

Module 9

Security

Issues

- Separation of
 - ➔ Security policies
 - ◆ Precise definition of which entities in the system can take what actions
 - ➔ Security mechanism
 - ◆ Means of enforcing that policy
- Distributed system security
 - ➔ Communication between users or processes that may be on different machines
 - ◆ Secure channel (authentication, message integrity, confidentiality)
 - ➔ Authorization to ensure that a user or process performs only those actions that is allowed under the security policy
 - ◆ Access Control
- Security services
 - ➔ What any distributed system should provide as part of its infrastructure to enable the implementation of different policies.

Basic Concepts

- **Security:** attempt to protect the services and data it offers against security threats.
- **Confidentiality:** the property of a computer system whereby its information is disclosed only to authorized parties
- **Integrity:** the characteristic that alterations to a system's assets can be made only in an authorized way.

Types of Threats

- **Interception**

- ➔ An unauthorized party has gained access to a service or data

- **Interruption**

- ➔ Services or data become unavailable, unusable, destroyed and so on.

- **Modification**

- ➔ Unauthorized changing of data or tampering with a service so that it no longer adheres to its original specifications

- **Fabrication**

- ➔ Refers to the situation in which additional data or activity are generated that would normally not exist

Methods of Attack

- **Eavesdropping**
 - ➔ Obtaining copies of messages without authority
- **Masquerading**
 - ➔ Sending or receiving messages using the identity of another principal without their authority
- **Message tampering**
 - ➔ Intercepting messages and altering their contents before passing them on to the intended recipient
 - ➔ Man-in-the-middle attack
- **Replaying**
 - ➔ Storing intercepted messages and sending them out at a later time
- **Denial of service**
 - ➔ Flooding a communication channel or a system resource with messages in order to deny access for others

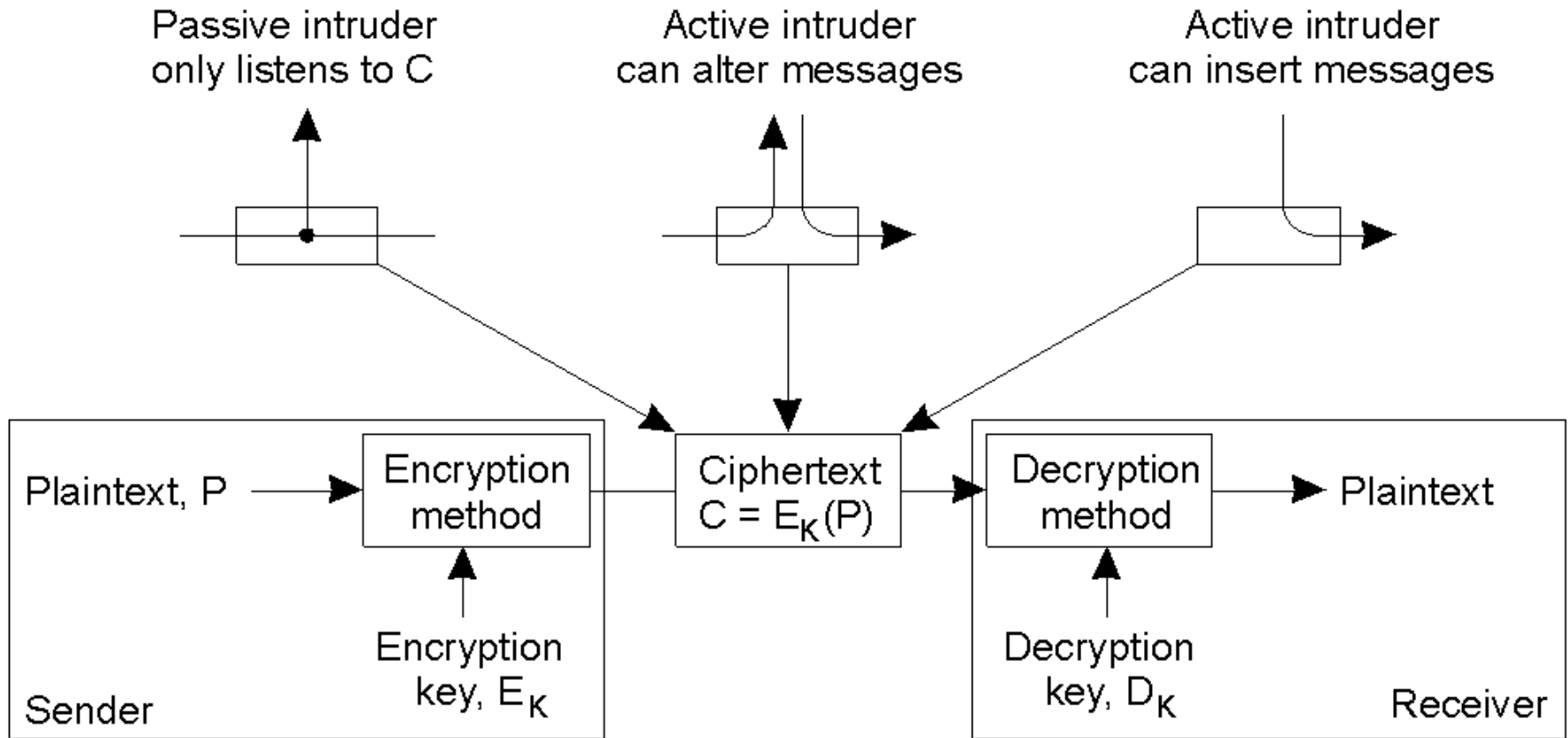
Security Mechanisms

- **Encryption**
 - ➔ transform data into something an attacker cannot understand
 - ➔ provides a means to implement confidentiality
 - ➔ provides support for integrity
- **Authentication**
 - ➔ verify the claimed identity of a user, client, server and so on
- **Authorization**
 - ➔ check whether the client is authorized to perform the action requested
- **Auditing**
 - ➔ auditing tools are used to trace which clients accessed what, and which way

Security Services - Design Issues

- Focus of Control
 - ➔ Decide the focus of control: data, operations or users
- Layering of security mechanisms
 - ➔ Decide at which level security mechanisms should be placed
- Simplicity
 - ➔ Simplicity will contribute to the trust that end users will put into the application and, more importantly, will contribute to convincing the designers that the system has no security holes.

Cryptography



The three different attacks that we need to protect against, and for which encryption helps.

Secure Channels

- Secure communication requires authentication of the communicating parties, but also ensuring message integrity and possibly confidentiality as well.
- A **secure channel** protects senders and receivers against **interception**, **modification**, and **fabrication** of messages.
- It does not necessarily protect against interruption.
- Protecting messages against interception is done by ensuring confidentiality
- Protecting messages against modification and fabrication is done through protocols for mutual authentication and message integrity.

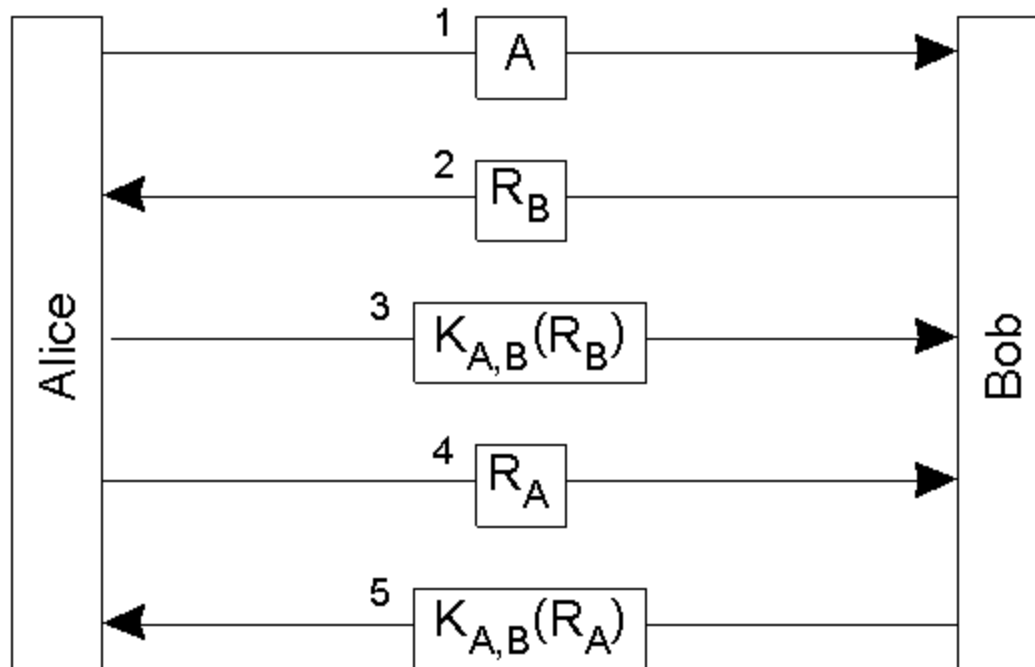
Authentication

- Authentication and message integrity cannot do without each other.
- The combination works as follows:
 - ➔ Alice starts by sending a message to Bob to set up a channel
 - ➔ Once the channel has been set up, Alice knows for sure that she is talking to Bob, and Bob knows for sure that he is talking to Alice, they can exchange messages
 - ➔ To subsequently ensure integrity it is common practice to use secret-key cryptography by means of **session keys**.

Notation	Description
$K_{A,B}$	Secret key shared by A and B
K_A^+	Public key of A
K_A^-	Private key of A

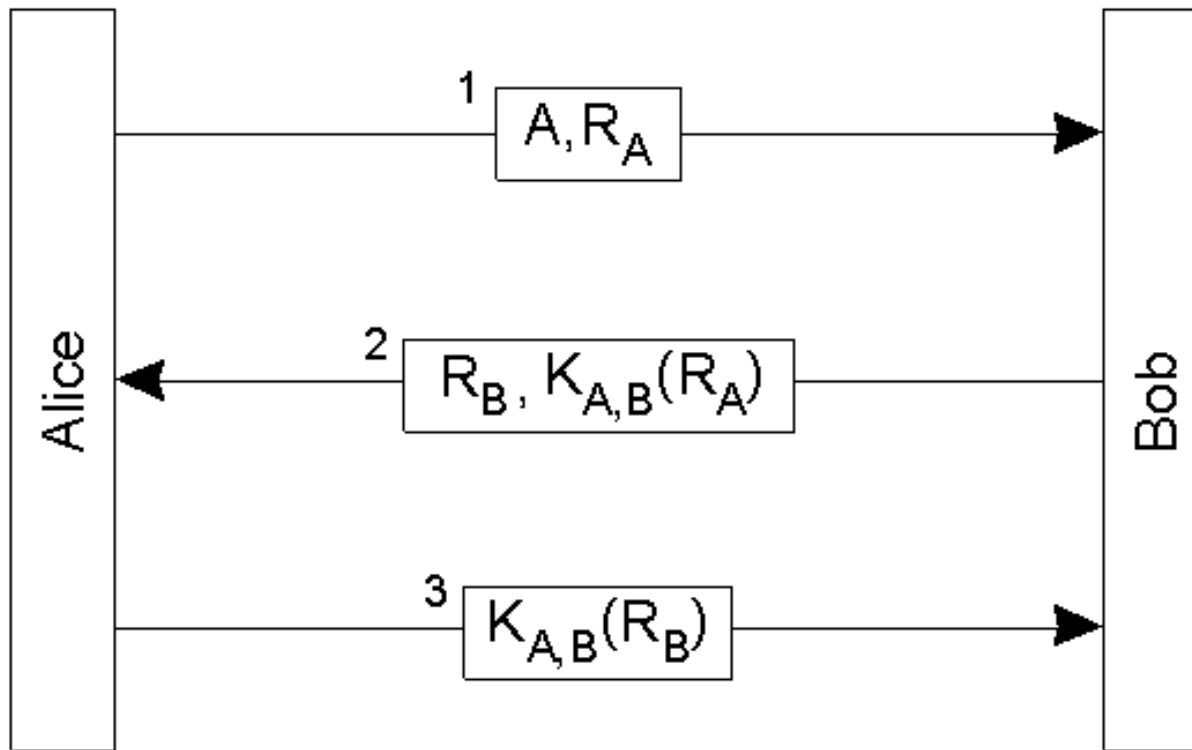
Authentication

- Authentication based on a shared secret key
- Also known as **challenge-response protocol**



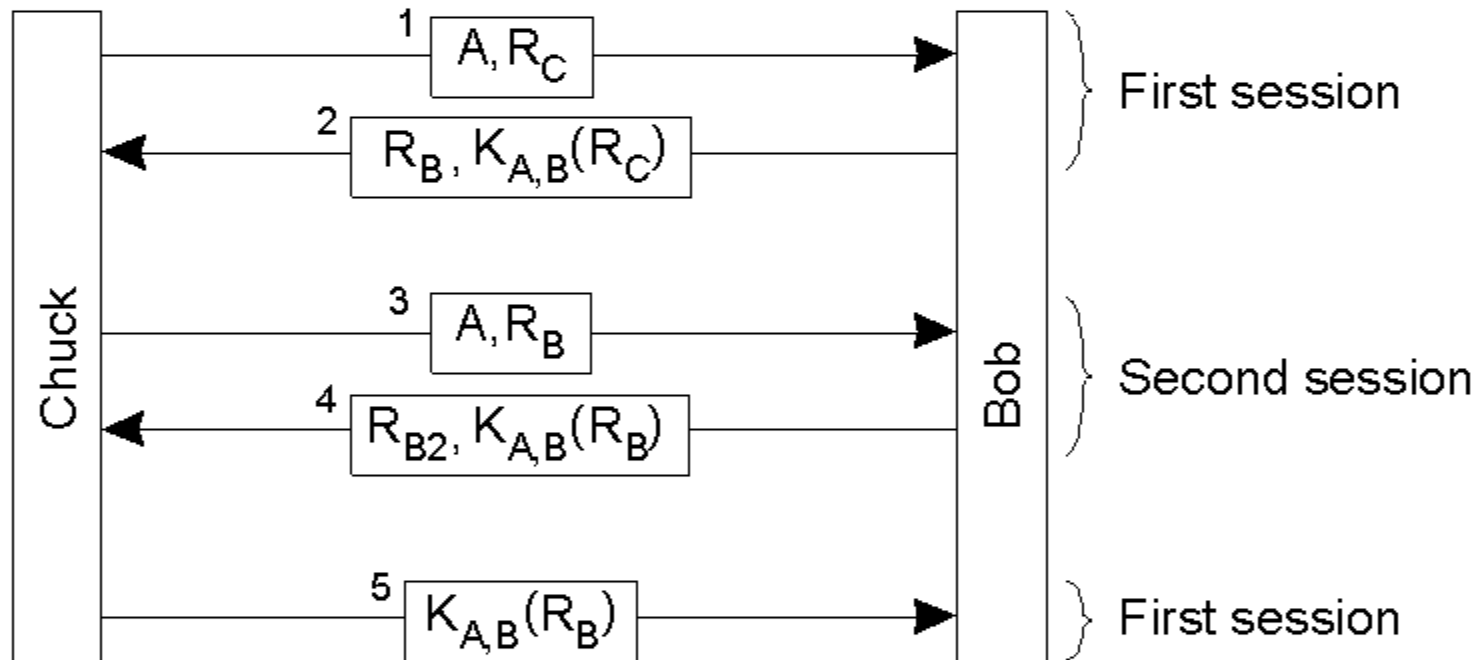
Authentication (2)

- Consider this “optimization”: Authentication based on a shared secret key, but using three instead of five messages.



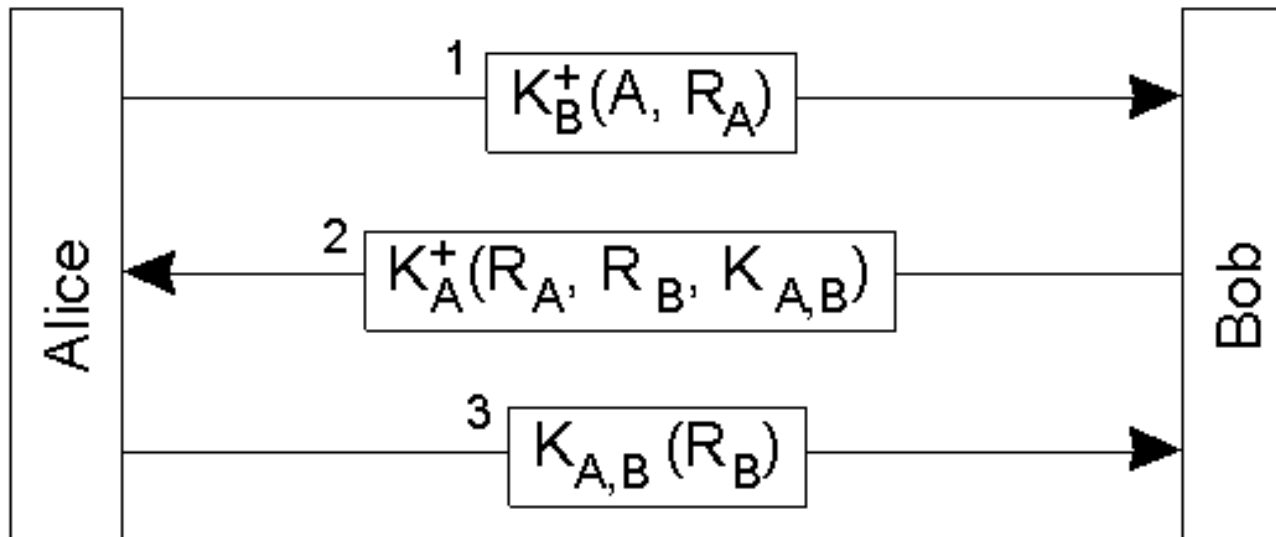
Authentication (3)

- The **reflection attack**.



- ➔ Tweaking an existing protocol to improve its performance, can easily affect its correctness

Authentication Using Public-Key Cryptography

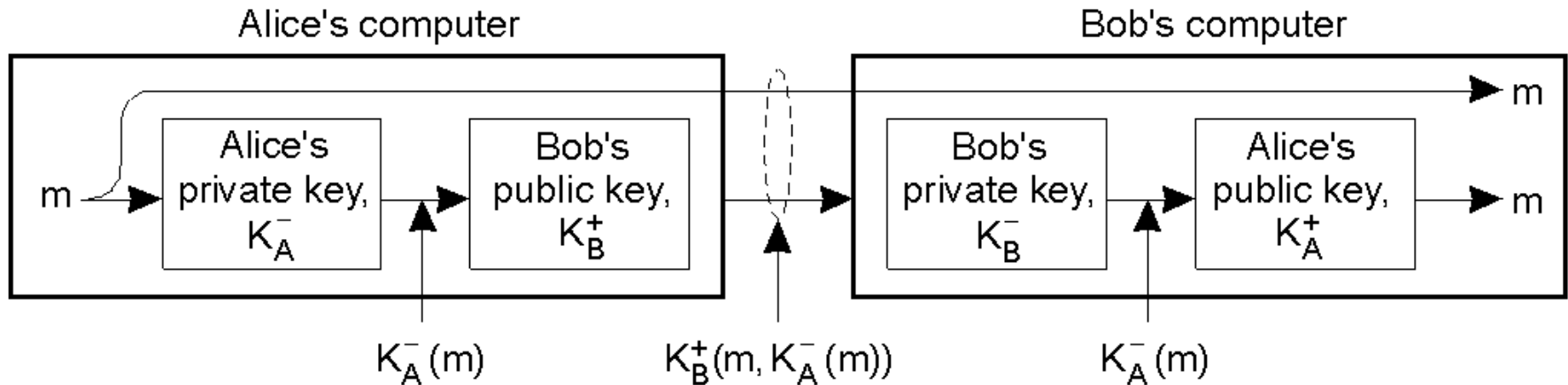


Message Integrity and Confidentiality

- Besides authentication, a secure channel should also provide guarantees for message **integrity** and **confidentiality**
- Confidentiality is easily established by simply encrypting a message before sending it.
- Protecting a message against modifications is somewhat more complicated
- Digital Signatures:
 - ➔ Digitally sign a message in such a way that the signature is uniquely tied to its content
 - ➔ Several ways to place digital signatures:
 - ♦ Use a public-key cryptosystem such as RSA
 - ♦ Use a message digest

Digital Signatures

Digital signing a message using public-key cryptography.

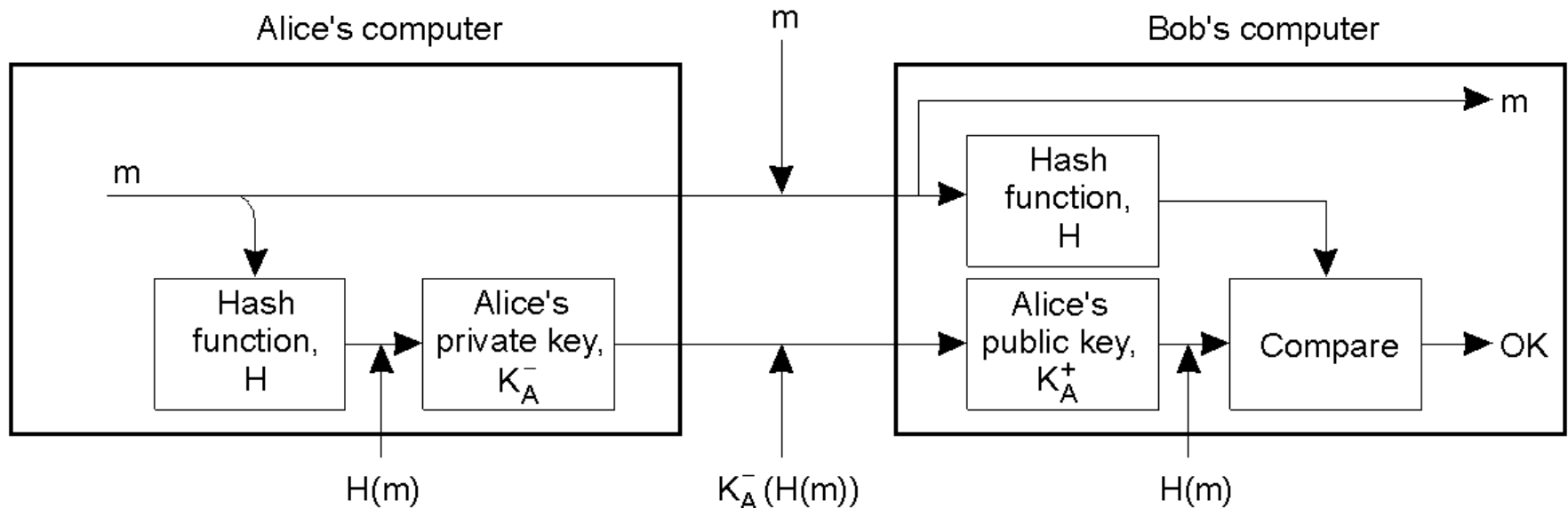


Problems with this scheme:

- the validity of the signature holds only as long as the private key remains secret
- what if Alice decides to change her private key
- encryption of the entire message may be costly in terms of processing requirements and is actually unnecessary

Digital Signatures

Digitally signing a message using a message digest.



Session Keys

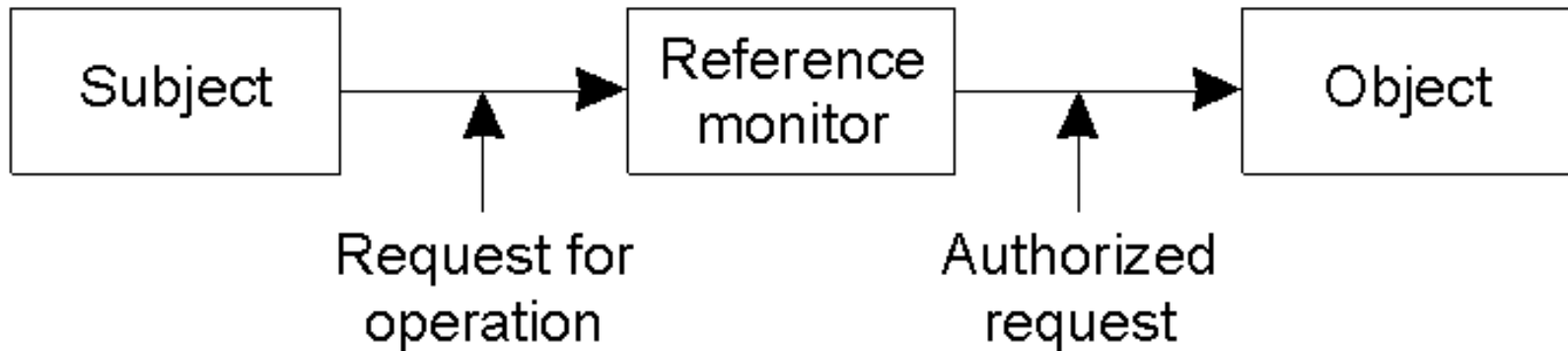
- During the establishment of a secure channel, after the authentication phase has completed, the communicating parties generally use a unique shared session key for confidentiality.
- The session key is safely discarded when the channel is no longer used.
- Why not use the same keys for confidentiality as those that are used for setting up the secure channel?
 - ➔ Cryptographic keys are subject to “wear and tear” just like ordinary keys.
 - ➔ Protection against replay attacks
 - ➔ If a key is compromised, only a single session is affected
- ➡ The combination of a long-lasting keys with the much cheaper and more temporary session keys is often a good choice for implementing secure channels for exchanging data

Access Control

- In the client-server model, once a client and a server have set up a secure channel, the client can issue requests that are to be carried out by the server.
- A request involve carrying out operations on resources that are controlled by the server.
- Such a request can be carried out only if the client has sufficient **access rights** for that request.
- Verifying access rights is referred to as **access control**, whereas **authorization** is about granting access rights.

General Issues in Access Control

- General model of controlling access to objects.



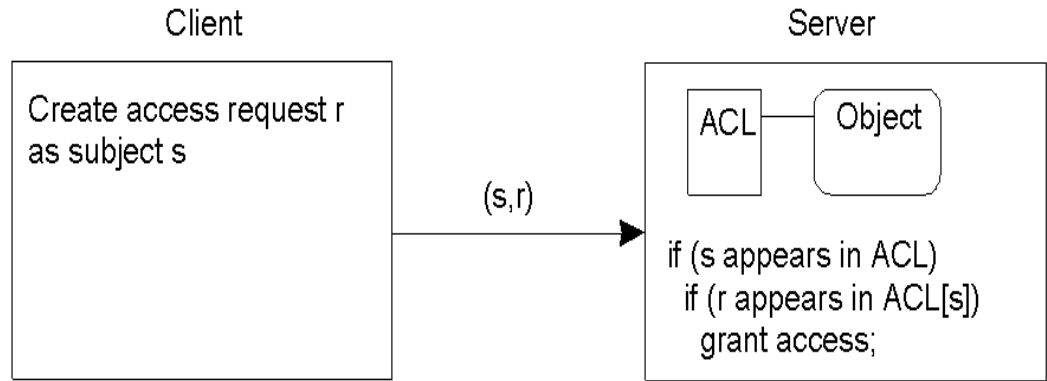
- Controlling the access to an object is all about protecting the object against invocations by subjects that are not allowed to have specific methods carried out
- Also, protection may include object management issues

Access Control Matrix

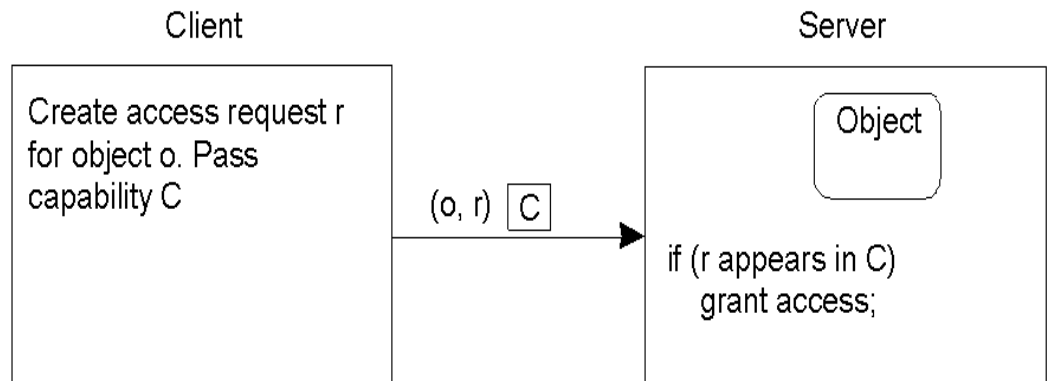
- A common approach to modeling the access rights of subjects with respect to objects, is to construct an **access control matrix**

$$M[s,o]=\{m_1,m_2,\dots\}$$

- **Access Control List**: The matrix is distributed column-wise
- **Capabilities**: The matrix is distributed row-wise



Using an ACL



(b)

Using capabilities

Role-Based Access Control

- Related to having groups as protection domains, it is also possible to implement protection domains as roles.
- In role-based access control, a user always logs into the system with a specific role, which is often associated with a function the user has in an organization.
- A user may have several functions.
- Depending on the role the user takes when logging in, he may be assigned different privileges (i.e. his role determines the protection domain in which he will operate).