17 Security

In this chapter, you will learn about

- how encryption works, including the use of public keys and private keys, plaintext and ciphertext, symmetric key cryptography and asymmetric key cryptography
- how keys can be used to send verified messages
- how data is encrypted using symmetric and asymmetric cryptography
- quantum cryptography and QKD
- Secure Sockets Layer (SSL) and Transport Layer Security (TLS)
- the use of SSL and TLS in a client/server communication
- examples of where SSL and TLS would be used
- how digital certificates are acquired
- how digital certificates are used to produce digital signatures.

WHAT YOU SHOULD ALREADY KNOW

In Chapter 6, you learnt about security. Try these four questions before you read this chapter.

1 Explain what is meant by the terms

a) data integrity

b) data privacy

c) data security.

2 Describe how it is possible to recover data after it has been lost accidentally or otherwise.

3 Describe **three** ways of protecting against data loss.

4 **a)** Explain the effect of these five security risks.

i) Hacking

ii) Malware

iii) Phishing

iv) Pharming

b) Explain how it is possible to guard against each of the five security risks named in part a).

17.1 Encryption

Key terms

Eavesdropper – a person who intercepts data being transmitted.

Plaintext – the original text/document/message before it is put through an encryption algorithm.

Ciphertext – the product when plaintext is put through an encryption algorithm.

Block cipher – the encryption of a number of contiguous bits in one go rather than one bit at a time.

Stream cipher – the encryption of bits in sequence as they arrive at the encryption algorithm.

Block chaining – form of encryption, in which the previous block of ciphertext is XORed with the block of plaintext and then encrypted thus preventing identical plaintext blocks producing identical ciphertext.

Symmetric encryption – encryption in which the same secret key is used to encrypt and decrypt messages.

Key distribution problem – security issue inherent in symmetric encryption arising from the fact that, when sending the secret key to a recipient, there is the risk that the key can be intercepted by an eavesdropper/hacker.

Asymmetric encryption – encryption that uses public keys (known to everyone) and private keys (secret keys).

Public key – encryption/decryption key known to all users.

Private key – encryption/decryption key which is known only to a single user/computer.

17.1.1 Encryption keys, plaintext and ciphertext

Why do we need encryption?

When data is transmitted over any public network (wired or wireless), there is a risk of it being intercepted by, for example, a hacker (sometimes referred to as an **eavesdropper**). Using encryption helps to minimise this risk.

Encryption alters data into a form that is unreadable by anybody for whom the data is not intended. It cannot prevent the data being intercepted, but it stops it from making any sense to the eavesdropper. This is particularly important if the data is sensitive (for example, medical or legal documents) or confidential (for example, credit card or bank details).

There are four main security concerns when data is transmitted: confidentiality, authenticity, integrity and non-repudiation.

1. **Confidentiality** is where only the intended recipient should be able to read or decipher the data.
2. **Authenticity** is the need to identify who sent the data and verify that the source is legitimate.
3. **Integrity** is that data should reach its destination without any changes.
4. **Non-repudiation** is that neither the sender nor the recipient should be able to deny that they were part of the data transmission which just took place.

Plaintext and ciphertext

The original data being sent is known as **plaintext**. Once it has gone through an encryption algorithm, it produces **ciphertext**. Figure 17.1 summarises what happens.

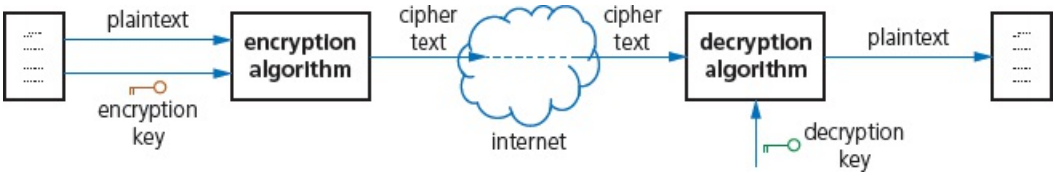

Figure 17.1

Note that, when encrypting text, **block cipher** is usually used. Here, the encryption algorithm is applied to a group of contiguous bits (for example, 128 bits) rather than one bit at a time (which is known as **stream cipher**). With block cipher, each plaintext block is XORed with the previous ciphertext block and then encrypted – this is known as **block chaining**. This prevents identical blocks of plaintext producing the same ciphertext each time they are encrypted.

Notice the use of encryption and decryption keys in Figure 17.1. These keys will be considered in the next section.

17.1.2 Symmetric encryption

Symmetric encryption uses a secret key; the same key is used to encrypt and decrypt the encoded message.

Consider a simple system which uses 10-denary-digit encryption (which gives about 10 billion possibilities). Suppose our secret key is 4 2 9 1 3 6 2 8 5 6, which means each letter in a word is shifted across the alphabet +4, +2, +9, and so on, places.

For example, here is the message, 'computer science is exciting' before and after the 10-denary-digit secret key is applied:

	C	O	M	P	U	T	E	R		S	C	I	E	N	C	E		I	S		E	X	C	I	T	I	N	G
Key	4	2	9	1	3	6	2	8		5	6	4	2	9	1	3		6	2		8	5	6	4	2	9	1	3
	G	Q	V	Q	X	Z	G	Z		X	I	M	G	W	D	H		O	U		M	C	I	M	V	R	O	J

Figure 17.2

However, modern computers could 'crack' this key (and, therefore, decrypt the message) in a few seconds. To combat this, we use 256-bit encryption (in other words, a 256-bit key) which gives 2^{256} possible combinations. Even this may not be enough, as computers become more powerful.

One issue with symmetric encryption is that both sender and recipient need to use the same secret key. This is a security risk here, since the sender has to supply the key to the recipient. This key could be intercepted. This is referred to as the **key distribution problem**.

So, how can both sender and receiver have the required secret key without sending it electronically? The following routine shows one possibility.

stage	sender	recipient
1	uses an encryption algorithm and chooses a secret value, such as X = 2	uses the same algorithm and also chooses a secret value, such as Y = 4
2	this value of X is put into a simple algorithm:	the value of Y is put into the same algorithm:
	Note: MOD gives the remainder when dividing a number by 11	
	7^X (MOD 11) = 7^2 (MOD 11) = 49 (MOD 11) = 4 remainder 5 So, the value is 5	7^Y (MOD 11) = 7^4 (MOD 11) = 2401 (MOD 11) = 218 remainder 3 So, the value is 3
3	the sender sends the value just calculated (5) to the recipient	the recipient sends the value just calculated (3) to the sender
4	the new value from the recipient replaces 7 in the original algorithm:	the new value from the sender replaces 7 in the original algorithm:

	3^X (MOD 11) = 3^2 (MOD 11) = 0 remainder **9** So, the new value is **9**	5^Y (MOD 11) = 5^4 (MOD 11) = 625 (MOD 11) = 56 remainder **9** So, the new value is **9**

Table 17.1

Both sender and recipient end up with the same encryption and decryption key of **9**. This is oversimplified; in practice, computers would generate much larger keys (possibly 256 bits – equivalent to 64 denary digits if using BCD).

There are many other ways to keep the encryption key secret. But the issue of security is always the main drawback of symmetrical encryption, since a single key is required for both sender and recipient.

EXTENSION ACTIVITY 17A

Using the following sender and receiver values to check that the method described above works.

a) Sender uses the value X = 3 and the receiver uses the value Y = 5

b) Sender uses the value X = 7 and the receiver uses the value Y = 6

17.1.3 Asymmetric encryption

Asymmetric encryption uses two keys – a **public key**, available to all users, and a **private key**, known to a specific person or computer.

Suppose Tom and Meera work for the same company. Tom wishes to send a confidential document to Meera. Here's how he could do it.

Step 1: Tom and Meera both use an algorithm to generate their own matching pairs of keys (private and public) which they keep stored on their computers. The matching pairs of keys are mathematically linked but cannot be derived from each other.

Step 2:

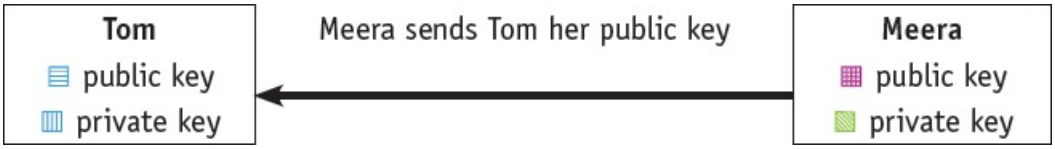

Figure 17.3

Step 3: Tom now uses Meera's public key (▦) to encrypt the document he wishes to send to her. He then sends his encrypted document (ciphertext) to Meera.

Step 4: Meera uses her matching private key (▧) to unlock Tom's document and decrypt it. This works because the public key used to encrypt the document and the private key used to decrypt it are a matching pair generated on Meera's computer.

Meera can exchange her public key with any number of people working in the company, so she is able to receive encrypted messages (which have been encrypted using her public key) and she can then decrypt them using her matching private key:

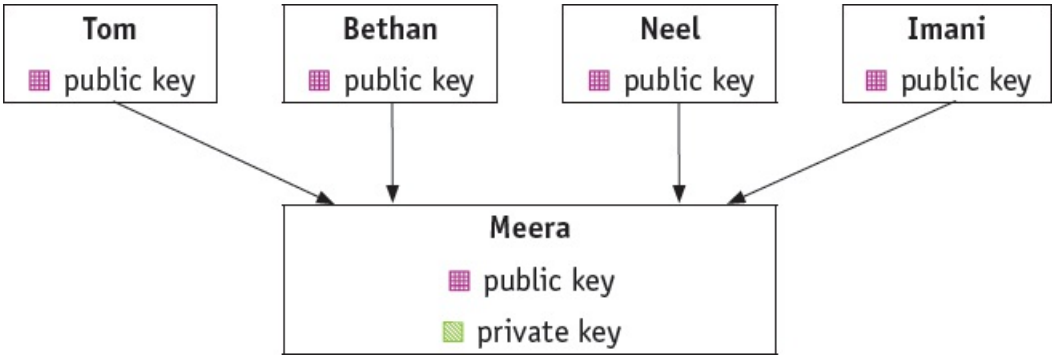

Figure 17.4

If a two-way communication is required between all five workers, then they all need to generate their own matching public and private keys. Once this is done, all users then need to swap public keys so that they can send encrypted documents, files or messages between each other. Each worker will then use their own private key to decrypt information being sent to them.

However, there are still issues. For example, how can Meera be certain that the document came from Tom, and that it has not been tampered with during transmission? Additional security is

required; this will be discussed in Section 17.4.

17.2 Quantum cryptography

Key terms

Quantum cryptography – cryptography based on the laws of quantum mechanics (the properties of photons).

Quantum key distribution (QKD) – protocol which uses quantum mechanics to securely send encryption keys over fibre optic networks.

Qubit – the basic unit of a quantum of information (**qu**antum **bit**).

Quantum cryptography utilises the physics of photons (light energy according to the formula E = hf) and their physical quantum properties to produce a virtually unbreakable encryption system. This helps protect the security of data being transmitted over fibre optic cables.

Photons oscillate in various directions and produce a sequence of random bits (0s and 1s) across the optical network.

Sending encryption keys across a network uses quantum cryptography – a **quantum key distribution (QKD)** protocol (one of the most common is BB84).

QKD uses quantum mechanics to facilitate the secure transmission of encryption keys. Quantum mechanics use a **qubit** (**qu**antum **bit**) as the basic unit of quantum data. Unlike normal binary (which uses discrete 0s and 1s), the state of a qubit can be 0 or 1, but it can also be **both** 0 and 1 simultaneously. Figure 17.5 shows a representation of a photon and how a photon can be affected by one of four types of polarising filter.

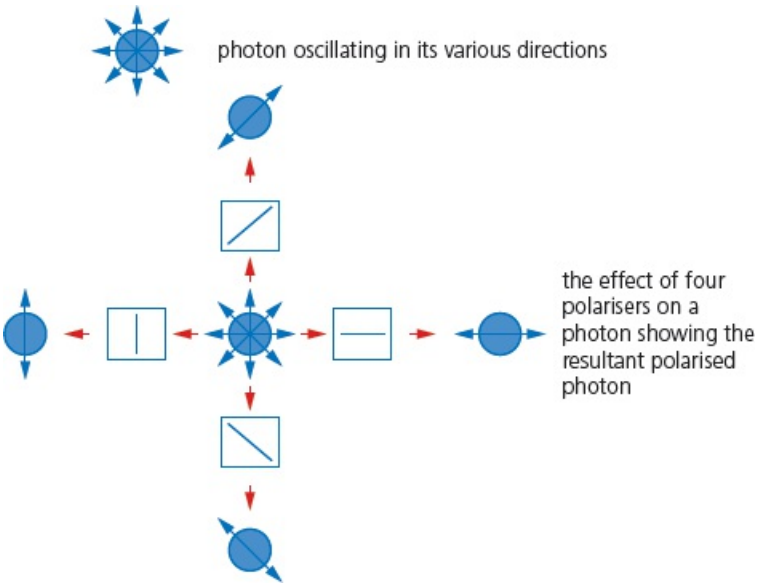

Figure 17.5

So, how do we use quantum cryptography to send an encryption key from ‘A’ to ‘B’ using the QKD protocol?

Stage 1: The sender uses a light source to generate photons.

Stage 2: The photons are sent through four random polarisers (see Figure 17.2) which randomly give one of four possible polarisations and bit values:

vertical polarisation	≡ 1 bit	
horizontal polarisation	≡ 0 bit	
45° right polarisation	≡ 1 bit	*diagonal polarisation shows 0 bit and 1 bit simultaneously*
45° left polarisation	≡ 0 bit	

Stage 3: The polarised photon travels along a fibre optic cable to its destination.

Stage 4: At the destination, there are two beam splitters:

diagonal splitter	⊠ 'X'
vertical/horizontal splitter	⊞ 'Y'

and two photon detectors.

Stage 5: One of the two beam splitters is chosen at random and the photon detectors are read.

Stage 6: The whole process is repeated until the whole of the encryption key has been transmitted from 'A' to 'B'.

Stage 7: The recipient sends back the sequence of beam splitters that were used (for example, XXXYYXXYYXXYYYY) to the sender.

Stage 8: The sender now compares this sequence to the polarisation sequence used at the sending station.

Stage 9: The sender now informs the recipient where in the sequence the correct beam splitters were used.

Stage 10: This now ensures that the sender and recipient are fully synchronised.

Stage 11: The encryption key can again be sent and received safely; even if intercepted, the eavesdropper would find it almost impossible to read the encryption key making the whole process extremely secure. Encrypted messages can now be sent along the fibre optic cable with the decryption key being used to decode all messages.

Despite the advantages of quantum cryptography, there are some potential drawbacks:

- It requires a dedicated line and specialist hardware, which can be expensive to implement initially.
- It still has a limited range (at the time of writing the limit is about 250 km).
- It is possible for the polarisation of the light to be altered (due to various conditions) while travelling down fibre optic cables.
- Due to the inherent security system generated by quantuin cryptography, terrorists and other criminals can use the technology to hide their activities from government law enforcers.

17.3 Protocols

Key terms

Secure Sockets Layer (SSL) – security protocol used when sending data over the internet.

Transport Layer Security (TLS) – a more up-to-date version of SSL.

Handshake – the process of initiating communication between two devices. This is initiated by one device sending a message to another device requesting the exchange of data.

Session caching – function in TLS that allows a previous computer session to be 'remembered', therefore preventing the need to establish a new link each time a new session is attempted.

Certificate authority (CA) – commercial organisation used to generate a digital certificate requested by website owners or individuals.

Public key infrastructure (PKI) – a set of protocols, standards and services that allow users to authenticate each other using digital certificates issued by a CA.

The two main protocols used to ensure security when using the internet are **Secure Sockets Layer (SSL)** and **Transport Layer Security (TLS)**; these are both part of the transport layer discussed in Chapter 14.

TLS is the more modern; it is based on SSL. The primary use of SSL and TLS is in the client/server application (see Chapter 2). They both use the standard cryptographic protocols to ensure there is a secure and authenticated communication between client and server. However, normally only the server is authenticated with the client remaining unauthenticated. Once a secure link between server and client is established, SSL or TLS protocols ensure no third party can eavesdrop.

17.3.1 Secure Sockets Layer (SSL)

When a user logs onto a website, SSL encrypts the data – only the client's computer and the web server are able to make sense of what is being transmitted. Two other functions of SSL are data compression (reducing the amount of data being transmitted), and data integrity checks. A user will know if SSL is being applied when they see the https protocol and/or the small green closed padlock.

The browser address display is different when the http or https protocol is used:

Figure 17.6

Similar banners will be seen when using TLS.

As mentioned in Chapter 14, TCP is used to establish a connection between the client and the server. A **handshake** takes place, thus enabling communication to begin between the client and server. One part of the SSL protocol is to agree which encryption algorithms are to be used; this is essential to ensure a secure, encrypted communication takes place. To be able to create an SSL connection, a web server requires an SSL digital certificate; the website owner needs to obtain this certificate to allow SSL protocols to be used (see Section 17.4).

Examples of where and when SSL (and TLS) would be used include

- online banking and all online financial transactions
- online shopping/commerce
- sending software to a restricted list of users
- sending and receiving emails
- using cloud storage facilities
- intranets and extranets (as well as the internet)
- using virtual private networks (VPNs)
- using Voice over Internet Protocols (VoIP) for video and/or audio chatting over the internet
- using instant messaging
- making use of a social networking site.

17.3.2 Transport Layer Security (TLS)

Transport Layer Security (TLS) is a modern, more secure version of SSL – it provides encryption, authentication and data integrity in a more effective way. It ensures the security and privacy of data between devices and users when communicating over a network (such as the internet). When a website and client communicate over the internet, TLS prevents third party eavesdropping.

TLS is formed of two main layers:

1. **Record protocol** can be used with or without encryption (it contains the data being transmitted over the network/internet).
2. **Handshake protocol** permits the web server and client to authenticate each other and to make use of encryption algorithms (a secure session between client and server is then established).

Only the most recent web browsers support both SSL and TLS, which is why the older, less secure, SSL is still used in many cases (although soon SSL will not be supported and users will have to adopt the newer TLS protocol if they wish to access the internet using a browser).

- It is possible to extend TLS by adding new authentication methods (unlike SSL).
- TLS can make use of **session caching** which improves the overall performance of the communication when compared to SSL (see below).
- TLS separates the handshaking process from the record protocol (layer) where all the data is held.

Session caching

When opening a TLS session, it requires considerable computer time (due mainly to complex cryptographic processes taking place). The use of session caching can avoid the need to utilise as much computer time for each connection. TLS can either establish a new session or attempt to resume an existing session; using the latter can considerably boost the system performance.

Summary

As already indicated, two of the main functions of SSL/TLS are

- the encryption of data
- the identification of client and server to ensure each knows who they are communicating with.

Stage 1: Once the client types in the URL into the browser and hits the <enter> key, several steps will occur before any actual encrypted data is sent; this is known as the handshaking stage.

Stage 2: The client's browser now requests secure pages (https) from the web server.

Stage 3: The web server sends back the SSL digital certificate (which also contains the public key) – the certificate is digitally signed by a third party called the **certificate authority (CA)** (see Section 17.4.2).

Stage 4: Once the client's browser receives the digital certificate, it checks

- the digital signature of the CA (is it one of those in the browser's trusted store – a list of trusted CAs is part of the browser which the client downloads to their computer)

- if the start and end dates shown on the certificate are still valid
- if the domain listed in the certificate is an exact match with the domain requested by the client in the first place.

Stage 5: Once the browser trusts the digital certificate, the public key (which forms part of the digital certificate) is used by the browser to generate a temporary session key with the web server; this session key is then sent back to the web server.

Stage 6: The web server uses its private key to decrypt the session key and then sends back an acknowledgement that is encrypted using the same session key.

Stage 7: The browser and web server can now encrypt all the data/traffic sent over the connection using this session key; a secure communication can now take place.

The **public key infrastructure (PKI)** is a set of protocols, standards and services that allow clients and servers to authenticate each other using digital certificates issued by the CA (for example, X509, PKI X.509); digital signatures also follow the same protocol. PKI requires the provider to use an encryption algorithm to generate public and private keys.

17.4 Digital signatures and digital certificates

Key terms

Digital signature – electronic way of validating the authenticity of digital documents (that is, making sure they have not been tampered with during transmission) and also proof that a document was sent by a known user.

Digest –a fixed-size numeric representation of the contents of a message produced from a hashing algorithm. This can be encrypted to form a digital signature.

Hashing algorithm (cryptography) – a function which converts a data string into a numeric string which is used in cryptography.

Digital certificate – an electronic document used to prove the identity of a website or individual. It contains a public key and information identifying the website owner or individual, issued by a CA.

17.4.1 Digital signatures

Digital signatures are a way of validating the authenticity of digital documents and identifying the sender (signing with a digital signature indicates that the original message, document or file is safe and has not been tampered with). As mentioned earlier on, there are four main purposes of digital signatures: authentication, non-repudiation, data integrity and confidentiality. A digital signature is a digital code which is often derived from the digital certificate (described below), although other methods of generating digital signatures will be described throughout this section.

The example used in Section 17.1 required Meera to send her public key to each of the workers, and she used her private key to decrypt their messages. However, the two keys can be reversed – the other workers can encrypt messages using their own private keys and then send these encrypted messages to other workers in the company, who use their matching public key to decrypt the messages. While this would be a bad idea if the messages were confidential, it could be used as a way of identifying or verifying who the sender of the message was (in other words, the private key would act like a digital signature, identifying the sender, since the private keys will be unique to the sender).

This also needs a lot of processing time to encrypt everything in the message. The following method, which is used to identify the sender and ensure the message was not tampered with, does not encrypt the messages but uses a generated numerical value known as a **digest**.

With this method, to actually identify the sender, it is not necessary to encrypt the whole message. The plaintext message is put through a **hashing algorithm** which produces the digest.

For example, if the first page of this chapter was going to be sent, we could put it through a hashing algorithm (such as MD4) and it would produce a digest, for example, it might produce the following digest:

873add9ed804fc5ce0338d2e9f7e0962

The sender's private key and digest are then put through an encryption algorithm to produce a digital signature.

Therefore, the plaintext and digital signature are sent to the recipient as two separate files. The recipient puts the digital signature through a decryption algorithm (using the sender's public key) to produce a digest. The recipient then puts the plaintext through the same hashing algorithm and also produces a digest.

If these two digests are the same, then the document has been sent correctly (and has not been tampered with). Since this process does not encrypt the document, if it needed to be kept confidential then it would be necessary to put the document through the asymmetric encryption process, as described earlier, before sending.

Note: a digest is a fixed-size numerical value which represents the content of a message. It is generated by putting the message through a hashing algorithm. The digest can be encrypted to produce a digital signature.

Figure 17.7 outlines the process.

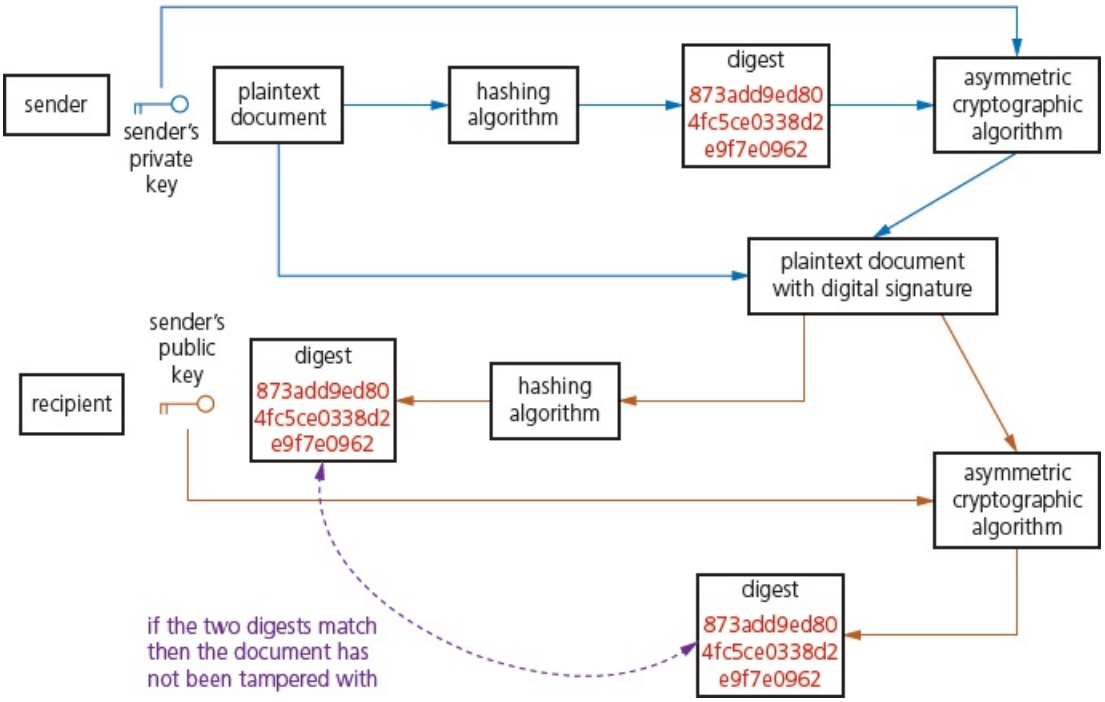

Figure 17.7

However, this method still is not safe enough, since the public key could be forged by a third party, which means the recipient still cannot be certain that the message came from a legitimate source. Therefore, an even more robust system is needed to give confidence that the sender is really who they claim to be.

17.4.2 Digital certificates

A **digital certificate** is an electronic 'document' used to prove the online identity of a website or an individual. The certificate contains a public key and other information identifying the owner of the certificate. A digital certificate is issued by the certificate authority (CA) – they independently validate the identity of the certificate owner.

This is a list of the items commonly found on a digital certificate

- version number
- serial number of certificate
- algorithm identification
- name of certificate issuer
- validity (start date and expiry date of certificate)
- company details
- public key
- issuer's identifier
- company's identifier
- signature algorithm used
- digital signature.

The digital signature is created by condensing all of the certificate details and then putting it through a hashing algorithm (such as MD4/5). The number generated is then put through an encryption algorithm, together with the CA's private key, thus producing a digital signature.

Figure 17.8 shows how a user can apply for a digital certificate. Figure 17.9 shows what a typical SSL digital certificate looks like.

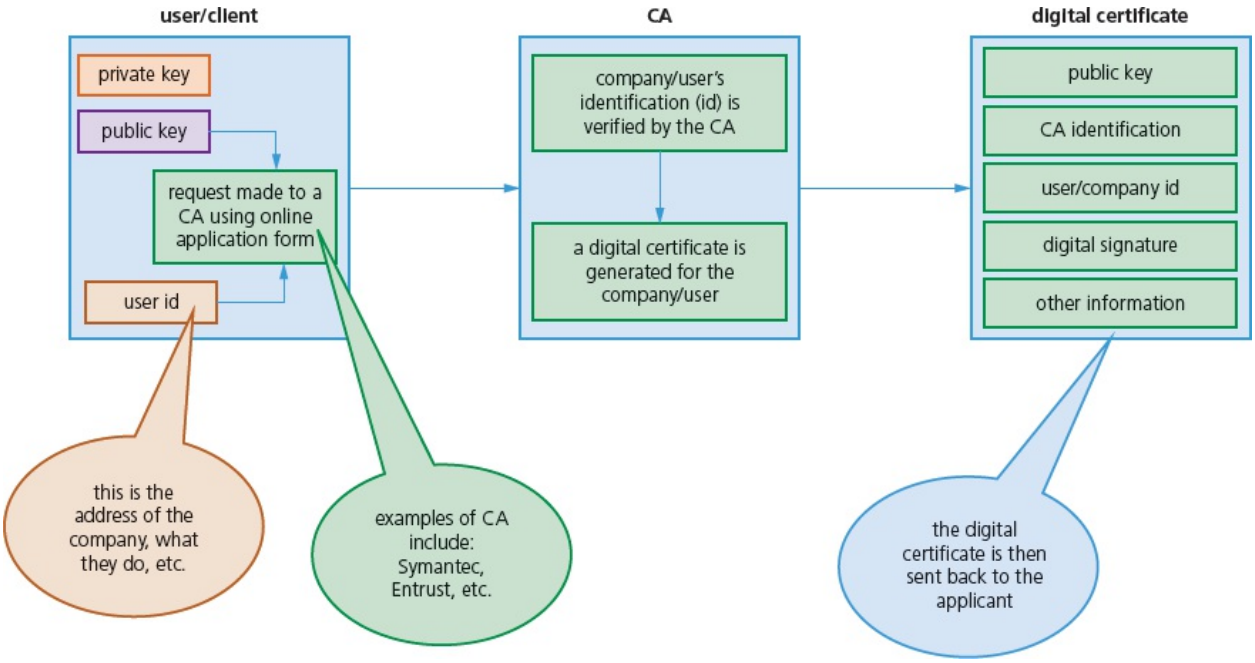

Figure 17.8

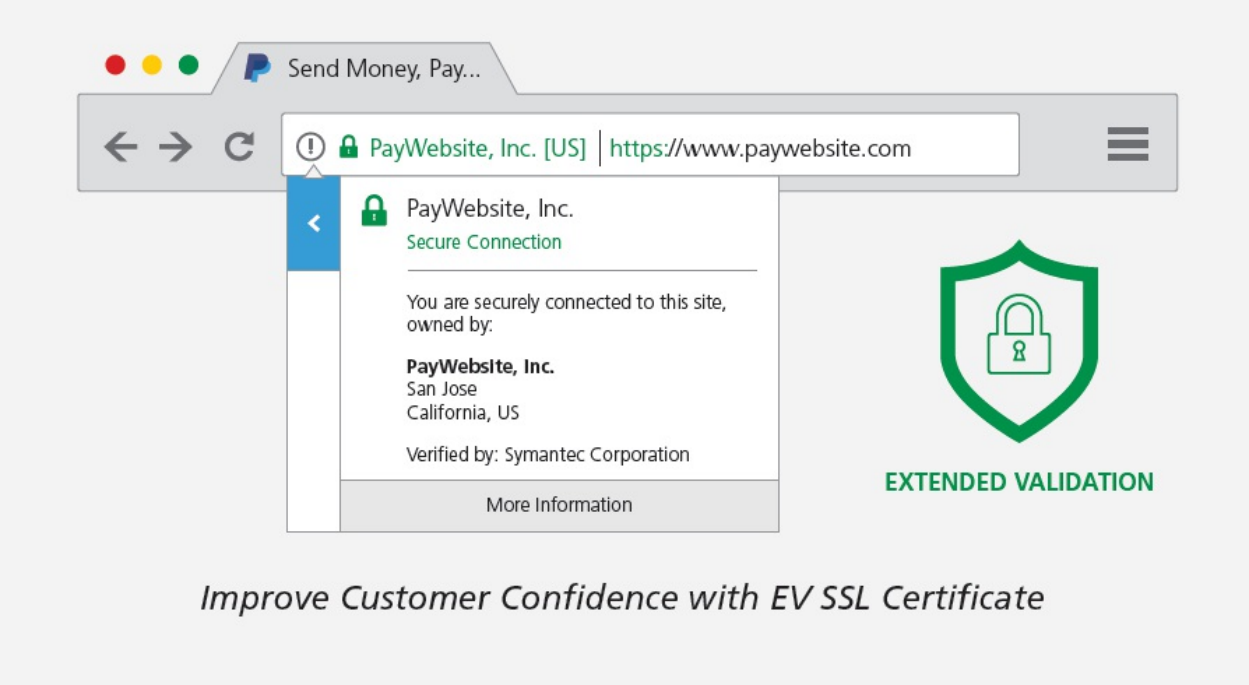

Figure 17.9

It is possible for a user to produce a self-signed digital certificate rather than use a commercial CA (for example, if an individual builds their own website, called my-site.com, and wants to make this generally available on the internet, they could produce their own digital certificate).

However, if a user attempts to log onto my-site.com they might see an error screen, like this:

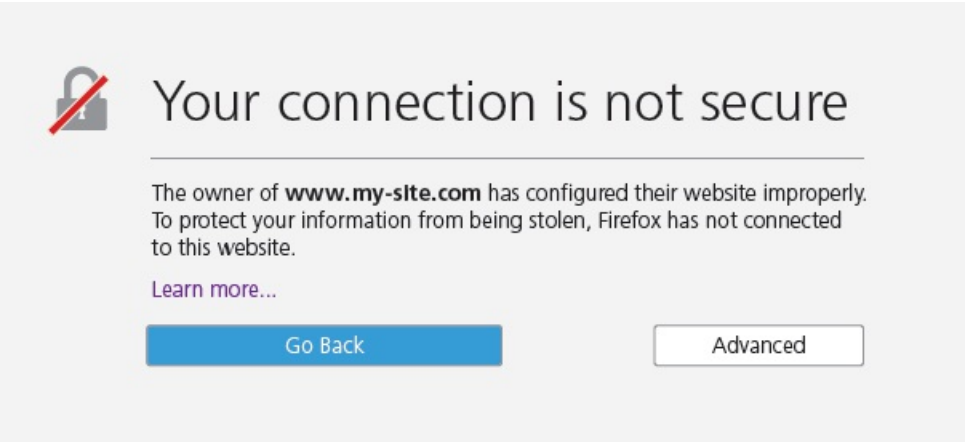

Figure 17.10

ACTIVITY 17A

For each of the following questions, choose the option which corresponds to the correct response.

1 What is meant by the term *cipher* when used in cryptography?

A an encryption or decryption algorithm
B an encrypted message
C a type of session key
D a digital signature
E text following an encryption algorithm

2 When carrying out asymmetric encryption, which of the following users would keep the private key?
A the sender
B the receiver
C both sender and receiver
D all recipients of the message
E none of the above

3 In cryptography, which of the following is the term used to describe the message before it is encrypted?
A simpletext
B plaintext
C notext
D ciphertext
E firsttext

4 Which of the following is the biggest disadvantage of using symmetric encryption?
A it is very complex and time consuming
B it is rarely used any more
C the value of the key reads the same in both directions
D it only works on computers with older operating systems
E there is a security problem when transmitting the secret key

5 Which of the following is the correct name for a form of encryption in which both the sender and the recipient use the same key to encrypt/decrypt?
A symmetric key encryption
B asymmetric key encryption
C public key encryption
D same key encryption
E block cipher encryption

6 Which of the following is involved in temporary key generation?
A session keys
B private key and certificate
C public key and certificate
D master keys
E public keys

7 Which of the following is a correct statement about PKIs?
A they use private and public keys but not digital certificates

B they use digital signatures and public keys

C they are a combination of digital certificates, public key cryptography and CAs

D they use asymmetric keys, hashing algorithms and certificate authorities

E they are a combination of digests, hashing algorithms and asymmetric cryptographic algorithms

8 SSL provides which of the following?

A message integrity only

B confidentiality only

C compression and authentication

D message integrity, confidentiality and compression

E authentication, encryption and digital signatures

9 Which of the following indicates a secure website?

A http and closed padlock

B http and open padlock

C https and closed padlock

D https and open padlock

E green closed padlock only

10 Which of the following is not part of security?

A non-repudiation

B bit streaming

C data integrity

D data privacy

E user authentication

End of chapter questions

1 a) Explain what is meant by QKD.

[2]

b) The following eleven statements refer to the transmission of an encryption key using quantum key distribution protocols.
Put each statement into its correct sequence, 1-11. The first one has been numbered for you.

[10]

sequence	statement
	the sender and receiver are now fully synchronised
	the photons are sent through four random polarisers which give one of four possible polarisations and bit values
	the process is repeated until the whole of the encryption key has been transmitted

1	the sender uses a light source to create the photons
	one of the two beam splitters is chosen at random and the photon detectors are read
	the sender now informs the recipient where, in the sequence, the correct beam splitters had been used
	the polarised photons travel along the fibre optic cable to the destination
	the encryption key can now be sent and received safely since eavesdroppers would find it impossible to crack the key code
	the sender now compares this sequence to the polarisation sequence used by the sending station
	at the destination, there are two beam splitters (diagonal and vertical/ horizontal) and two photon detectors
	the recipient sends back the sequence of beam splitters to the sender

2 **a)** Explain the terms *SSL* and *TLS*.

[3]

b) Explain the following terms used in TLS.

i) Record protocol

ii) Handshake protocol

iii) Session caching

[5]

c) Give **two** differences between SSL and TLS.

[2]

3 A user keys a URL into their browser and hits the <enter> key.
Re-order the following stages, 1-6, to show how an SSL digital certificate is used to set up a secure connection between client (user) and website.

[6]

order	**stage**
	browser and web server now encrypt all data/traffic sent over the connection using the session key and a secure communication can now take place
	client's browser requests secure pages (https) from the web server
	once trusted, the browser uses public key to agree temporary session key with web server; session key is sent back to web server
	the web server uses its private key to decrypt the session key and then sends back an acknowledgement that is encrypted using the session key
	once the client's browser gets the SSL digital certificate it checks the digital

	signature, validity of start and end dates and whether the domain listed in the certificate matches the domain requested by the user
	the web server sends back the SSL digital certificate containing the public key; this is digitally signed by a third party called the Certificate Authority (CA)

b) List **four** items found on a digital certificate.

[4]

c) Explain how a digital signature can be formed from a digital certificate.

[2]