Define "security" with its attributes "availability", "confidentiality", and "integrity" and give examples in your explanation. Describe the four security threats and mechanisms to ensure security.

Book p. 377-380; Slides p. 4-9

Attributes (slides p. 4)

- **Confidentiality:** Only authorized users and processes may have access to a system and its stored data and files. This is e.g. accomplished by using authentications services like Kerberos or Active Directory.
- **Integrity:** Modifications of parts of the system (HW/SW/data) must not be unauthorized. I.e. it should be possible to detect and recover abusive modifications.
- **Availability:** The probability of a service being available. Security mechanisms should be available everywhere and round the clock. If access control is used only on one part of the system, the system may be vulnerable from other parts w/o access control.

Threats (slides p. 5-6)

- **Interception:** Eavesdropping operations; unauthorized access to a service or data.
- **Interruption:** A break in communication leading in corrupted or lost files; services or data becomes unavailable or unusable.
- **Modification:** Unauthorized modification of data or service.
- **Fabrication:** Additional data or activity are generated that would normally not exist. Blocking the system by generating corrupt and faulty data.

Mechanisms (slides p. 8-9)

Security mechanisms are used implement security policies.

- **Encryption:** Transform code or data into something an attacker cannot understand.
- **Authentication:** Is used to verify the identity of a user, client, server, hosts, etc. (**Who is accessing**)
- **Authorization:** Is used to determine if an entity is allowed to do a specific operation. (**Who is allowed to access**)
- **Auditing:** Trace the activity of the entities accessing something. E.g. log files.

Question 2 Slide 9: p. 15-21

Give a (valid) definition for "cryptography" and explain the core principle of symmetric and asymmetric encryption mechanisms. Further explain the specific advantages and disadvantages of the different algorithms.

Discipline or techniques employed in protecting integrity or secrecy of electronic messages by converting them into unreadable (cipher text) form. Only the use of a secret key can convert the cipher text back into human readable (clear text) form. Cryptography software and/or hardware devices use mathematical formulas (algorithms) to change text from one form to another.

Symmetric (p. 17)

Symmetric-key algorithms are a class of algorithms for cryptography that use the same cryptographic keys for both encryption of plaintext and decryption of cipher text. The keys may be identical or there may be a simple transformation to go between the two keys. The keys, in practice, represent a shared secret between two or more parties that can be used to maintain a private information link. This requirement that both parties have access to the secret key is one of the main drawbacks of symmetric key encryption, in comparison to public-key encryption.

Types
Symmetric-key encryption can use either stream ciphers or block ciphers

- Stream ciphers encrypt the digits (typically bits) of a message one at a time.
- Block ciphers take a number of bits and encrypt them as a single unit, padding the plaintext so that it is a multiple of the block size. Blocks of 64 bits have been commonly used.

Implementations

Twofish, Serpent, AES (Rijndael), Blowfish, CAST5, RC4, 3DES

Asymmetric (Public Key encryption) (p. 18)

Public-key cryptography refers to a cryptographic system requiring two separate keys, one of which is secret and one of which is public. Although different, the two parts of the key pair are mathematically linked. One key locks or encrypts the plaintext, and the other unlocks or decrypts the ciphertext. Neither key can perform both functions (however, the private key can generate the public key). One of these keys is published or public, while the other is kept private. Public-key cryptography uses asymmetric key algorithms (such as RSA), and can also be referred to by the more generic term "asymmetric key cryptography."

Implementations

RSA, GPG (OpenPGP), TLS, SSH, DH (Diffie-Hellman)

Symmetric

Advantage	Dis
Speed	the secret keys must be transmitted (either manually or through a communication channel) since the same key is used for

	encryption and decryption

Asymmetric

Advantage	Dis
not necessary in a single-user environment.	Speed
private keys never need to be transmitted or revealed to anyone	vulnerable to impersonation, even if users' private keys are not available
can provide digital signatures that cannot be repudiated.	not meant to replace secret-key cryptography, but rather to supplement it, to make it more secure.

Exercise 03: Secure channel (slide 09, p. 24 - 26)

Question: Which kind of protection can be provided by a secure channel? Give examples of insecure communication where different requirements to a secure channel are not satisfied. Explain how two communication partners can perform mutual authentication based on public key cryptography.

Secure channel (slide p. 24)

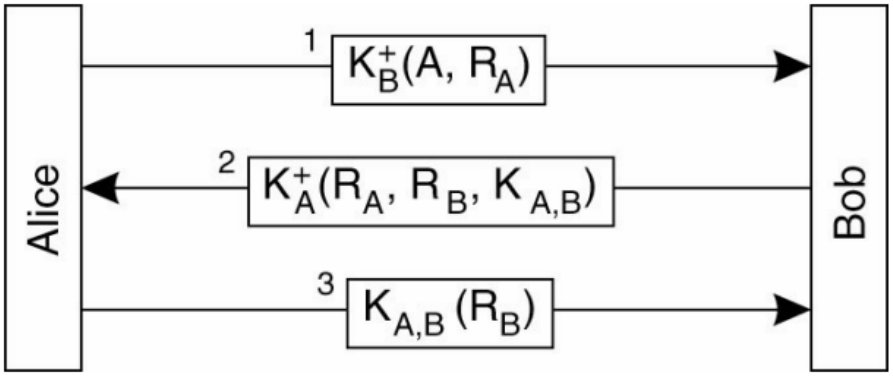
Secure channel between two or more participants, which is protected against attacks (e.g. Eavesdropping, spoofing, message tampering, …). Most attacks can be prevented by using an encrypted channel (Public- and Private-Key encryption). BUT: doesn't protect against interruption!

Give examples of insecure communication where different requirements to a secure channel are not satisfied

See slide 25! Spoofing and message tampering!

Explain how two communication partners can perform mutual authentication based on public key cryptography

See slide 26!



The communication participants own the public key (K+) of each one another.

- Alice sends a message to Bob, containing 'A' ("I am Alice") and a challenge RA (e.g. a random number). This message encrypted using Bob's public key (KB+).
- Due to the fact that only Bob is able to decrypt, he receives the challenge RA. He generates a session key KA,B which is used for the further session of the secure channel. Bob replies with RA, the new challenge RB and the session key (encrypted by the public key of Alice (KA+)).
- Now Alice owns the Session key and confirms Bob's challenge RB, which is encrypted by the session key.

Question4: Explain how two communication partners can perform mutual authentication based on symmetric key cryptography. Show an example of what can happen if an authentication protocol is "optimized" incorrectly.

Authentication: Slide 27

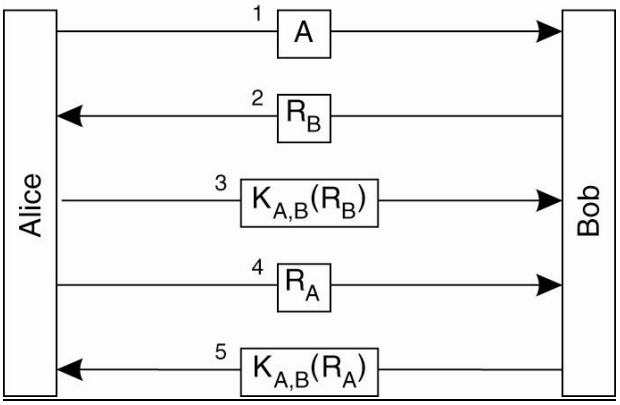
Alice sends her identity to bob in order to open a communication channel.

Bob sends a request "Rb" (e.g. a random number) to Alice.

Alice encrypts Rb with the shared secret key and sends it to Bob. Bob decrypts it and checks if it corresponds to what he sent before.

Now Bob knows that Alice is Alice indeed.

Alice repeats these steps to authenticate Bob.



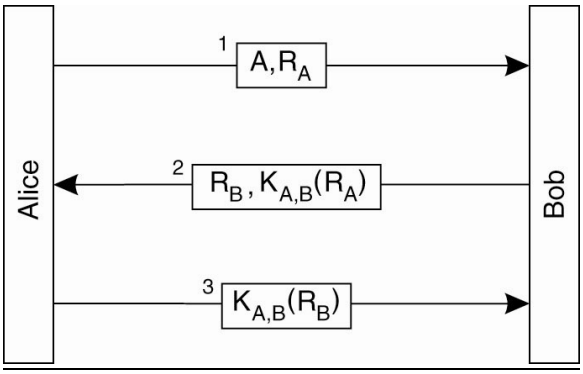
Only 3 messages: Slide 28

Alice sends her identity and a random number to Bob.

Bob encrypts the random number that was sent by Alice and sends it along with another random number that he generates.

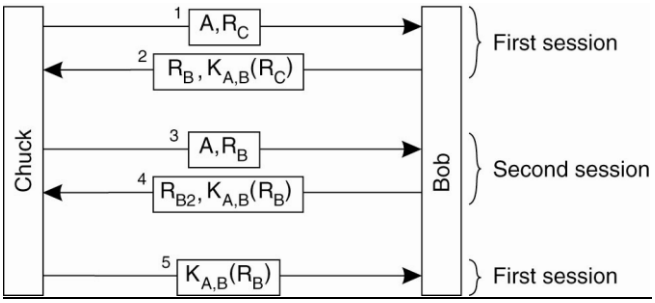
Alice now decrypts the message and is able to check whether Bob is her communication partner.

Alice encrypts Bob's random number and sends it to him in order to complete the authentication process.



Problem: Reflection Attack:

Chuck impersonates Alice. To achieve this, he opens a second communication channel and sends the random number that Bob has asked him to encrypt to Bob. He gets an encryption as response and can use it as response for the first communication channel. Bob now thinks that Chuck is Alice.



What is meant by the term "digital signature" and which guarantees are given by such a signature? What is a hash function and which properties does it have to provide in order to be usable in the construction of a digital signature. Why is the use of a hash function an advantage?

Book p. unknown; Slides: none

Digital signature

A digital signature is an attachment to a message which confirms the authentication of the sender A, i.e. that the message was really sent by A, to the recipient B. At the same time it also verifies that the message has not been modified by another person, because the signature is really bound to the content of the message.

How does it work

There is a so called hash function (that is explained later) that calculates a hash value from the content of the message. This hash value is then encrypted with the private key of the sender A (afterwards it is called the signature) and sent along with the message. The recipient B also calculates the hash value of the message content, decrypts the signature of A with the public key of A and compares it to its own calculated hash value. If the two hash values are equal, the message was really sent by A and the message content has not been modified.

Hash functions

A hash function *h(.)* calculates a value *h* with constant length from a given input *m*. This value is the so called hash value.
A good hash function should have the following properties:

- There must not exist a reverse function *h'(.)* that allows someone to calculate the input *m* from the hash value *h*. Hash functions have to be one-way functions.
- The hash function has to be resistant to collisions. I.e. it should be hard to find any *x* and *y* such that *h(x) = h(y)*.
- The length of the hash value should be constant and independent from the input *m*.