Security and Privacy

Software security

- Software security should always be a high priority for product developers and their users.
- If you don’t prioritize security, you and your customers will inevitably suffer losses from malicious attacks.
- In the worst case, these attacks could can put product providers out of business.
  - If their product is unavailable or if customer data is compromised, customers are liable to cancel their subscriptions.
- Even if they can recover from the attacks, this will take time and effort that would have been better spent working on their software.

Figure 7.1 Types of security threat

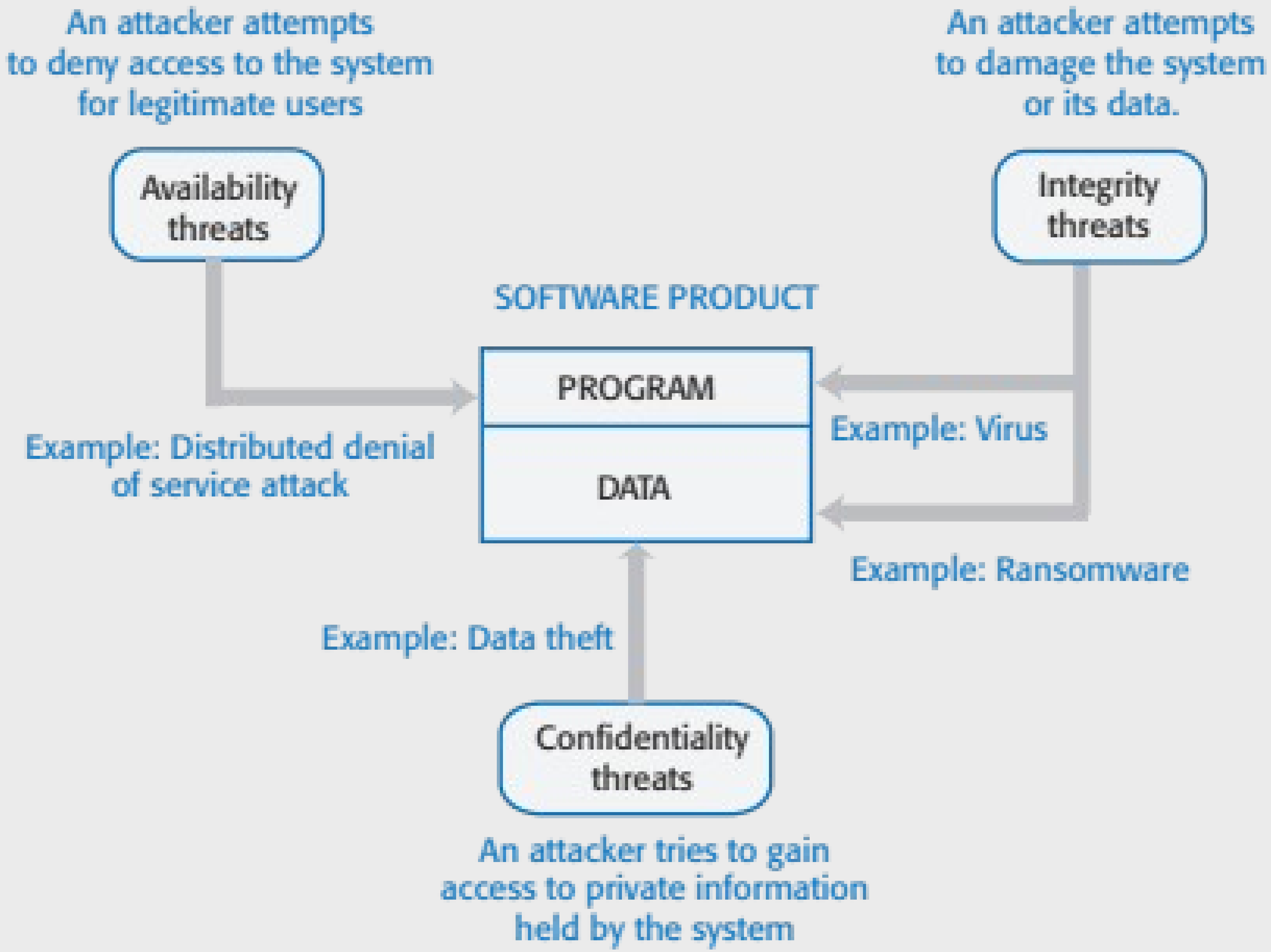

An attacker attempts to deny access to the system for legitimate users
Availability threats
An attacker attempts to damage the system or its data.
Integrity threats
SOFTWARE PRODUCT
PROGRAM
DATA
Example: Distributed denial of service attack
Example: Virus
Example: Ransomware
Example: Data theft
Confidentiality threats
An attacker tries to gain access to private information held by the system

Figure 7.2 System infrastructure stack

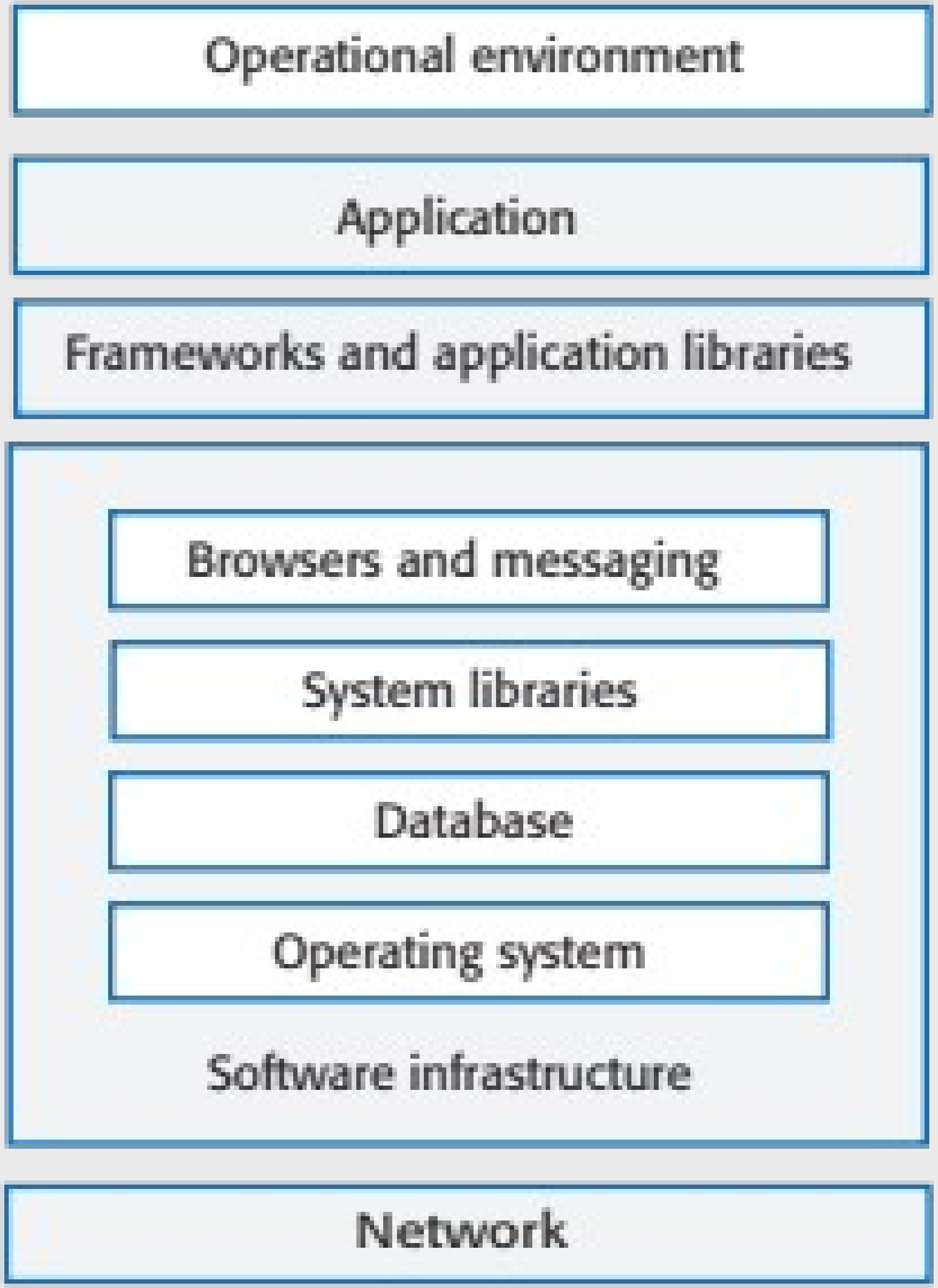


Table 7.1 Security management

Authentication and authorization
You should have authentication and authorization standards and procedures that ensure that all users have strong authentication and that they have properly access permissions properly. This minimizes the risk of unauthorized users accessing system resources.

System infrastructure management
Infrastructure software should be properly configured and security updates that patch vulnerabilities should be applied as soon as they become available.

Attack monitoring
The system should be regularly checked for possible unauthorized access. If attacks are detected, it may be possible to put resistance strategies in place that minimize the effects of the attack.

Backup
Backup policies should be implemented to ensure that you keep undamaged copies of program and data files. These can then be restored after an attack.

Operational security

- Operational security focuses on helping users to maintain security. User attacks try to trick users into disclosing their credentials or accessing a website that includes malware such as a key-logging system.

- Operational security procedures and practices

  - *Auto-logout*, which addresses the common problem of users forgetting to logout from a computer used in a shared space.

  - *User command logging*, which makes it possible to discover actions taken by users that have deliberately or accidentally damaged some system resources.

  - *Multi-factor authentication*, which reduces the chances of an intruder gaining access to the system using stolen credentials.

Injection attacks

- Injection attacks are a type of attack where a malicious user uses a valid input field to input malicious code or database commands.
- These malicious instructions are then executed, causing some damage to the system. Code can be injected that leaks system data to the attackers.
- Common types of injection attack include buffer overflow attacks and SQL poisoning attacks.

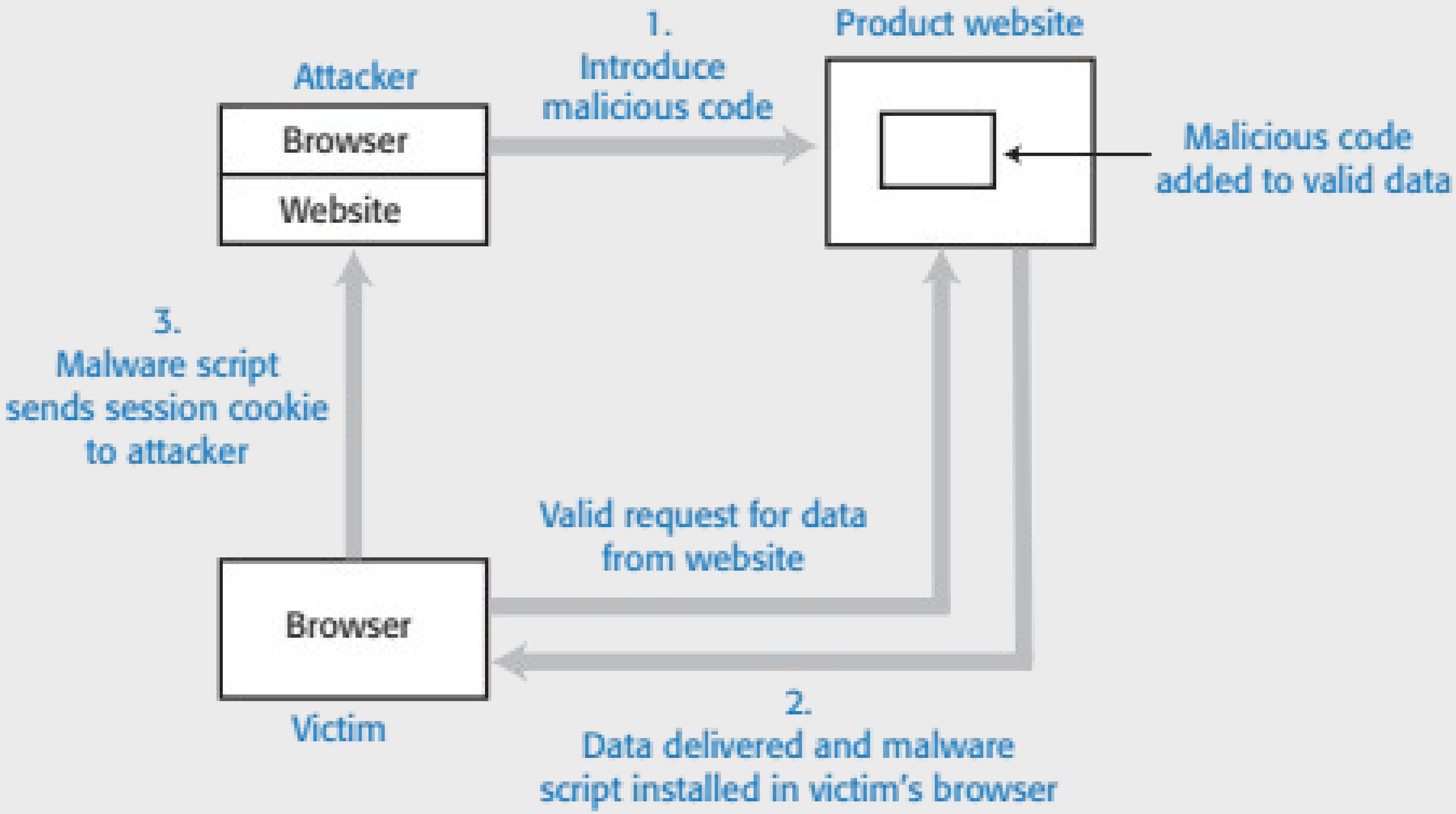
SQL poisoning attacks

- SQL poisoning attacks are attacks on software products that use an SQL database.
- They take advantage of a situation where a user input is used as part of an SQL command.
- A malicious user uses a form input field to input a fragment of SQL that allows access to the database.
- The form field is added to the SQL query, which is executed and returns the information to the attacker.

Cross-site scripting attacks

- Cross-site scripting attacks are another form of injection attack.
- An attacker adds malicious Javascript code to the web page that is returned from a server to a client and this script is executed when the page is displayed in the user's browser.
- The malicious script may steal customer information or direct them to another website.
  - This may try to capture personal data or display advertisements.
  - Cookies may be stolen, which makes a session hijacking attack possible.
- As with other types of injection attack, cross-site scripting attacks may be avoided by input validation.

Figure 7.3 Cross-site scripting attack



Session hijacking attacks

- When a user authenticates themselves with a web application, a session is created.
  - A session is a time period during which the user's authentication is valid. They don't have to re-authenticate for each interaction with the system.
  - The authentication process involves placing a session cookie on the user's device
- Session hijacking is a type of attack where an attacker gets hold of a session cookie and uses this to impersonate a legitimate user.
- There are several ways that an attacker can find out the session cookie value including cross-site scripting attacks and traffic monitoring.
  - In a cross-site scripting attack, the installed malware sends session cookies to the attackers.
  - Traffic monitoring involves attackers capturing the traffic between the client and server. The session cookie can then be identified by analysing the data exchanged.

Table 7.2 Actions to reduce the likelihood of hacking

Traffic encryption
Always encrypt the network traffic between clients and your server. This means setting up sessions using https rather than http. If traffic is encrypted it is harder to monitor to find session cookies.

Multi-factor authentication
Always use multi-factor authentication and require confirmation of new actions that may be damaging. For example, before a new payee request is accepted, you could ask the user to confirm their identity by inputting a code sent to their phone. You could also ask for password characters to be input before every potentially damaging action, such as transferring funds.

Short timeouts
Use relatively short timeouts on sessions. If there has been no activity in a session for a few minutes, the session should be ended and future requests directed to an authentication page. This reduces the likelihood that an attacker can access an account if a legitimate user forgets to log off when they have finished their transactions.

Denial of service attacks

- Denial of service attacks are attacks on a software system that are intended to make that system unavailable for normal use.

- Distributed denial of service attacks (DDOS) are the most common type of denial of service attacks.

  - These involve distributed computers, that have usually been hijacked as part of a botnet, sending hundreds of thousands of requests for service to a web application. There are so many service requests that legitimate users are denied access.

- Other types of denial of service attacks target application users.

  - User lockout attacks take advantage of a common authentication policy that locks out a user after a number of failed authentication attempts. Their aim is to lock users out rather than gain access and so deny the service to these users.

  - Users often use their email address as their login name so if an attacker has access to a database of email addresses, he or she can try to login using these addresses.

- If you don't lock accounts after failed validation, then attackers can use brute-force attacks on your system. If you do, you may deny access to legitimate users.

Brute force attacks

- Brute force attacks are attacks on a web application where the attacker has some information, such as a valid login name, but does not have the password for the site.

- The attacker creates different passwords and tries to login with each of these. If the login fails, they then try again with a different password.

  - Attackers may use a string generator that generates every possible combination of letters and numbers and use these as passwords.

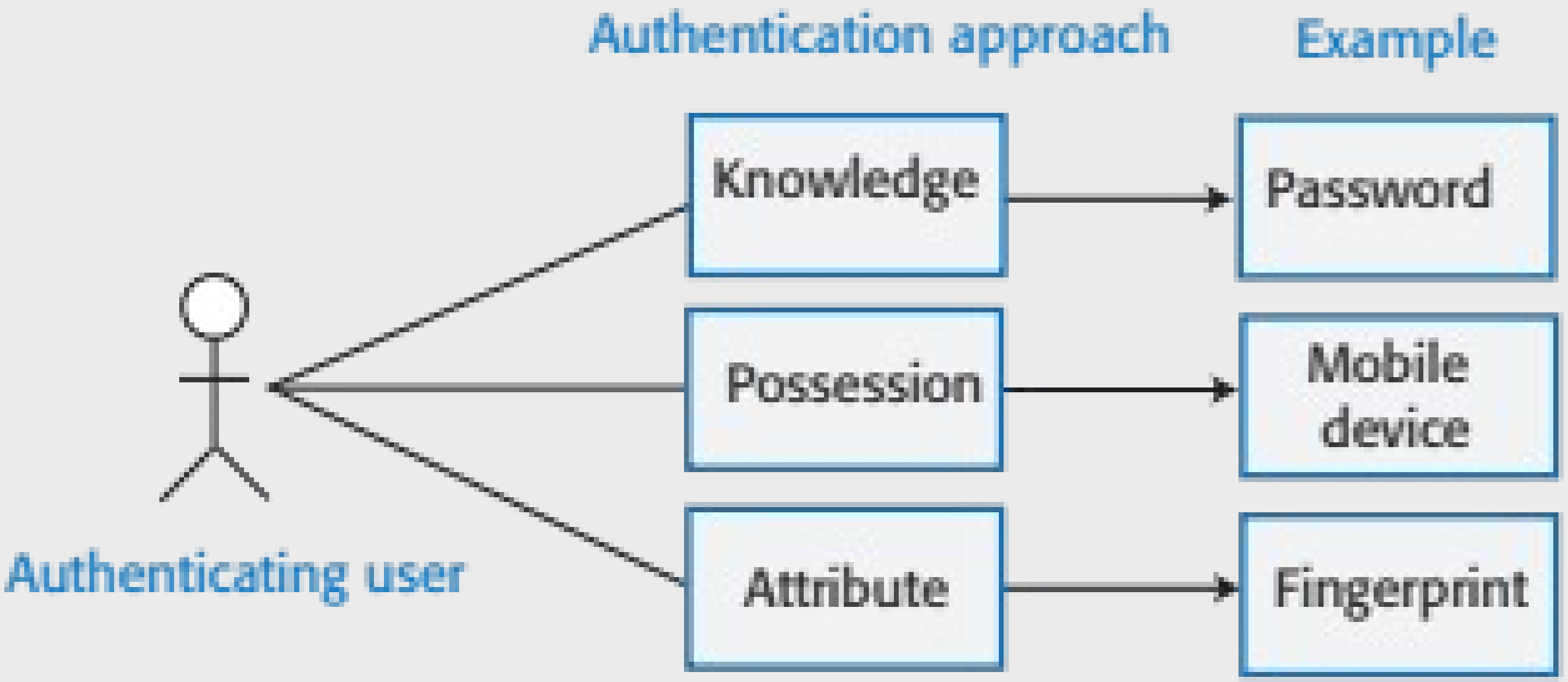
  - To speed up the process of password discovery, attackers take advantage of the fact that many users choose easy-to-remember passwords. They start by trying passwords from the published lists of the most common passwords.

- Brute force attacks rely on users setting weak passwords, so you can circumvent them by insisting that users set long passwords that are not in a dictionary or are common words.

Authentication

- Authentication is the process of ensuring that a user of your system is who they claim to be.
- You need authentication in all software products that maintain user information, so that only the providers of that information can access and change it.
- You also use authentication to learn about your users so that you can personalize their experience of using your product.

Figure 7.4 Authentication approaches



Authentication methods

- Knowledge-based authentication
  - The user provides secret, personal information when they register with the system. Each time they log on, the system asks them for this information.
- Possession-based authentication
  - This relies on the user having a physical device (such as a mobile phone) that can generate or display information that is known to the authenticating system. The user inputs this information to confirm that they possess the authenticating device.
- Attribute-based authentication is based on a unique biometric attribute of the user, such as a fingerprint, which is registered with the system.
- Multi-factor authentication combines these approaches and requires users to use more than one authentication method.

Table 7.3 Weaknesses of password-password-based authentication

Insecure passwords

Users choose passwords that are easy to remember. However, it is also easy for attackers to guess or generate these passwords, using either a dictionary or a brute force attack.

Phishing attacks

Users click on an email link that points to a fake site that tries to collect their login and password details.

Password reuse

Users use the same password for several sites. If there is a security breach at one of these sites, attackers then have passwords that they can try on other sites.

Forgotten passwords

Users regularly forget their passwords so that you need to set up a password recovery mechanism to allow these to be reset. This can be a vulnerability if users' credentials have been stolen and attackers use it to reset their passwords.

Federated identity

- Federated identity is an approach to authentication where you use an external authentication service.

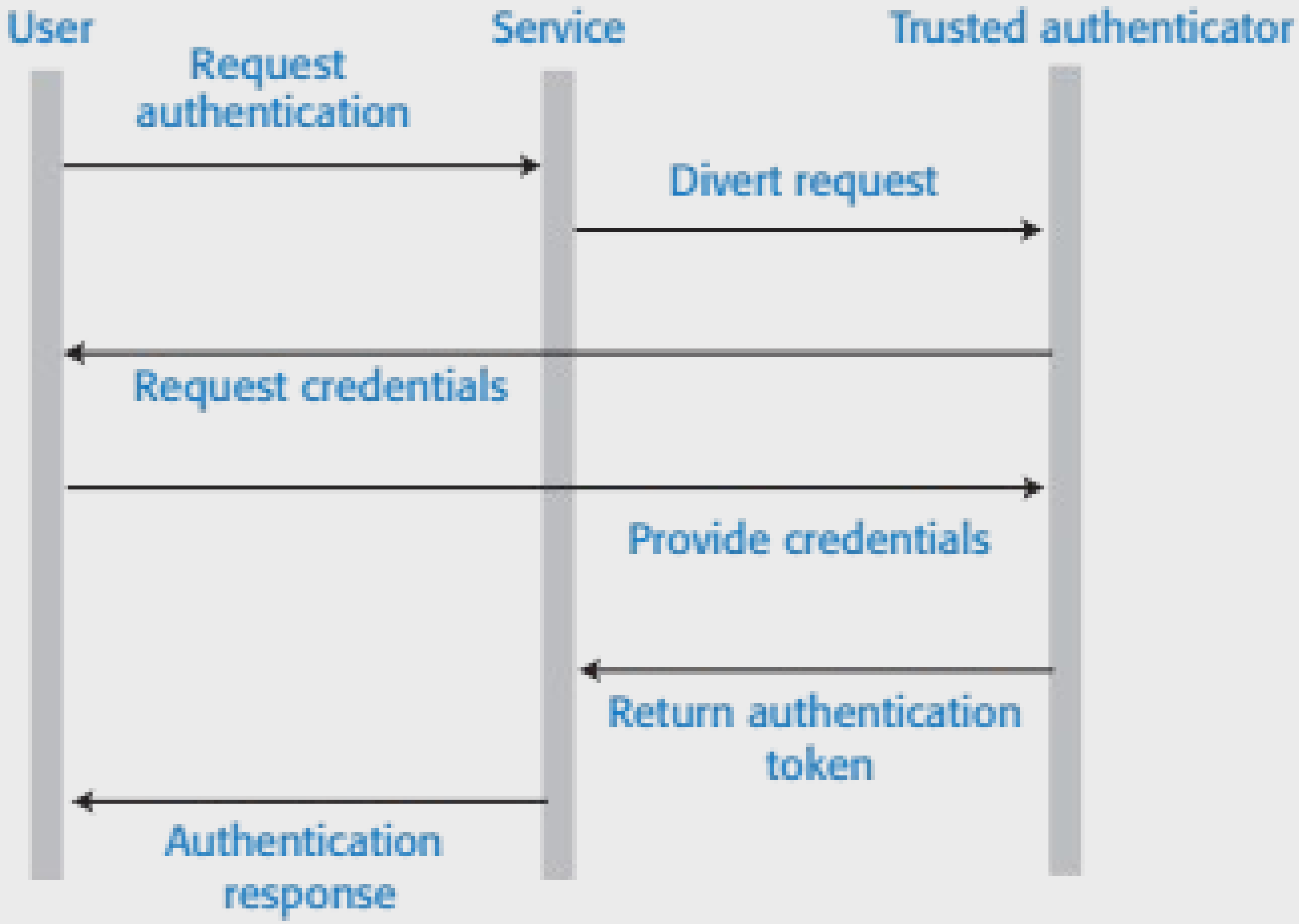
- 'Login with Google' and 'Login with Facebook' are widely used examples of authentication using federated identity.

- The advantage of federated identity for a user is that they have a single set of credentials that are stored by a trusted identity service.

- Instead of logging into a service directly, a user provides their credentials to a known service who confirms their identity to the authenticating service.

- They don't have to keep track of different user ids and passwords. Because their credentials are stored in fewer places, the chances of a security breach where these are revealed is reduced.

Figure 7.5 Federated identity



Authorization

- Authentication involves a user proving their identity to a software system.
- Authorization is a complementary process in which that identity is used to control access to software system resources.
  - For example, if you use a shared folder on Dropbox, the folder's owner may authorize you to read the contents of that folder, but not to add new files or overwrite files in the folder.
- When a business wants to define the type of access that users get to resources, this is based on an access control policy.
- This policy is a set of rules that define what information (data and programs) is controlled, who has access to that information and the type of access that is allowed

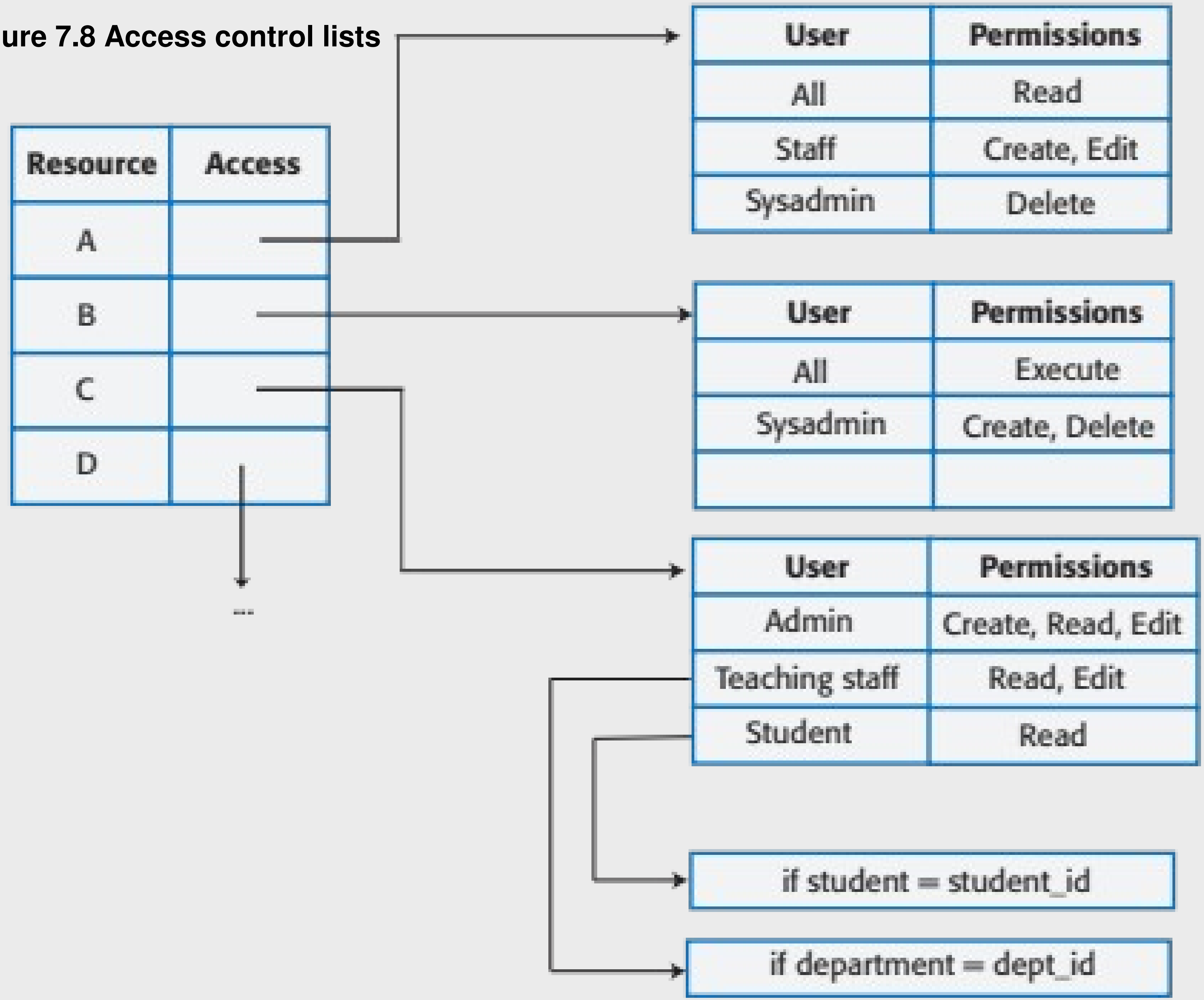
Access control policies

- Explicit access control policies are important for both legal and technical reasons.
  - Data protection rules limit the access the personal data and this must be reflected in the defined access control policy. If this policy is incomplete or does not conform to the data protection rules, then there may be subsequent legal action in the event of a data breach.
  - Technically, an access control policy can be a starting point for setting up the access control scheme for a system.
  - For example, if the access control policy defines the access rights of students, then when new students are registered, they all get these rights by default.

Access control lists

- Access control lists (ACLs) are used in most file and database systems to implement access control policies.
- Access control lists are tables that link users with resources and specify what those users are permitted to do.
  - For example, for this book I would like to be able to set up an access control list to a book file that allows reviewers to read that file and annotate it with comments. However, they are not allowed to edit the text or to delete the file.
- If access control lists are based on individual permissions, then these can become very large. However, you can dramatically cut their size by allocating users to groups and then assigning permissions to the group

Figure 7.8 Access control lists



Resource	Access
A	
B	
C	
D	

...

User	Permissions
All	Read
Staff	Create, Edit
Sysadmin	Delete

User	Permissions
All	Execute
Sysadmin	Create, Delete

User	Permissions
Admin	Create, Read, Edit
Teaching staff	Read, Edit
Student	Read

if student = student_id

if department = dept_id

Encryption

- Encryption is the process of making a document unreadable by applying an algorithmic transformation to it.

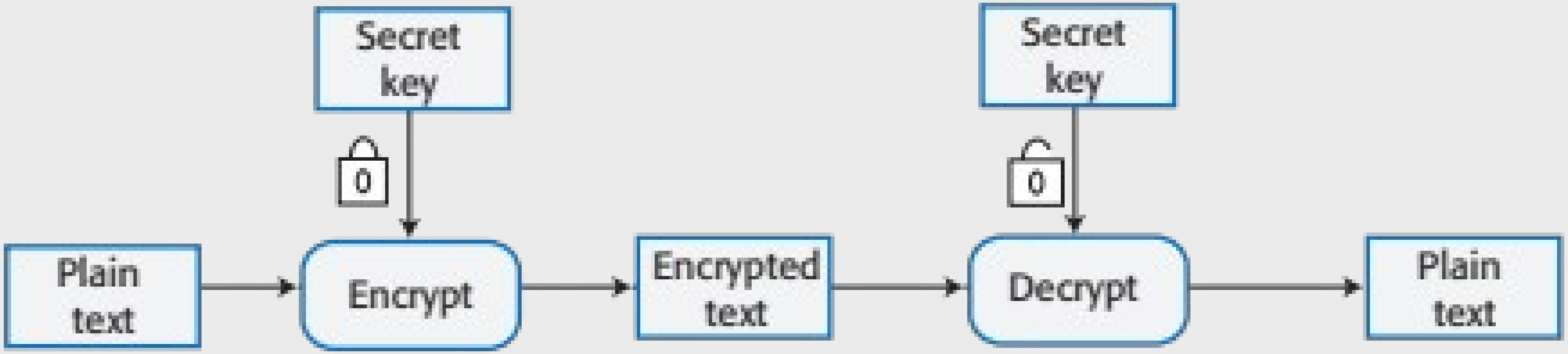
- A secret key is used by the encryption algorithm as the basis of this transformation. You can decode the encrypted text by applying the reverse transformation.

- Modern encryption techniques are such that you can encrypt data so that it is practically uncrackable using currently available technology.

- However, history has demonstrated that apparently strong encryption may be crackable when new technology becomes available.

- If commercial quantum systems become available, we will have to use a completely different approach to encryption on the Internet.

Figure 7.9 Encryption and decryption



Symmetric encryption

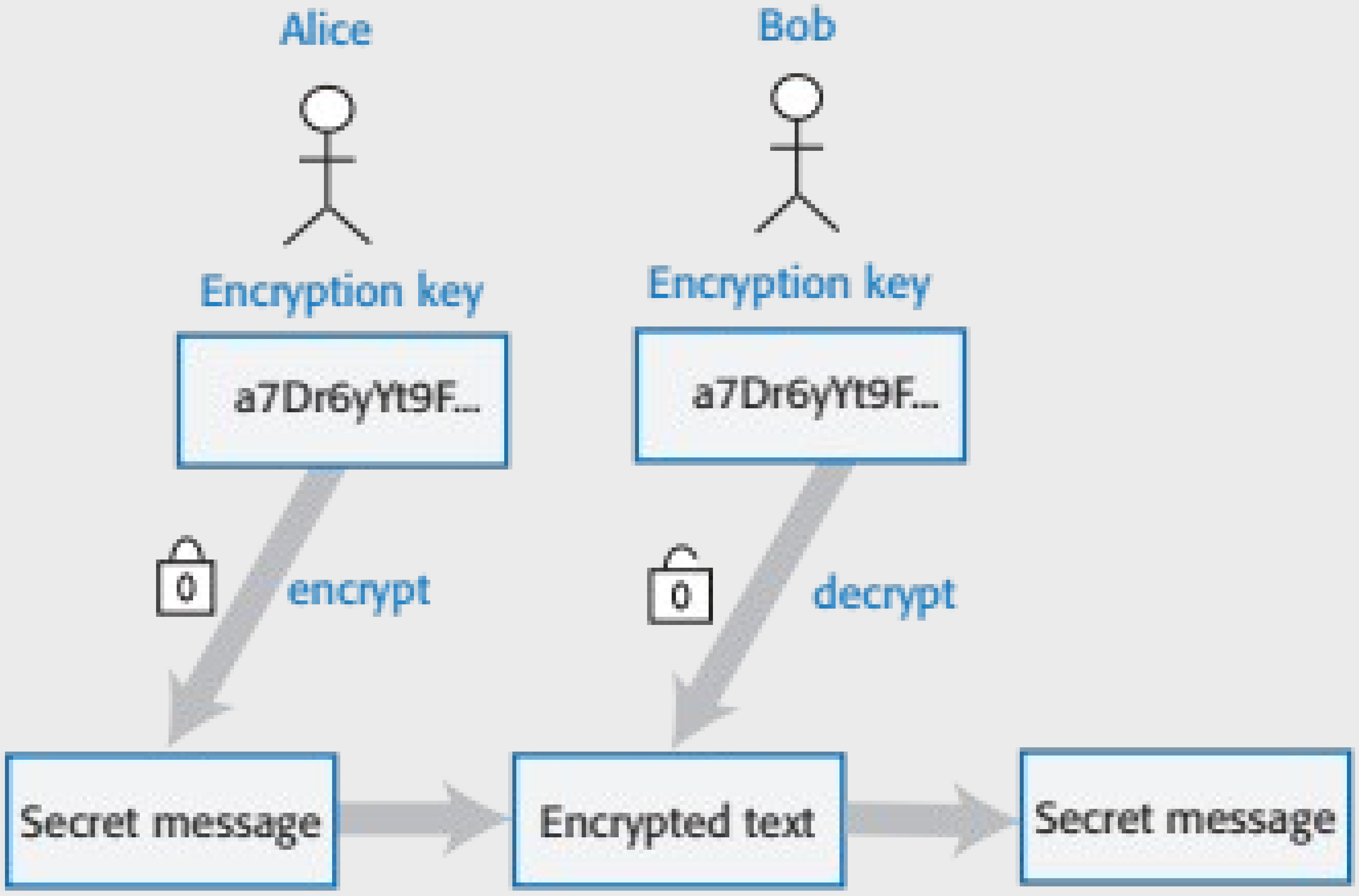
- In a symmetric encryption scheme, the same encryption key is used for encoding and decoding the information that is to be kept secret.

- If Alice and Bob wish to exchange a secret message, both must have a copy of the encryption key. Alice encrypts the message with this key. When Bob receives the message, he decodes it using the same key to read its contents.

- The fundamental problem with a symmetric encryption scheme is securely sharing the encryption key.

- If Alice simply sends the key to Bob, an attacker may intercept the message and gain access to the key. The attacker can then decode all future secret communications.

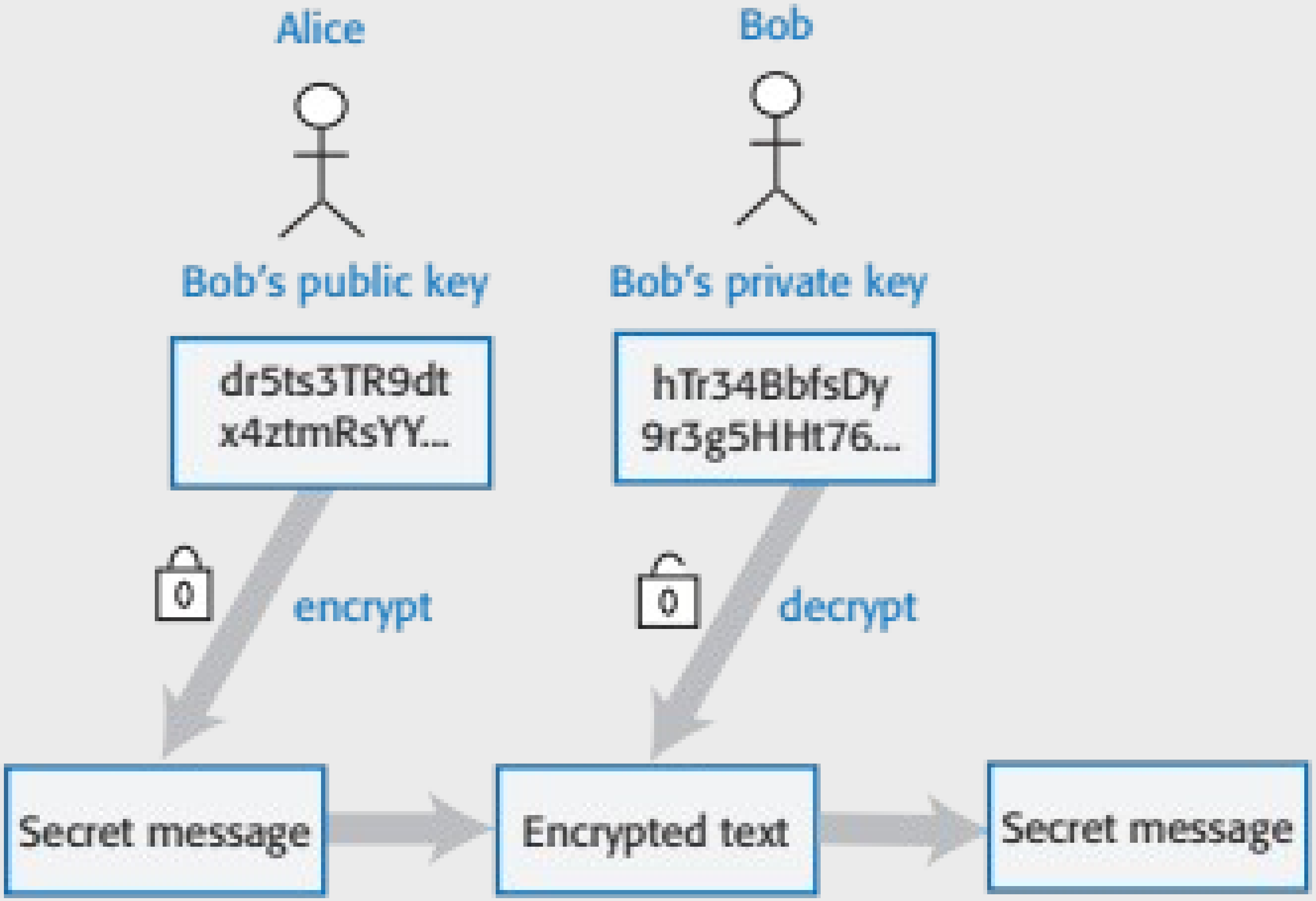
Figure 7.10 Symmetric encryption



Asymmetric encryption

- Asymmetric encryption, does not require secret keys to be shared.
- An asymmetric encryption scheme uses different keys for encrypting and decrypting messages.
- Each user has a public and a private key. Messages may be encrypted using either key but can only be decrypted using the other key.
- Public keys may be published and shared by the key owner. Anyone can access and use a published public key.
- However, messages can only be decrypted by the user's private key so is only readable by the intended recipient

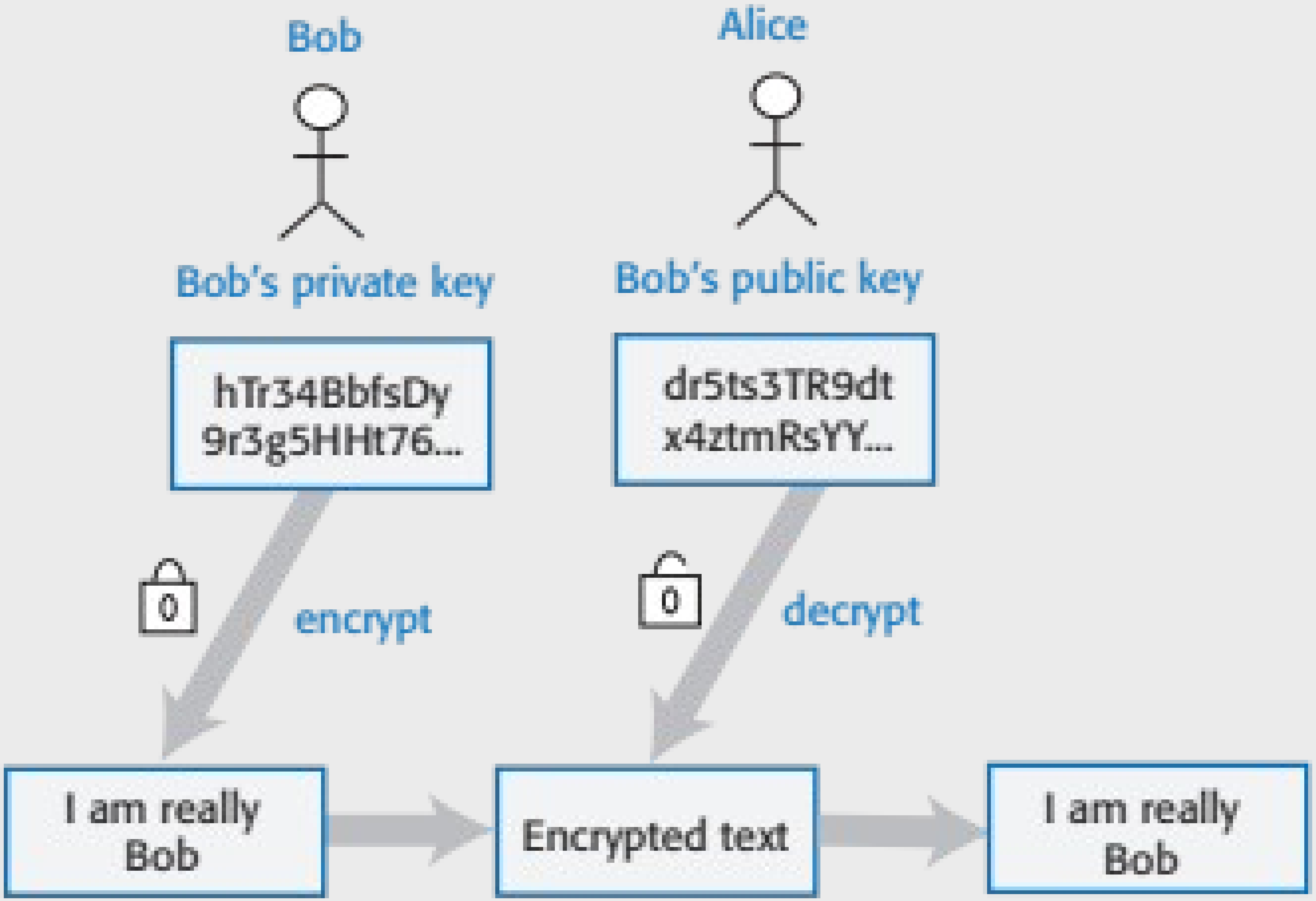
Figure 7.11 Asymmetric encryption



Encryption and authentication

- Asymmetric encryption can also be used to authenticate the sender of a message by encrypting it with a private key and decrypting it with the corresponding public key.
- Say Alice wants to send a message to Bob and she has a copy of his public key.
- However, she is not sure whether or not the public key that she has for Bob is correct and she is concerned that the message may be sent to the wrong person.
- Private/public key encryption can be used to verify Bob’s identity.
    - Bob uses his private key to encrypt a message and sends this to Alice. If it can be decrypted using Bob’s public key, then Alice has the correct key.

Figure 7.12 Encryption for authentication



TLS and digital certificates

- The https protocol is a standard protocol for securely exchanging texts on the web. It is the standard http protocol plus an encryption layer called TLS (Transport Layer Security). This encryption layer is used for 2 things:
  - to verify the identity of the web server;
  - to encrypt communications so that they cannot be read by an attacker who intercepts the messages between the client and the server
- TLS encryption depends on a digital certificate that is sent from the web server to the client.
  - Digital certificates are issued by a certificate authority (CA), which is a trusted identity verification service.
  - The CA encrypts the information in the certificate using their private key to create a unique signature. This signature is included in the certificate along with the public key of the CA. To check that the certificate is valid, you can decrypt the signature using the CA's public key.

Table 7.5 Digital certificates

Subject information
Information about the company or individual whose web site is being visited. Applicants apply for a digital certificate from a certificate authority who checks that the applicant is a valid organization.

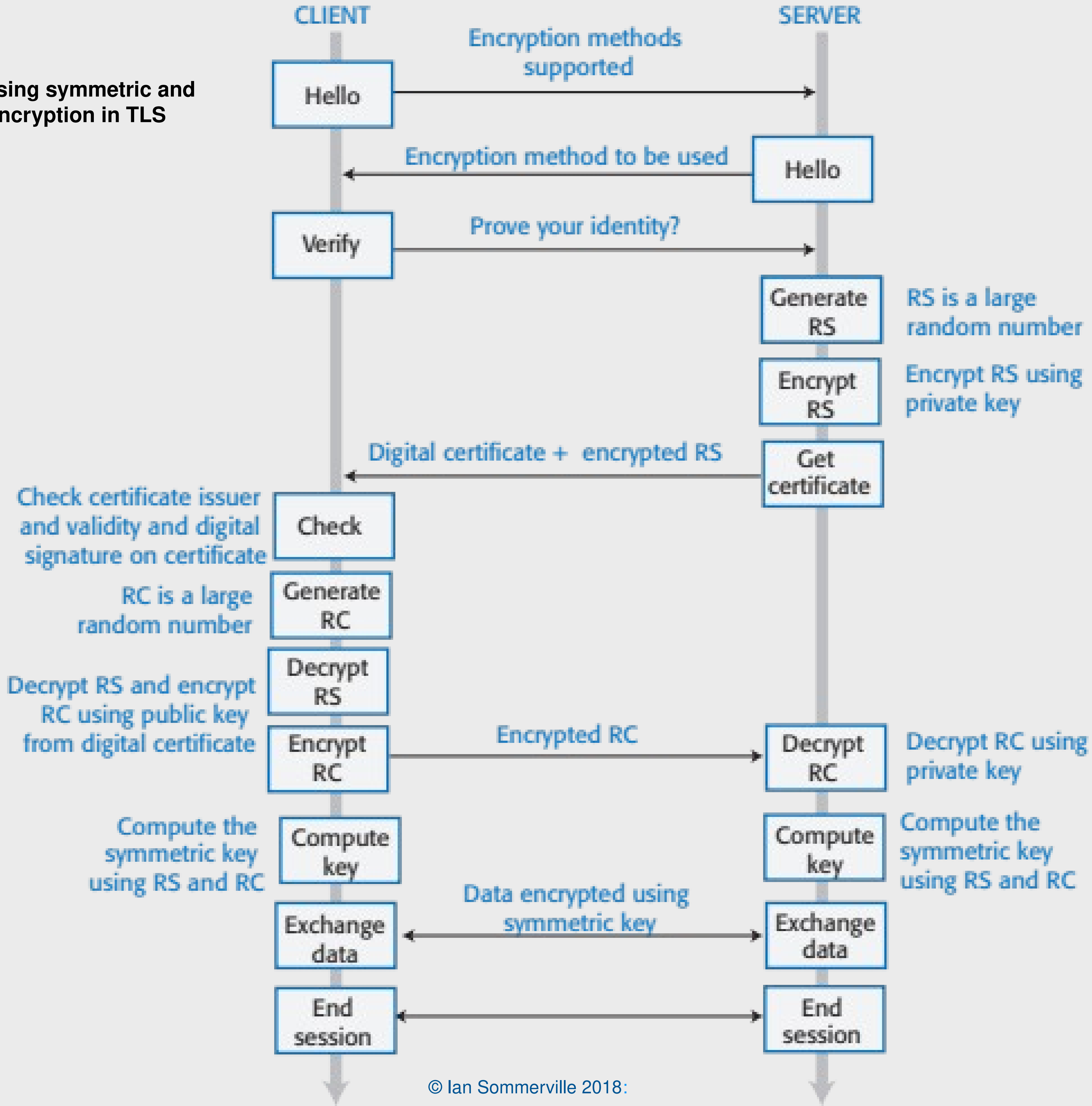
Certificate authority information
Information about the certificate authority (CA) who has issued the certificate.

Certificate information
Information about the certificate itself, including a unique serial number and a validity period, defined by start and end dates.

Digital signature
The combination of all of the above data uniquely identifies the digital certificate. The signature data is encrypted with the CA’s private key to confirm that the data is correct. The algorithm used to generate the digital signature is also specified.

Public key information
The public key of the CA is included along with the key size and the encryption algorithm used. The public key may be used to decrypt the digital signature.

Figure 7.13 Using symmetric and asymmetric encryption in TLS


CLIENT
SERVER
Encryption methods supported
Hello
Encryption method to be used
Hello
Prove your identity?
Verify
Generate RS
RS is a large random number
Encrypt RS
Encrypt RS using private key
Digital certificate + encrypted RS
Get certificate
Check certificate issuer and validity and digital signature on certificate
Check
RC is a large random number
Generate RC
Decrypt RS and encrypt RC using public key from digital certificate
Decrypt RS
Encrypt RC
Encrypted RC
Decrypt RC
Decrypt RC using private key
Compute the symmetric key using RS and RC
Compute key
Compute key
Compute the symmetric key using RS and RC
Data encrypted using symmetric key
Exchange data
Exchange data
End session
End session

TLS explained

- The digital certificate that the server sends to the client includes the server’s public key. The server also generates a long random number, encrypts it using its private key and sends this to the client.
- The client can then decrypt this using the server’s public key and, in turn, generates its own long random number. It encrypts this number using the server’s public key and sends it to the server, which decrypts the message using its private key. Both client and server then have two long random numbers.
- The agreed encryption method includes a way of generating an encryption key from these numbers. The client and server independently compute the key that will be used to encrypt subsequent messages using a symmetric approach.
- All client-server traffic is encrypted and decrypted using that computed key. There is no need to exchange the key itself.

Data encryption

- As a product provider you inevitably store information about your users and, for cloud-based products, user data.

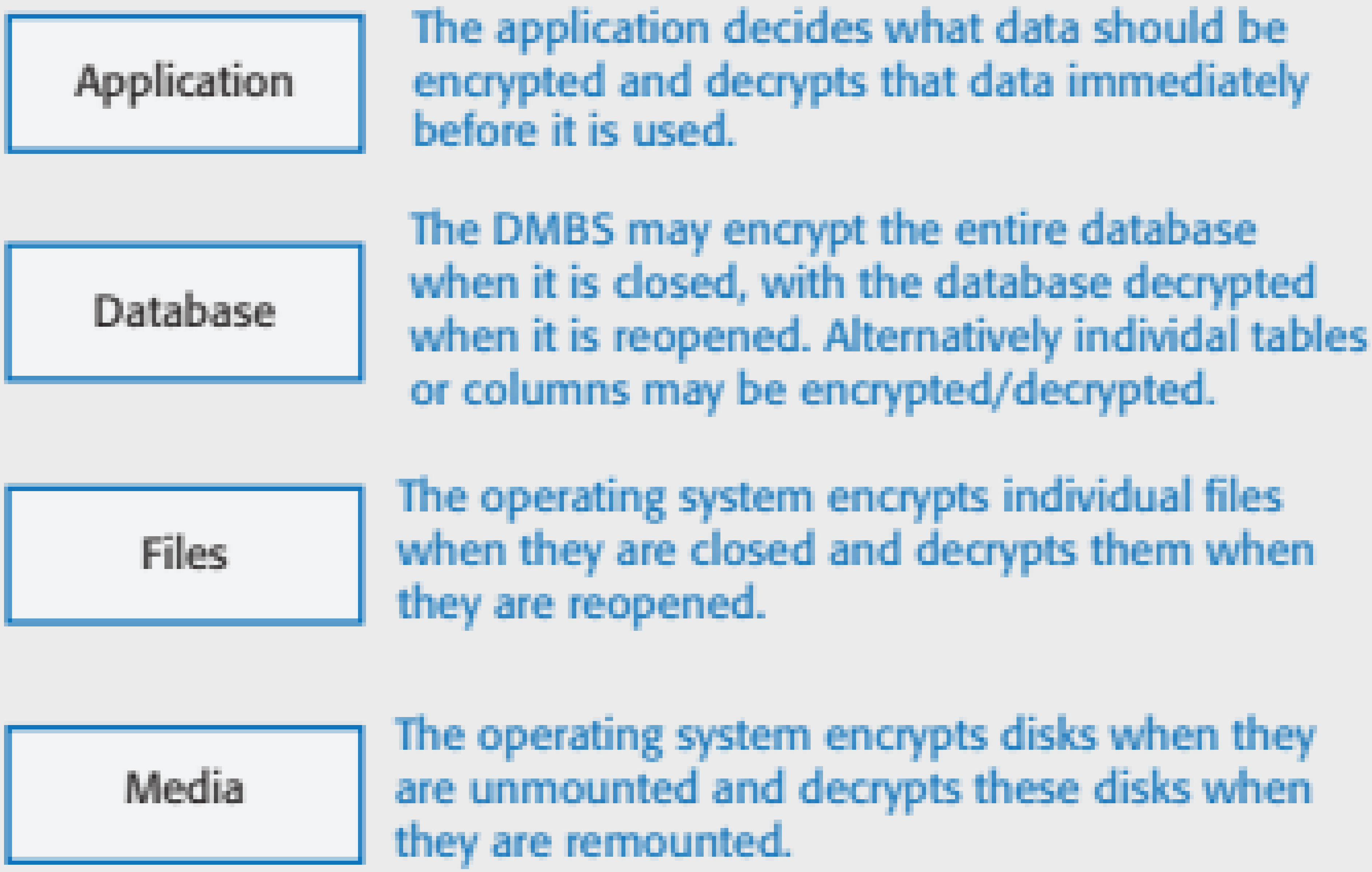
- Encryption can be used to reduce the damage that may occur from data theft. If information is encrypted, it is impossible, or very expensive, for thieves to access and use the unencrypted data.

  - Data in transit.
    When transferring the data over the Internet, you should always use the https rather than the http protocol to ensure encryption.

  - Data at rest.
    If data is not being used, then the files where the data is stored should be encrypted so that theft of these files will not lead to disclosure of confidential information.

  - Data in use
    The data is being actively processed. Encrypting and decrypting the data slows down the system response time. Implementing a general search mechanism with encrypted data is impossible,

Figure 7.14 Encryption levels



Application

The application decides what data should be encrypted and decrypts that data immediately before it is used.

Database

The DMBS may encrypt the entire database when it is closed, with the database decrypted when it is reopened. Alternatively individal tables or columns may be encrypted/decrypted.

Files

The operating system encrypts individual files when they are closed and decrypts them when they are reopened.

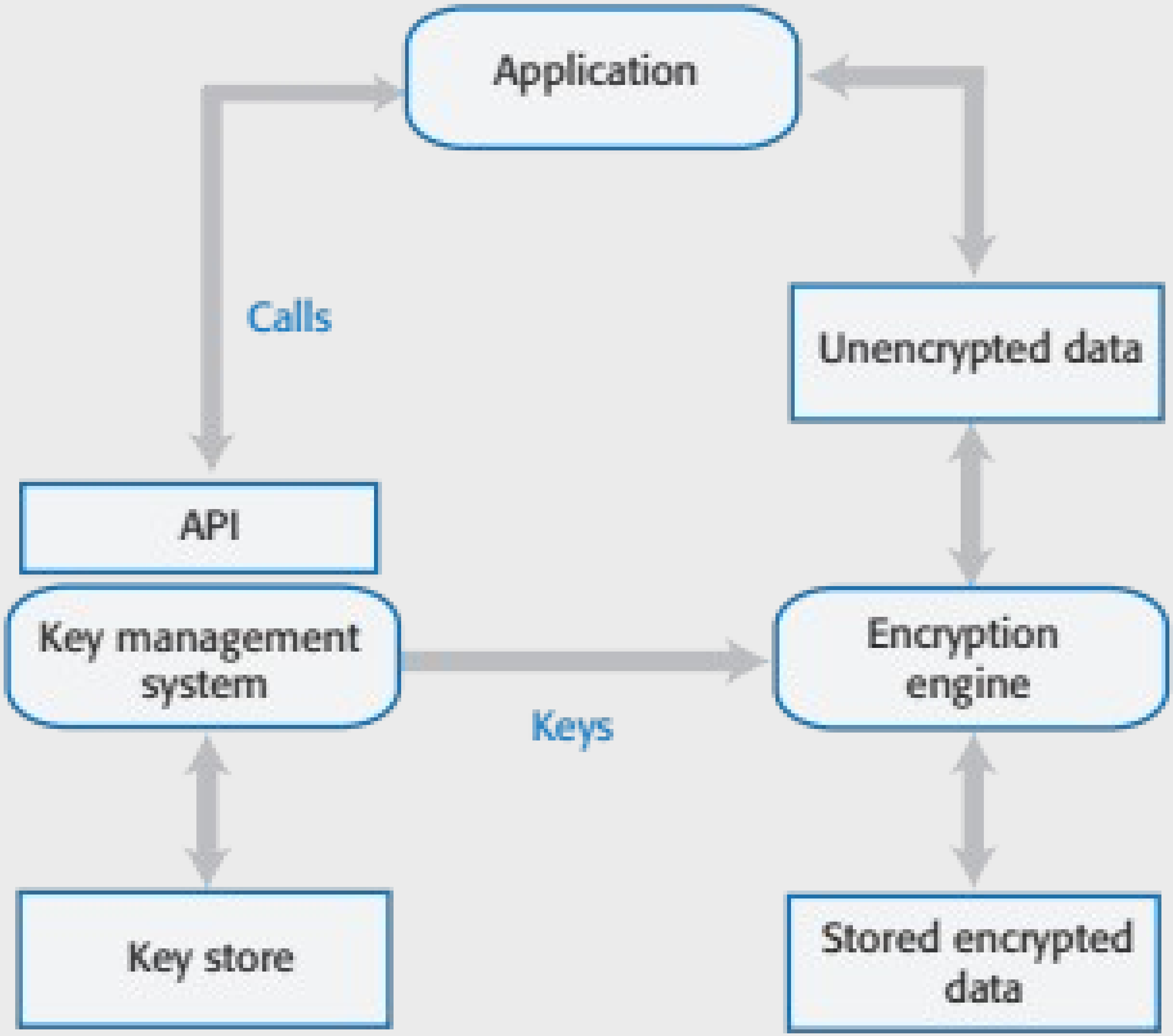
Media

The operating system encrypts disks when they are unmounted and decrypts these disks when they are remounted.

Key management

- Key management is the process of ensuring that encryption keys are securely generated, stored and accessed by authorized users.
- Businesses may have to manage tens of thousands of encryption keys so it is impractical to do key management manually and you need to use some kind of automated key management system (KMS).
- Key management is important because, if you get it wrong, unauthorized users may be able to access your keys and so decrypt supposedly private data. Even worse, if you lose encryption keys, then your encrypted data may be permanently inaccessible.
- A key management system (KMS) is a specialized database that is designed to securely store and manage encryption keys, digital certificates and other confidential information.

Figure 7.15 Using a KMS for encryption management



Long-term key storage

- Business may be required by accounting and other regulations to keep copies of all of their data for several years.
  - For example, in the UK, tax and company data has to be maintained for at least six years, with a longer retention period for some types of data. Data protection regulations may require that this data be stored securely, so the data should be encrypted.
- To reduce the risks of a security breach, encryption keys should be changed regularly. This means that archival data may be encrypted with a different key from the current data in your system.
- Therefore, key management systems must maintain multiple, timestamped versions of keys so that system backups and archives can be decrypted if required.

Privacy

- Privacy is a social concept that relates to the collection, dissemination and appropriate use of personal information held by a third-party such as a company or a hospital.

- The importance of privacy has changed over time and individuals have their own views on what degree of privacy is important.

- Culture and age also affect peoples' views on what privacy means.

  - Younger people were early adopters of the first social networks and many of them seem to be less inhibited about sharing personal information on these platforms than older people.

  - In some countries, the level of income earned by an individual is seen as a private matter; in others, all tax returns are openly published.

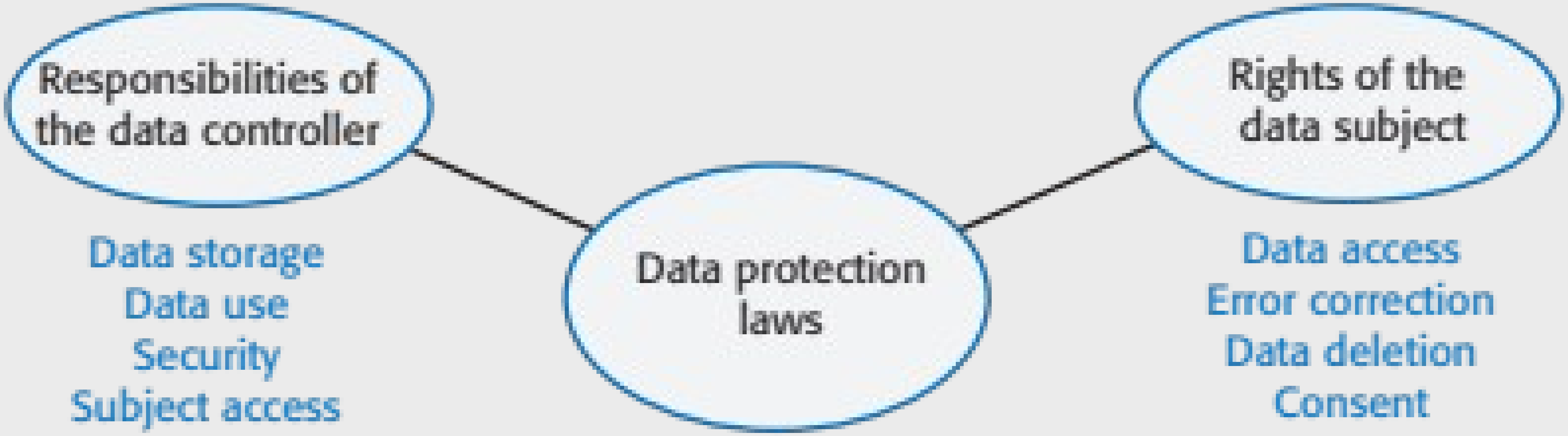
Business reasons for privacy

- If you are offering a product directly to consumers and you fail to conform to privacy regulations, then you may be subject to legal action by product buyers or by a data regulator. If your conformance is weaker than the protection offered by data protection regulations in some countries, you won’t be able to sell your product in these countries.
- If your product is a business product, business customers require privacy safeguards so that they are not put at risk of privacy violations and legal action by users.
- If personal information is leaked or misused, even if this is not seen as a violation of privacy regulations, this can lead to serious reputational damage. Customers may stop using your product because of this

Data protection laws

- In many countries, the right to individual privacy is protected by data protection laws.
- These laws limit the collection, dissemination and use of personal data to the purposes for which it was collected.
  - For example, a travel insurance company may collect health information so that they can assess their level of risk. This is legal and permissible.
  - However, it would not be legal for those companies to use this information to target online advertising of health products, unless their users had given specific permission for this.

Figure 7.16 Data protection laws



Responsibilities of the data controller

Data storage
Data use
Security
Subject access

Data protection laws

Rights of the data subject

Data access
Error correction
Data deletion
Consent

Table 7.6 Data protection principles (1)

Awareness and control
Users of your product must be made aware of what data is collected when they are using your product, and must have control over the personal information that you collect from them.

Purpose
You must tell users why data is being collected and you must not use that data for other purposes.

Consent
You must always have the consent of a user before you disclose their data to other people.

Data lifetime
You must not keep data for longer than you need to. If a user deletes their account, you must delete the personal data associated with that account.

Data protection principles (2)

Secure storage
You must maintain data securely so that it cannot be tampered with or disclosed to unauthorized people.

Discovery and error correction
You must allow users to find out what personal data that you store. You must provide a way for users to correct errors in their personal data.

Location
You must not store data in countries where weaker data protection laws apply unless there is an explicit agreement that the stronger data protection rules will be upheld.

Privacy policy

- You should to establish a privacy policy that defines how personal and sensitive information about users is collected, stored and managed.
- Software products use data in different ways, so your privacy policy has to define the personal data that you will collect and how you will use that data.
- Product users should be able to review your privacy policy and change their preferences regarding the information that you store.
- Your privacy policy is a legal document and it should be auditable to check that it is consistent with the data protection laws in countries where your software is sold.
- Privacy policies should not be expressed to users in a long 'terms and conditions' document that, in practice, nobody reads.
- The GDPR now require software companies to include a summary of their privacy policy, written in plain language rather than legal jargon, on their website.

Key points 1

- Security is a technical concept that relates to a software system's ability to protect itself from malicious attacks that may threaten its availability, the integrity of the system and/or its data, and the theft of confidential information.

- Common types of attack on software products include injection attacks, cross-site scripting attacks, session hijacking attacks, denial of service attacks and brute force attacks.

- Authentication may be based on something a user knows, something a user has, or some physical attribute of the user.

- Federated authentication involves devolving responsibility for authentication to a third-party such as Facebook or Google, or to a business's authentication service.

- Authorization involves controlling access to system resources based on the user's authenticated identity. Access control lists are the most commonly-used mechanism to implement authorization.

- Symmetric encryption involves encrypting and decrypting information with the same secret key. Asymmetric encryption uses a key pair – a private key and a public key. Information encrypted using the public key can only be decrypted using the private key.

Key points 2

- A major issue in symmetric encryption is key exchange. The TLS protocol, which is used to secure web traffic, gets around this problem by using asymmetric encryption for transferring information used to generate a shared key.
- If your product stores sensitive user data, you should encrypt that data when it is not in use.
- A key management system (KMS) stores encryption keys. Using a KMS is essential because a business may have to manage thousands or even millions of keys and may have to decrypt historic data that was encrypted using an obsolete encryption key.
- Privacy is a social concept that relates to how people feel about the release of their personal information to others. Different countries and cultures have different ideas on what information should and should not be private.
- Data protection laws have been made in many countries to protect individual privacy. They require companies who manage user data to store it securely, to ensure that it is not used or sold without the permission of users, and to allow users to view and correct personal data held by the system.