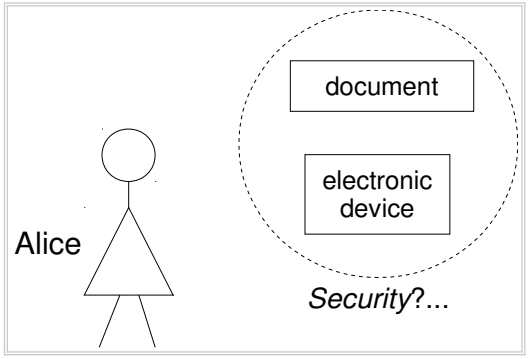
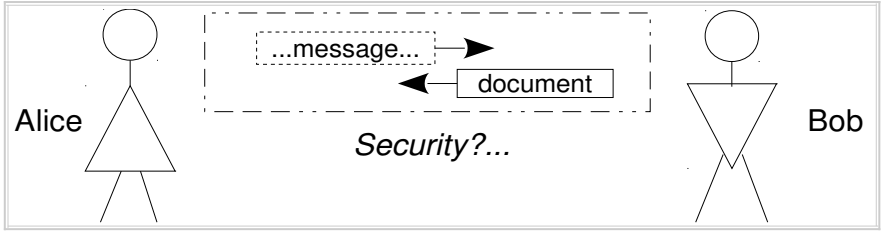
INFORMATION SECURITY

General Protection Techniques





Protection

- providing **access control** to resources (e.g. users' information)
  - by building secure channels
    - for communication
    - for storage
  - with properties
    - main: confidentiality, integrity, authentication
    - secondary: anonymity, forward secrecy, non-repudiation, etc.

Secure channel for communication

- cryptographically-protected conversation line between two identified subjects
- base, expected properties:
  - Authentication – assuring that each subject is talking to the genuine other
  - Integrity – assuring that deletion, change or addition of data is detected
  - Confidentiality – assuring that data is not understandable by anybody else

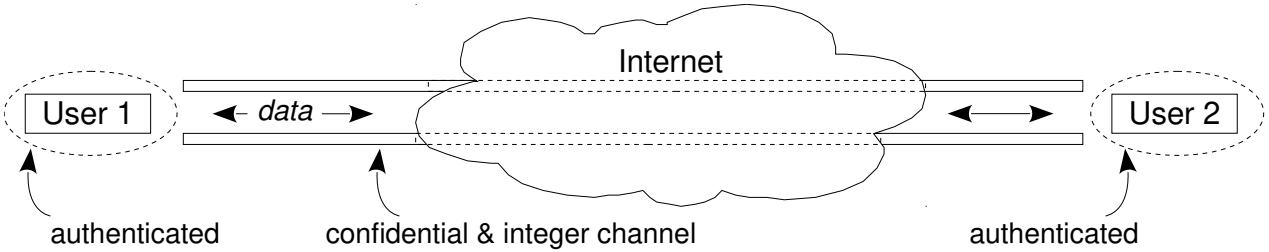


Fig. General secure (communication) channel.

Utilization of a secure channel

- 1st: Authentication of one or both subjects and probable parameter negotiation
  - -> usually, an asymmetrical cipher is used and a session key is created
- 2nd: Utilization proper, maybe also with protection for
  - integrity
  - confidentiality
  - -> usually, a symmetrical cipher is used (with above session key)

Protecting Communication Channels

Integrity

- assuring that a change in "document"[1] is detected[2]
  - implies Authentication of the entities involved!

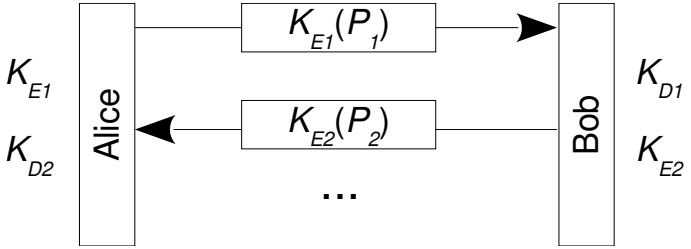
Solutions

- encipher the document
  - with symmetric or asymmetric algorithms
- use integrity code
  - with shared key
- digitally sign the document
  - with private key of sender
  - with digest (+ private key of sender)

1 file, message,...
2 if detected, change cannot be corrected (in general!)

Simple solution: encipher everything!

- exchange ciphered information
  - detection of alteration of message (e.g. intelligibility affected)!
  - confidentiality also granted (but not relevant here).



Problems

- symmetric cipher: no origin authenticity (repudiation is possible)!
- asymmetric cipher: low efficiency!
- in any case, alterations can go unnoticed:
  - in applications with general binary data (numbers, pictures...)
  - with some algorithms that guarantee confidentiality but not integrity!

Better solution: use Message Integrity Codes, MIC[1]

- build an *hash* of the message "added" to a shared key: that is the MIC!
  - e.g. MIC = $h\ (m \parallel K)$[2]
- send both message and MIC
- receiver can check message's integrity with knowledge of the key

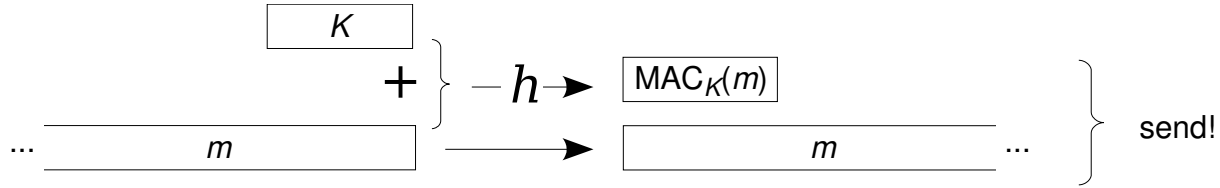


Fig. General construction principle and usage of Message Integrity Codes.

1 The designation Message Integrity Code (MIC), is currently not much used; instead, the designation in fashion is Message Authentication Code, MAC. Some authors make a slight distinction between the two; we will not. Also, we will prefer MIC, as it is more clear.

2 || means concatenation

...Integrity Protection with message integrity codes (cont.)

Problem

- uses a shared key (set in advance)
  - so, does not prevent:
    - message alteration or forging by the recipient
    - message repudiation by the sender!

Exercise:

Review the nearby image, that was first shown in the support slides of the Chapter on basic Cryptography.

Explain how the Message Integrity Codes technique relates to the pictured "protected "channel".

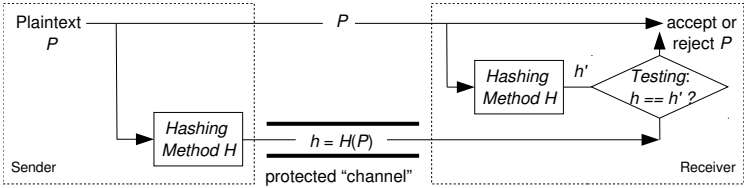


Fig. Modern Cryptography: basic model for the validation of info (e.g. integrity protection). Note the need for a protected channel!

Great solution: use digital signatures

- then we can:
  - check a document for alteration
  - associate a document to its author
- and so:
  - no one but the author can change the original document
  - the readers are assured of the identity of the author
  - the author is not able to later deny the authorship of the document (i.e. repudiate it)

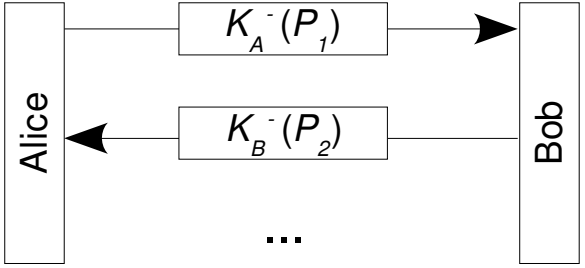
Techniques

- public key[1]
- message digest (with public key!)

1 In reality, a digital signature is made with a *private* key!

Digital signatures: public key technique

- encipherment with sender's private key
- decipherment with sender's public key



Problem

- asymmetric cipher: low efficiency

Minor problems

- sender has to ensure the secrecy of his/her private key
- sender's public key must be known in advance
- longevity of protection of sent document implies safe keeping of key pair

Digital signatures: message digest[1] technique

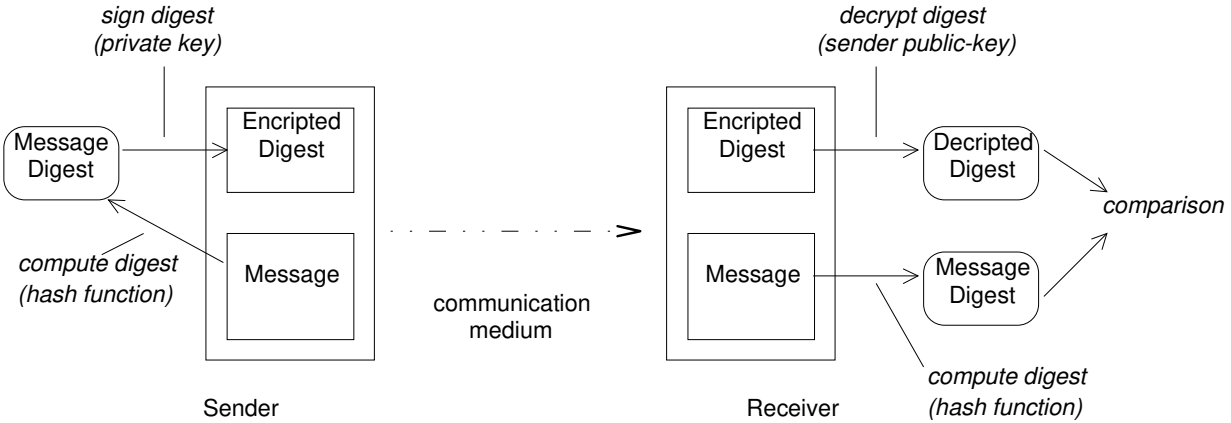


Fig. Integrity protection with digital signatures: message digest technique.

1 PT: sumário

...Digital signatures: message digest technique (cont.)

Notice:

- typical hashing efficiency is much greater than asymmetric encipherment
- the digest itself is a “fingerprint” of the document
- the signing guarantees digest is not tampered with

Minor problems

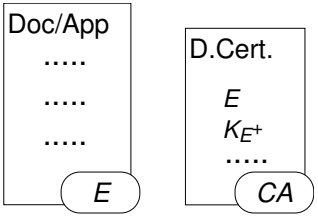
- same as public key's technique

...Digital signatures (cont.)

Example: Secure distribution of documents or software

Part I: Emission

- Emitter *E* of application/document *APP*:
  - digitally signs *APP*
    - public-key technique, digest technique...
    - generates $[APP]_E$ [1]
  - appends to $[APP]_E$ a digital certificate $[DC(E)]_{CA}$
    - certificate has K_E^+
    - is signed by *CA* (also trusted by Receiver!)
  - sends everything to Receiver
    - $APP + [APP]_E + [DC(E)]_{CA}$

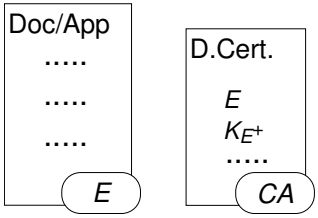


1 Notation of digital signature: $[DOC]_E <==> K_E^-$ (DOC) or $[DOC]_E <==> K_E^-$ (h(DOC))

Example (cont.) : Secure distribution of documents or software

Part II: Reception

- Receiver *R* of application/document:
  - gets K_E^+ of Emitter (if she does not yet have it)
    - by processing the digital certificate $[DC(E)]_{CA}$
      - must already know, or somehow get, K_{CA}^+
      - checks the integrity of $[DC(E)]_{CA}$
  - checks the integrity of $[APP]_E$
  - uses *APP* with confidence!

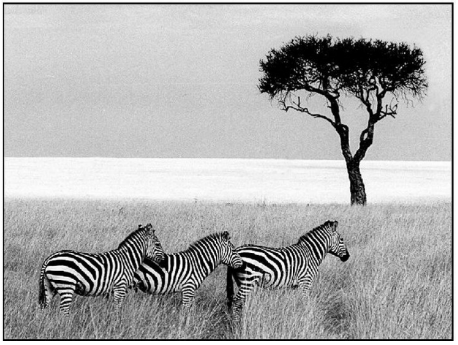


Confidentiality

- assurance of limited disclosure of information
  - implies Authentication of the entities involved!

Solutions

- hide the sensitive documents
  - physically protecting them
  - cunningly disguising them
    - steganography! [FIG[1]]
- encipher documents
  - parties need appropriate keys



1 Presumably, the original of this picture (colored, 1024×768 pixel), contains in compressed form the complete unabridged text of five Shakespeare's plays, totaling more that 700kB of text. (Tanenbaum, Modern Operating Systems)

Hiding of documents

- see steganography examples in practical classes

Encipherment of documents

- symmetrical technique [FIG a)]
- asymmetrical technique [FIG b)]

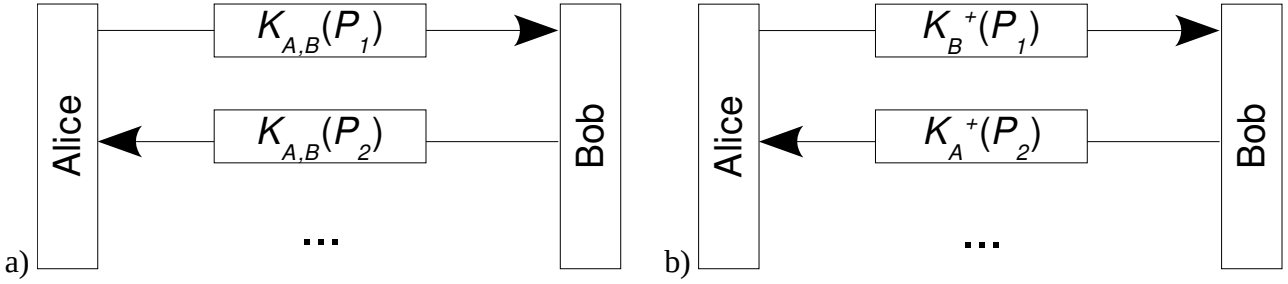
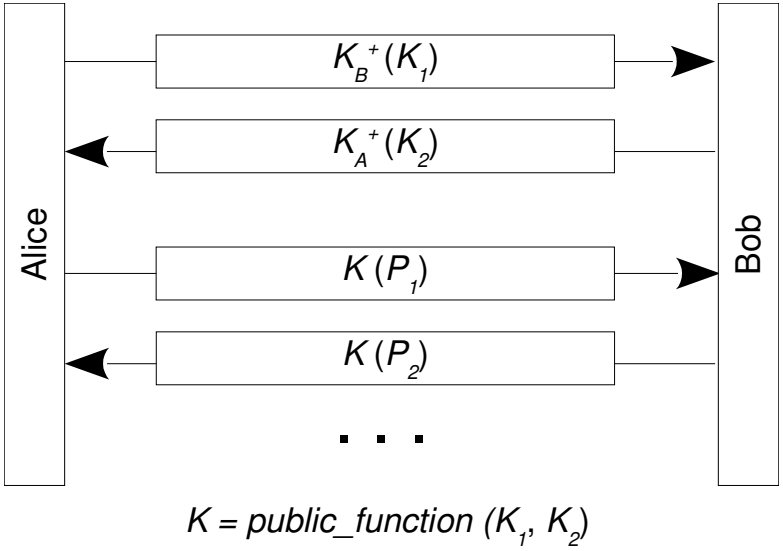


Fig. Base encipherment techniques: a) shared key; b) public key.

Practical problems:

- symmetrical keys are difficult to manage
- asymmetrical operations are very inefficient
- So, usual solution: [FIG]
  - create (symmetric) session key by public-key means[1]
  - encipher documents with (short-term) session key



1 both parties cooperate on the creation of key!

If additional integrity and authenticity is necessary:

- add digital signatures [FIG]
- use authenticated encipherment protocols[1]

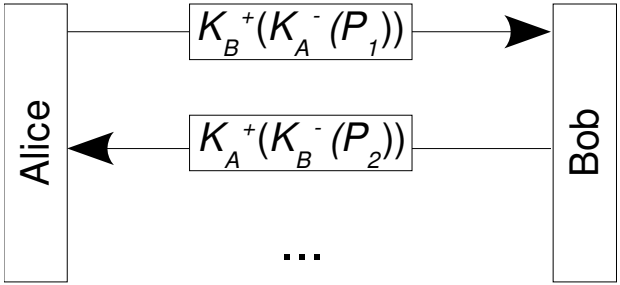


Fig. Confidentiality with integrity protection.

1 See, for example, https://en.wikipedia.org/wiki/Authenticated_encryption.

Authentication

- assuring the identity of the entities involved
  - *topic to be presented!*