# Hospital Information Systems

## Unit 5: Information Security, Data Protection and Privacy
## Part I: Information Security

*Master in Biomedical Engineering*

Rodrigo García Carmona

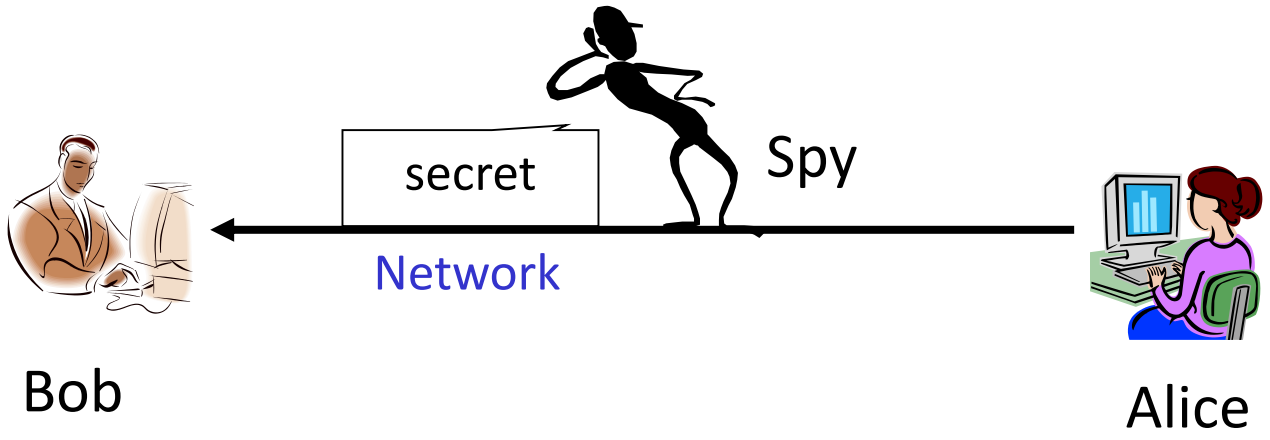

*Universidad CEU San Pablo*
*Escuela Politécnica Superior*
*Departamento de Tecnologías de la Información*

# *Table of Contents*

# *ENCRYPTION AND SECURE COMMUNICATIONS*

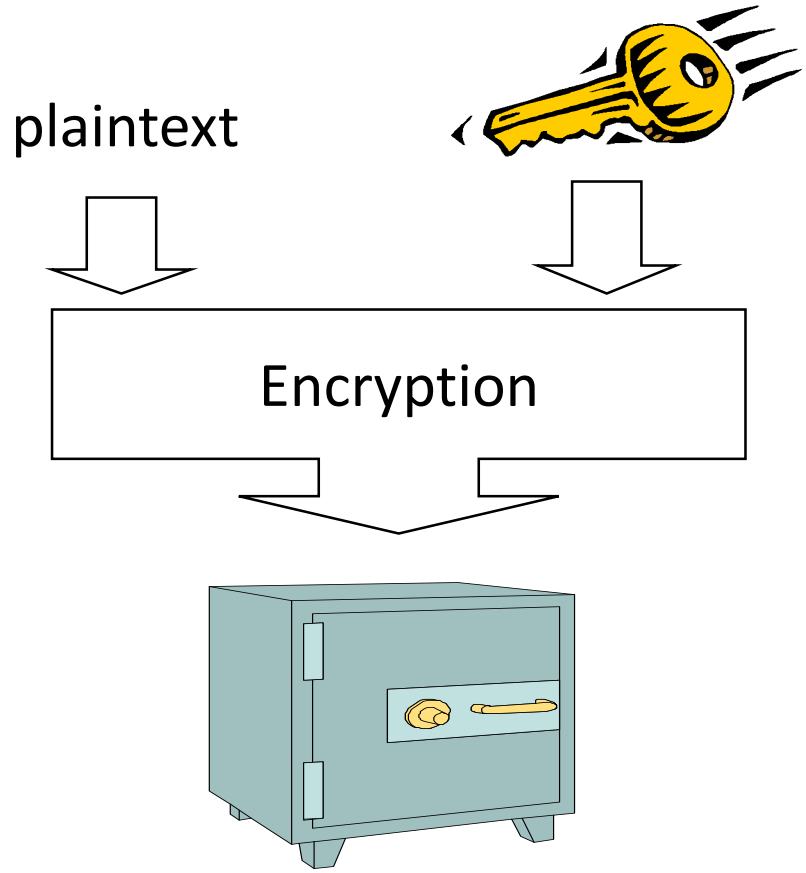
# *Secure Communications*

- How can we protect the information that is being sent through a network?
  - Specially when the network is the Internet or another heterogeneous or non-centralized network.
  - Also important in wireless networks, where it is very easy to eavesdrop.



# *Encryption*

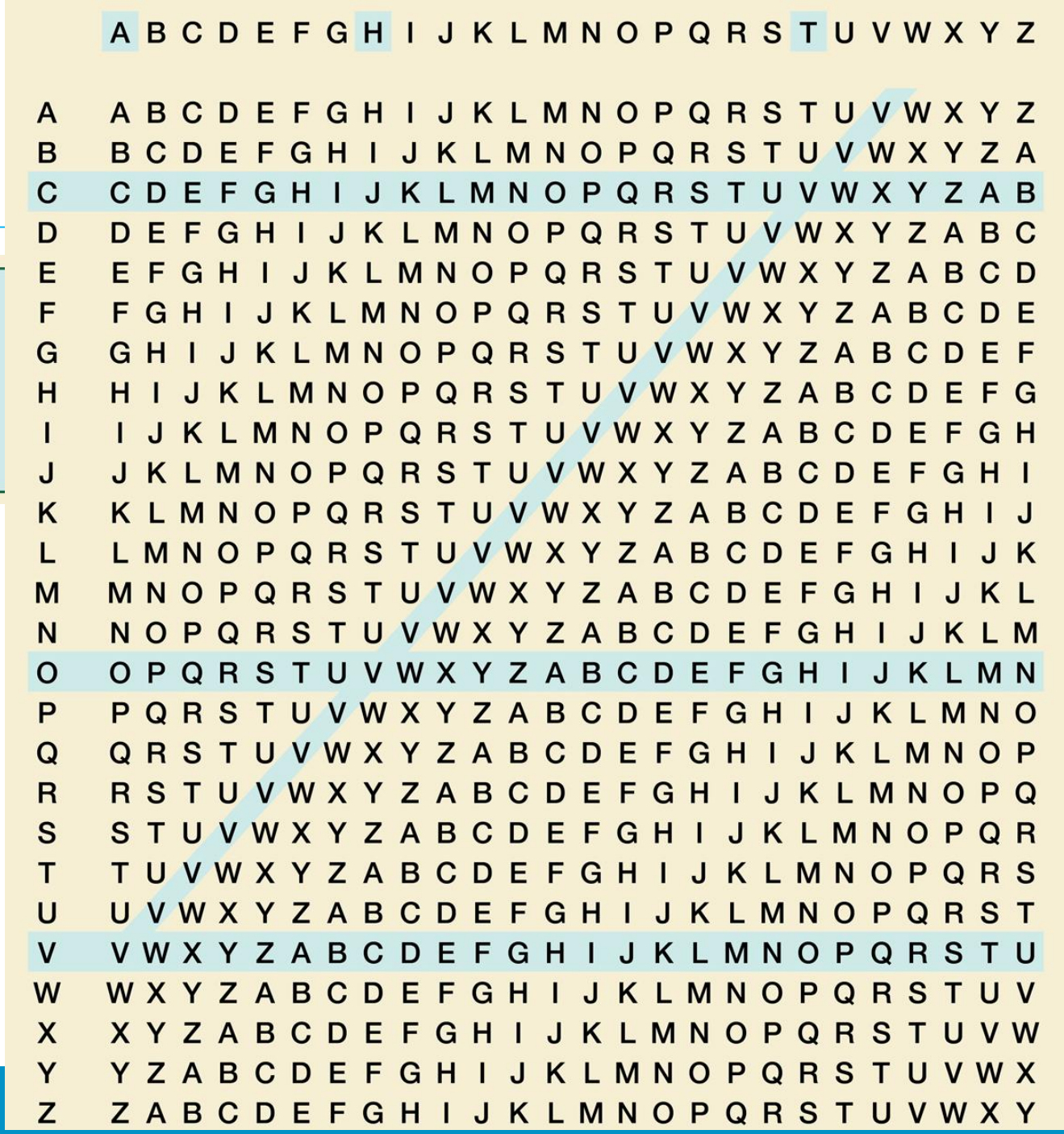
- Input:
  - Message (plaintext).
  - Encryption method.
  - Key.
- Output:
  - Encrypted message.



# *Sample Encryption*

| | |
|---|---|
| cipher | VVVRBACP |
| key | COVERCOVER... |
| plaintext | THANKYOU |

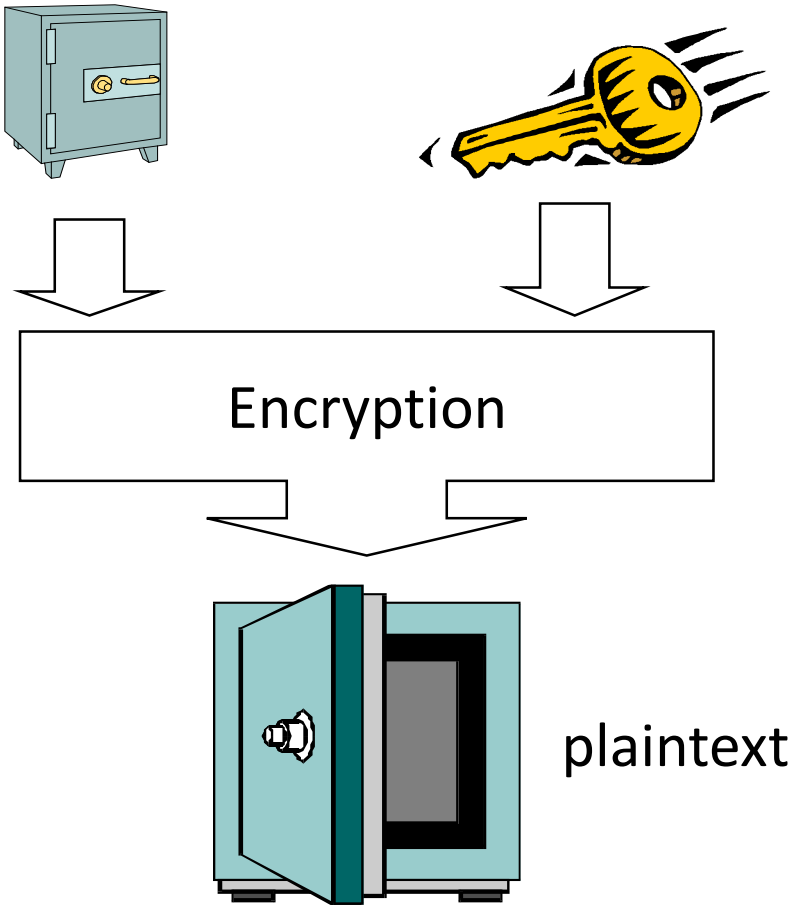
- Encryption methods are public.
- Their strength lies in the key, which only the participants know.
- Security through obscurity is bad security.




CEU

# *Decryption*

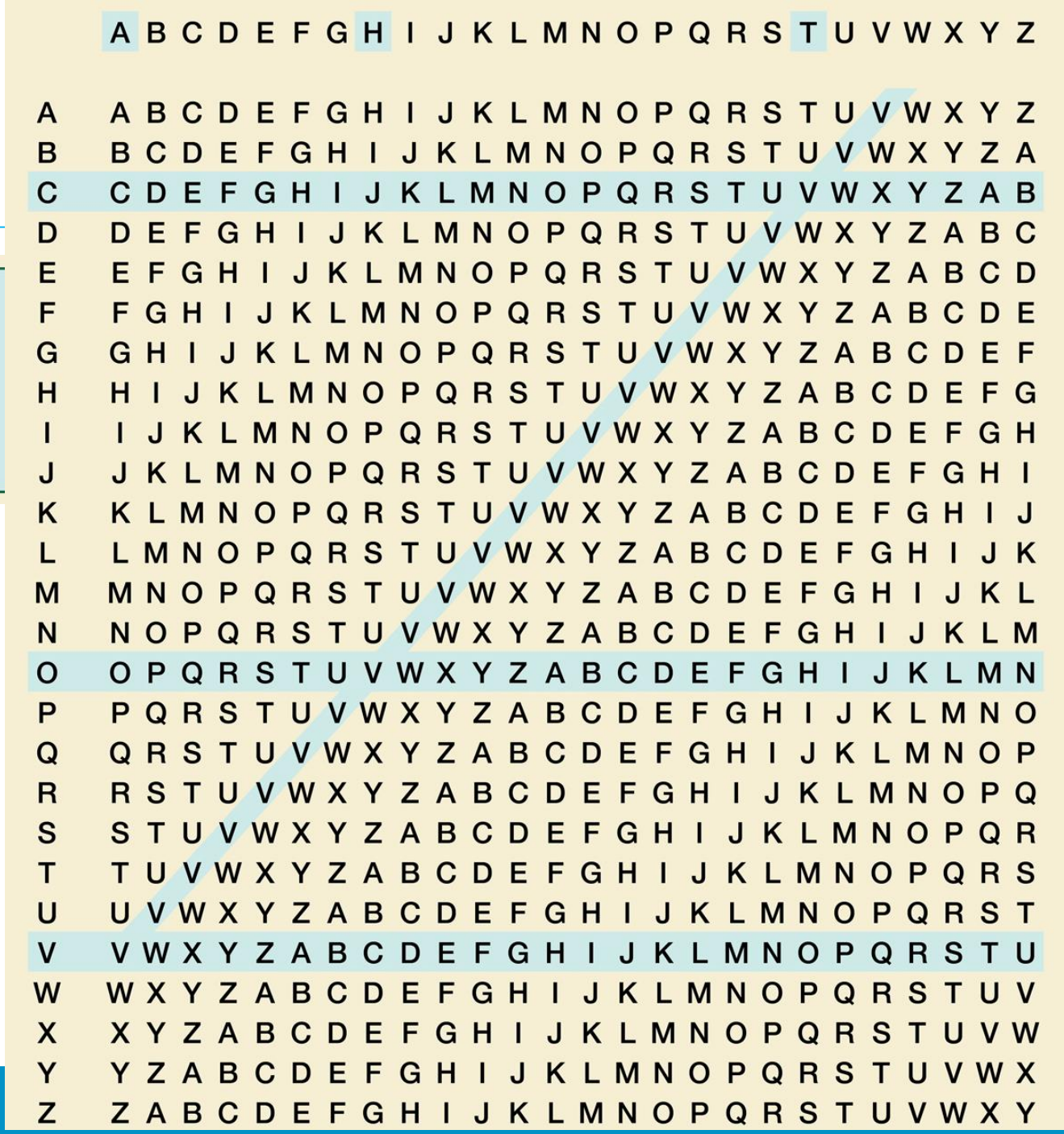
- Input:
  - Encrypted message.
  - Encryption method.
  - Key.
- Output:
  - Message.



# *Sample Decryption*

| | |
|---|---|
| cipher | VVVRBACP |
| key | COVERCOVER... |
| plaintext | THANKYOU |

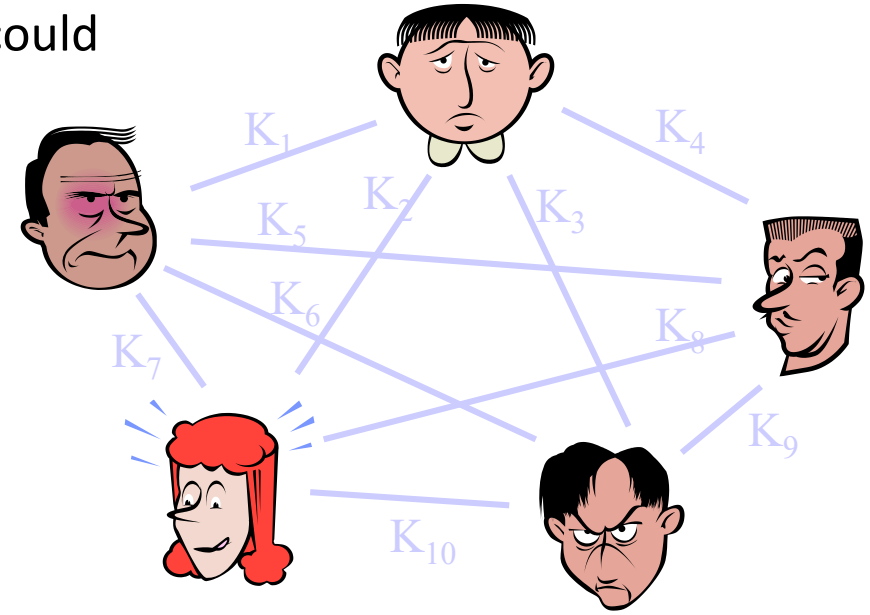
- Anybody with the key who knows the encryption method (public) can decrypt the message.




CEU

# *Key Sharing*

- An important problem is how to share and store keys.
  - Specially if we need to communicate with different people.
  - A key needs to be sent, and could be intercepted.
  - Or the storage could be compromised, specially if it is shared.

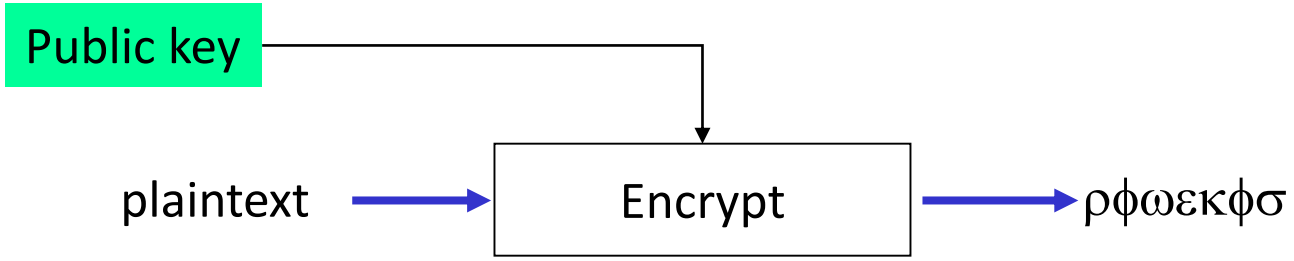


# *Symmetric and Asymmetric Cryptography*

- **Symmetric cryptography:**
  - The same key is used to both encrypt and decrypt.
  - More computationally efficient.
  - The previous example.
- **Asymmetric cryptography:**
  - One key is used to encrypt (*public key*) and another, different one, to decrypt (*private key*).
  - Made possible by some interesting mathematical properties.
    - $P \neq NP$
  - Less computationally efficient.
  - Public keys can be shared freely.

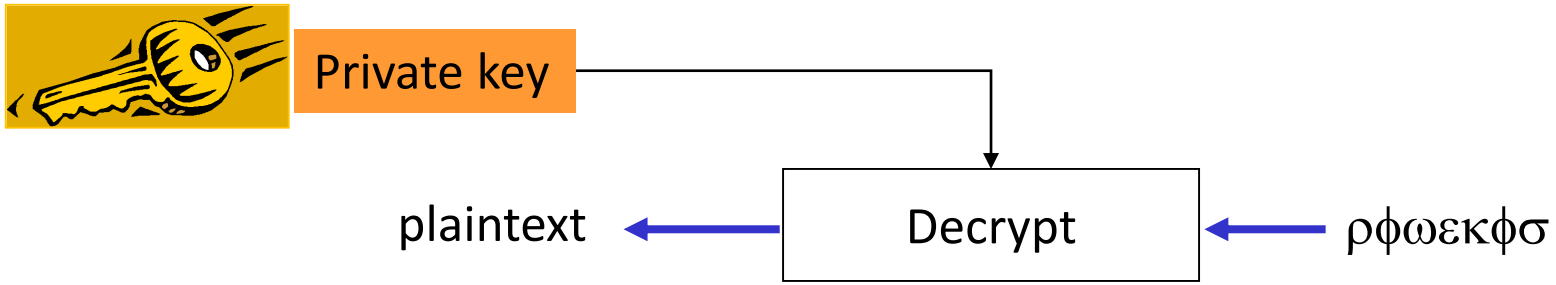
# *Asymmetric Cryptography - Encryption*

- The public key is made public.
- Everybody can access it.
- The sender uses it to encrypt the message.

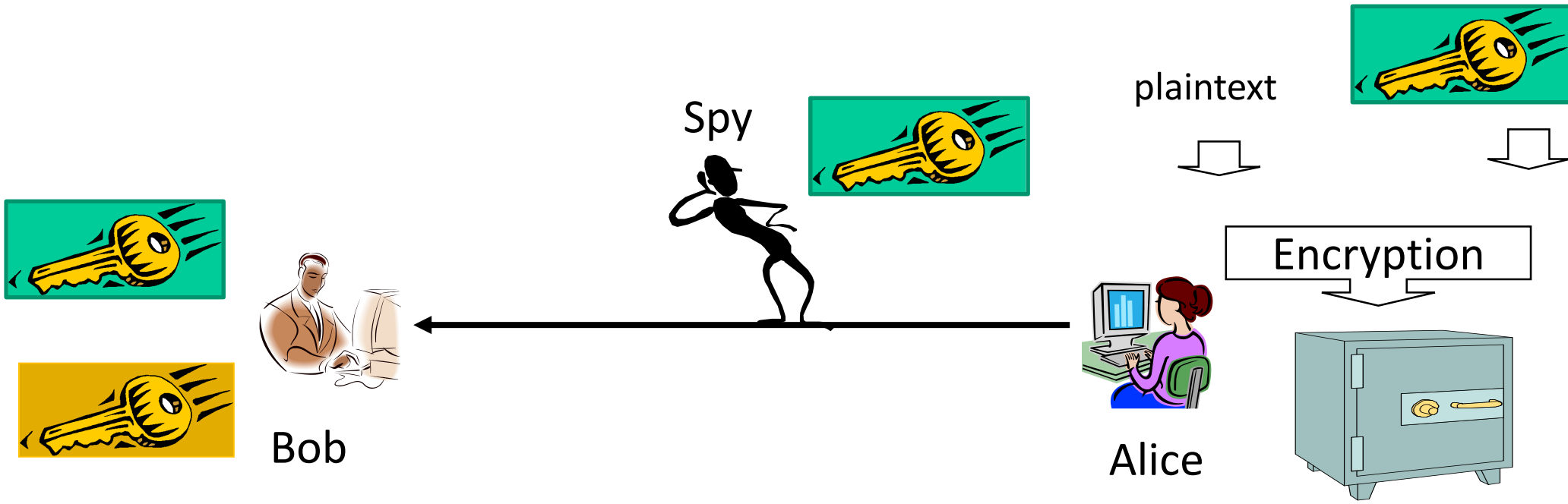


# *Asymmetric Cryptography - Decryption*

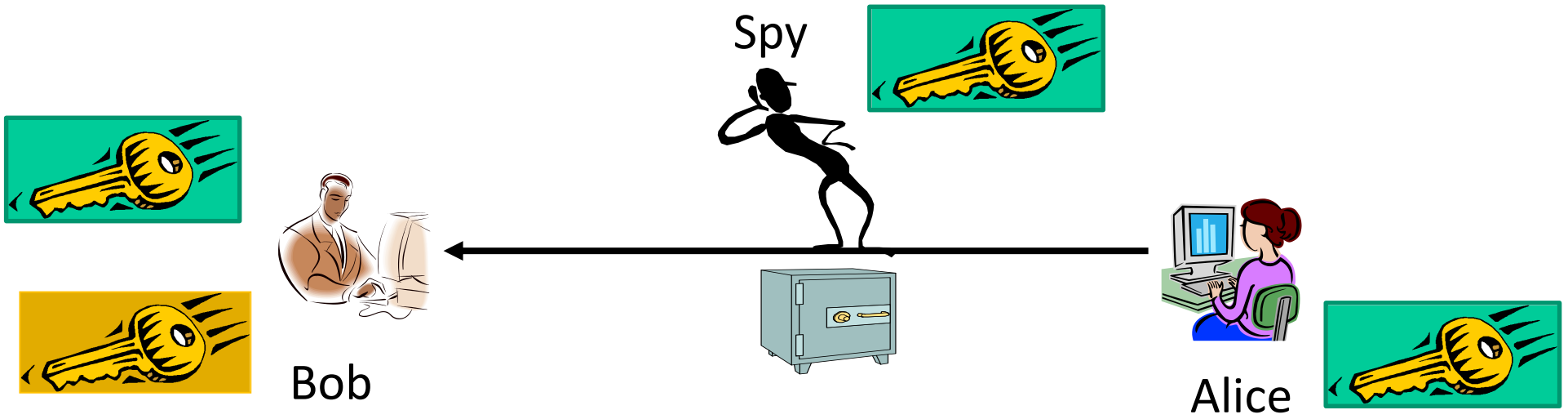
- Only the receiver has access to the private key.
- The private key is paired with a public key.
- The receiver uses it to decrypt a message encrypted with the corresponding public key.



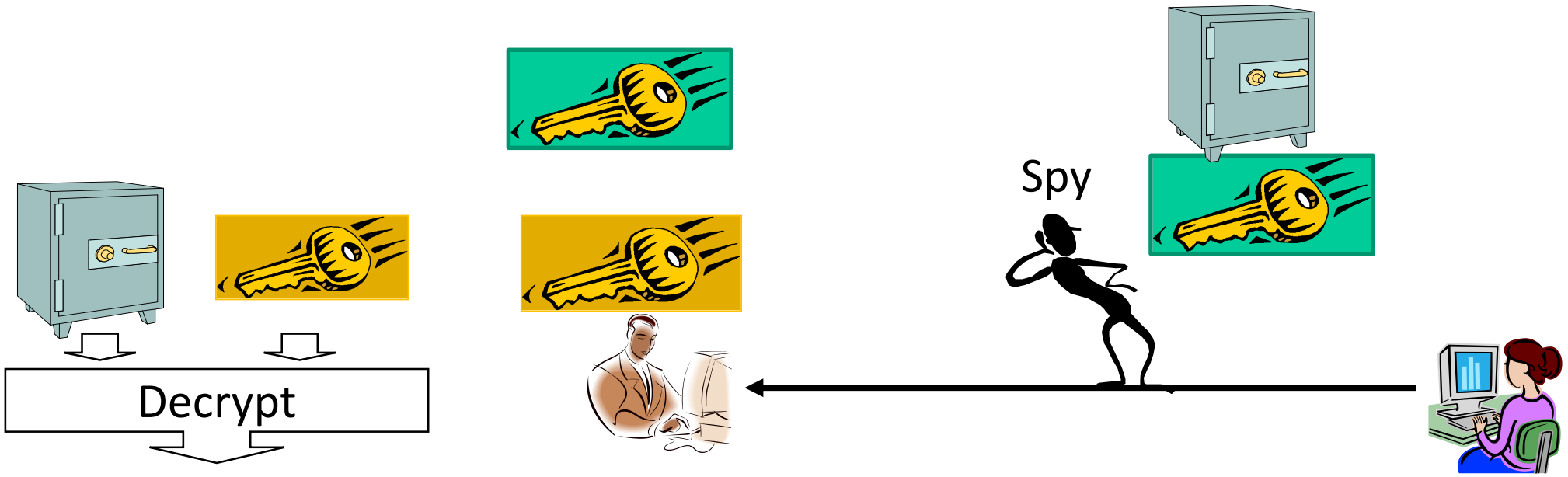
# *Asymmetric Cryptography – Process (I)*



# *Asymmetric Cryptography – Process (II)*



# *Asymmetric Cryptography – Process (III)*




Only Bob can decrypt the message.

# *SECURITY DIMENSIONS - TECHNICAL*

# *Security Dimensions - Technical*

- **Confidentiality**
  - Nobody can see the message.
  - We have already studied this.
- **Authentication**
  - Login: This is who I am.
  - Digital signature: this is something I have created.
    - Made possible by asymmetric cryptography.
- **Authorization**
  - This is what I'm authorized to do.
- **Integrity**
  - The message has not been modified.
  - Made possible by asymmetric cryptography.
- **Non repudiation**
  - I can not retract myself.
  - Made possible by asymmetric cryptography.

# *SECURITY DIMENSIONS - LEGAL*

# *Security Dimensions - Legal*

- Built on the following ideas:
  - Information is the **most valuable asset** of a company.
  - Part of this information is not the company's property. It has been entrusted to it by a third party.

- **Confidentiality**
  - Information can not be shown to non-authorized people.
- **Availability**
  - Information must be available to the authorized people in the same moment they want to access it.
- **Integrity**
  - Information must be maintained exactly as it is, without modification by non-authorized people or processes.

# *Information Security*

- From a legal standpoint, security must be maintained at all levels, not only from the “software viewpoint”.
- It is information security, not just computer security.
- Security is designed and managed around the idea of **risks.**
- **Risk management** is realized through security audits:
  - Policies.
  - Procedures.
  - Specific controls.

# *Management of Information Security*

- **Management of information security implies:**
  - Physical security and access control
  - Information classification (public, restricted, reserved…)
  - User access and human resources
    - Social engineering
  - Devices (including mobile devices and workstations)
  - Vulnerabilities (monitoring, characterization and patching)
  - Event response (logging of an event is sensible information)
  - Back up and disaster recovery
  - Security during operations
- Certificates and laws are designed around this management model.
- From now on, we will focus on the computer security dimension.

CEU

# *SECURITY ELEMENTS AND PRODUCTS*

# *Security Elements*

- The following **security elements** are actual hardware that form part of the network or datacenter that they intend to protect:
  - Firewalls
  - Load balancers
  - Intrusion Detection Systems (IDS)
  - Intrusion Prevention Systems (IPS)
  - Network Access Control (NAC) systems
- The **Demilitarized Zone** (DMZ) comprises the elements of a network that are exposed to another, untrusted network (usually the Internet).

# *Security Products*

- The following **security products** are software that a company can buy and then install in their machines to protect the organization's information:
  - Antivirus products
  - Endpoint security solutions
    - Malware removal
    - Antispyware
    - Application control
  - Data Leak Prevention (DLP)
  - Privileged Access Manager (PAM)
  - Full disk encryption
  - Identity management

# *NETWORK AND TRANSPORT LEVEL SECURITY*

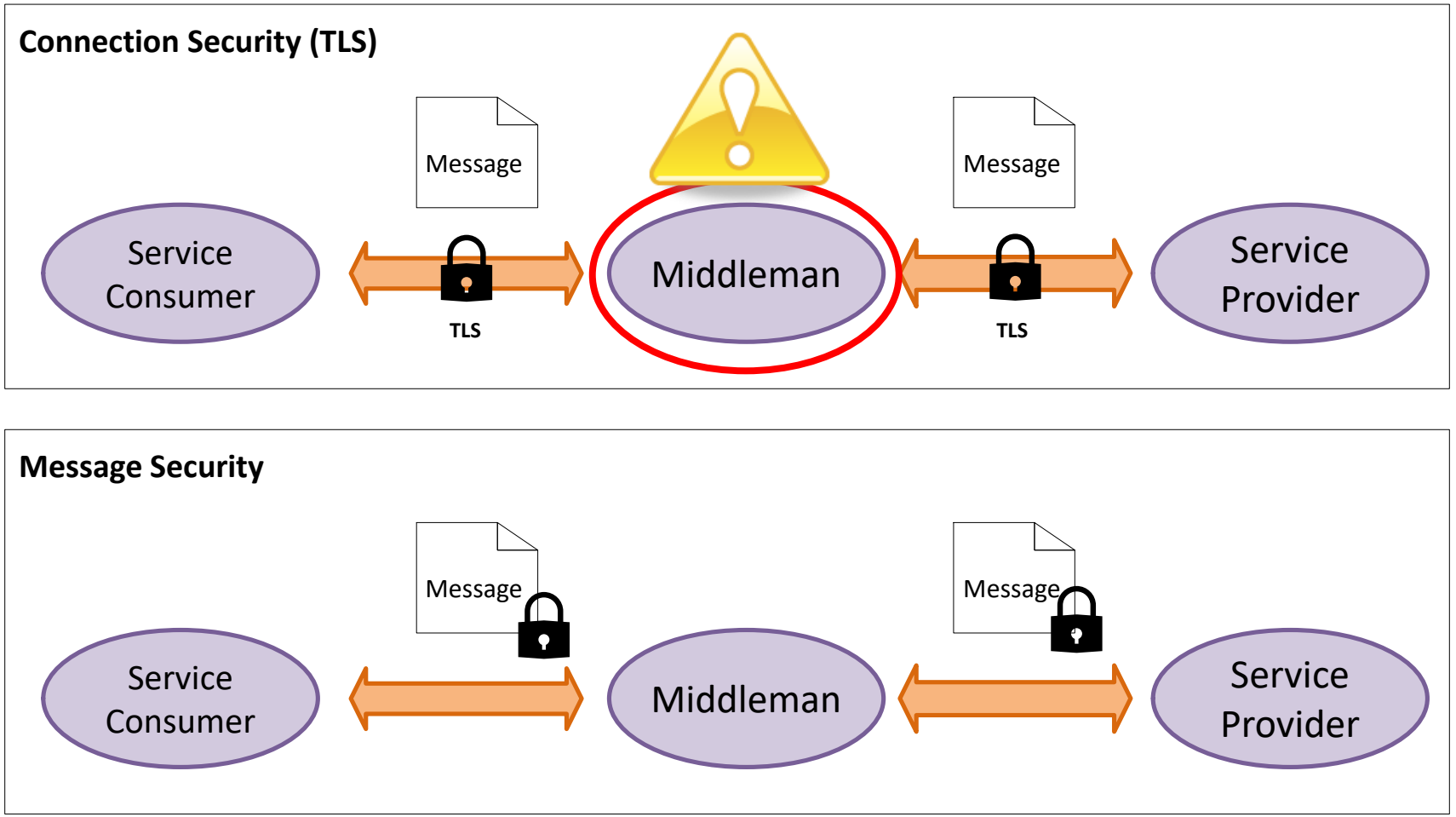
# *Network and Lower Level Security*

- Communications can be secures at almost any level of the TCP/IP architecture.
- Usually makes sense to secure at network, transport and/or application levels.
- Cabled networks traditionally do not provide network level security, since the medium is not shared or is only shared by a few parties.
  - However, shared medium networks, like Wi-Fi need to be secure at the network or a lower level.
  - Wi-Fi supports several security standards:
    - WEP (Wire Equivalent Protection): Broken.
    - WPA (Wi-Fi Protected Access): Intermediate measure. Deprecated.
    - WPA2 (Wi-Fi Protected Access II): Broken.
    - WPA3 (Wi-Fi Protected Access III): Announced January 2018. Not available yet.

# *Transport Level Security*

- Transport is the first end-to-end layer of the TCP/IP stack.
- Therefore, it makes sense to secure communications at the transport level. If done so, no intermediate point in the connection can spy the communications.
- The most widespread security protocol for the network layer is TLS:
  - Transport Layer Security.
  - Successor to the broken SSL.
  - Designed to run over a reliable communication protocol (typically TCP).
  - Most Internet security protocols (FTPS, HTTPS, …) are just an unsecure protocol running on top of TLS.
- However, some times a connection-level security is not enough, since there exist middlemen that should be able to modify and look at some parts of a message, but not at others…
  - The contents of the message itself must be encrypted. The complete message or just some parts.
  - Sometimes, it is advisable to use encryption at several levels at the same time.

# *Connection VS. Message Security*

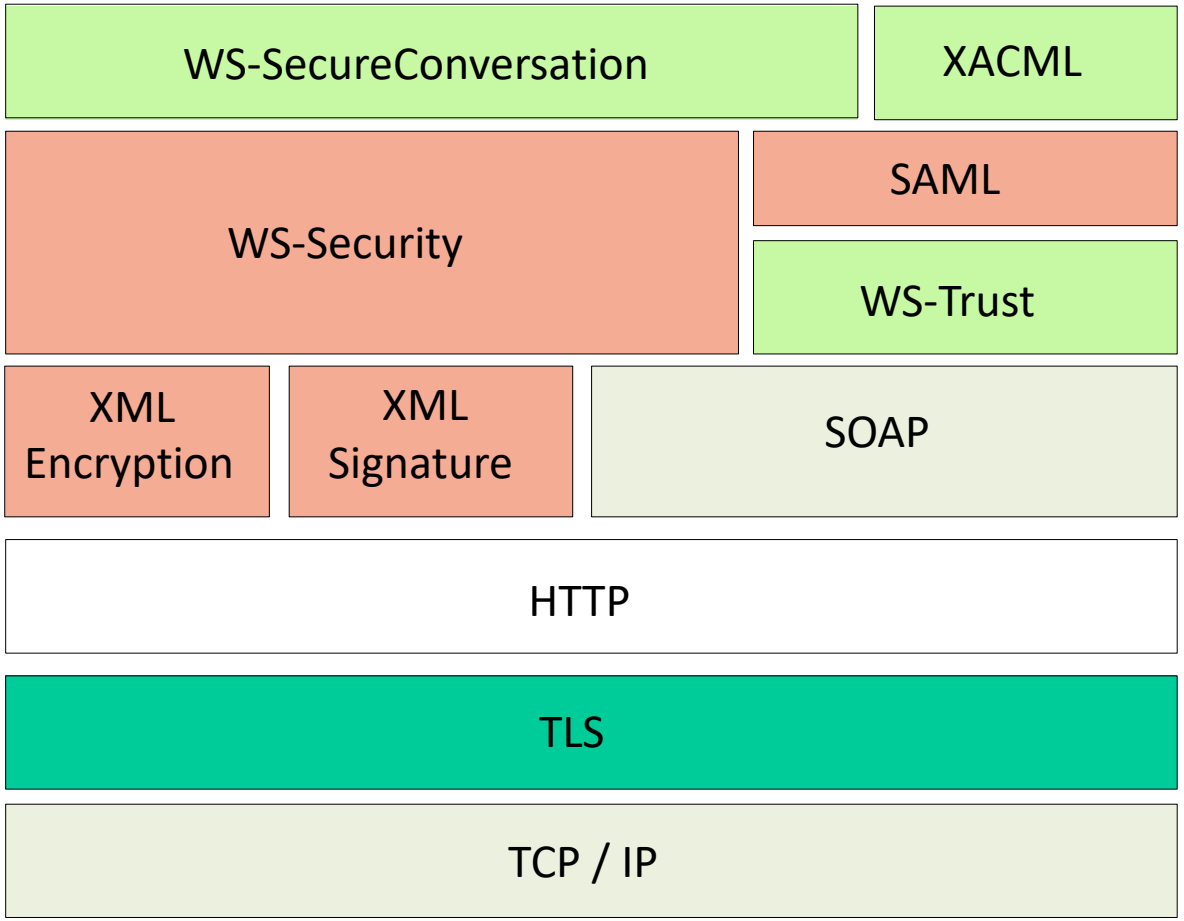


# *SECURITY STANDARDS FOR SOA*

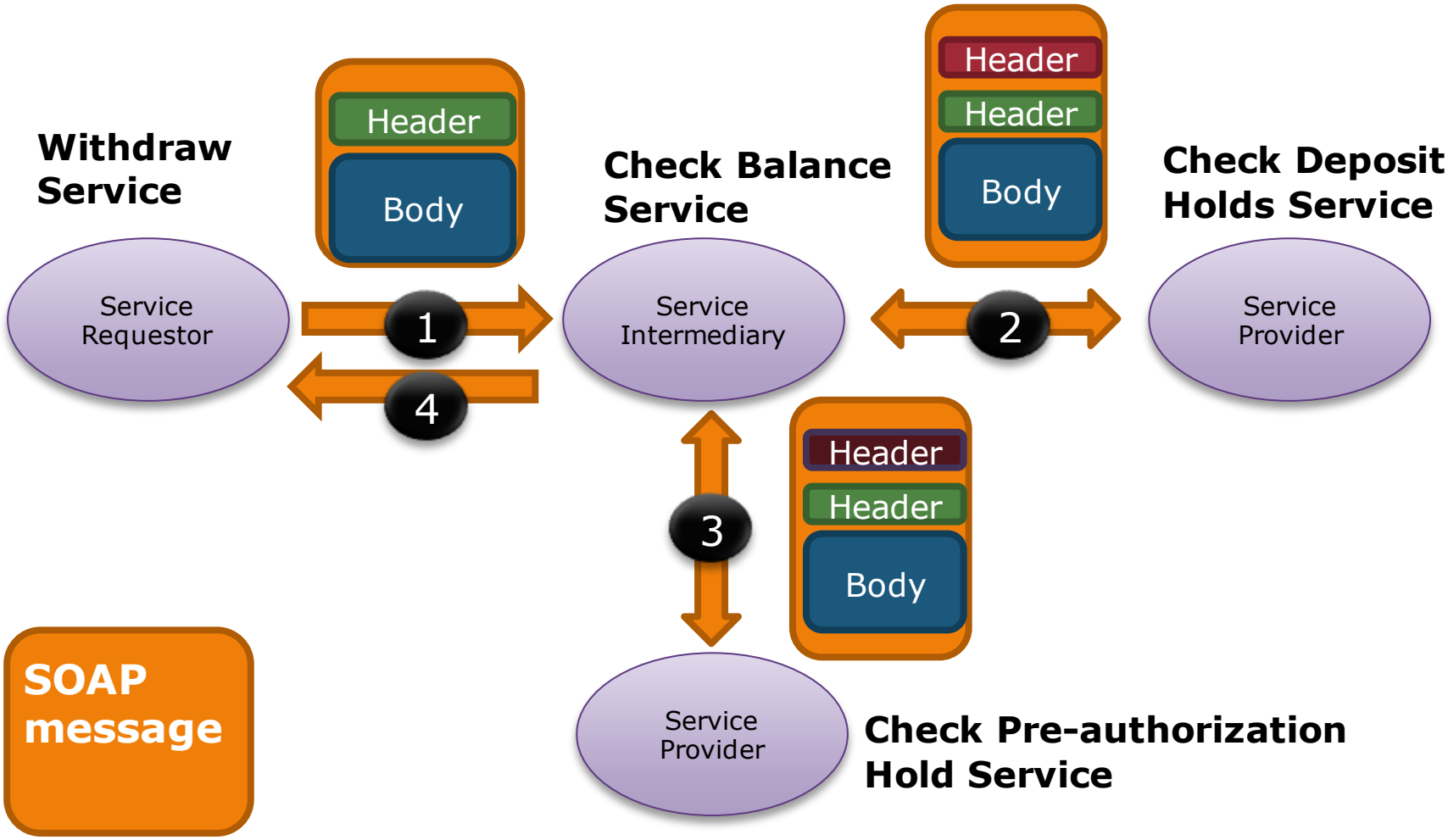
# *REST vs. SOAP*

- REST and SOAP follow different approaches to security implementation.
- REST is simple and straightforward…
  - …but does not provide any security standard at all.
  - Relies in:
    - HTTPS (with TLS) for connection-level security.
    - An external sign on service for authentication and authorization.
- SOAP is more complex and very business oriented.
  - Can use the same security services that REST uses…
  - …but also has its own set of standards for more complex (business) security scenarios.

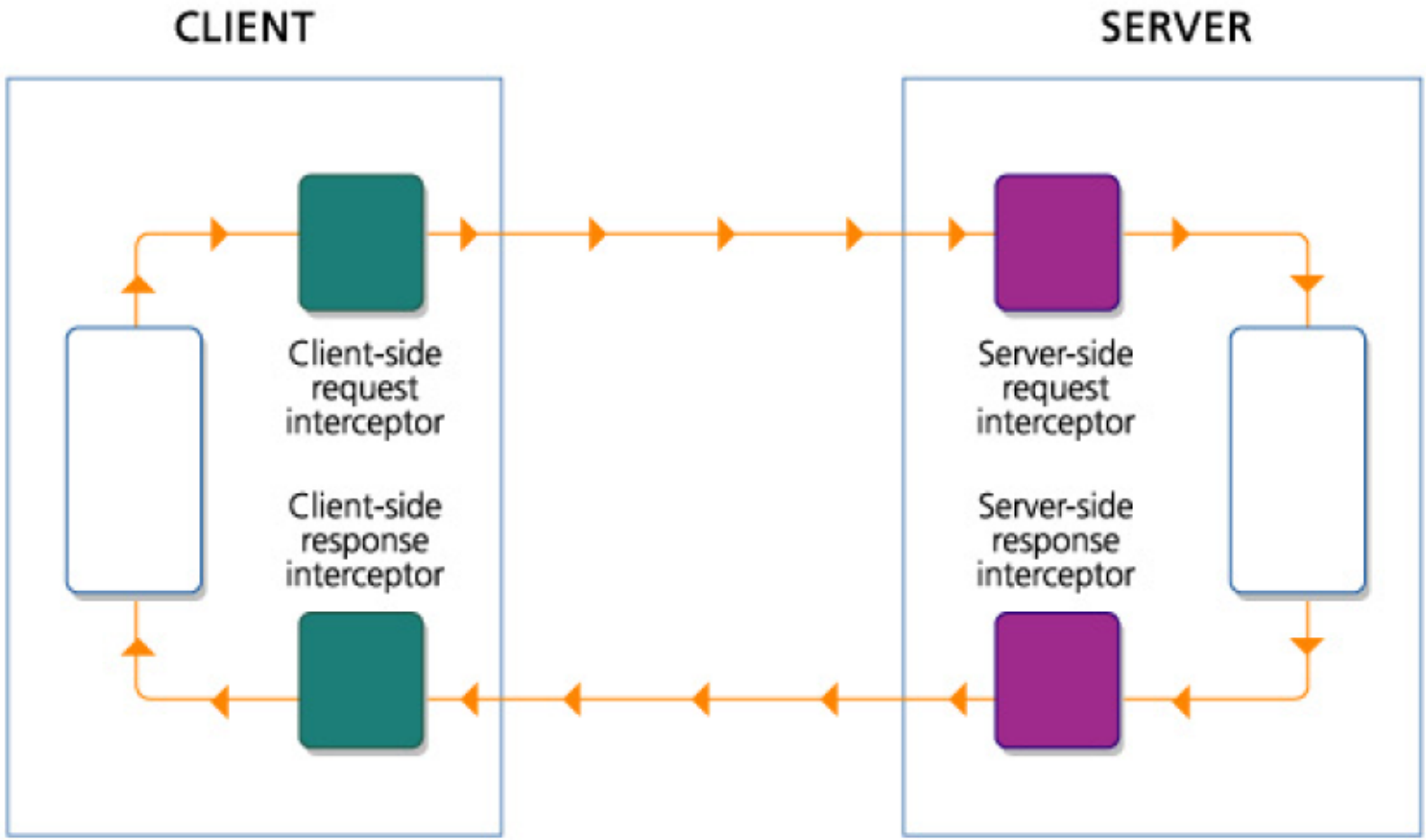
# *SOAP Security Standards*



# *Security Through SOAP Headers*



# *Security Management Through Interceptors*



# *WS-Security*

- Security extension for SOAP messages;
  - WS-Security 1.1, OASIS standard since 2006.
- Defines an additional header that contains all the security information.
  - Does not provide any security features by itself.
  - Provides support for specific extensions that implement such features:
    - Authentication modes (simple, certificate, tokens…)
    - Authorization
    - Negotiation
    - Signature
    - Encryption

# *Confidentiality in SOAP*

- XML Encryption 1.0
  - W3C standard since 2002
  - Specifies how to encrypt fragments of XML messages
  - Currently broken
- Used to encrypt parts of a SOAP message
  - The header contains the encryption method, the algorithm and the key
  - Several parts can be signed independently (and for different receivers)
  - Headers to avoid unwanted receivers
  - Segments in body to protect sensible information

# *Authentication in SOAP*

- Credentials are managed through a WS-Security header included in the SOAP messages.
- Several authentication modes:
  - User / password
  - Kerberos ticket
  - X.509 Certificates / PKI Infraestructure
  - SAML assertions
  - SSO (Single Sign On)

# *Example of Basic Authentication*

```
<soapenv:Envelope>
<soapenv:Header>
    <wsse:Security soapenv:actor="..." >
        <wsse:UsernameToken>
            <wsse:Username>usuario</wsse:Username>
        <wsse:Password>1234</wsse:Password>
        </wsse:UsernameToken>
    </wsse:Security>
</soapenv:Header>
<soapenv:Body>
...
</soapenv:Body>
</soapenv:Envelope>
```

CEU

# *Authorization in SOAP*

- SAML (Security Assertion Markup Language) is an XML-based language used to send security assertions.
  - SAML 2.0 is an OASIS standard since 2005.
  - **Authorization** / Authentication / Attributes.
  - WS-Security headers are transmitted.
  - It is the result or invoking the services of an identity provider (WS-Trust).

# *Integrity in SOAP*

- Digital signature of the whole message or parts of it
  - Non modification guarantee.
  - The message is exactly as it was sent.
- XML Signature 1.1
  - W3C Standard since 2013
  - Defines what and how SOAP messages are signed.
  - The whole message or just some parts can be signed.
  - To avoid false positives the canonical XML form must be used
    - XML Canonical 1.1, W3C Standard since 2008

# *Non Repudiation in SOAP*

- The XML signature that accompanies the SOAP message certifies the integrity of the signed content.
  - To guarantee non repudiation we must check that the sender's certificate is valid.
- To avoid repetition attacks it is recommended that the signature includes the timestamp and nonce of the message.

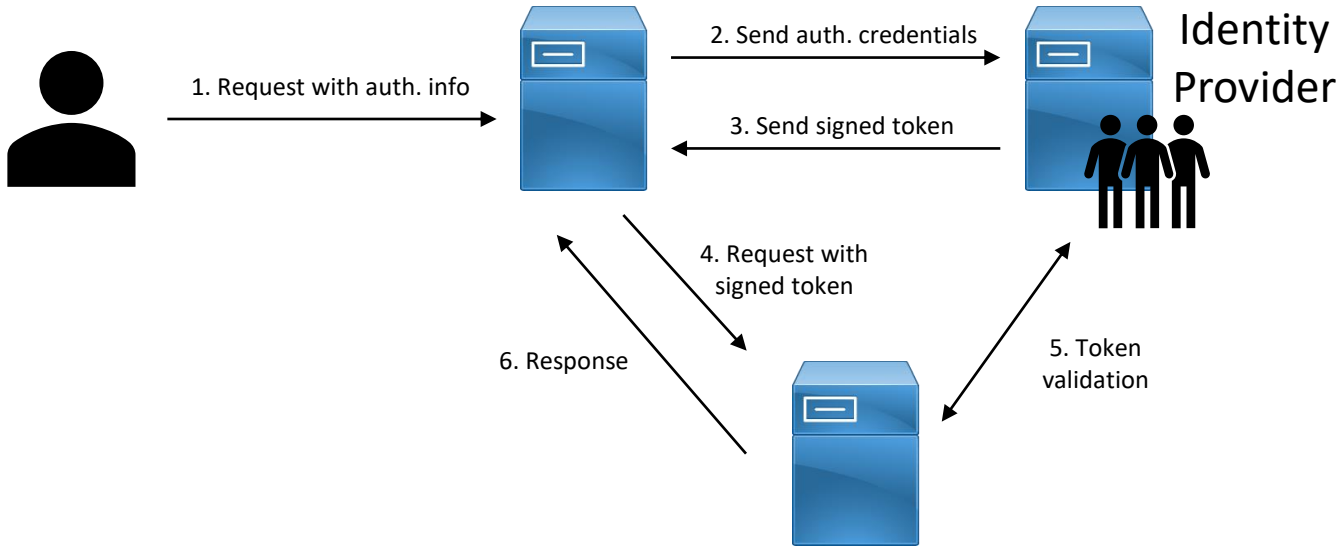
# *SOA SECURITY SCENARIOS*

# *The Security Service*

- Managing all the security features imposes a huge toll in every element of a SOA architecture.
- **Alternative:** use a security service.
  - Simplifies the management of some aspects of the security (authorization and authentication) of each specific service (they only need to know how to communicate with the service).
  - Service interfaces specially designed to provide security.
  - Communication of security information.

# *WS-Trust*

- WS-Trust is an specification that defines the services used to obtain and communication security information.
  - WS-Trust 1.4 is an OASIS standard since 2009.
- Can issue, validate, renew and cancel security tokens.
  - Implemented using a Security Token Service (Identity Provider)



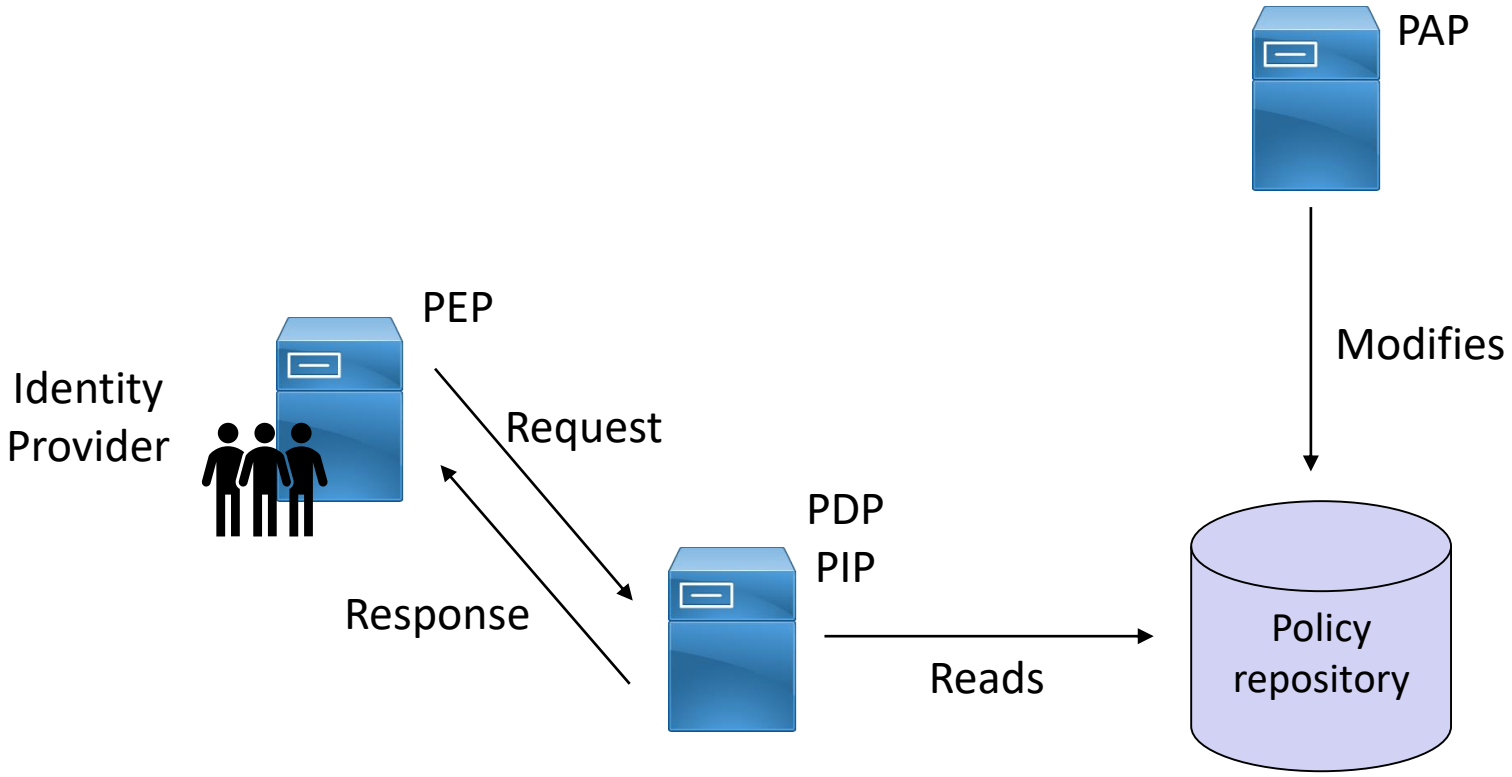
# *WS-SecureConversation*

- Asymmetric encryption:
  - More secure, there is no need to share a key.
  - More arduous, has a higher resource usage.
- WS-SecureConversation 1.4
  - An OASIS standard since 2009.
  - Establishes a symmetric encryption using an asymmetric encryption.
  - Creates a Security Context Token (SCT).
  - The SCT (for the symmetric encryption) is sent using asymmetric encryption.
  - Uses SAML assertions and a WS-Trust server.

# *Authorization Through Policies*

- Authorization can be defined using security policies.
  - XACML (eXtensible Access Control Markup Language) 3.0 is an OASIS standard since 2009.
  - Policies are defined as rules:
    - Who can access which resources.
    - Under which obligations and conditions.
  - Permissions are codified using SAML assertions.
- Roles:
  - PAP (Policy Administration Point): *Manages*
  - PDP (Policy Decision Point): *Evaluates*
  - PEP (Policy Enforcement Point): *Executes*
  - PIP (Policy Information Point): *Reports*

# *Policy Management in the Identity Provider*



# *CLOUD SECURITY*

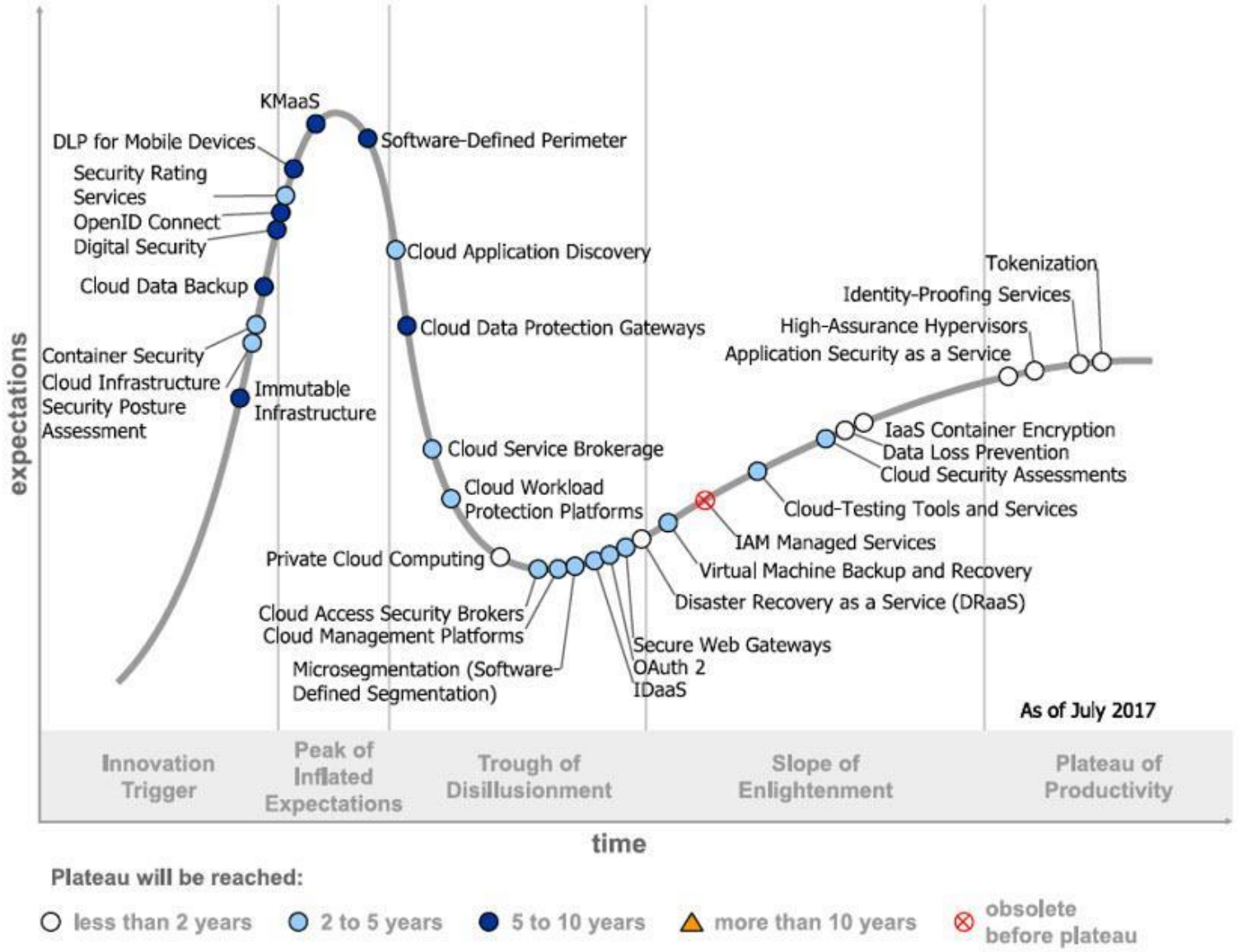
# *Cloud Security*

- Security in the cloud is complicated and difficult to understand.
  - Security continues to be the most commonly cited reason for avoiding the use of public cloud.
  - But, paradoxically, the organizations already using public cloud infrastructures consider security to be one of the primary benefits.
- The attack resistance of the majority of cloud service providers has not proven to be a major weakness so far…
  - …but customers of these services may not know how to use them securely.
- If the information is in another party's servers, how can it be secure?
  - Sometimes it is, in fact, more secure.
  - Though regulations can be tricky.

# *Cloud Security Improvements*

- Cloud computing can offer several security improvements:
  - **Disaster Recovery as a Service (DRaaS)** can help small companies without specialized staff and/or a recovery datacenter. It is part of the more generic **Security as a Service (SECaaS)**.
  - **Private cloud computing** can leverage most benefits of the cloud while meeting the regulatory, functionality and intellectual property protection needs of some companies.
  - **Data loss protection services** in the cloud can help mitigate or prevent the disclosure of sensitive or regulated information. It helps to identify broken business processes and enforce some policies and procedures.
  - **Container encryption** allows the protection of data stored in cloud providers, like a full disk encryption would do.
  - **Tokenization** allows a piece of sensitive data to be replaced by a surrogate value known as a token, which in turn is securely stored in a centralized, more secure, location.
  - **Identity proofing services** can automate background checks and be used as an additional step for an authentication method.

**Figure 1. Hype Cycle for Cloud Security, 2017**




*Source: Gartner (September 2017)*