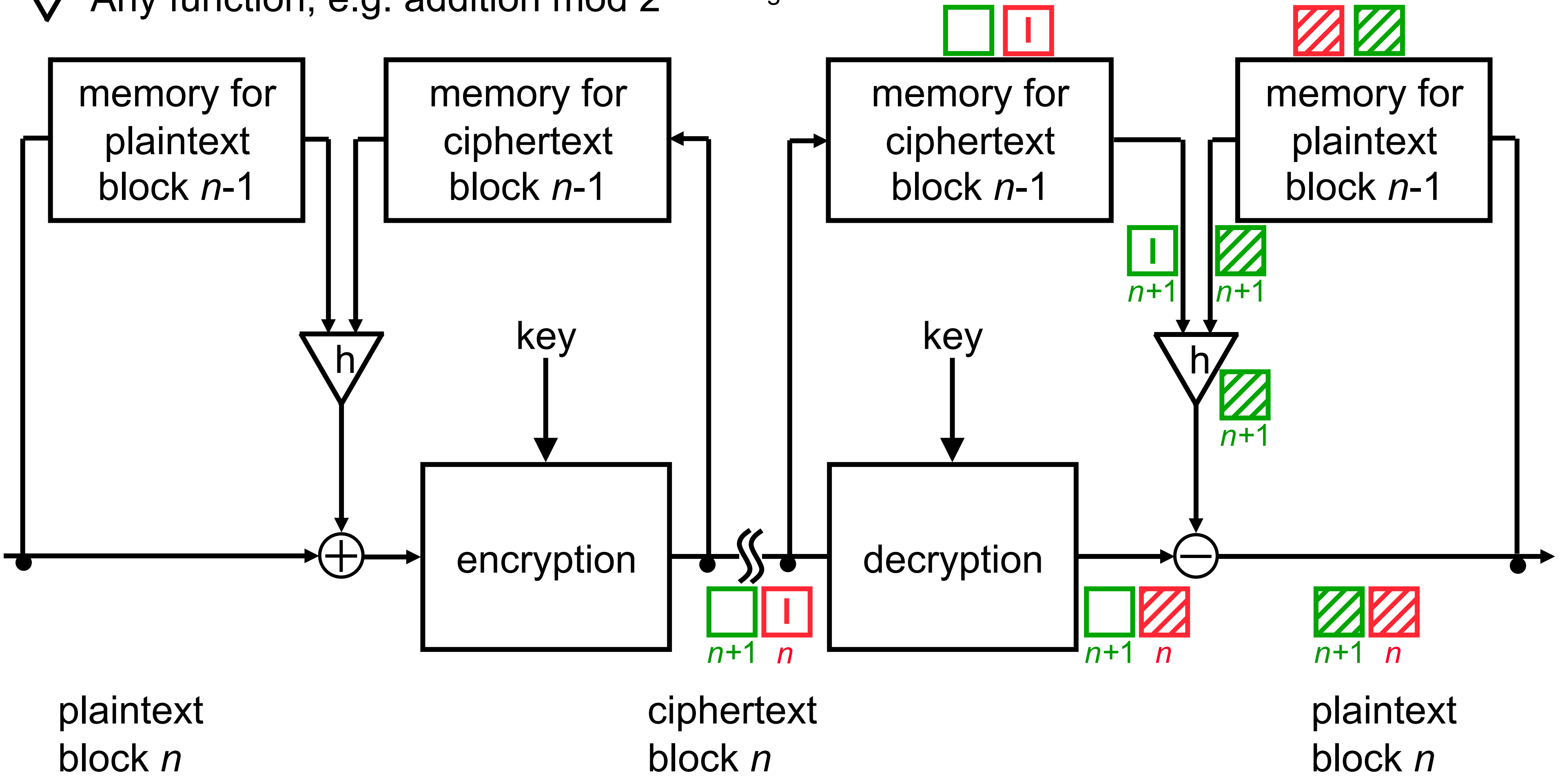
Plain Cipher Block Chaining (PCBC)

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

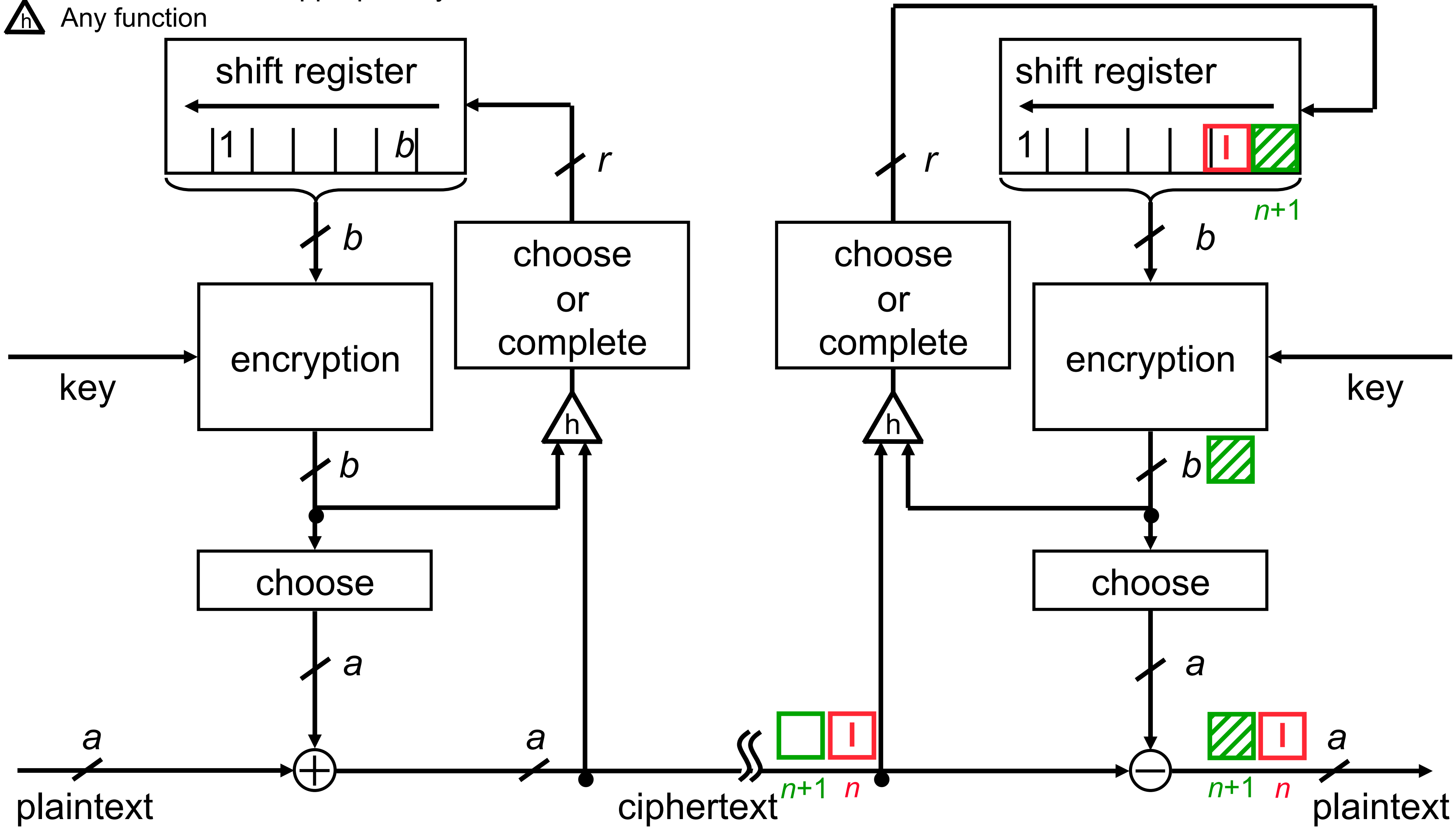
h Any function, e.g. addition mod $2^{\text{Block length}}$


memory for plaintext block *n*-1
memory for ciphertext block *n*-1
memory for ciphertext block *n*-1
memory for plaintext block *n*-1
n+1
n+1
h
key
key
h
n+1
encryption
decryption
n+1
n
n+1
n
n+1
n
plaintext block *n*
ciphertext block *n*
plaintext block *n*

Output Cipher FeedBack (OCFB)

b Block length
a Length of the output unit, $a \leq b$
r Length of the feedback unit, $r \leq b$
⊕ Addition mod appropriately chosen modulus
⊖ Subtraction mod appropriately chosen modulus
h Any function

symmetric;
synchronous



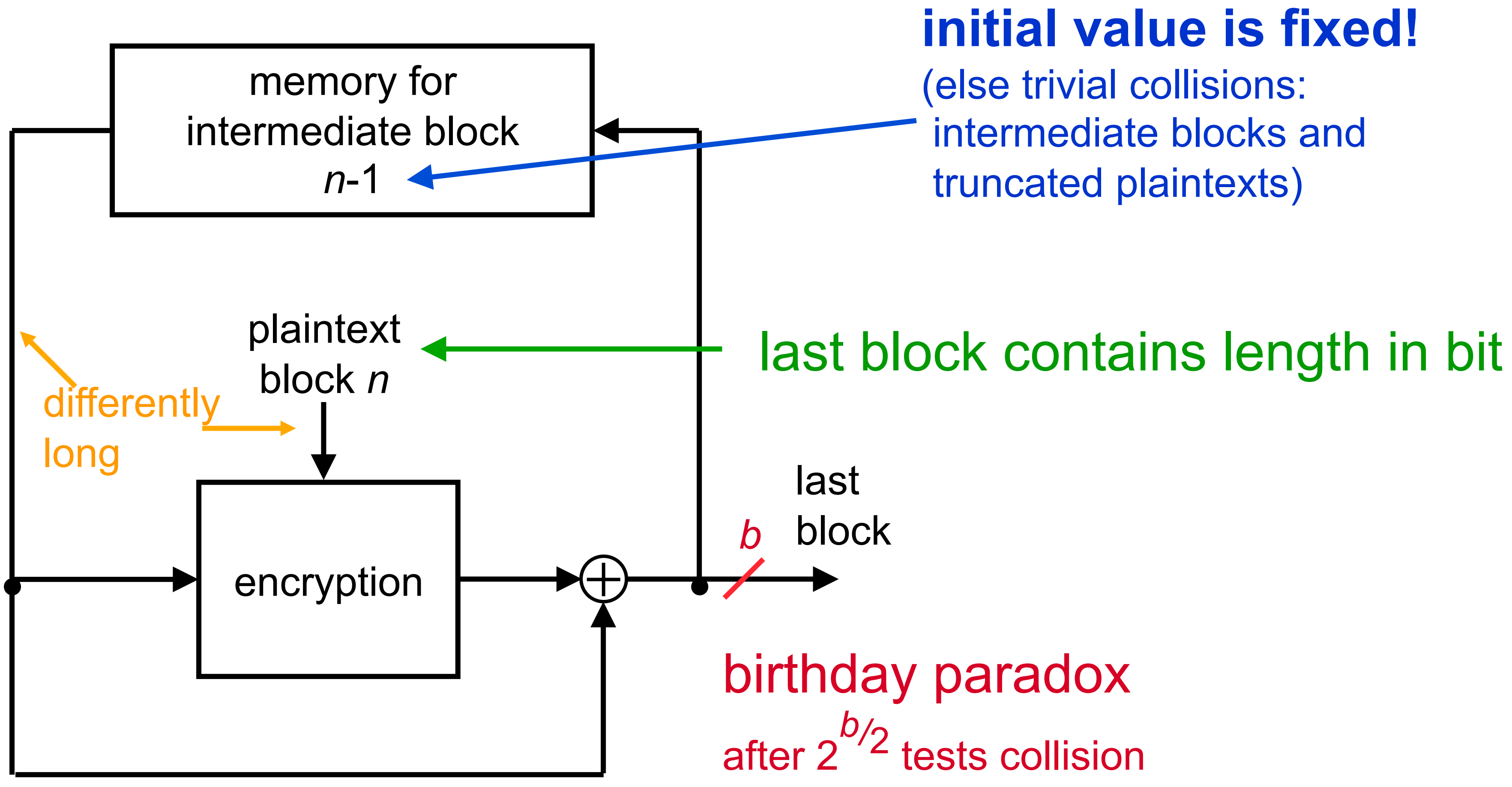
Properties of the operation modes

	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 ! any nearly

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

Diffie-Hellman key agreement (1)

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

based on difficulty to calculate **discrete logarithms**

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

$$\boldsymbol{g}^{x} = \boldsymbol{h} \bmod \boldsymbol{p}$$

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

$$\boldsymbol{x} = \log_{\boldsymbol{g}}(\boldsymbol{h}) \mod \boldsymbol{p}$$

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 $g^x = h \mod p$

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

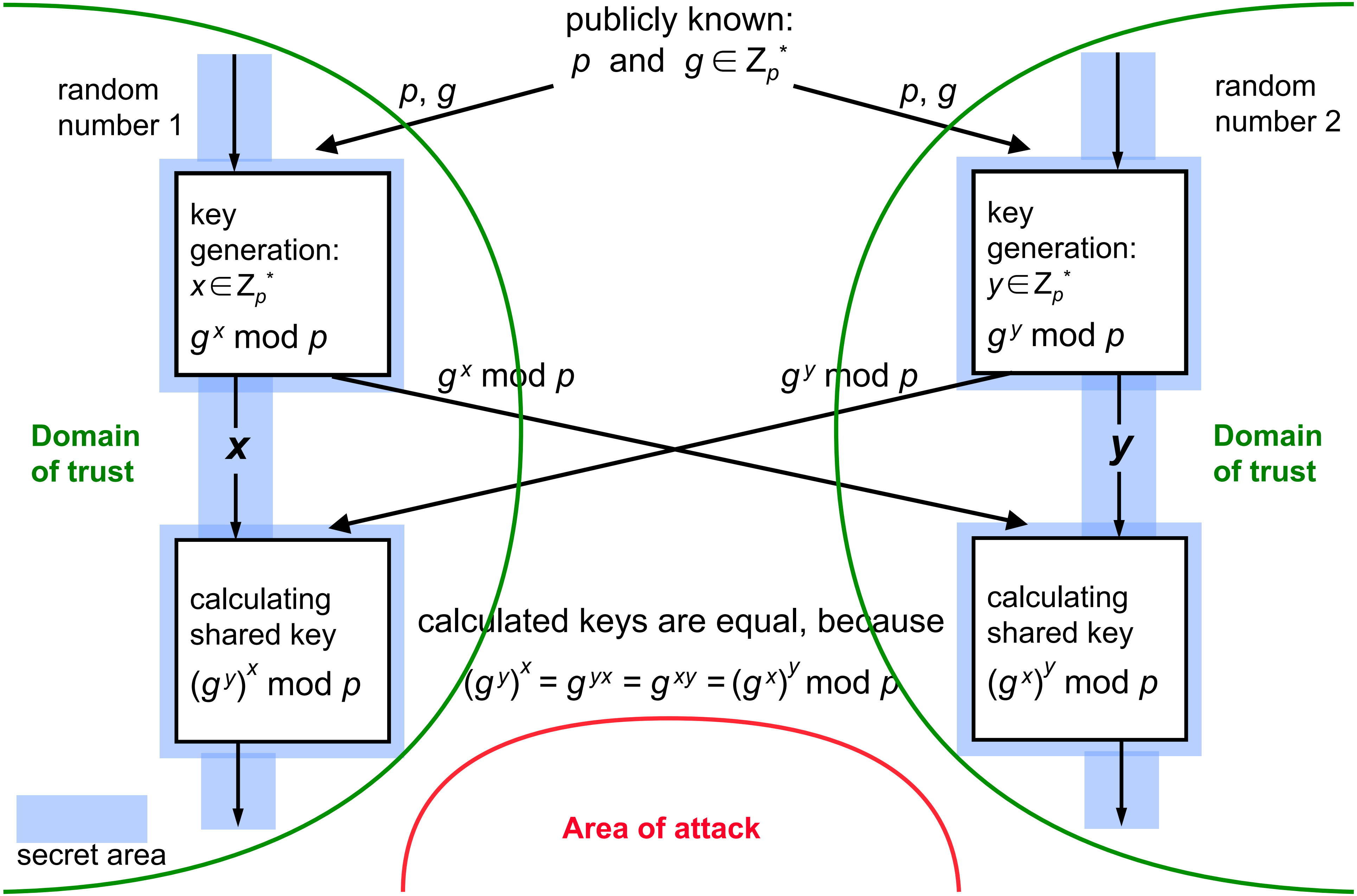
(probability that $\mathcal{DL}$ really calculates the discrete logarithm,

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

trustworthy ??

practically as well analyzed as the assumption factoring is hard

Diffie-Hellman key agreement (2)



publicly known:
p and $g \in Z_p^*$

p, g

p, g

random number 1

random number 2

key generation:
$x \in Z_p^*$
$g^x \bmod p$

key generation:
$y \in Z_p^*$
$g^y \bmod p$

$g^x \bmod p$

$g^y \bmod p$

Domain of trust

Domain of trust

x

y

calculating shared key
$(g^y)^x \bmod p$

calculating shared key
$(g^x)^y \bmod p$

calculated keys are equal, because

$$(g^y)^x = g^{yx} = g^{xy} = (g^x)^y \bmod p$$

Area of attack

🟦 secret area

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

Find a generator in cyclic group Z_p^*

Find a generator of a cyclic group Z_p^*

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

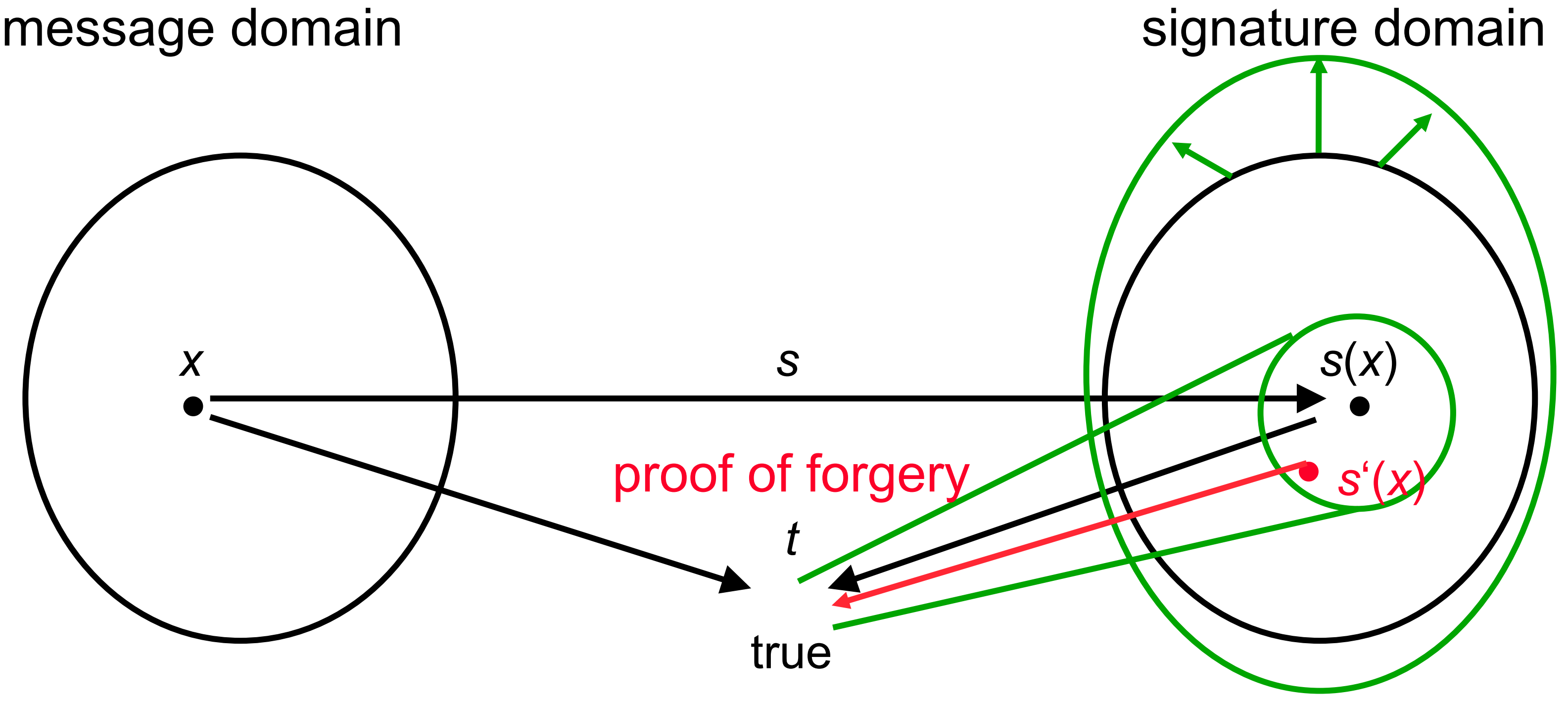
```
1. Choose a random element $g$ in $Z_p^*$
2. For $i$ from 1 to $k$:
        $b := g^{\frac{p-1}{p_i}} \bmod p$
        If $b=1$ go to 1.
```

Digital signature system

Security is asymmetric, too

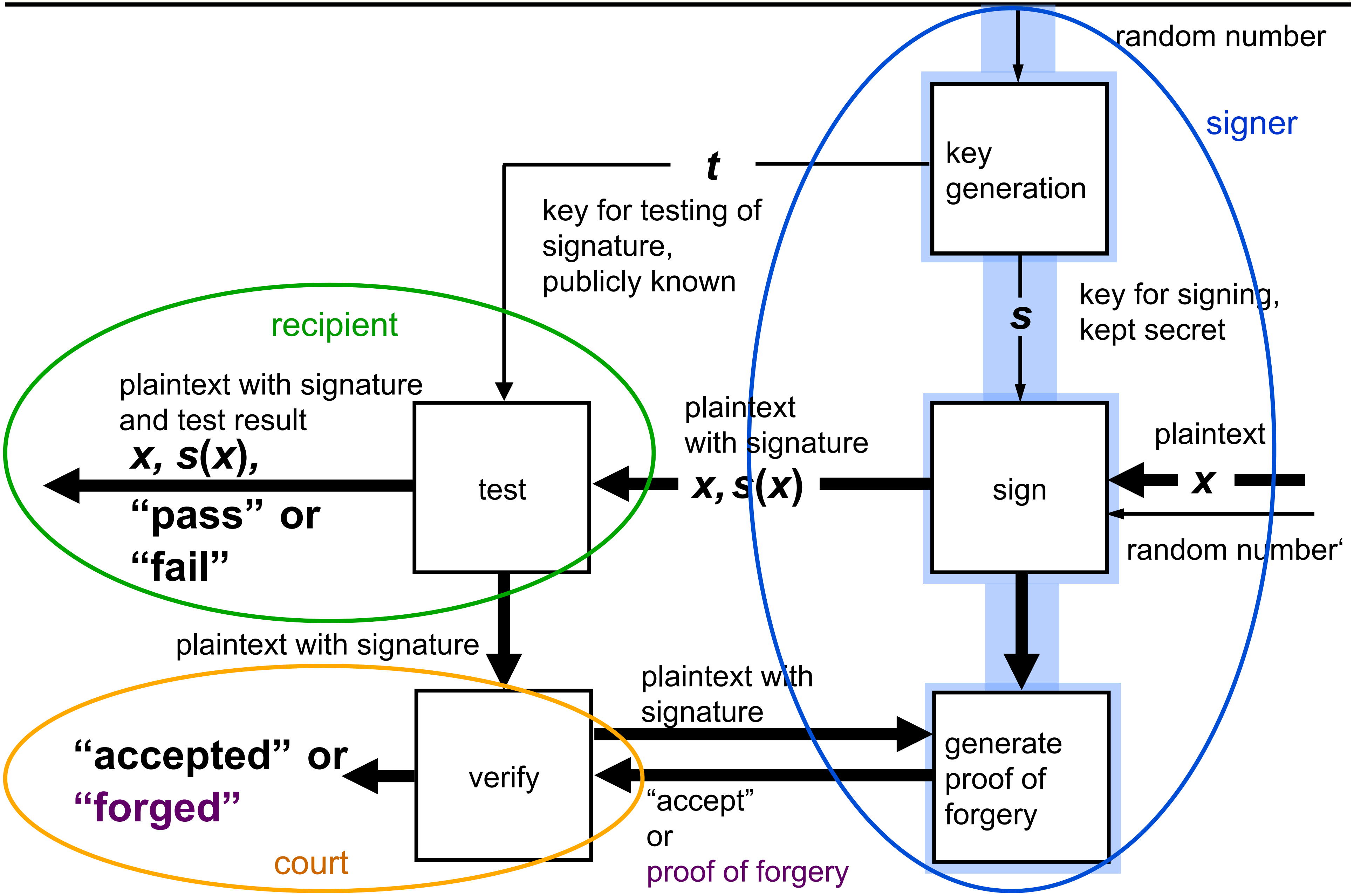
usually: unconditionally secure for recipient
only cryptographically secure for signer

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

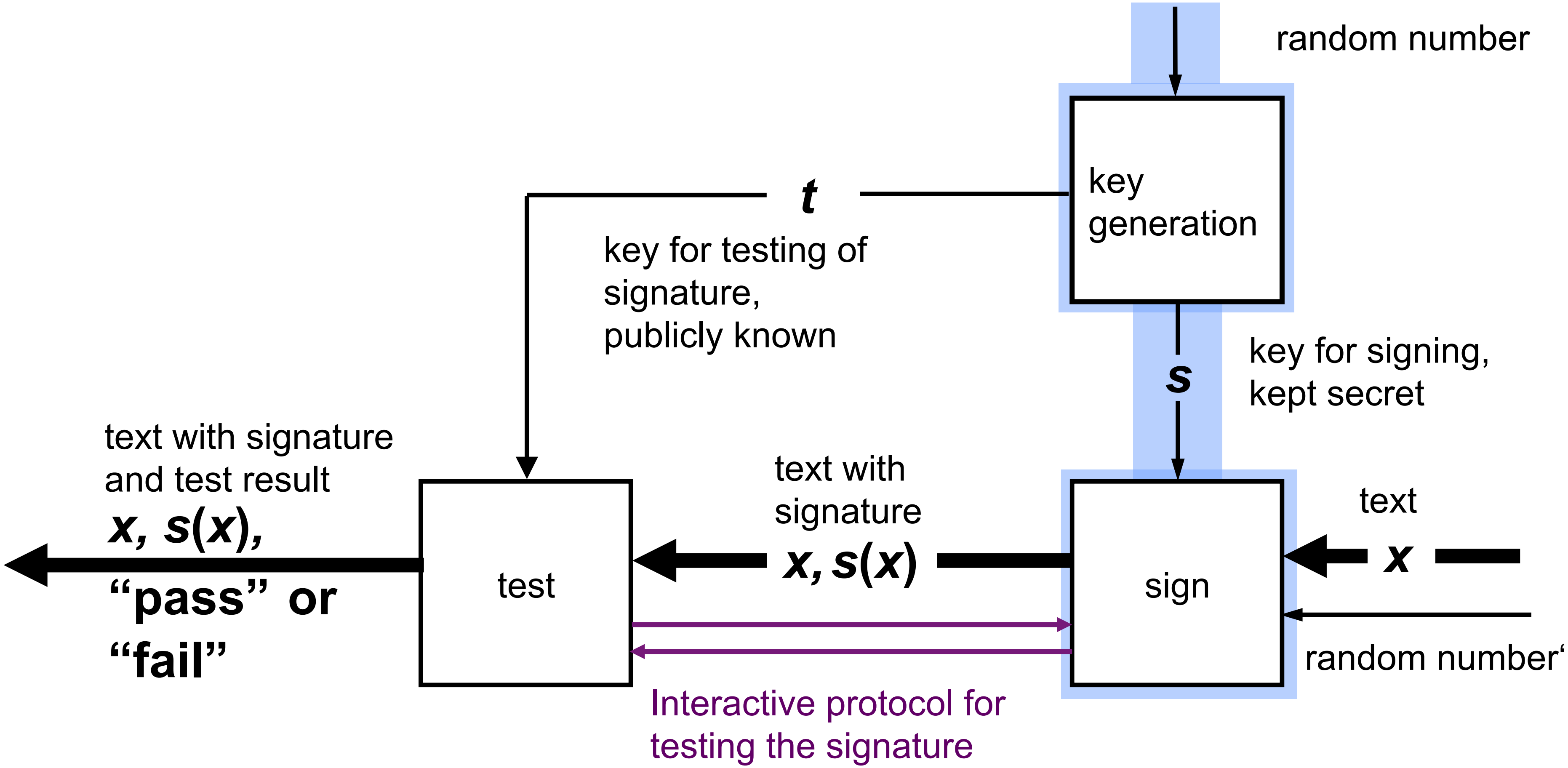


distribution of risks if signature is forged: 1. recipient
2. insurance or system operator
3. signer

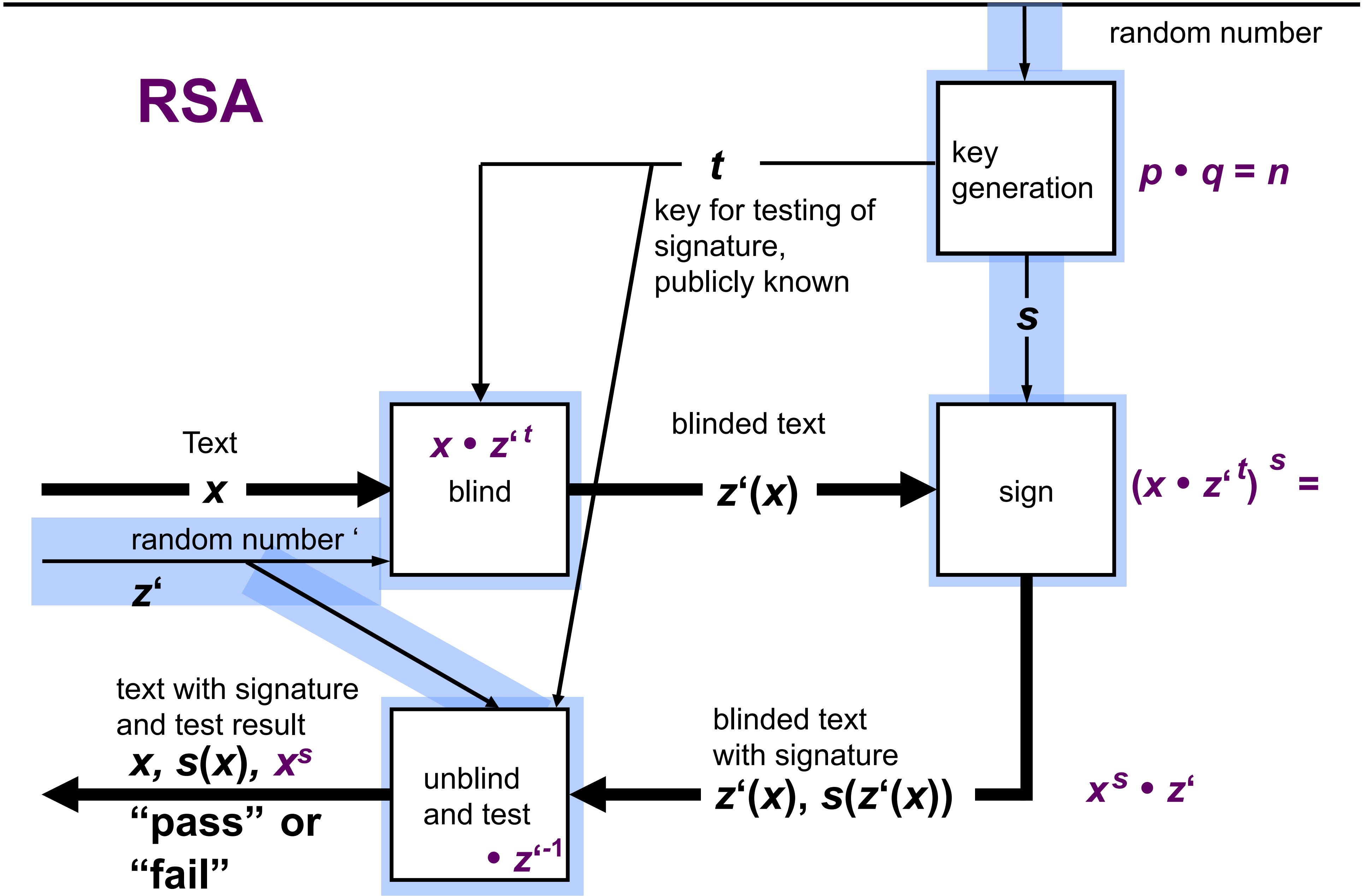
Fail-stop signature system



Undeniable signatures


random number
key generation
t
key for testing of signature, publicly known
s
key for signing, kept secret
text with signature and test result
x, s(x),
"pass" or "fail"
test
text with signature
x, s(x)
sign
text
x
random number'
Interactive protocol for testing the signature

Signature system for blindly providing of signatures



Threshold scheme (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

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

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

Threshold scheme (2)

Reconstruction of the secret:

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

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

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 of the parts ⇒ Addition of the secrets

Share refreshing

1.) Choose random polynomial q' for $S' = 0$

2.) Distribute the n parts $(i, q'(i))$

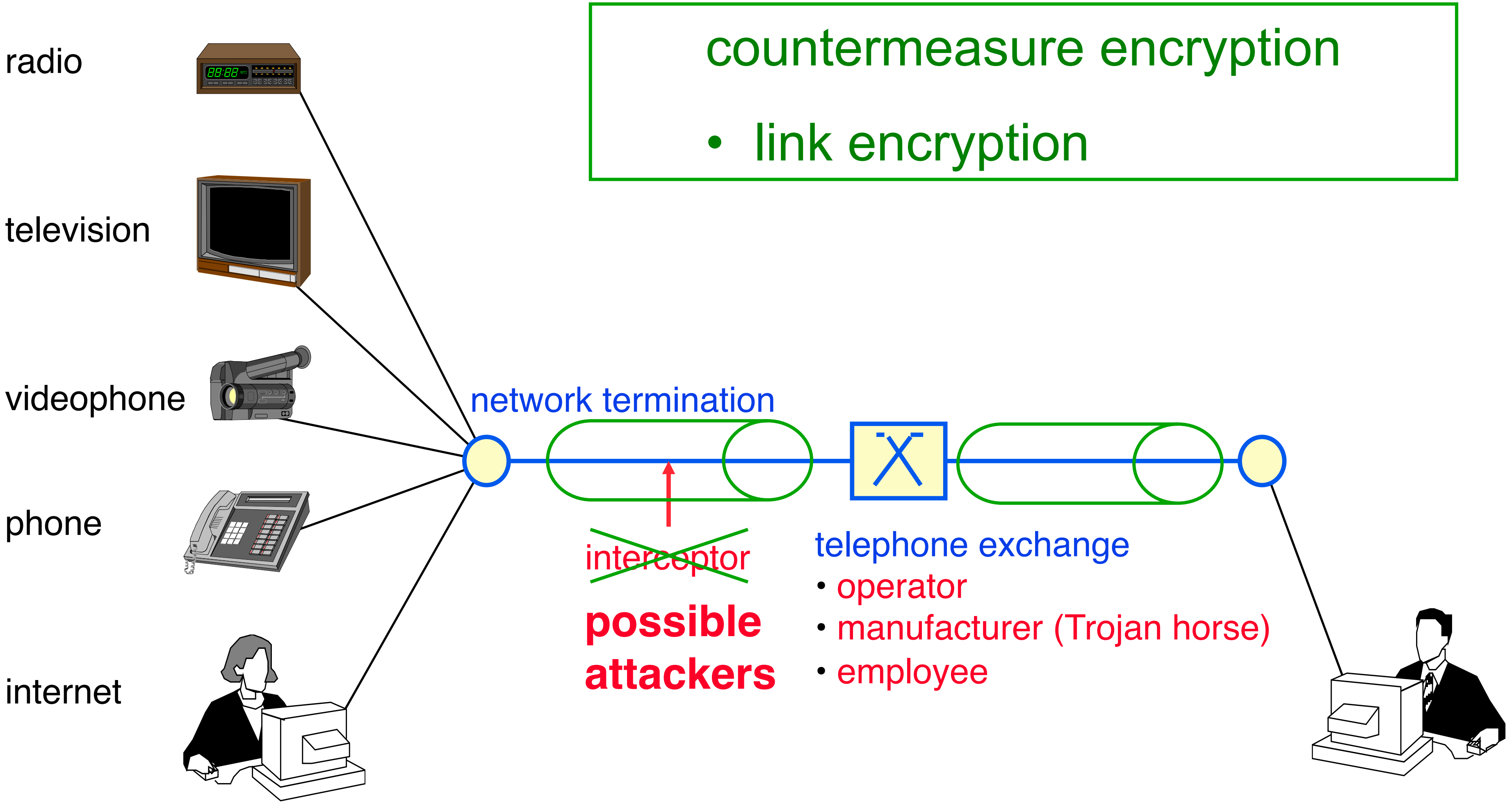
3.) Everyone adds his “new” part to his “old” part
→ “new” random polynomial $q+q'$ with “old” secret S

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

Observability of users in switched networks

countermeasure encryption

- link encryption



radio

television

videophone

phone

internet

network termination

~~interceptor~~

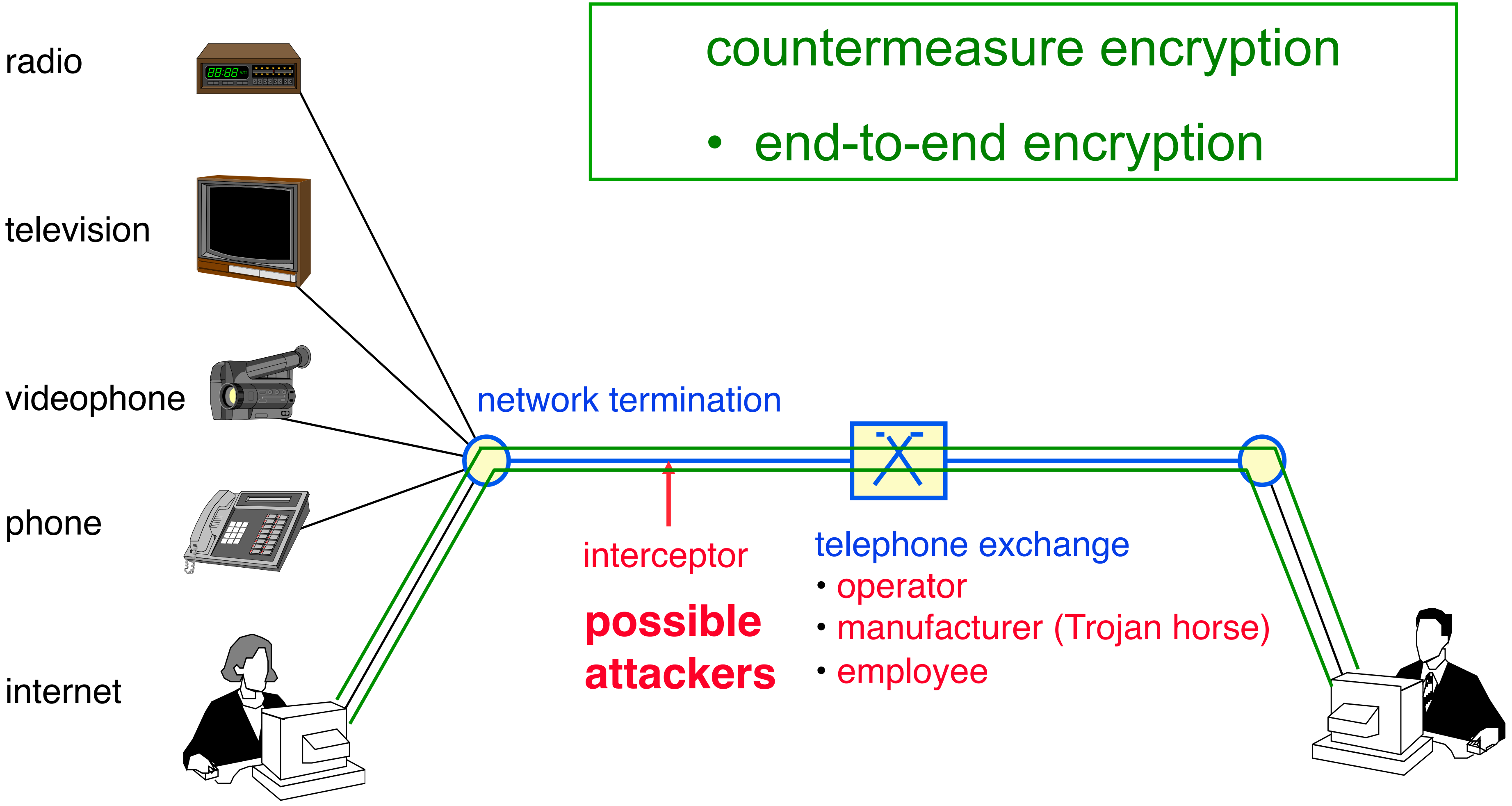
possible attackers

telephone exchange
- operator
- manufacturer (Trojan horse)
- employee

Observability of users in switched networks

countermeasure encryption

- end-to-end encryption



radio

television

videophone

phone

internet

network termination

interceptor

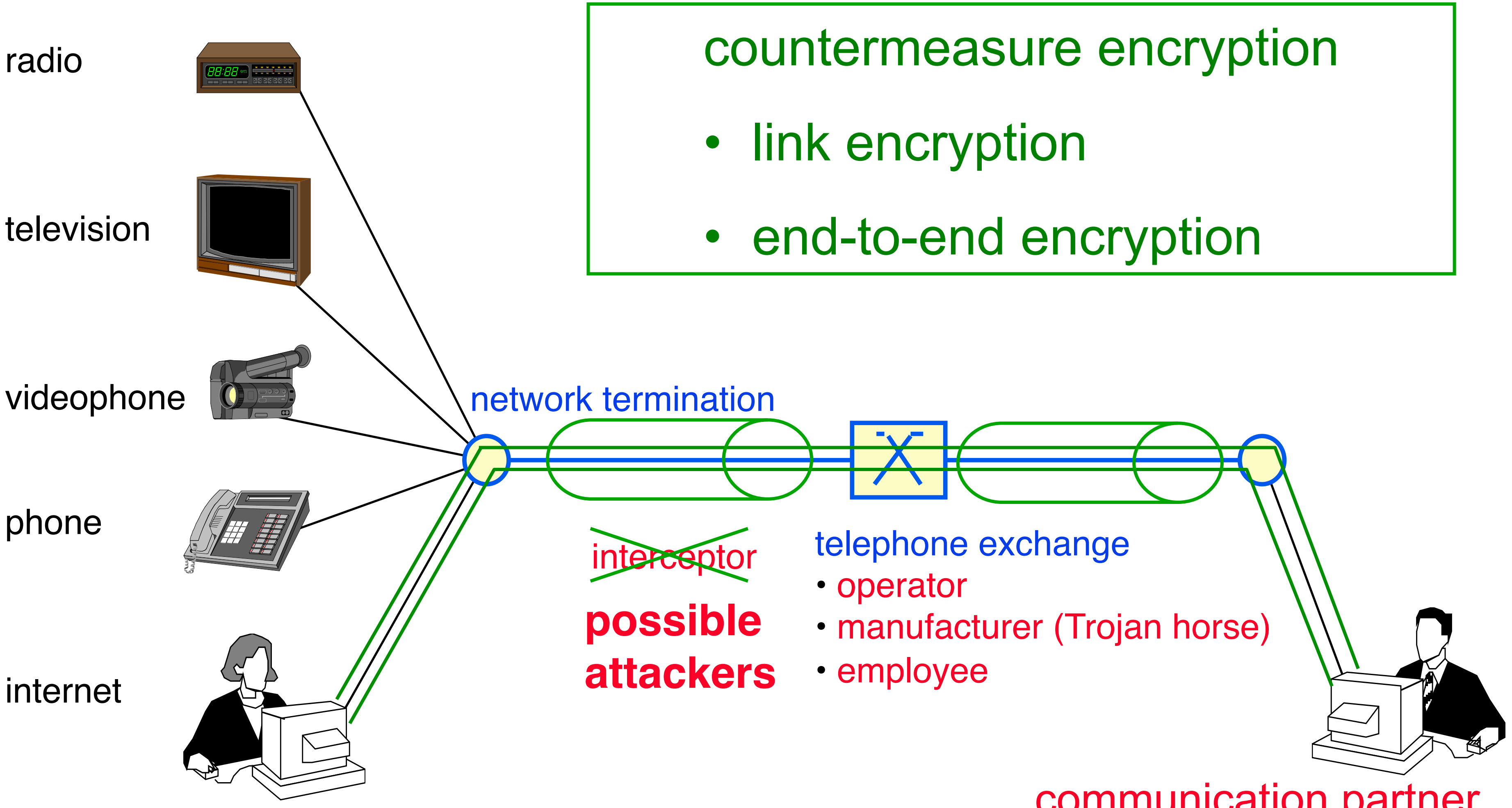
possible attackers

telephone exchange
- operator
- manufacturer (Trojan horse)
- employee

Observability of users in switched networks

countermeasure encryption

- link encryption
- end-to-end encryption



radio

television

videophone

phone

internet

network termination

~~interceptor~~

possible attackers

telephone exchange
- operator
- manufacturer (Trojan horse)
- employee

communication partner

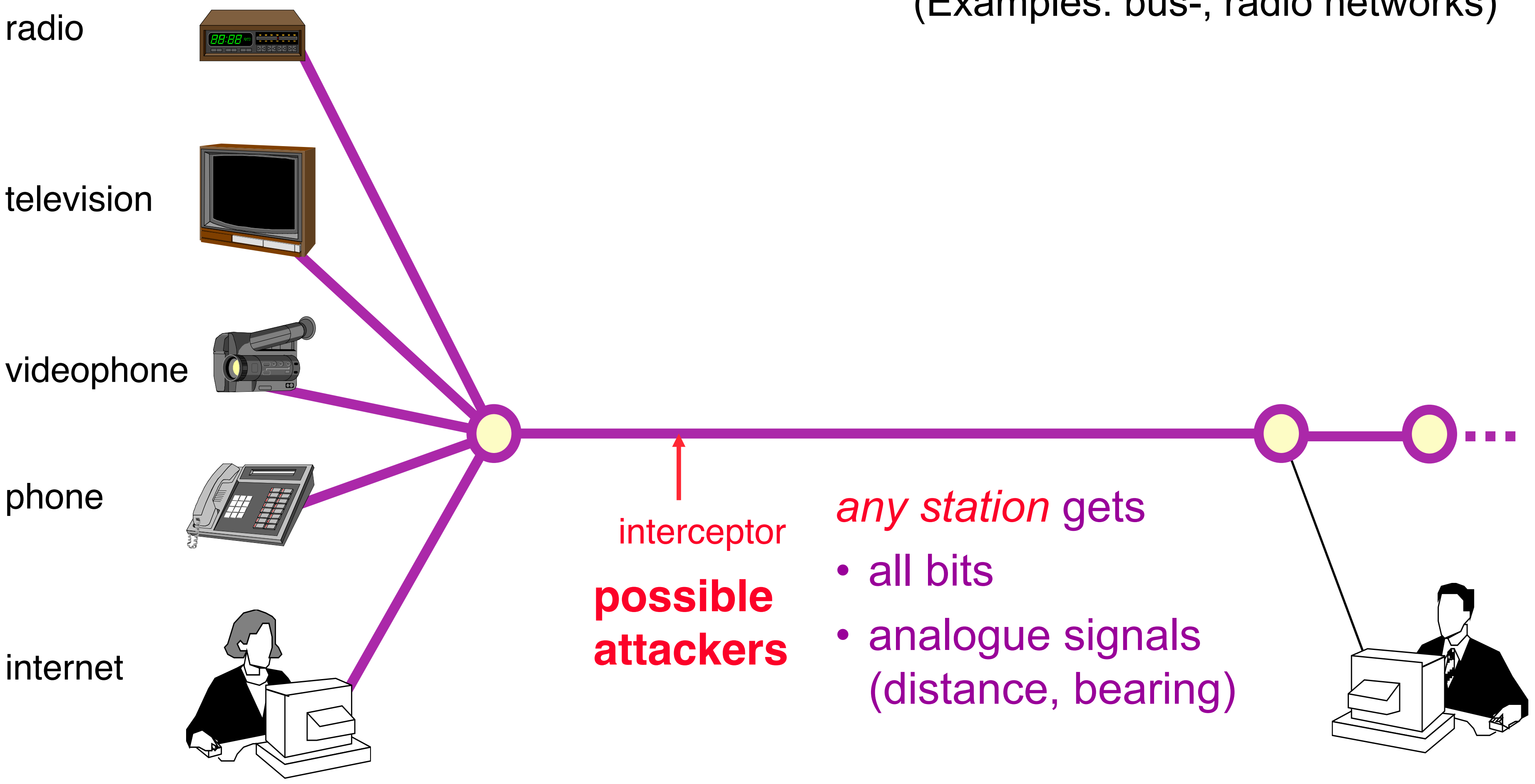
Problem: traffic data
who with whom?
when? how long?
how much information?

data on interests: Who? What?

Aim: "protect" traffic data (and so data on interests, too) so that they couldn't be captured.

Observability of users in broadcast networks

(Examples: bus-, radio networks)



interceptor

possible attackers

any station gets

- all bits
- analogue signals (distance, bearing)

Reality or fiction?

Since about 1990 reality

Video-8 tape 5 Gbyte

= 3 * all census data of 1987 in Germany

memory costs < 25 EUR

100 Video-8 tapes (or in 2003: 2 hard drive disks each with 250 G-Byte for < 280 EUR each) store all telephone calls of one year:

Who with whom ?

When ?

How long ?

From where ?

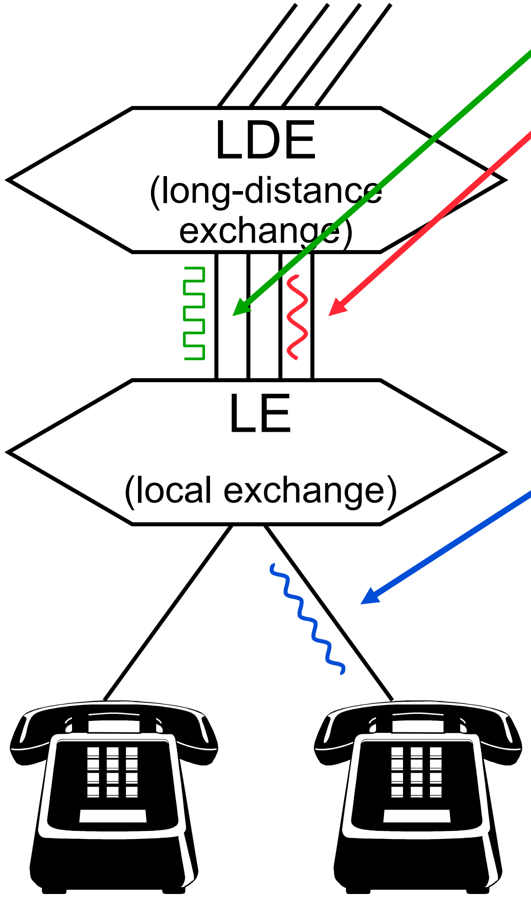
Excerpt from: 1984

With the development of television,
and the technical advance which
made it possible to receive and transmit
simultaneously on the same instrument,
private life came to an end.

George Orwell, 1948

Problems with exchanges

Unsolved problems by dedicated design of separate exchange:



+ encryption:

– message contents

– connection data, if speaker identification or ⊂ message contents

Trojan horse vs. add-on equipment: see below

Interception of participant’s terminal line (to scramble the signals is expensive and ineffective, encryption of the analogue signals is not possible):

- – message contents (content of calls)
- – connection data
  - number of the callee
  - speaker identification or ⊂ message contents

Mechanisms to protect traffic data

Protection outside the network

Public terminals

– use is cumbersome

Temporally decoupled processing

– communications with real time properties

Local selection

– transmission performance of the network

– paying for services with fees

Protection inside the network

Attacker (-model)

Questions:

- How widely distributed ? (stations, lines)
- observing / modifying ?
- How much computing capacity ? (computationally unrestricted, computationally restricted)

Unobservability of an event E

For attacker holds for all his observations B: $0 < P(E|B) < 1$

perfect: $P(E) = P(E|B)$

Anonymity of an entity

Unlinkability of events

if necessary: partitioning in classes