OPODIS 2003: 7th International Conference on Principles of Distributed Systems
December 10-13 2003
La Martinique, France

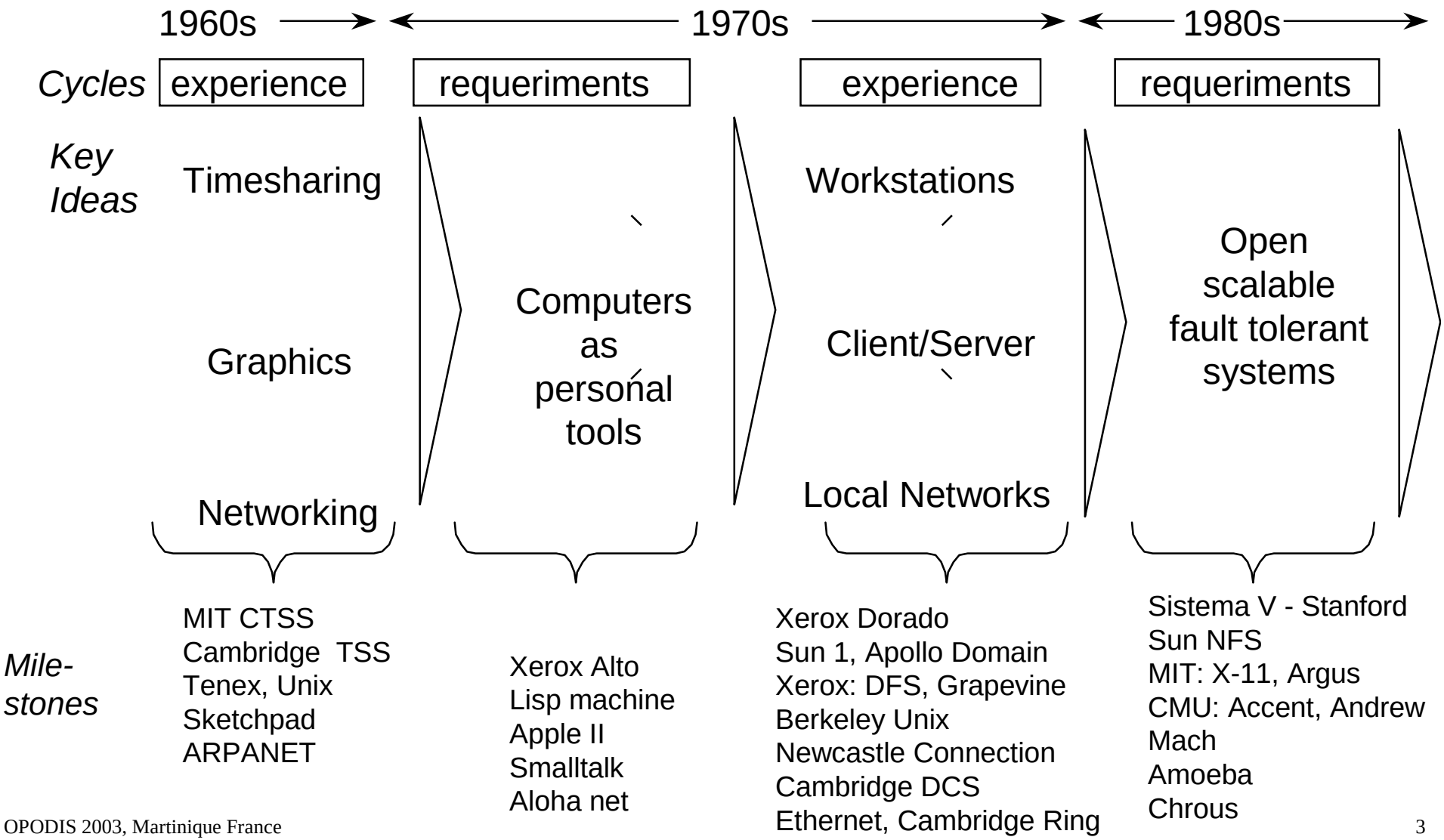
Distributed computing and information security

Roberto Gómez Cárdenas
rogomez@itesm.mx
http://webdia.cem.itesm.mx/ac/rogomez

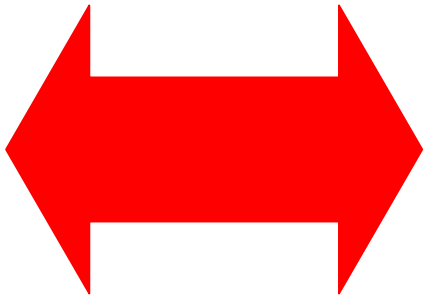
Presentation Outline

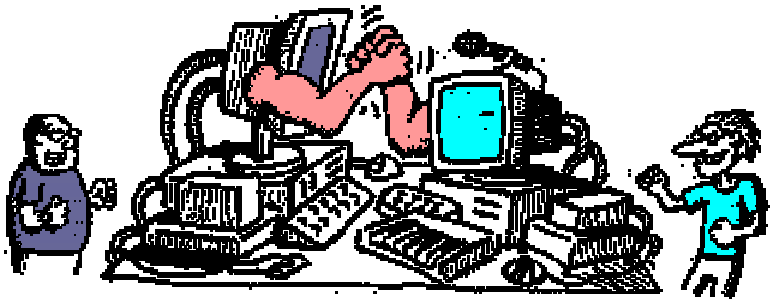
- Distributed systemas concepts
- Security features
- Cryptography basics
- Distributed protocols used for secure computer networks.

History


1960s
1970s
1980s
Cycles
experience
requeriments
experience
requeriments
Key Ideas
Timesharing
Graphics
Networking
Computers as personal tools
Workstations
Client/Server
Local Networks
Open scalable fault tolerant systems
Mile-stones
MIT CTSS
Cambridge TSS
Tenex, Unix
Sketchpad
ARPANET
Xerox Alto
Lisp machine
Apple II
Smalltalk
Aloha net
Xerox Dorado
Sun 1, Apollo Domain
Xerox: DFS, Grapevine
Berkeley Unix
Newcastle Connection
Cambridge DCS
Ethernet, Cambridge Ring
Sistema V - Stanford
Sun NFS
MIT: X-11, Argus
CMU: Accent, Andrew
Mach
Amoeba
Chrous

Computer Security and Distributed Computing





Distributed Computing

- Probably means many things to different people.
  - Client/Server
  - Cooperative processing
  - Distributed database
  - Protocols

A definition

A collection of autonomous computers linkeb by a network, with sofware designed to produce and integrated computing facility (i.e. distributed software).

Others definitions

"A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusuable"

Leslie Lamport

"A distributed system is one that stops from getting any work done when a machine you've never never heard of crashes"

Distributed Systems (Ed. Sape Mullender)
edition 1, ACM Press 1989

Key Characteristics

- Resource sharing
  - hardware components
    - disks and printers
    - shared for convenience and to reduce costs
  - software defined entities
    - files, windows, databases and other data objects
    - essencial requiriment in many computer applications
- Openness
  - characteristic that determines wheter the system can be extended in varius ways
  - a system can be open or closed with respect to hardware extentions

Key Characteristics

- Concurrency
  - when several processes exist in a single computer we say that they are excuted concurrently
- Scalability
  - distributed systems operate effectively and efficiently at many different sacles
    - smallest practicable distributed system: two workstations
- Fault tolerance
  - it the one of the system component's fails, this continues to work
  - two approaches
    - hardware redundancy
    - software recovery

Key Characteristics

- Transparency
  - concealment from the user and the application programmer of the separation of components in a distributed sytem
  - the system is perceived as a whole rather than as a collection of indepnedent components

Examples

- Distributed Unix
  - extension of the original Unix system design to include support for interprocess communication
  - BSD: sockets
- Wide area network applications
  - internet: IPV6, Internet-2, Internet-3
- Multimedia information access and conferencing applications
  - computer aide
- Commercial applications
  - airlines sytems for seat reservation and ticketing
  - security!!!

Computer Security

- Computer security protects your computer and everything associated with it
  - protects information you've stored in your system
  - computer security is often colled informaticon security
- X/OPEN definition of information technology security
  - IT security is the state of an IT system in which the risks of the IT system's applications because of the relevant threats are reduced to an acceptable level by taking appropriate measures
  - The porpouse of IT security is to protect assets against threats

Computer security aspects

- Confidentiality (secrecy)
  - do not allow information to be disclosed to anyone who is not authorized to access oit
- Integrity (accuracy)
  - the system must not corrupt the information or allow any unathorized malicious or accidental changes to it
  - network communications variant: authenticity
- Availability
  - the computer system's hardware and software keeps working efficiently and the system is able to recover quickly and completely if a disaster occurs

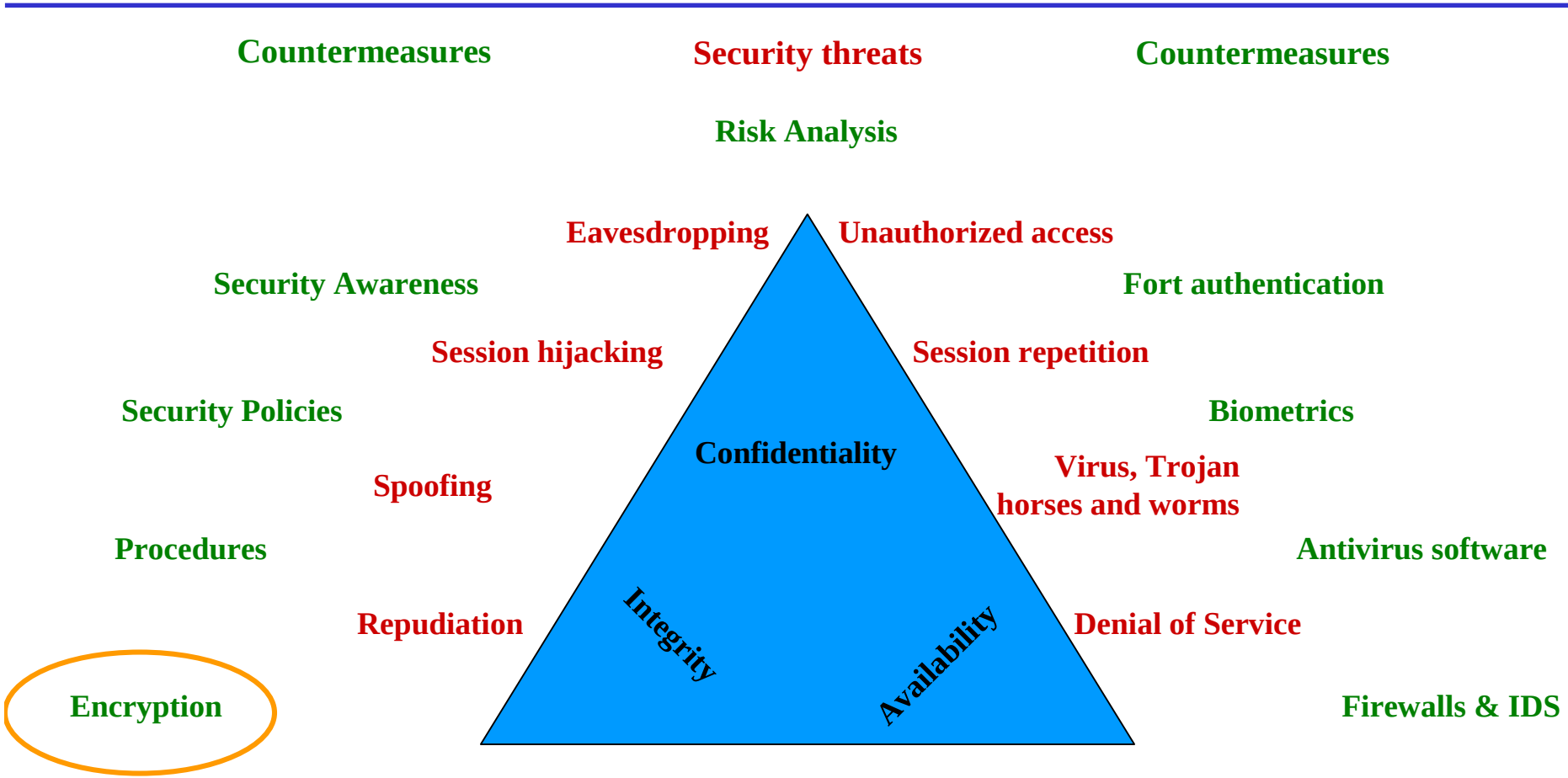
Other problems to be solved

- authentication - protect info origin (sender)
- non repudiation - protect from deniability
- user identification - ensure identity of users
- access control - control access to info/resources

Attacks

- interception - of information-traffic, breaches confidentiality
- interruption - of service, availability
- modification - of info, i.e. integrity
- fabrication - of information, destroys authenticity

Network Security



Cryptology

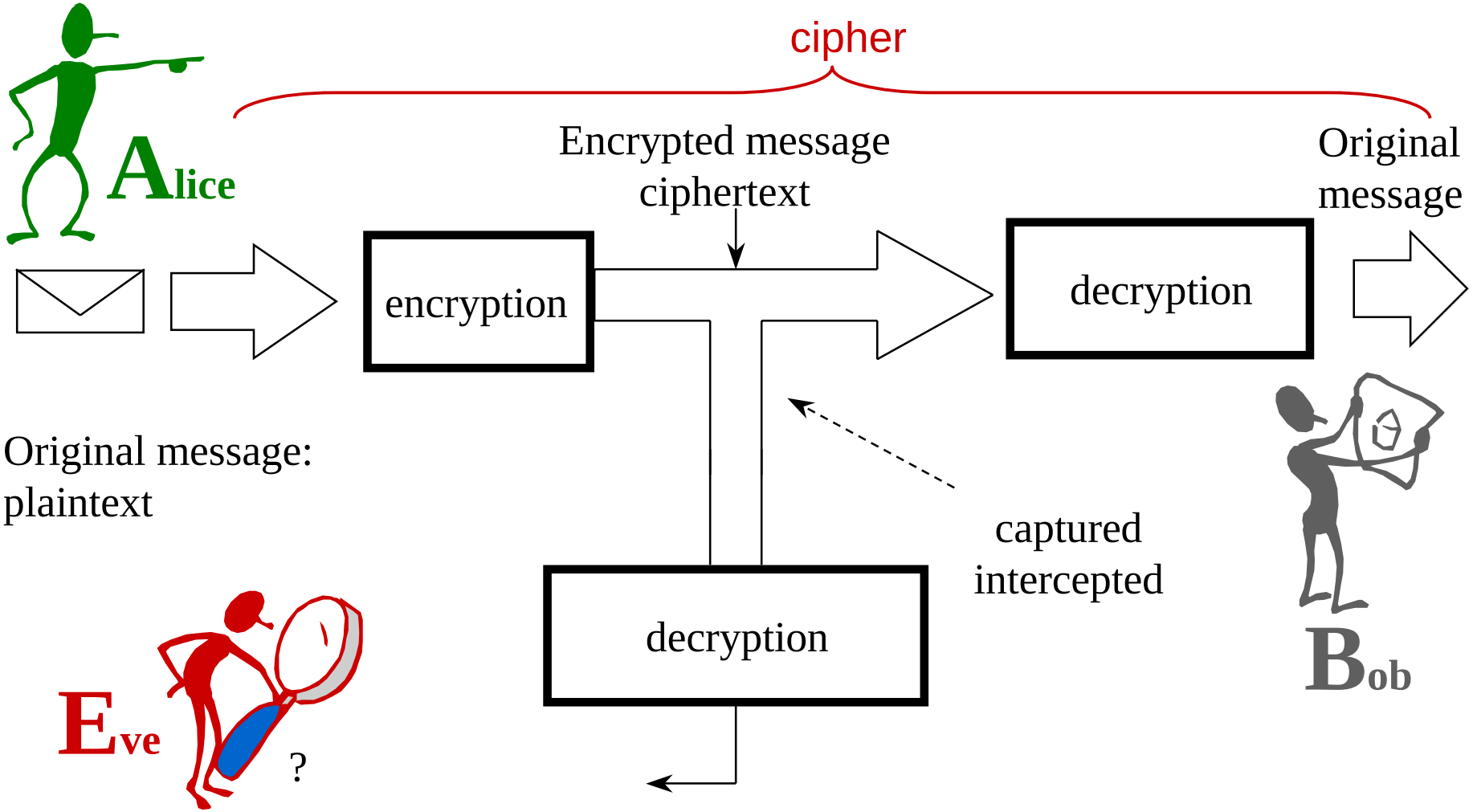
- The science of developing secret codes and/or the use of those codes in encryption systems.
- Divided in
  - cryptography
  - cryptanalysis

Steganography (covert channels)

What Cryptography can do?

- Secrecy (encryption)
- Authenticity (signature/encryption)
- Integrity (signature/encryption)
- Non-repudiation (signature)
- Why?
  - Encryption: only the authorized party can understand the encrypted message.
  - Signature: allow people to verify the authenticity of the message.

Terminology



Classical Ciphers

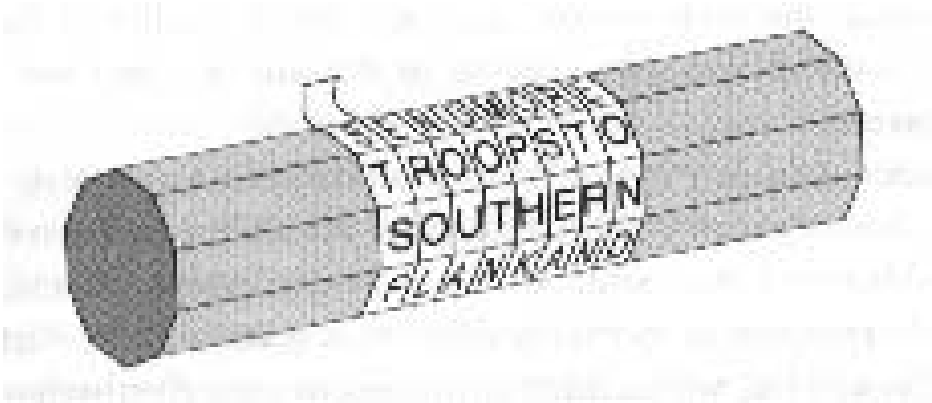
- First ciphers
  - Romans time – XX century
- Two techniques
  - transposition
  - substitution

Transposition ciphers

T R A N S P O S I T I O N

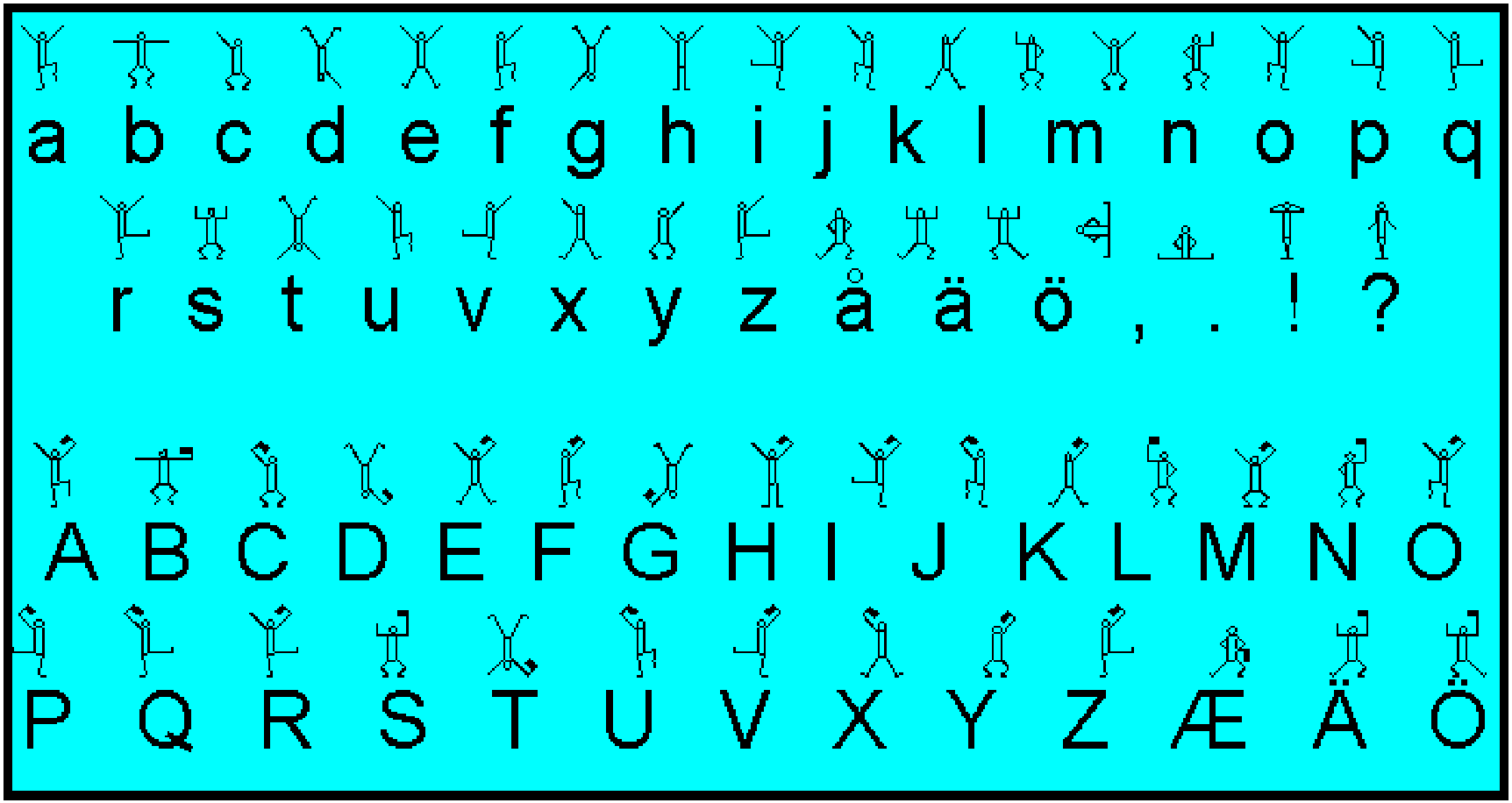
S I N O I O N A C T R P T

Example:

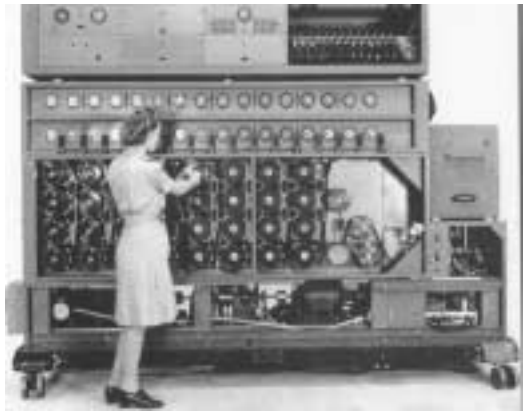


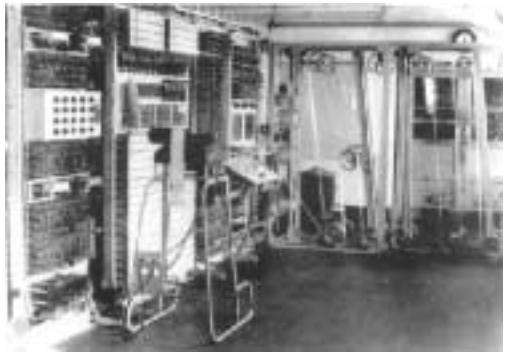
Plaintext: **SENDMORETROOPSTOSOUTHERNFLANKAND…**
Ciphertext: **STSFEROLNOUADOTNMPHKOSEARTRNEOND...**

Susbtitution ciphers



Cipher machines





Computer encryption

- ASCII

Plaintext:

HELLO = 1001000 1000101 1001100 1001100 1001111

- Transposition:

Plaintext:

HELLO = 1001000100010110011001001100 1001111

Ciphertext:

LHOEL = 1001100100100010011111000101100 1100

- Bits transposition:

Original letter: 1001000 **Encrypted letter: 0010010**

Using a key

- We can use a key to encrypt the message
- Example: key = DAVID.

DAVID = 1000100 1000001 1010110 1001001 1000100

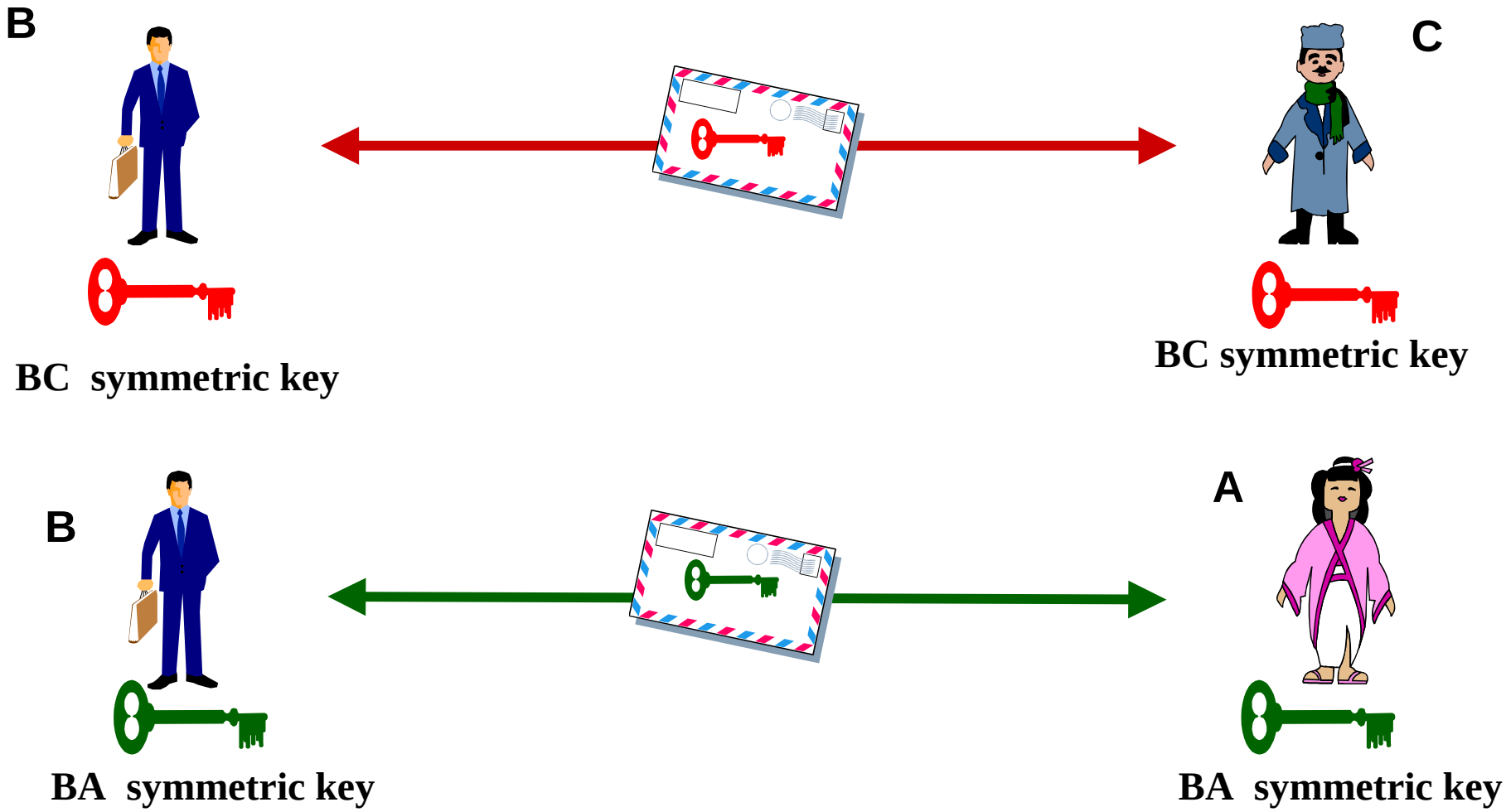
- Using xor

Plaintext:	**HELLO**
ASCII Plaintext:	**1001000100001011001100100110010011111**
Key:	**1000100100000110101101001001100010 0**
Ciphertext:	**0001100000010000110100000101000101 1**

Modern Cryptography

- Symmetric ciphers
  - classical ciphers
  - Shared-key cryptography
- Asymmetric ciphers
  - Diffie y Hellman, 1976
  - public-key cryptography

Shared-key cryptography



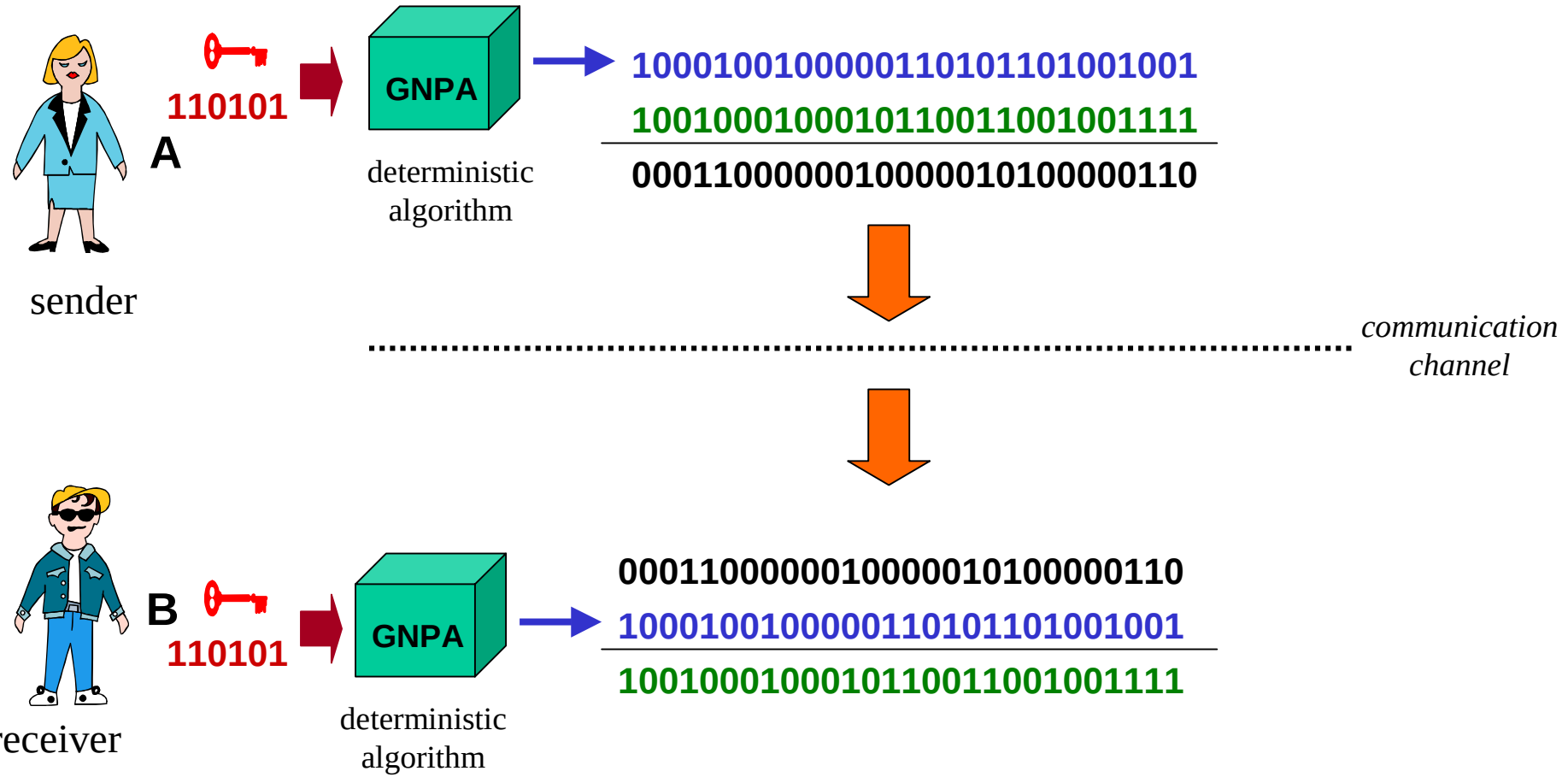
Symmetric Ciphers

Private-key (symmetric) ciphers are usually divided into two classes.

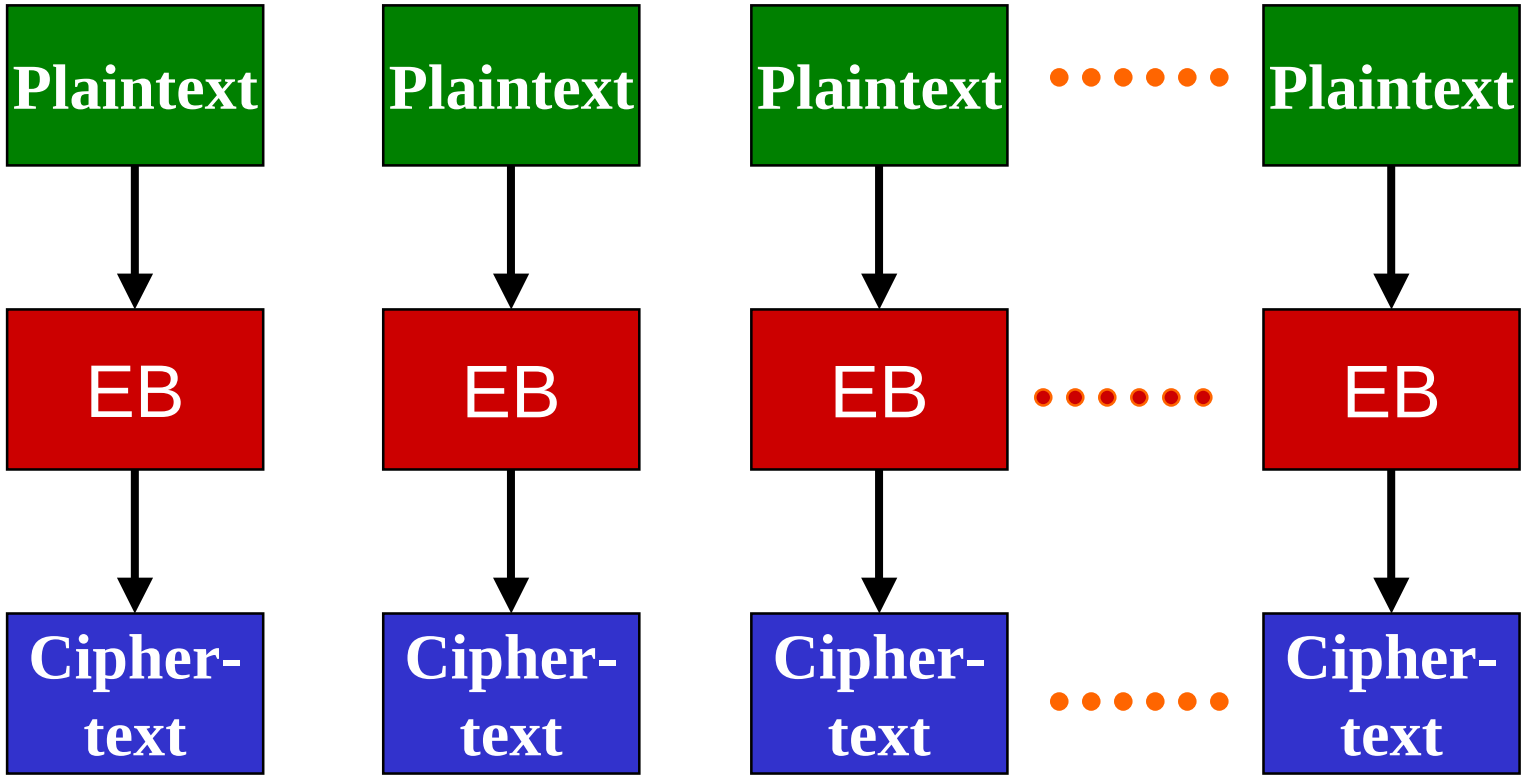
- Stream ciphers
- Block ciphers

Stream ciphers

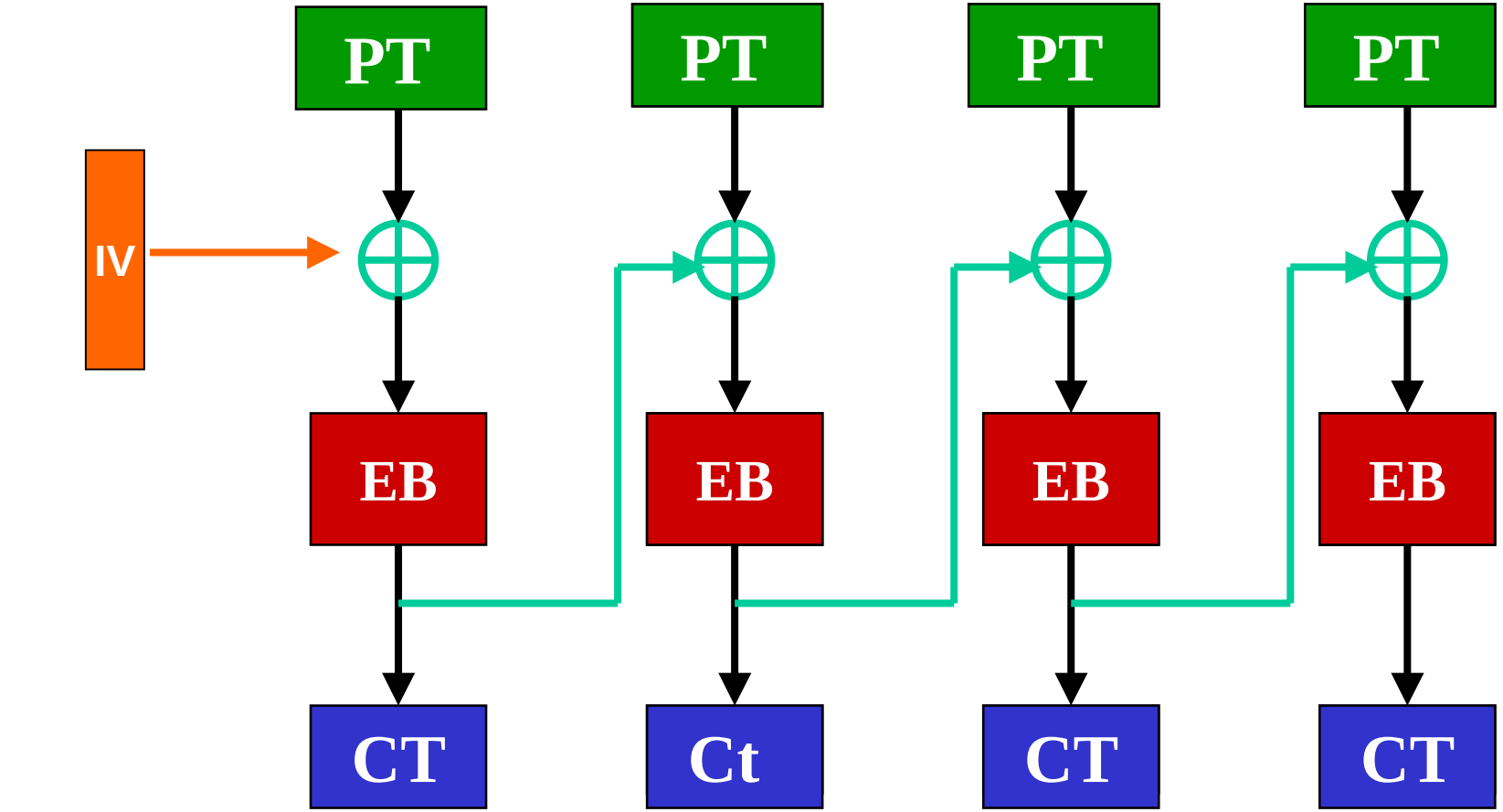
Plaintext: HOLA
HOLA = 10010001000101100110010011111



Block ciphers



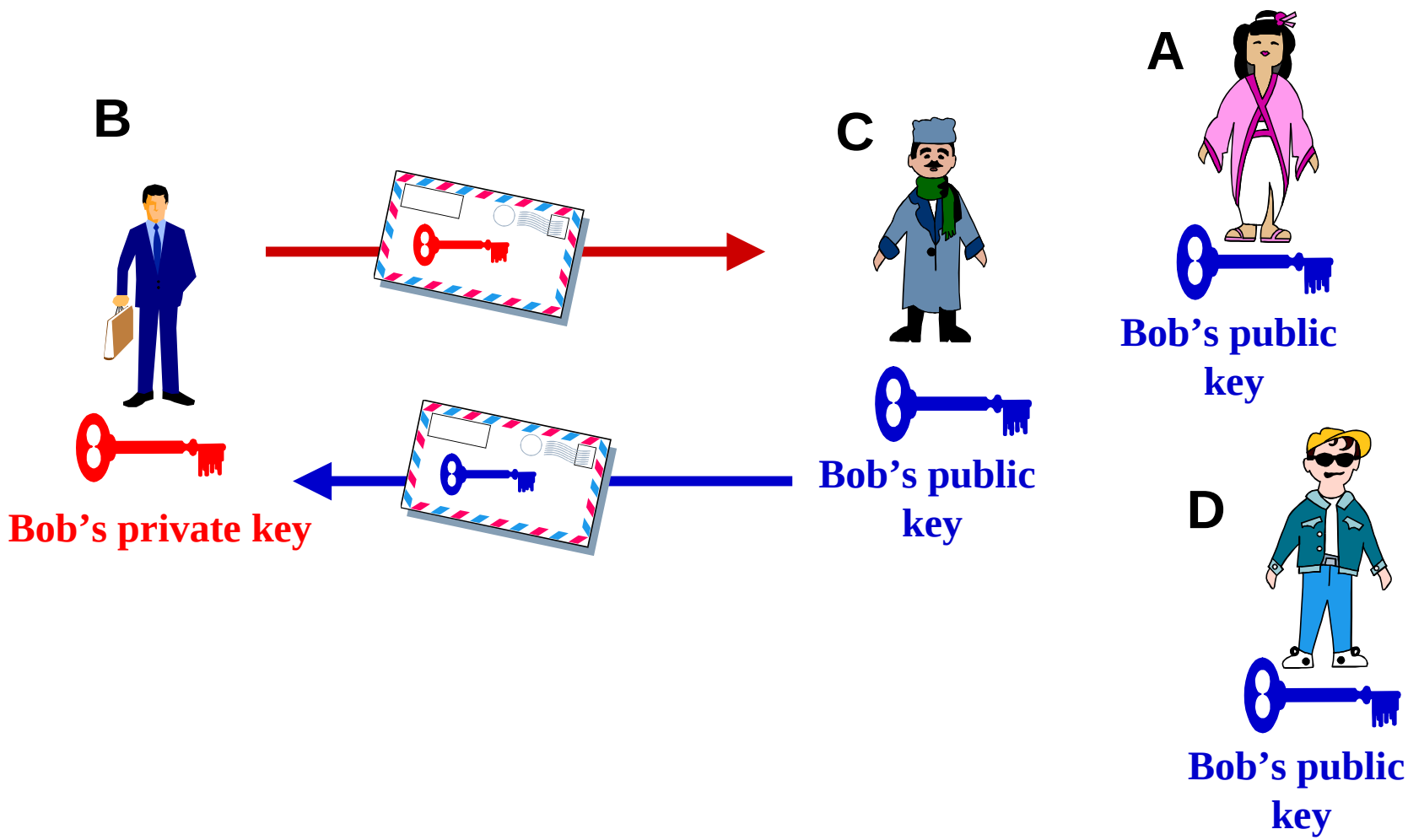
Cipher Block Chaining (CBC)



Symmetric encryption problems

- Key distribution
- Key management
- No digital signature

Public-key cryptography



One way function

- A function that takes m buts as input and produces n bits.
- A hash (message digest) is a one way function
- It is considered a function because it takes an input message and produces and output
- It is considered one-way because it’s not practical to figure out what input corresponds to a given product.

One-way functions

- MD2
- MD4
- MD5
- SHA-1
- SHA-256
- RIPE MD-160
- HMAC
- N-Hash
- Havalk

MD5 example

```
rogomez@armagnac:464>more toto
ULTRA SECRETO

Siendo las 19:49 hrs del dia 19 de noviembre de 1999
pretendo anunciar que se termino el presente texto
para pruebas de programas hash.

Atte;

RGC
rogomez@armagnac:465>md5 toto
MD5 (toto) = 0c60ce6e67d01607e8232bec1336cbf3
rogomez@armagnac:466>
```

rogomez@armagnac:467>more toto
ULTRA SECRETO

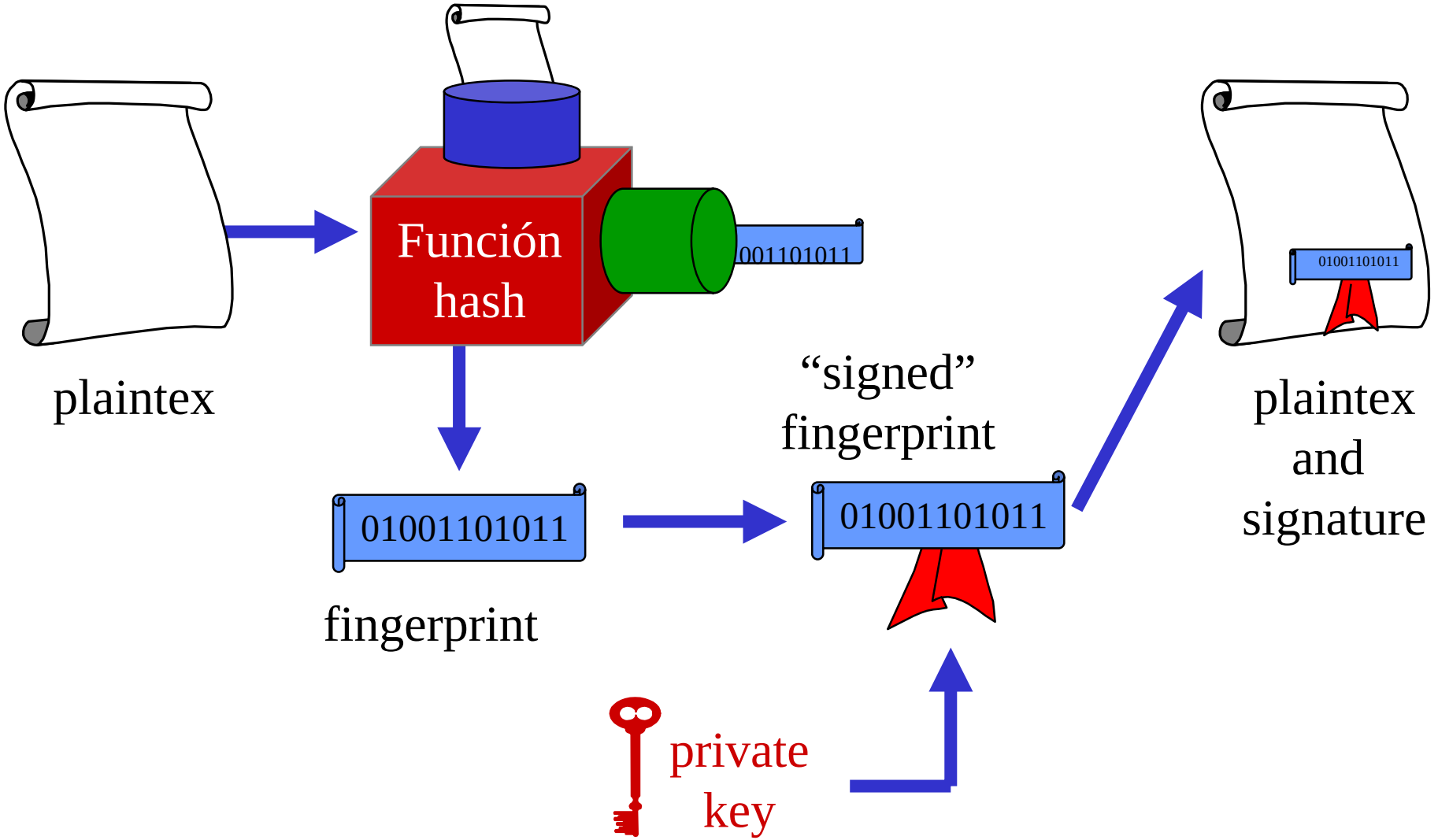
Siendo las 19:49 hrs del dia 19 de noviembre de 1999
pretendo anunciar que se termino el presente texto
para pruebas de programas hash.

Atte

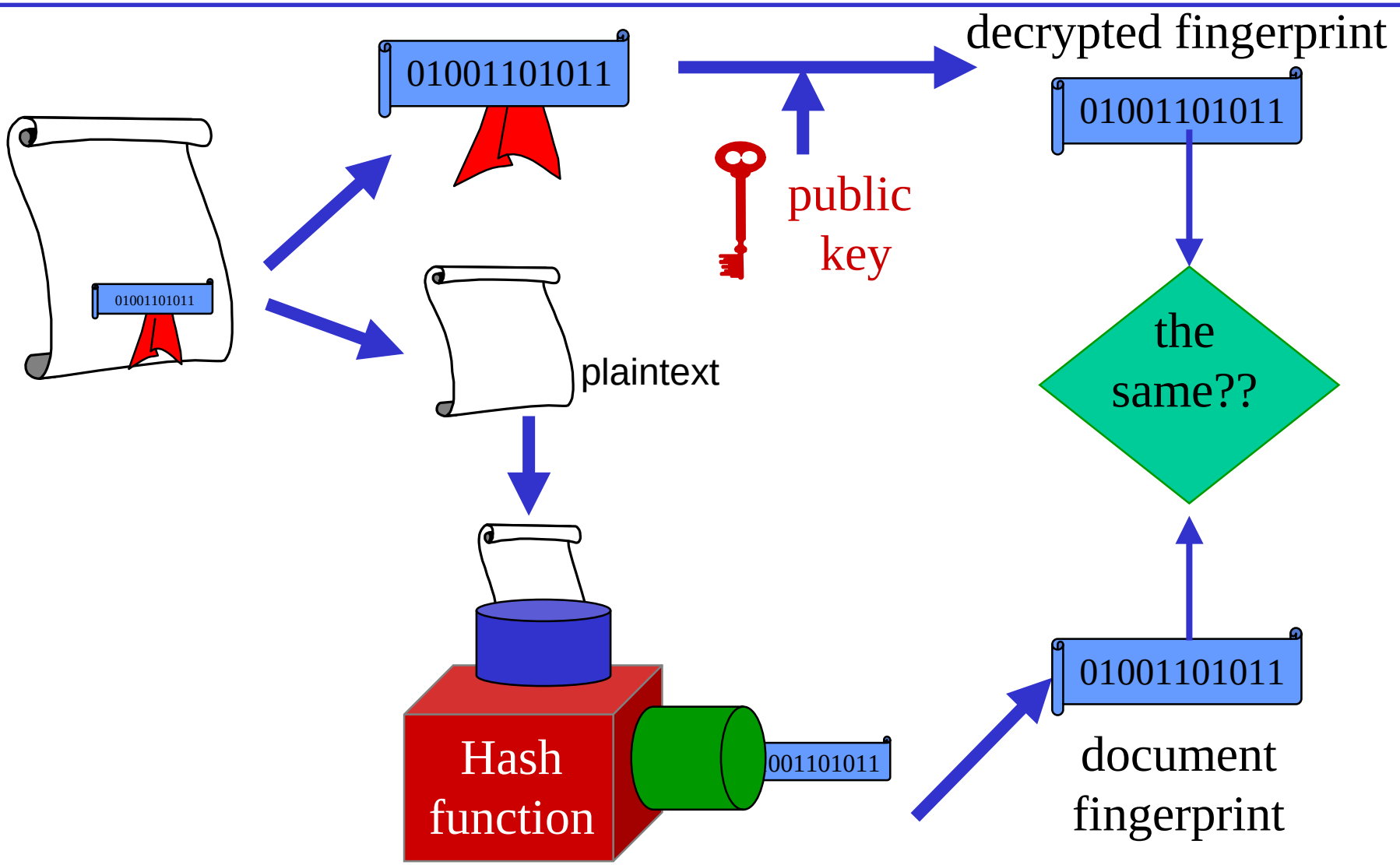
RGC
rogomez@armagnac:468>md5 toto
MD5 (toto) = 30a6851f7b8088f45814b9e5b47774da
rogomez@armagnac:469>

Integrity and fingerprints

Digital signature (send)



Digital signature (receive)

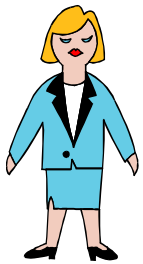


Distributed security

- How can distributed systems help?
  - Key distribution
  - Private communication
  - Authentication protocols

Diffie Hellman

q (prime number) and α $(\alpha < q)$



A

Key for Alice and Bob: K

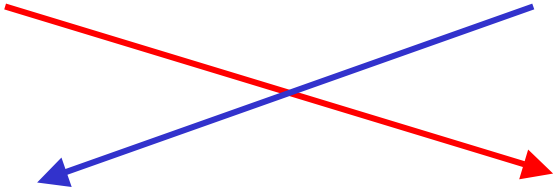
B

X_A $(X_A < q)$

$Y_A = \alpha^{X_A} \bmod q$

X_B $(X_B < q)$

$Y_B = \alpha^{X_B} \bmod q$



Y_A Y_B

Alice computes the key:

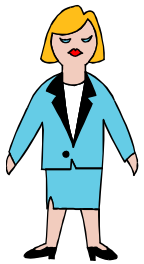
$$K = (\mathrm{Y_B})^{X_A} \bmod q$$

Bob computes the key

$$K = (\mathrm{Y_A})^{X_B} \bmod q$$

Diffie Hellman example

$q = 53 \quad \alpha \quad = 2 \; (2 < 53)$



A

Key for Alice and Bob: 21

B

$X_A = 29 \;\; (29 < 53)$

$Y_A = \; 2^{29} \bmod 53$

$= 45 \quad \bmod 53$

$X_B = 19 \quad (19 < 53)$

$Y_B = \; 2^{19} \bmod 53$

$= 12 \bmod 53$

$Y_B \; (12)$

$Y_A \; (45)$

Alice computes:

$K = \; 12^{29} \bmod 53 = 21 \bmod 53$

Bob computes:

$K = 45^{19} \bmod 53 = 21 \bmod 53$

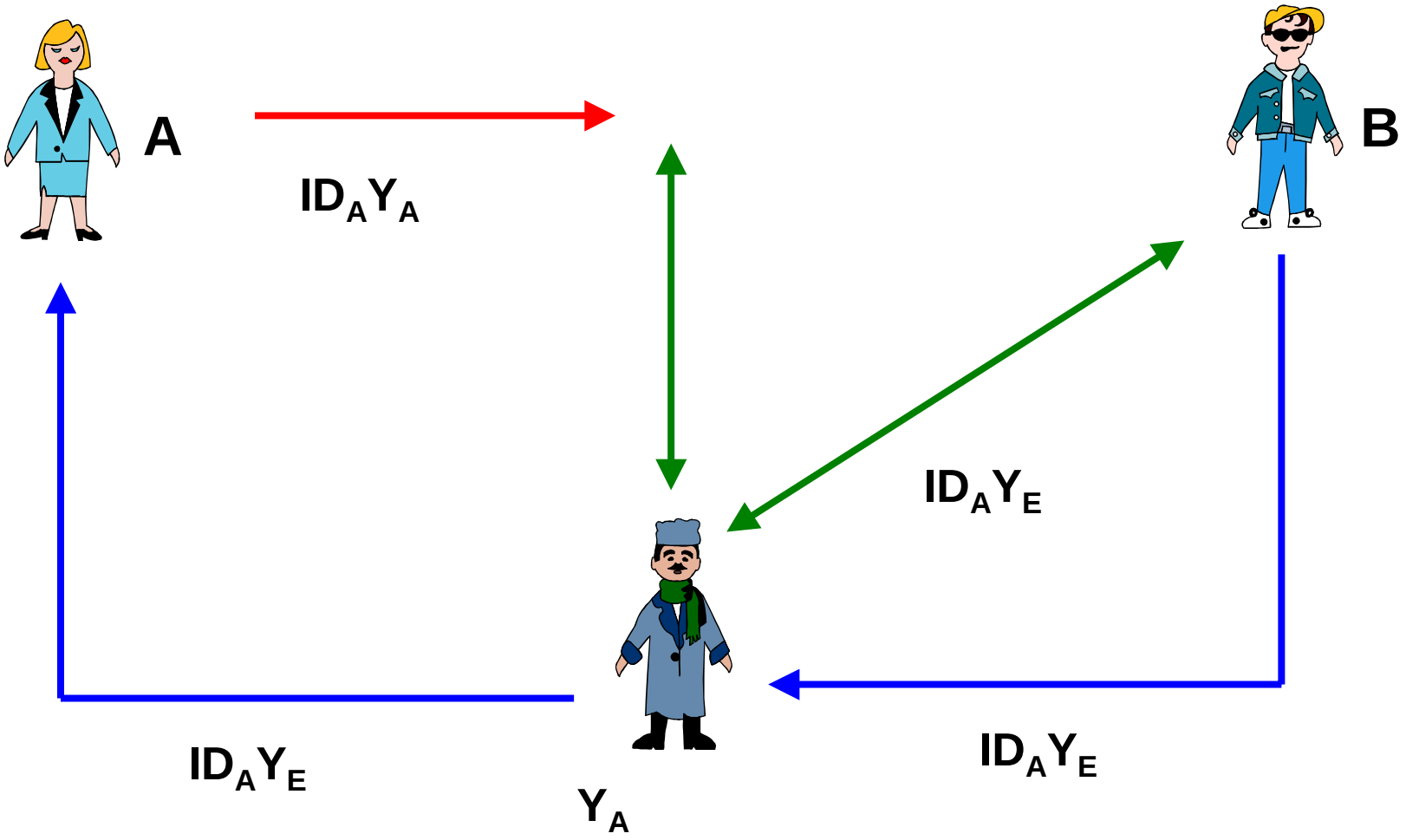
Aventages

- Secret keys are created only when needed.
  - There is no need to store secret keys for a long period of time, exposing them to increased vulnerability
- The exchange requires no preexisting infrastructure other than an agreement on the global parameters

Weakness

- It does not provide any information about the identities of the parties.
- It is computationally intensive
  - as a result is vulnerable to a clogging attack in which an oponnent request a high number of keys
  - the victim spends considerable computing resources doing useless modular exponentation rather than real work
- It is subject to a man in the middle attack

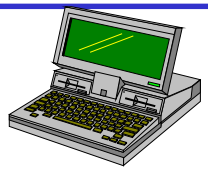
DF man in the middle attack



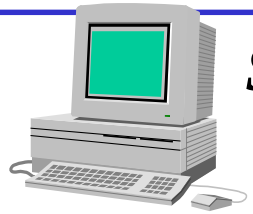
Oakley Key Determination Protocol

- Is a refinament of the DH key exchange algorithm
- It employs a mechanism known as cookies to thwart attacks
- It enables two parties to negotiate a group; this, in essence, specifies the global parametets of the Diffie-Hellman key exchange
- It uses nonees to ensure against replay attacks
- It enables the exchange of Diffie-Hellman public key value
- It authenticates the DH exchange to thwart MIM attack

Private communications



Client



Server

Hello

Hello

How are you

Fine

No authentication
No privacy
No encryption

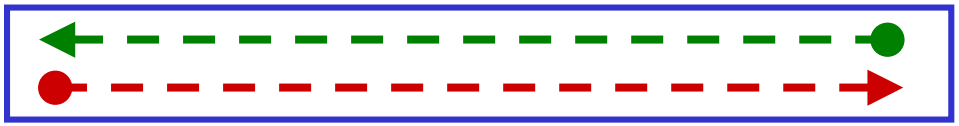
Creating
a symmetric
key

I want to pay

I send you my public key

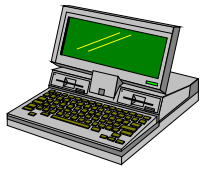
I send you a key encrypted with your public key

Lets talk in a secure way

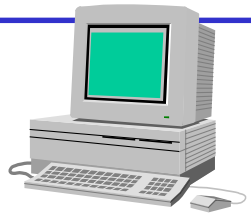


Encrypted communication with the client's generated key

Another scenario



Cliente

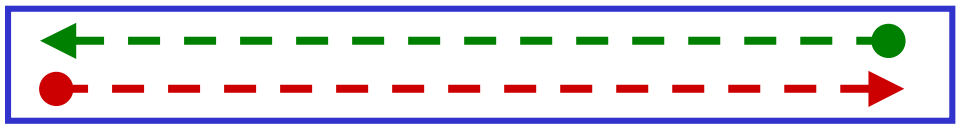


Servidor

Lets talk in a secure way
these are the protocols and ciphers I understand

I choose these protocol and cipher. I send you my public key, a digital cetificate and a random number

Using your public key I encypted a random symmetric key



Encrypted communication with the key sent by the client and a hash for messaje authentication

Authentication of People

- User authentication consists of a computer verifying that you are who claim to be
- There are three main techniques:
  - what you know
    - passwords
  - what you have
    - physical keys or ATM cards
  - what you are
    - biometric devises

Cryptographic authentication protocols

- Can be much more secure than either password-based in adrress-based authentication.
- Basic idea
  - Alice proves her identity to Bob by performing a cryptographic operation on a quantity Bob supliers
  - The cryptographic operation performed by Alice is based on Alice's secret

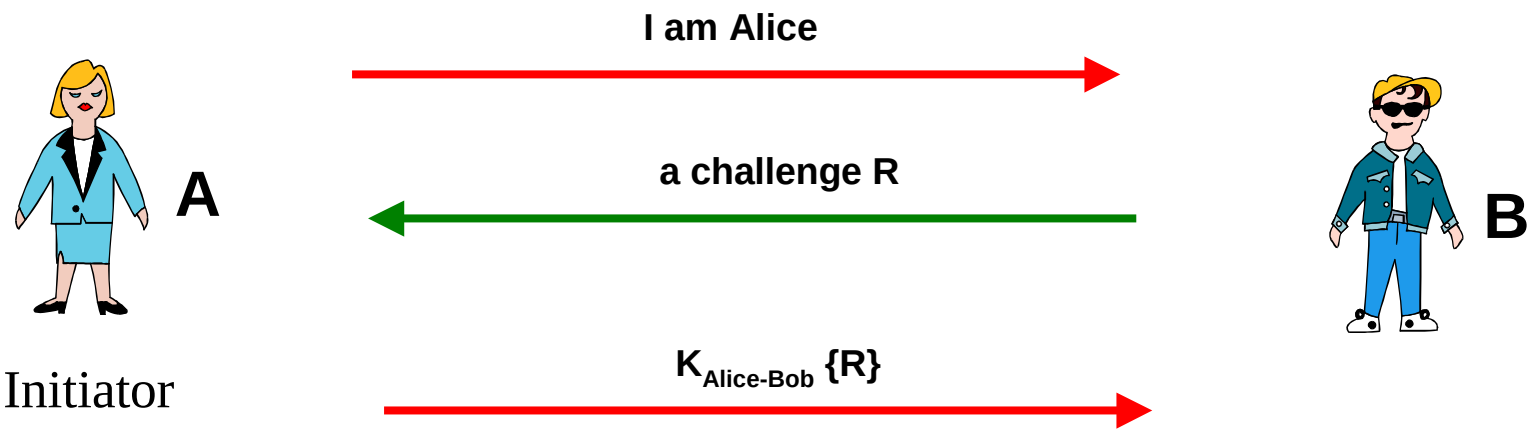
Authentication in communications

- Protocols includes an initial authentication handshake, and sometimes, in addition, integrity protection and/or encryption of the data.
- Alice and Bob wish to communicate
  - they need to know some information about themselves and about the other party
  - some of this information is secret
  - some is not, such as the names Alice and Bob
- Cryptographic authentication protocols are exemples of security handshakes
  - minor variants of secure protocols can have security holes

Only login

- A lot of existing protocols were designed in an environment where eavesdropping was not a concern and bad guys were not expected to be very sophisticated
- The authentication in such protocols generally consists of
  - Alice (the initiator) sends her name and password (in the clear) accross the network to Bob
  - Bob verifies the name and the password, and then communication occurs, with no further attention to security – no encryption, no cryptographic integrity protection
- A very common enhancement is to replace the transmission of the password with a cryptographic challenge/response

Shared secret

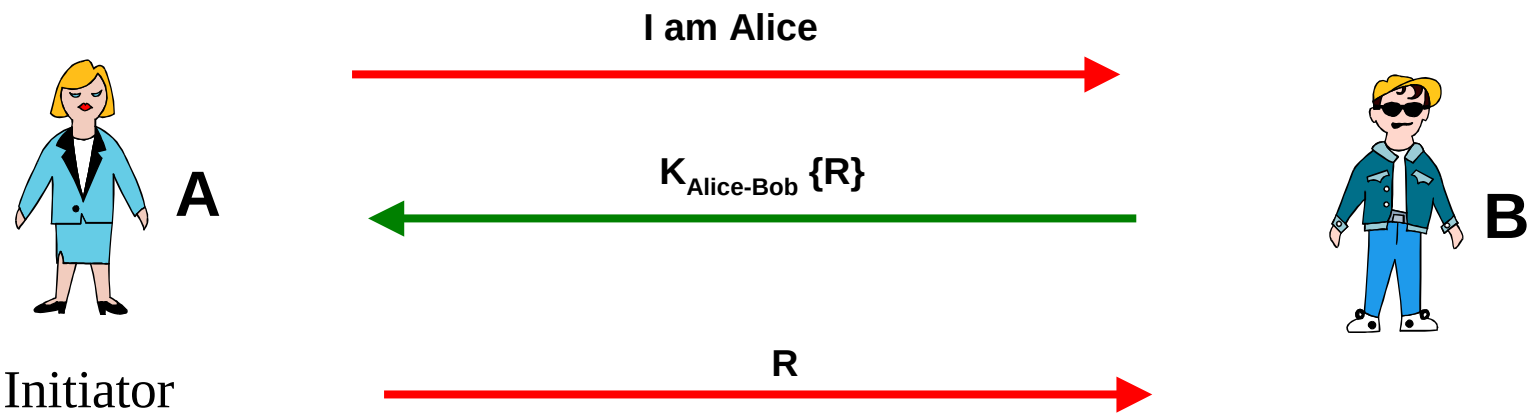


- $K_{Alice-Bob}\{R\}$: encryption of R with key $K_{Alice-Bob}$
- K is a symetric key used with some algorithm like DES or IDEA
- Is a big improvement over passwords in the clear.
- An eavesdropper cannot impersonate Alice based on overhearing the exchange, since next time will be a different challenge

Protocol weaknesses

- Authentication is not mutual
  - Bob authenticates Alice, but Alice does not authenticate Bob
- Eve can hijack the conversation after the initial exchange
  - assuming she can generate packets with Alice's source address
- Someone who reads the database at Bob can later impersonate Alice
  - in many cases it is difficult to protect the database at Bob
  - there might be many servers where Alice uses the same password

Shared secret first variant

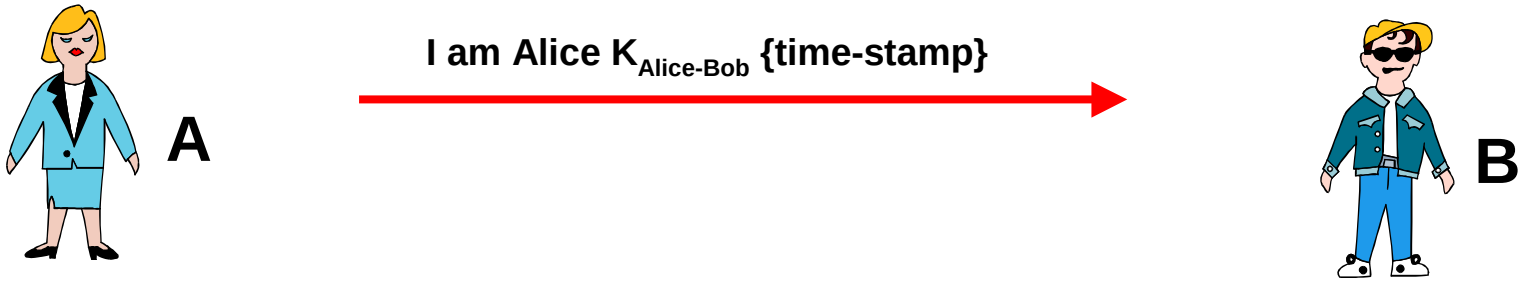


- It is a minor variant of previous protocol
- Requieres reversible cryptography
  - previous protocol can be done using a hash function
  - Alice has to be able to reverse what Bob has donde to R in order to retrieve R

Protocol features

- If R is a recognizable quantity, for instance a 32 bit random number padded with 32 zero bits to fill our encyption block
  - Eve can mount a password guessing attack by merely sending the message I am Alice and obtaining the $K_{Alice\text{-}Bob}\{R\}$
- If R is a recognizable quantity with lifetime, such as a random number conca

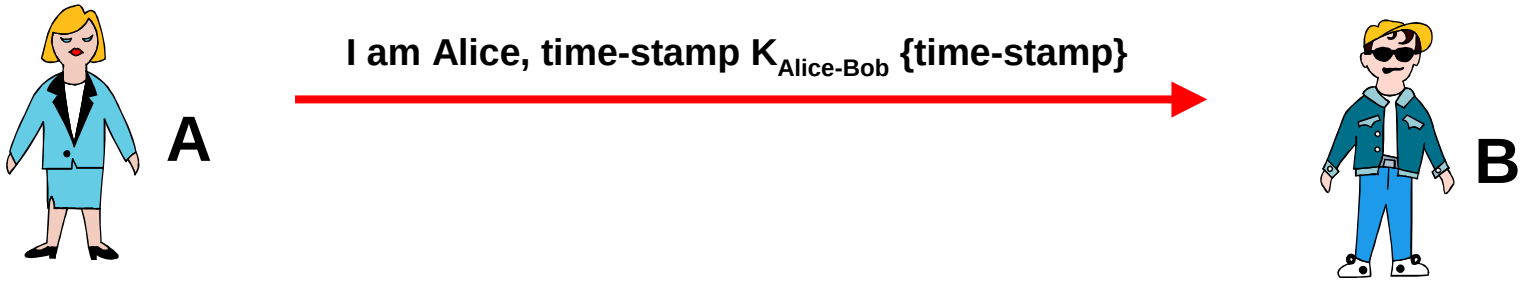
Shared secret third variant



Initiator

- It is a minor variant of previous protocol
- Requieres reversible cryptography
  - previous protocol can be done using a hash function
  - Alice has to be able to reverse what Bob has donde to R in order to retrieve R

Authentication based on high resolution time



Initiator

- Differences:
  - a hash function is used, rather than a reversible encryption scheme

One way public key

- Previous protocols are based on shared secrets
- Eve can impersonate Alice if she can read Bob's database
- If protocols are based in public key technology instead this cab be avoided
- Terminology

$$[R]_{Alice} = \text{Alice signs } R$$

transforms R using her private key

Two one way authenticacion protocolos

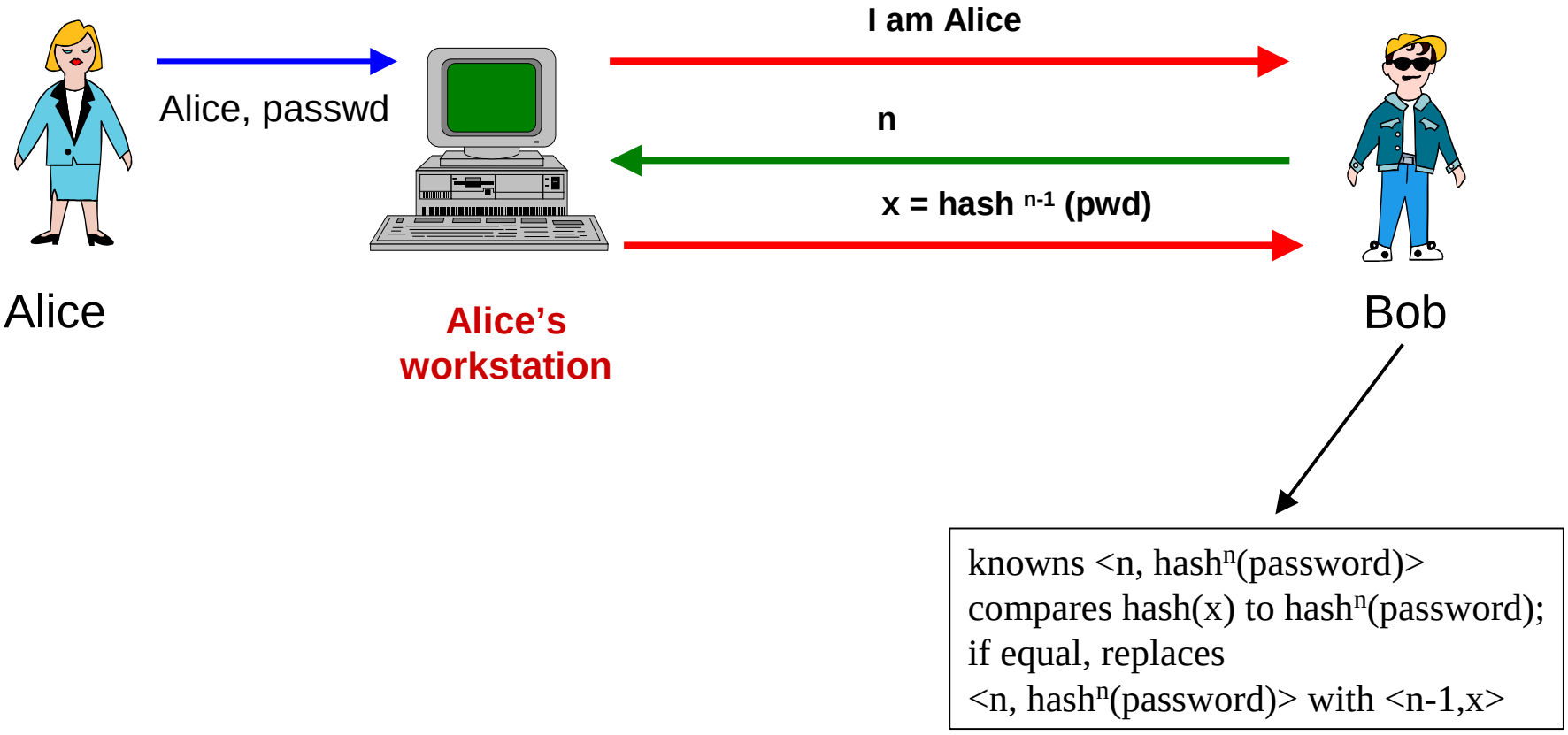
Initiator

Bob authenticates Alice based on her public key signature

Initiator

Bob authenticates Alice if she can decrypt a message encrypted with her public key

Lamport hash



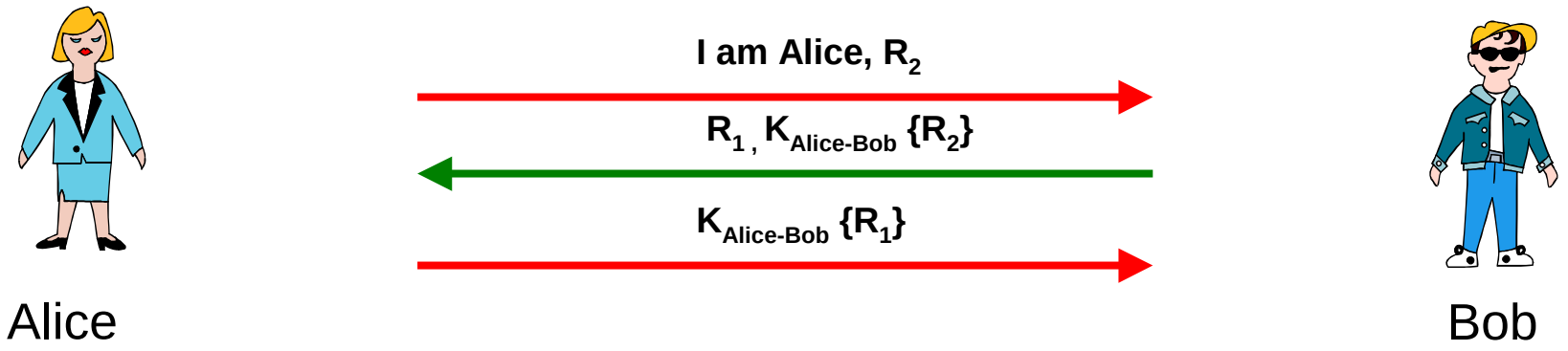
Human and paper environment

- The information <n, $hash^n$(password)> is installed at the server, all the values of $hash^i$(password) for i<n are:
  - computed
  - encoded into a typeable string
  - printed on a paper and given to Alice
- When Alice logs in, she uses the string and the top of the page, ande the crosses the value otr, using the next value the next time
- There is a depolyed version of Lamport's hash, known as S/KEY

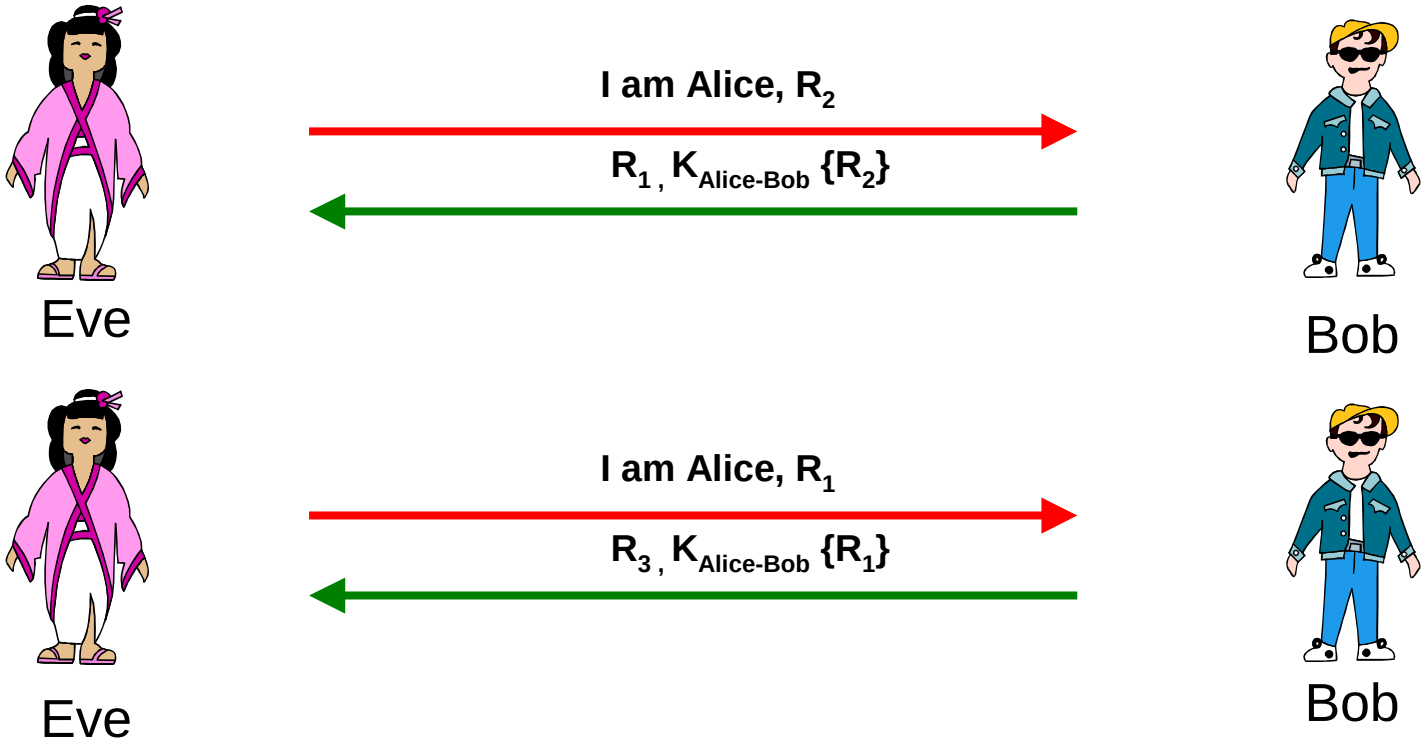
Mutual Authentication

Mutual authentication based on a shared secret $K_{Alice\text{-}Bob}$

Optimized mutual authentication based on a shared secret $K_{Alice\text{-}Bob}$



The reflection attack

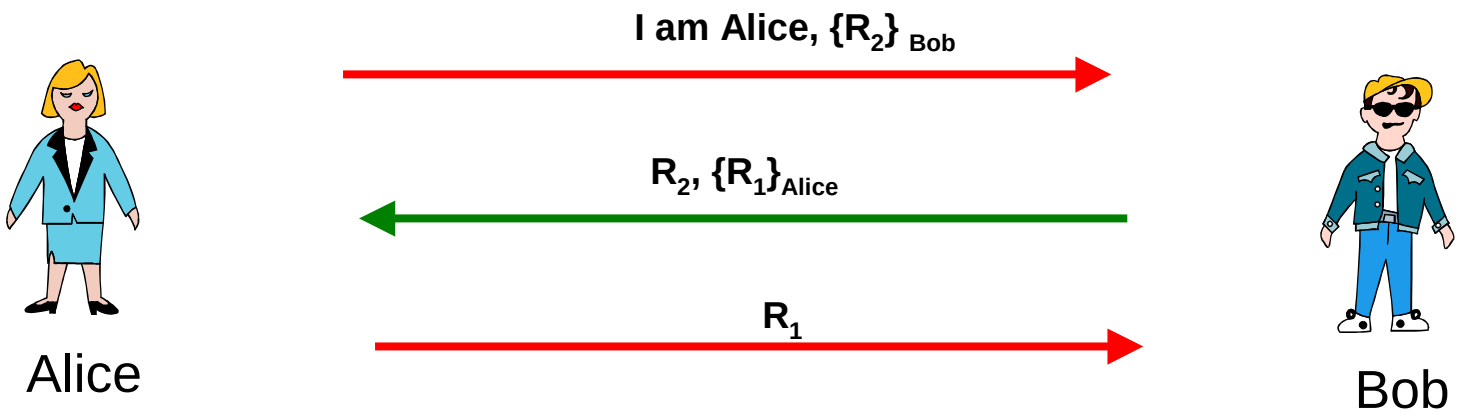


- Notes
  - Eve can't go any further with this session, because she can't encrypt R3
  - but now she knows KAlice-Bob {R1], so she can complete the first session

Fixing the protocol

- Principle
  - don’t have Alice and Bob do exactly the same thing
- Different keys
  - the key used to authenticate Alice be different from the key used to authenticate Bob
  - use two different keys used by Alice anda Bob at the cost of additional configuration and storage
- Different challengs
  - the challenge from the initiator (Alice) look different from the challenge from the responder
  - initator challenge be an odd number and responder challenge be an even number

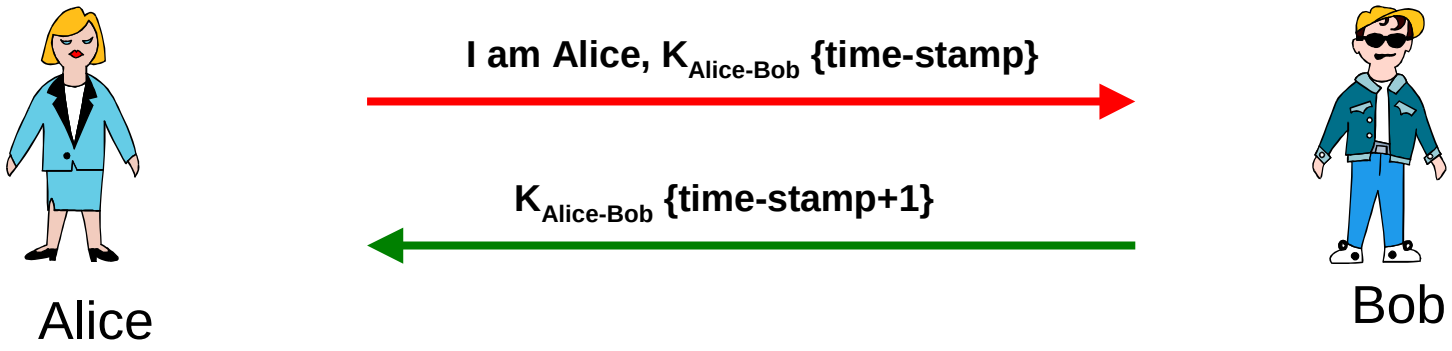
Public keys



- Assumption
  - Bob and Alice know each other's public key
- Challenges
  - How does Alice know Bob's public key
  - Alice's workstarion obtain Alic'es private key (password -> key)

Timestamps

- We can reduce the mutual authentication down to two messages by using timestamps instead of random numbers for the challenges.

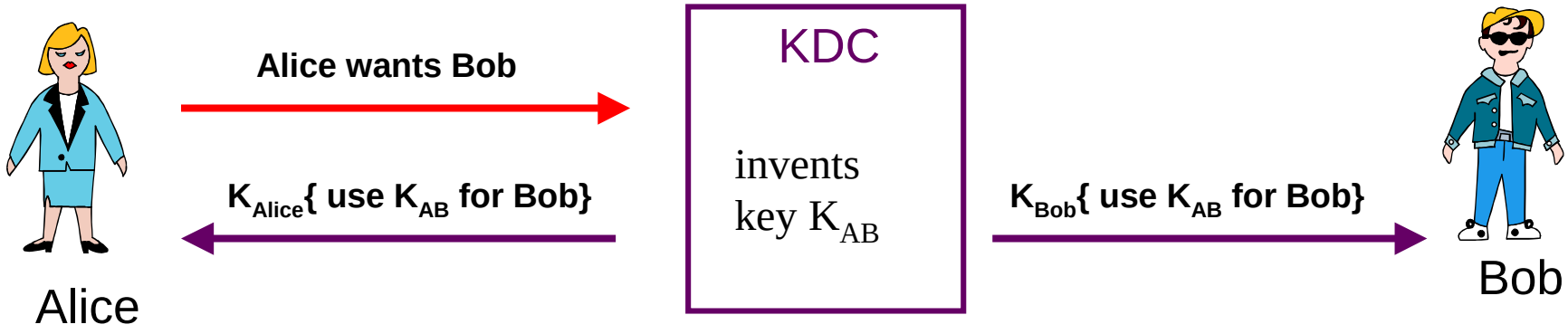


Trusting intermediaries

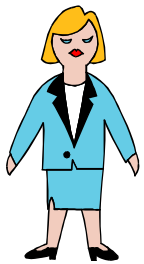
- Network: n nodes
  - each computer migth need to authenticate each other computer
  - each computer would need to know n-1 keys, one for each sytem in the network
  - if a new node were added to the network, then n keys would need to be generated,
- Solutions
  - KDC: Key Distribution Center
  - Certification Authorities
  - Multitrusted intermediaries

Key Distribution Center

- The KDC konws keys for all the nodes
- If a new code is installed in the network, only that new mode and the KDC need to be configured with a key for that node



Using a ticket



Alice

Alice wants Bob

K_{Alice}{ use K_{AB} for Bob}
ticket to Bob = K_{Bob}{use K_{AB} for Alice}

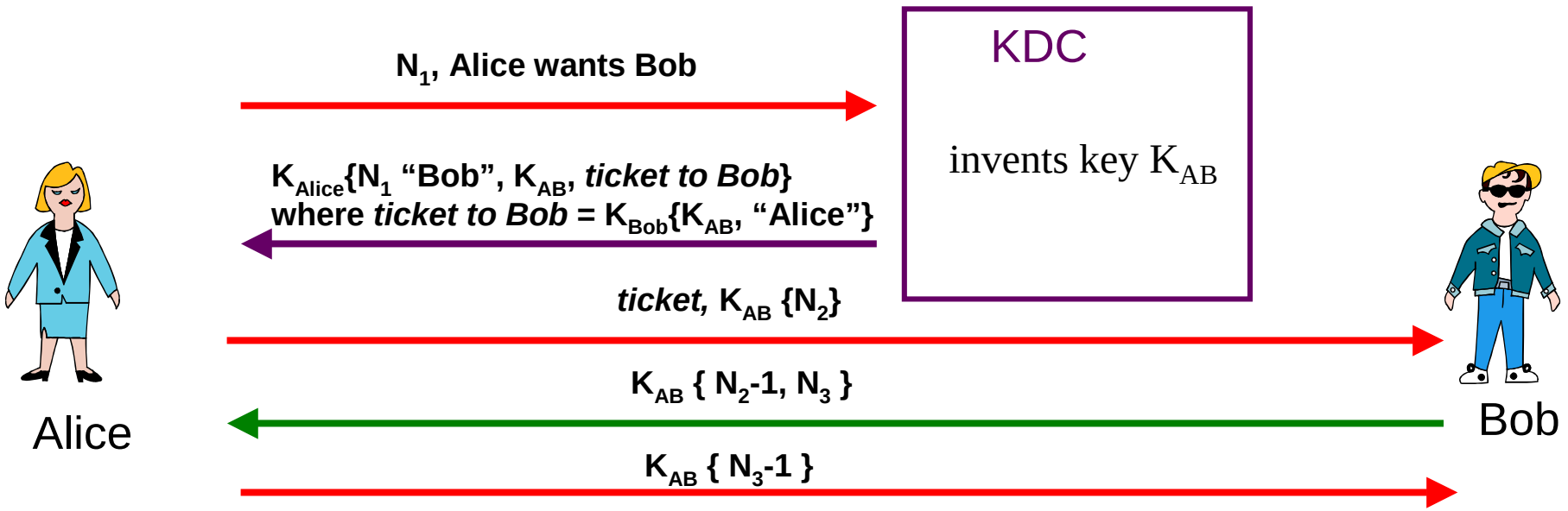
KDC

invents key K_{AB}

Bob

"I'm Alice", ticket = K_{Bob}{ use K_{AB} for Alice}

Neddham-Schroeder protocol



- N_i : nonce (number that is used only once)
- Purpose N_1: assure Alice that she is really talking to KDC
  - avoid case where Eve has stolen and old key of Bob and the message where Alice has requested a key for Bob

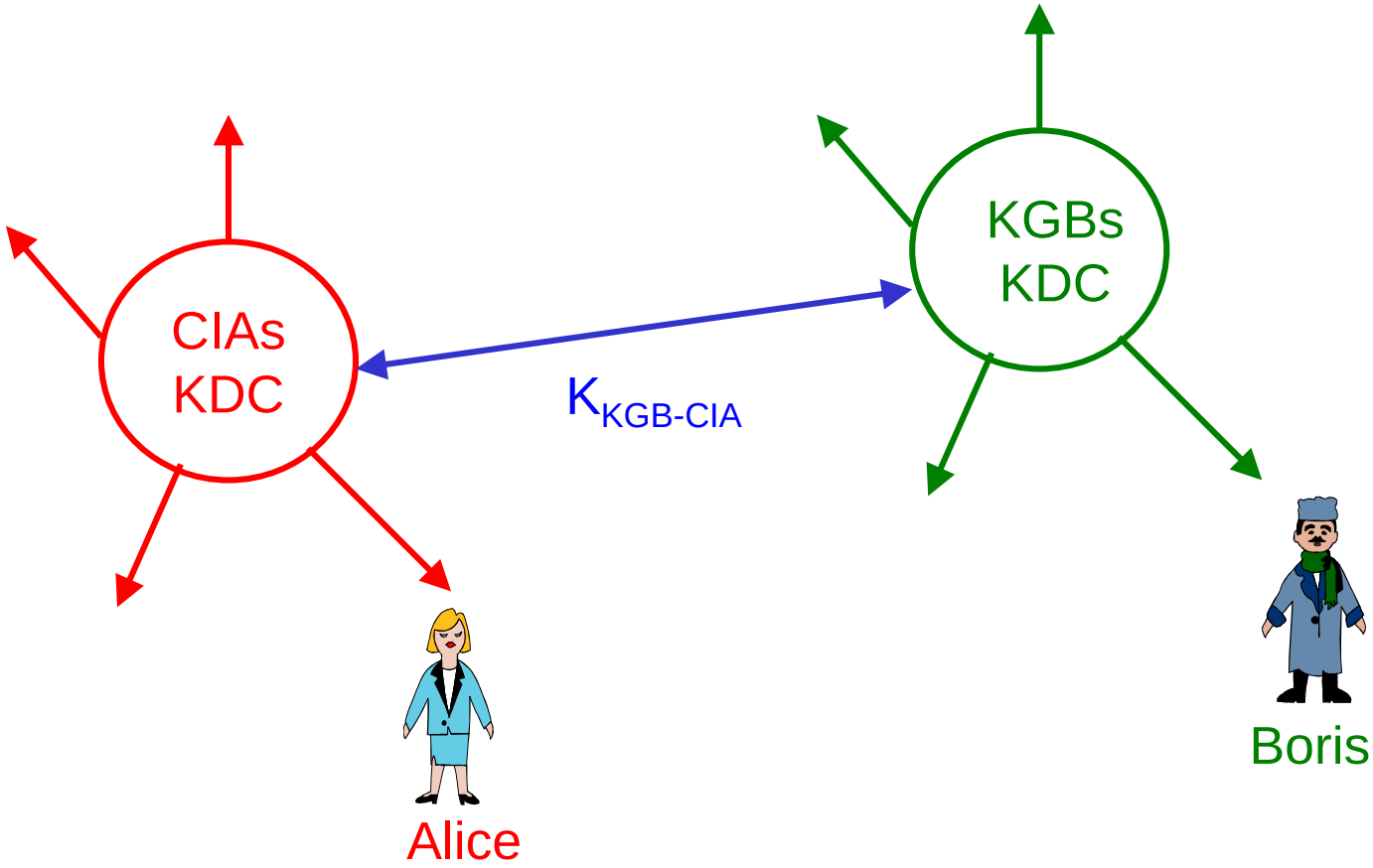
Certification Authorities

- Generates certificates which are signed messages specifying a name (Alice) and the corresponding public key
- All nodes need to be configured with the CA's public key
  - they can verify its signature on certificates
- Is the only key they need to know a priori
- Certificates can be stores in any convinient location
  - directory services or
  - each node can store its own certificate and furnish it as a part of the authentication exchange

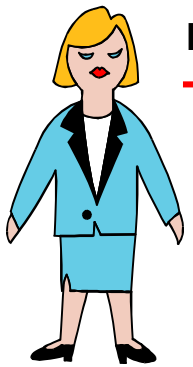
Multitrusted Intermediaries

- KDCs and CAs require a single administration trusted by all principlas in the system
- Problems
  - compromising KDC or CA can impersonate anyone to anyone
  - scale authentication schemes
- Solution
  - break thw world into domains
  - each domian has one trusted administration
  - if Alice and Boris are in the same domain, they authenticate as described previously, if not the authentication is still possible

Multiple domain example



Multidomaine protocol



Alice

let me talk to KGB's KDC

CIA's KDC

CIA's KDC generates K_{new}

$K_{CIA-KGB}$ {Alice from my domain wants to talk to you use K_{new}}

K_{Alice} {use key K_{new}}

KGB's KDC

Boris

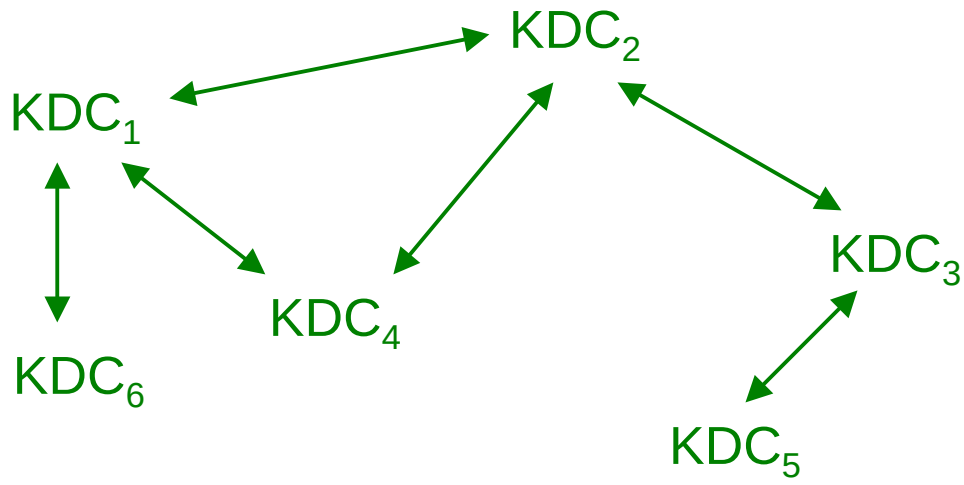
let me talk to Boris

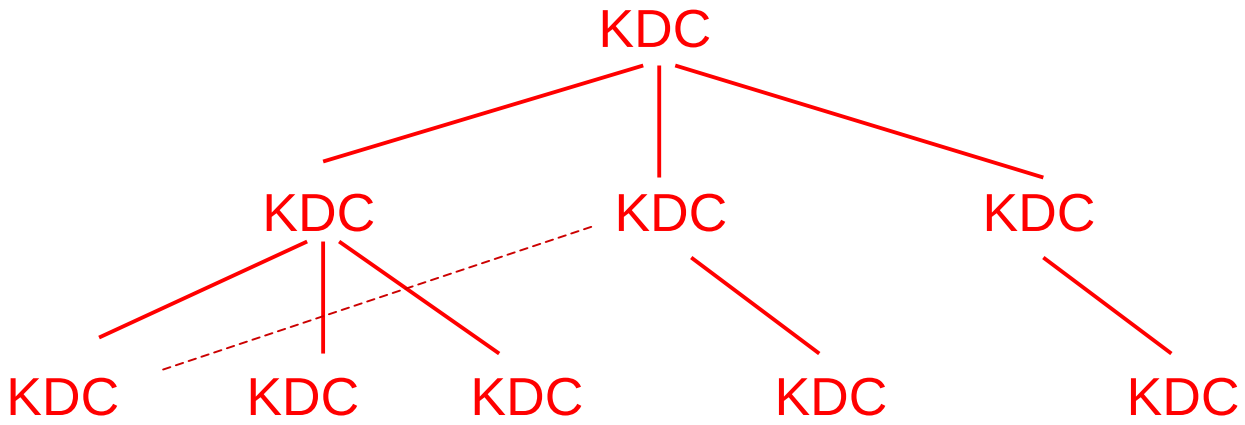
KDC generates $K_{Alice-Bob}$

K_{Boris} { talk to Alice from CIA; use $K_{Alice-Boris}$ }

K_{new} { use $K_{Alice-Boris}$ }

Topological KDC structures





What else?

- Distributed firewalls
- IDS sensor’s communication
- Formal proofs
- Integration of heterogeneous software
- Distributed steganography?

Conclusions

- Distributed protocols are required
- It is possible to use distributed alfgorithms in order to integrate security in information systems.

Old conception:
security enforces distributed systems

New conception:
distributed systems enforces security

OPODIS 2003: 7th International Conference on Principles of Distributed Systems
December 10-13 2003
La Martinique, France

Distributed computing and information security

Roberto Gómez Cárdenas
rogomez@itesm.mx
http://webdia.cem.itesm.mx/ac/rogomez