OpenLearn

The Open University

Network security

The Open University

Network security

OpenLearn | Free learning from The Open University

About this free course

This free course provides a sample of level 3 study in Computing & IT
http://www.open.ac.uk/courses/find/computing-and-it

This version of the content may include video, images and interactive content that may not be optimised for your device.

You can experience this free course as it was originally designed on OpenLearn, the home of free learning from The Open University:
http://www.open.edu/openlearn/science-maths-technology/computing-and-ict/systems-computer/network-security/content-section-0 .

There you'll also be able to track your progress via your activity record, which you can use to demonstrate your learning.

The Open University

Walton Hall, Milton Keynes

MK7 6AA.

Intellectual property

Head of Intellectual Property, The Open University

The Open University

United Kingdom by The Charlesworth Group, Huddersfield

Contents

Introduction

Communication networks are used to transfer valuable and confidential information for a variety of purposes. As a consequence, they attract the attention of people who intend to steal or misuse information, or to disrupt or destroy the systems storing or communicating it. In this unit you will study some of the main issues involved in achieving a reasonable degree of resilience against network attacks. Some attacks are planned and specifically targeted, whereas others may be opportunistic, resulting from eavesdropping activities.

Threats to network security are continually changing as vulnerabilities in both established and newly introduced systems are discovered, and solutions to counter those threats are needed. Studying this unit should give you an insight into the more enduring principles of network security rather than detailed accounts of current solutions.

The aims of this unit are to describe some factors that affect the security of networks and data communications, and their implications for users; and to introduce some basic types of security service and their components, and indicate how these are applied in networks.

This OpenLearn course provides a sample of level 3 study in Computing & IT

Learning Outcomes

After studying this course, you should be able to:

- identify some of the factors driving the need for network security
- identify and classify particular examples of attacks
- define the terms vulnerability, threat and attack
- identify physical points of vulnerability in simple networks
- compare and contrast symmetric and asymmetric encryption systems and their vulnerability to attack, and explain the characteristics of hybrid systems.

1 Terminology and abbreviations

1.1 Terminology

Throughout this unit I shall use the terms ‘vulnerability’, ‘threat’ and ‘attack’. It is worthwhile clarifying these terms before proceeding:

- A **vulnerability** is a component that leaves a system open to exploitation (e.g. a network cable or a protocol weakness).
- A **threat** indicates the potential for a violation of security.
- The term **attack** is applied to an attempted violation.

When you have finished studying this unit you should be able to explain the meaning of all the terms listed below:

active attack

application layer encryption

application level gateway

asymmetric key system

attack

authentication

availability

bastion host

block cipher

brute force attack

Caesar cipher

certification authority

ciphertext

circuit level gateway

collision-free

confidentiality

cryptanalysis

cryptography

cryptosystem

decryption

demilitarised zone

denial-of-service attacks

digital signature

encryption

end-to-end encryption

filtering rules

firewall

freshness
hash value
integrity
key
keyspace
keystream
link layer encryption
masquerade attack
message authentication code
message digest
message modification
message replay
network layer encryption
nonce
one-time pad
one-way hash function
passive attack
password
password cracker
plaintext
private key
protocol analyser
proxy server
public key
public key infrastructure
public key system
registration authority
screened sub-net
sequence number
session key
shared key system
sniffer
stream cipher
symmetric key system
threat
time stamp
traffic analysis
Trojan
virus
vulnerability
worm

These terms will be highlighted in bold throughout the unit.

1.2 Abbreviations

The table below shows the abbreviations that are used throughout this unit, and their meanings.

Table 1 Abbreviations

ADSL	asymmetric digital subscriber line	PGP	Pretty Good Privacy
		PING	packet internet groper
DES	Data Encryption Standard	PSTN	public switched telephone network
DMZ	demilitarised zone		
DNS	domain name system	RC2	Rivest cipher 2
DSS	Digital Signature Standard	RC4	Rivest cipher 4
FTP	file transfer protocol	RSA	Rivest, Shamir and Adleman block cipher
IANA	Internet Assigned Numbers Authority	S-HTTP	secure hypertext transfer protocol
ICMP	internet control message protocol	S/MIME	secure/multipurpose internet mail extensions
IDEA	International Data Encryption Algorithm	SET	secure electronic transaction
IP	internet protocol	SHA	secure hash algorithm
IPSec	internet protocol security	SIM	subscriber identity module
ISDN	integrated services digital network	SMTP	simple mail transfer protocol
ISO	International Organization for Standardization	TCP	transmission control protocol
LAN	local area network	UDP	user datagram protocol
MD5	message digest 5	VPN	virtual private network
MSP	message security protocol	XOR	exclusive-OR
NSA	National Security Agency	3DES	Triple Data Encryption Standard
OSI	open systems interconnection		

2 Background to network security

2.1 Introduction

An effective security strategy will of necessity include highly technical features. However, security must begin with more mundane considerations which are often disregarded: for example, restricting physical access to buildings, rooms, computer workstations, and taking account of the 'messy' aspects of human behaviour, which may render any security measures ineffective. I shall remind you of these issues at appropriate points in the unit.

The need for security in communication networks is not new. In the late nineteenth century an American undertaker named Almon Strowger (Figure 1) discovered that he was losing business to his rivals because telephone operators, responsible for the manual connection of call requests, were unfairly diverting calls from the newly bereaved to his competitors. Strowger developed switching systems that led to the introduction of the first automated telephone exchanges in 1897. This enabled users to make their own connections using rotary dialling to signal the required destination.

Figure 1 Almon Strowger (Source: courtesy of http://www.privateline.com)

2.2 The importance of effective network security strategies

In more recent years, security needs have intensified. Data communications and e-commerce are reshaping business practices and introducing new threats to corporate activity. National defence is also vulnerable as national infrastructure systems, for example transport and energy distribution, could be the target of terrorists or, in times of war, enemy nation states.

On a less dramatic note, reasons why organisations need to devise effective network security strategies include the following:

- Security breaches can be very expensive in terms of business disruption and the financial losses that may result.

- Increasing volumes of sensitive information are transferred across the internet or intranets connected to it.
- Networks that make use of internet links are becoming more popular because they are cheaper than dedicated leased lines. This, however, involves different users sharing internet links to transport their data.
- Directors of business organisations are increasingly required to provide effective information security.

For an organisation to achieve the level of security that is appropriate and at a cost that is acceptable, it must carry out a detailed risk assessment to determine the nature and extent of existing and potential threats. Countermeasures to the perceived threats must balance the degree of security to be achieved with their acceptability to system users and the value of the data systems to be protected.

Activity 1

Think of an organisation you know and the sort of information it may hold for business purposes. What are the particular responsibilities involved in keeping that information confidential?

Answer

Any sizeable organisation has information that needs to be kept secure, even if it is limited to details of the employees and the payroll. I first thought of the National Health Service and the particular responsibility to ensure patients' medical records are kept secure. Academic institutions such as The Open University too must ensure that student-related information such as personal details and academic progress is kept confidential and cannot be altered by unauthorised people.

Box 1 : Standards and legislation

There are many standards relating to how security systems should be implemented, particularly in data communication networks, but it is impractical to identify them all here. A visit to the British Standards Institution website (http://www.bsigroup.com/) is a suitable point of reference.

ISO/IEC 17799 (2000) *Information Technology – Code of Practice for Information Security Management* sets out the management responsibility for developing an appropriate security policy and the regular auditing of systems. BS 7799–2 (2002) *Information Security Management Systems – Specification with Guidance for Use* gives a standard specification for building, operating, maintaining and improving an information security management system, and offers certification of organisations that conform. Directors of UK businesses should report their security strategy in annual reports to shareholders and the stock market; lack of a strategy or one that is ineffective is likely to reduce the business share value.

Organisations in the UK must conform to the Data Protection Act of 1998. This requires that information about people, whether it is stored in computer memory or in paper systems, is accurate and protected from misuse and also open to legitimate inspection.

2.3 Network security when using your computer

Before considering the more technical aspects of network security I shall recount what happens when I switch my computer on each morning at The Open University. I hope you will compare this with what happens when you use a computer, and relate it to the issues discussed in this unit.

After pressing the start button on my PC, certain elements of the operating system load before I am asked to enter a password. This was set by the IT administrators before I took delivery of my PC. The Microsoft Windows environment then starts to load and I am requested to enter another password to enable me to access the Open University's network. Occasionally I am told that the password will expire in a few days and that I shall need to replace it with another one. A further password is then requested because I have chosen to restrict access to the files on my machine, although this is optional. While I am waiting I see a message telling me that system policies are being loaded. These policies in my workplace are mainly concerned with providing standard configurations of services and software, but could be used to set appropriate access privileges and specify how I might use the services. Sometimes the anti-virus software begins an automatic update on my machine to counter new threats that have recently been identified.

I can now start my work on my computer, although if I decide to check my email account, or access some information on the Open University intranet, or perhaps seek to purchase a textbook from an online retailer, I may need to enter further user names, account details or passwords. This sequence of events is likely to be fairly typical of the requirements of many work environments and you will, no doubt, appreciate the profusion of password and account details that can result.

In this short narrative I have omitted an essential, yet easily forgotten, dimension of security that affects access to networks – the swipe card on the departmental entrance door and the lock on the door to my room. Although these may be considered mundane and unimportant, they are essential aspects of network security and a common oversight when the focus is on more sophisticated electronic security measures.

An important criterion, which is generally applicable, is that a system can be considered secure if the cost of illicitly obtaining data from it is greater than the intrinsic value of the data. This affects the level of security that should reasonably be adopted to protect, for instance, multi-million pound transfers between banks or a student's record at The Open University.

In this unit I shall introduce some of the fundamental concepts that underpin approaches to achieving network security, rather than provide you with the knowledge to procure and implement a secure network. The Communications-Electronics Security Group is the government's national technical authority for information assurance. If you need to investigate matters relating to procurement and implementation, you should refer to its website (https://www.gov.uk/government/organisations/cesg), from which you can find an introduction to the Information Assurance and Certification Service and also the Information Technology Security Evaluation and Certification Scheme. The latter scheme enables you to identify products that have undergone security evaluation.

In the next section I shall introduce the categories of attacks that can be launched against networks, before discussing appropriate countermeasures.

3 Threats to communication networks

3.1 How have network security measures developed over the past fifty years?

To start this section it is useful to reflect on the different obstacles that an intruder, intending to eavesdrop on a telephone conversation, might face today compared with fifty years ago, before electronic processing as we now know it. I shall consider an attack first between a home telephone and its local exchange (of the order of a mile or less), and then beyond the local exchange.

Eavesdropping on a telephone conversation has never been technically difficult. In particular, ‘tapping’ the wires of the target telephone in the local circuit would have been straightforward fifty years ago, provided that physical access could be gained to the wires. In many old films eavesdropping was carried out by an intruder in the basement of an apartment block or in a wiring cabinet in the street, using a basic set of equipment that included a pair of crocodile clips making a connection to some simple listening equipment. Today, in principle, a similar approach could still be successful over the last mile of the telephone distribution system. Much of the technology is still analogue, and signals can be detected by either direct contact with the twisted-pair wires or by sensing fields radiating from the transmissions. However, where ADSL (asymmetric digital subscriber line) or ISDN (integrated services digital network) services are provided, separating a telephone conversation from data traffic would need an ADSL modem or ISDN telephone and the knowledge to connect them correctly. This information is commonly available, so should not be a major obstacle in itself.

Beyond the local exchange, signals are combined (multiplexed) for carrying over transmission links, so to eavesdrop on a particular telephone message it must be ‘unpicked’ from other multiplexed messages. In the 1950s the multiplexing of analogue voice messages relied on the use of different frequency bands (frequency division multiplexing) within a link's available bandwidth, but today time division multiplexing is widely employed to accommodate a mix of digitised voice and data traffic. In digital networks, greater difficulty may be experienced in identifying or selecting individual channels. However, agencies with an interest in selecting potentially valuable information from a mass of traffic can identify key words that are spoken or written in data streams. Digital technology makes it much easier to search for and access data in a structured manner.

Another complication is the coding algorithms that are applied for a variety of purposes, but a determined intruder should not find it difficult to reverse these processes, given that many software tools are available from the internet. In fact, it is probably the wide availability of tools that can assist intrusion that makes modern networks susceptible, despite their use of increasingly sophisticated technology.

Activity 2

What fundamental security measures have been traditionally used in organisations such as banks or government departments, apart from those involving computer networks, and are they relevant to network security?

Answer

Banks have always needed secure areas such as vaults protected by security codes, locks and keys, and have been concerned with the authorisation and identification of staff empowered to carry out certain activities. The honesty of staff is an important issue and careful selection and screening procedures are needed. At the appointment stage references are usually requested and other checks made on potential employees, sometimes using 'positive vetting' procedures for sensitive appointments. In terms of day-to-day activities, a need-to-know policy might be followed to ensure that information is not needlessly disseminated within the organisation, and that sensitive paperwork such as drawings, reports and accounts is securely locked up to minimise risk.

Customers, too, could present security concerns. Banks need to assess security threats arising from customer interactions, and government departments involved in taxation and benefits will have similar concerns. The principles behind these issues have not diminished in importance in the electronic environment of today's business world, although many of the 'locks' and other countermeasures take a different form.

3.2 Important terminology and information for making the most of this section

Before we move on to consider specific issues of network security, I need to introduce some important terms that I shall use when describing how data is stored, processed or transmitted to other locations. These are:

- **Confidentiality**, in terms of selecting who or what is allowed access to data and systems. This is achieved through encryption and access control systems. Even knowledge of the existence of data, rather than the information that it contains, may be of significant value to an eavesdropper.
- The **integrity** of data, where modification is allowed only by authorised persons or organisations. The modifications could include any changes such as adding to, selectively deleting from, or even changing the status of a set of data.
- The **freshness** of data contained in messages. An attacker could capture part or all of a message and re-use it at a later date, passing it off as a new message. Some method of incorporating a freshness indicator (e.g. a time stamp) into messages minimises the risk of this happening.
- The **authentication** of the source of information, often in terms of the identity of a person as well as the physical address of an access point to the network such as a workstation.
- The **availability** of network services, including security procedures, to authorised people when they are needed.

In general, attacks on data networks can be classified as either passive or active as shown in Figure 2.

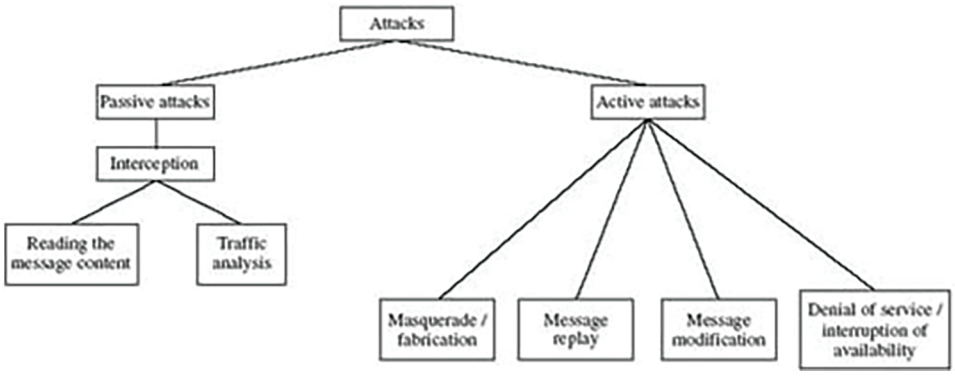

Figure 2 Forms of attack (Source: based on Stallings, 2001, p. 64)

This is a suitable point at which to listen to the audio track 'Digital dangers'. This provides some additional perspectives to supplement your study of this unit.

Audio content is not available in this format.
Hacking Audio

3.3 Passive attacks

A **passive attack** is characterised by the interception of messages without modification. There is no change to the network data or systems. The message itself may be read or its occurrence may simply be logged. Identifying the communicating parties and noting the duration and frequency of messages can be of significant value in itself. From this knowledge certain deductions or inferences may be drawn regarding the likely subject matter, the urgency or the implications of messages being sent. This type of activity is termed **traffic analysis**. Because there may be no evidence that an attack has taken place, prevention is a priority.

Traffic analysis, however, may be a legitimate management activity because of the need to collect data showing usage of services, for instance. Some interception of traffic may also be considered necessary by governments and law enforcement agencies interested in the surveillance of criminal, terrorist and other activities. These agencies may have privileged physical access to sites and computer systems.

Activity 3

Suppose that, in a passive attack, an eavesdropper determined the telephone numbers that you called, but not the message content, and also determined the websites that you visited on a particular day. Compare in relative terms the intelligence value of each approach. **Hint:** you will find some help here on the audio track 'Digital dangers'.

Answer

I suspect that an attacker could easily discover the identities of the parties you telephone, for example by simply telephoning the numbers you called. However, information about what was said in your calls may be more difficult to determine without an enquirer's interest becoming conspicuous. An investigation into websites that you visited, in contrast, may enable an attacker to build up a stronger picture of your interests and intentions based on the content of the pages, without the need to break cover.

3.4 Active attacks

An **active attack** is one in which an unauthorised change of the system is attempted. This could include, for example, the modification of transmitted or stored data, or the creation of new data streams. Figure 2 (see Section 3.2) shows four sub-categories here: masquerade or fabrication, message replay, message modification and denial of service or interruption of availability.

Masquerade attacks, as the name suggests, relate to an entity (usually a computer or a person) taking on a false identity in order to acquire or modify information, and in effect achieve an unwarranted privilege status. Masquerade attacks can also incorporate other categories.

Message replay involves the re-use of captured data at a later time than originally intended in order to repeat some action of benefit to the attacker: for example, the capture and replay of an instruction to transfer funds from a bank account into one under the control of an attacker. This could be foiled by confirmation of the freshness of a message.

Message modification could involve modifying a packet header address for the purpose of directing it to an unintended destination or modifying the user data.

Denial-of-service attacks prevent the normal use or management of communication services, and may take the form of either a targeted attack on a particular service or a broad, incapacitating attack. For example, a network may be flooded with messages that cause a degradation of service or possibly a complete collapse if a server shuts down under abnormal loading. Another example is rapid and repeated requests to a web server, which bar legitimate access to others. Denial-of-service attacks are frequently reported for internet-connected services.

Because complete prevention of active attacks is unrealistic, a strategy of detection followed by recovery is more appropriate.

Activity 4

What example of a replayed message could lead to a masquerade attack?

Answer

If an attacker identified and captured a data sequence that contained a password allowing access to a restricted service, then it might be possible to assume the identity of the legitimate user by replaying the password sequence.

In this unit I shall not deal with the detailed threats arising from computer viruses, but just give a brief explanation of some terms. The word 'virus' is used collectively to refer to Trojans and worms, as well as more specifically to mean a particular type of worm.

- A **Trojan** is a program that has hidden instructions enabling it to carry out a malicious act such as the capture of passwords. These could then be used in other forms of attack.
- A **worm** is a program that can replicate itself and create a level of demand for services that cannot be satisfied.
- The term **virus** is also used for a worm that replicates by attaching itself to other programs.

SAQ 1

How might you classify a computer virus attack according to the categories in Figure 2 (see Section 3.2)?

Answer

A virus attack is an active attack, but more details of the particular virus mechanism are needed for further categorisation. From the information on computer viruses, Trojans can lead to masquerade attacks in which captured passwords are put to use, and worms can result in loss of the availability of services, so denial of service is appropriate here. However, if you research further you should be able to find viruses that are implicated in all the forms of active attack identified in Figure 2.

SAQ 2

An attack may also take the form of a hoax. A hoax may consist of instructions or advice to delete an essential file under the pretence, for instance, of avoiding virus infection. How would you categorise this type of attack?

Answer

Denial of service will result if the instructions are followed and an essential file is removed.

Threats to network security are not static. They evolve as developments in operating systems, application software and communication protocols create new opportunities for attack.

During your study of this unit it would be a good idea to carry out a web search to find the most common forms of network attack. A suitable phrase containing key words for searching could be:

- most common network security vulnerabilities

Limit the search to reports within a year. Can you relate any of your findings to the general categories discussed above? What areas of vulnerability predominate? When I searched in early 2003, the most commonly reported network attacks were attributable to weaknesses in software systems (program bugs) and protocol vulnerabilities. Poor discipline in applying passwords rigorously and failure to implement other security provision were also cited. Another particular worry was the new opportunities for attack created by wireless access to fixed networks.

3.5 Potential network vulnerabilities

Whatever the form of attack, it is first necessary to gain some form of access to the target network or network component. In this section I shall take a brief look at a hypothetical network to see where an attacker may achieve this in the absence of appropriate defence measures.

Figure 3 shows the arrangement of a typical local area network (LAN), in which repeater hubs provide interconnections between workstations for a group of users (a work group), with different work groups being interconnected through backbone links and having access to the public switched telephone network (PSTN) and a packet-switched network. Repeater hubs are also used to refresh the transmitted signal to compensate for the attenuation caused by the transmission medium. I shall use Figure 3 to explore the potential for security breaches in a basic network.

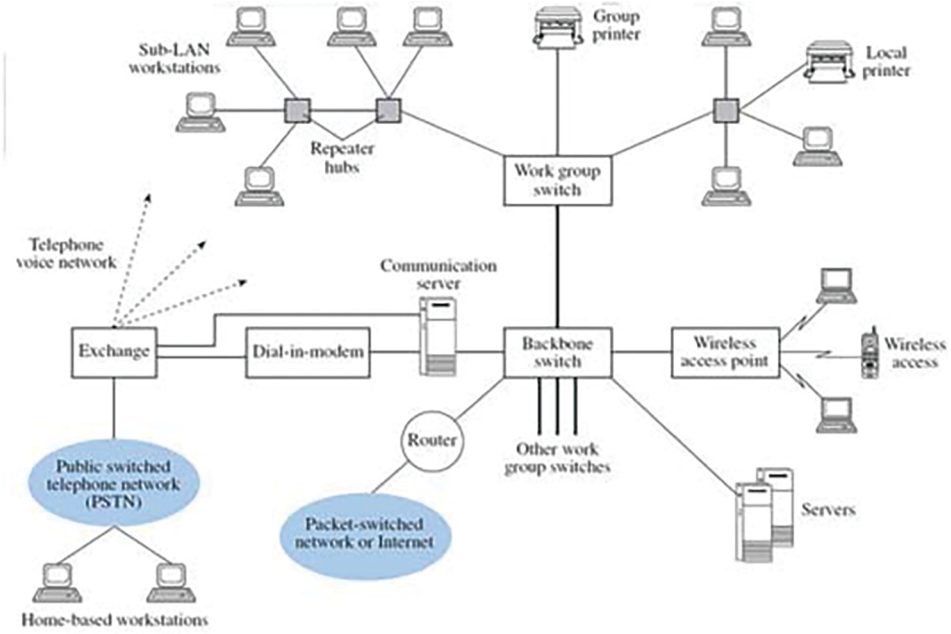

Figure 3 A typical LAN

Typically, the LAN will be Ethernet based, operating on the broadcast principle, whereby each packet (or strictly frame) is presented to all workstations on the local segment, but is normally accepted and read only by the intended receiving station. (Segment in this context refers to part of a local area network.) A broadcast environment is advantageous to someone attempting to carry out either a passive or an active attack locally. Switches that separate work groups on a LAN incorporate a bridge function, and bridges learn where to send packets by noting the source addresses of packets they receive and

recording this information in a forwarding table. The contents of the forwarding table then determine on which port on the switch packets will be sent. An attacker could corrupt entries in forwarding tables by modifying the source address information of incoming packets presented to a bridge. Future packets could then be forwarded to inappropriate parts of the network as a result.

SAQ 3

How would you categorise this form of attack?

Answer

Referring to Figure 2 (see Section 3.2), the modification of the source address corresponds to a message modification attack. This could then lead to a masquerade/fabrication attack if the purpose was for the attacker to receive messages intended for another computer. Alternatively, the misdirection of messages could result in a denial-of-service attack.

Routers also have forwarding tables that may be vulnerable to attack. However, they are configured in a different way from switches and bridges, depending on the security policies applied to the network. In some networks human network managers load forwarding tables that may remain relatively unchanged over some time. In larger networks, routing algorithms may update forwarding tables automatically to reflect current network conditions. The policy may be to route packets along a least-cost path, where the criterion for least cost may be, for example, path length, likely congestion levels or error rate. As routers share information they hold with neighbouring routers, any breach of one router's tables could affect several other routers. Any interference could compromise the delivery of packets to their intended destination and cause abnormal loading in networks, leading to denial of service. Routing tables or policies can be changed legitimately by, for example, network management systems, and this highlights the need for the management systems themselves to be protected against manipulation by intruders.

Returning to Figure 3, some organisations need to provide dial-in facilities for employees located away from the workplace. Dial-in access may present further opportunities for attack, and measures to protect this vulnerability will need to be considered. A wireless access point is also indicated, providing access for several wireless workstations to the wired LAN. Reports of security breaches on wireless LANs are widespread at the time of writing (2003), giving rise to the term ‘drive-by hacking’, although many instances have occurred quite simply because basic security features have not been activated. As a result, all the attacks on the LAN that I have described so far can be performed by someone who does not even have physical access to the site, but could be on the street nearby, for example in a parked car.

Some places on a LAN are particularly vulnerable. For example, connections to all internal data and voice communication services would be brought together in a wiring cabinet for patching across the various cable runs and for connecting to one or more external networks. An intruder would find this a convenient point for tapping into selected circuits or installing eavesdropping equipment, so mechanical locks are essential here.

3.6 Tapping into transmission media

Venturing beyond the organisation's premises in Figure 3 (see Section 3.5), there are many opportunities for interception as data passes through external links. These could be cable or line-of-sight links such as microwave or satellite. The relative ease or difficulty of achieving a connection to external transmission links is worth considering at this point.

Satellite, microwave and wireless transmissions can provide opportunities for passive attack, without much danger of an intruder being detected, because the environment at the point of intrusion is virtually unaffected by the eavesdropping activity. Satellite transmissions to earth generally have a wide geographic spread with considerable over-spill of the intended reception area. Although microwave links use a fairly focused beam of radiated energy, with appropriate technical know-how and some specialist equipment it is relatively straightforward physically to access the radiated signals.

I have already mentioned vulnerabilities arising from wireless LANs. In general, detecting and monitoring unencrypted wireless transmissions is easy. You may have noticed that when you switch a mobile telephone handset on, an initialisation process starts, during which your handset is authenticated and your location registered. The initial sequence of messages may be picked up by other circuits such as a nearby fixed telephone handset or a public address system, and is often heard as an audible signal. This indicates how easy it is to couple a wireless signal into another circuit. Sensing a communication signal may be relatively straightforward, but separating out a particular message exchange from a multiplex of many signals will be more difficult, especially when, as in mobile technology, frequency hopping techniques are employed to spread the spectrum of messages and so avoid some common transmission problems. However, to a determined attacker with the requisite knowledge, access to equipment and software tools, this is all possible.

Tapping into messages transmitted along cables without detection depends on the cable type and connection method. It is relatively straightforward to eavesdrop on transmitted data by positioning coupling sensors close to or in direct contact with metallic wires such as twisted pairs. More care would be needed with coaxial cables owing to their construction. Physical intrusion into physical media such as metallic wires may cause impedance changes, which in principle can be detected by time domain reflectometry. This technique is used to locate faults in communication media and is commonly applied to metallic cables or optical fibres for maintenance purposes. In practice, however, the levels of disturbance may be too slight to be measurable. The principle can also be applied to optical fibres.

Activity 5

Can you think of any difficulties in the interception of signals at a point along an optical fibre?

Answer

Optical fibres rely on a process of total internal reflection of the 'light' that represents the data stream. This means that no residual electrical signal is available under normal circumstances, but coupling into a fibre can be achieved for legitimate purpose by bending the fibre so that the angle of 'rays' inside it no longer conforms to the conditions for total internal reflection. A portion of the fibre protective cladding would need to be removed to allow access to the data stream. This would be a delicate operation for an attacker to perform and without suitable equipment the likely outcome would be a fractured fibre.

So far I have discussed the possibilities of gaining physical access to communication networks and hence the data that is carried on them. However, many users are interconnected through the internet or other internetworks, and these wider networks (particularly the internet) offer a broad range of opportunities without the need for intruders to move away from their desks. Many software tools have been developed for sound, legitimate purposes. For example, *protocol analysers* (or *sniffers)* analyse network traffic and have valid use in network management activities. Network discovery utilities based on the PING (packet internet groper) and TRACEROUTE commands are widely included in many PC operating systems and allow IP (internet protocol) addresses to be probed and routes through networks to be confirmed. The very same tools, however, can be used equally effectively for attacks on networks. If much of the traffic on the large public networks can be intercepted by determined attackers, how is network security to be achieved? It is the role of encryption to hide or obfuscate the content of messages, and this is the subject of the next section.

4 Principles of encryption

4.1 An introduction to encryption and cryptography

Section 3 has introduced you to the main threats to network security. Before I begin to examine the countermeasures to these threats I want to introduce briefly one of the fundamental building blocks of all network security. This is **encryption** – a process that transforms information (the **plaintext**) into a seemingly unintelligible form (the **ciphertext**) using a mathematical algorithm and some secret information (the encryption **key**). The process of **decryption** undoes this transformation using a mathematical algorithm, in conjunction with some secret value (the decryption key) that reverses the effects of the encryption algorithm. An encryption algorithm and all its possible keys, plaintexts and ciphertexts is known as a **cryptosystem** or cryptographic system. Figure 4 illustrates the process.

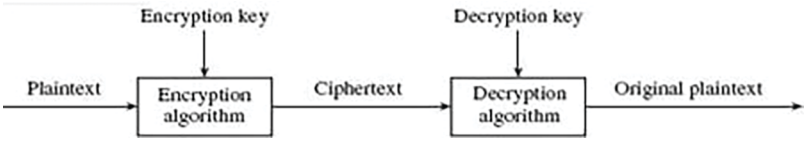

Figure 4 Encryption and decryption

Cryptography is the general name given to the art and science of keeping messages secret. It is not the purpose here to examine in detail any of the mathematical algorithms that are used in the cryptographic process, but instead to provide a general overview of the process and its uses.

Modern encryption systems use mathematical algorithms that are well known and have been exposed to public testing, relying for security on the keys used. For example, a well-known and very simple algorithm is the **Caesar cipher**, which encrypts each letter of the alphabet by shifting it forward three places. Thus A becomes D, B becomes E, C becomes F and so on. (A cipher that uses an alphabetic shift for any number of places is also commonly referred to as a Caesar cipher, although this isn't strictly correct since the Caesar cipher is technically one in which each character is replaced by one three places to the right.) I could describe this mathematically as $p + 3 = c$, where p is the plaintext and c the ciphertext. For a more general equation I could write $p + x = c$ where x could take any integer value up to 25. Selecting different values for x would obviously produce different values for c, although the basic algorithm of a forward shift is unchanged. Thus, in this example the value x is the key. (The Caesar cipher is of course too simple to be used for practical security systems.)

There are two main requirements for cryptography:

1. It should be computationally infeasible to derive the plaintext from the ciphertext without knowledge of the decryption key.
2. It should be computationally infeasible to derive the ciphertext from the plaintext without knowledge of the encryption key.

Both these conditions should be satisfied even when the encryption and decryption algorithms themselves are known.

The reason for the first condition is obvious, but probably not the second, so I shall briefly explain. In Section 3, the need to confirm authenticity was introduced. This is often also a requirement for information that is sent 'in the clear', that is, not encrypted. One method of authentication is for the sender and recipient to share a secret key. The sender uses the key to encrypt a copy of the message, or a portion of it, which is included with the data transfer and, on receipt, the recipient uses the key to decrypt the encrypted data. If the result matches the plaintext message, this provides a reasonable assurance that it was sent by the other key owner, and thus a check on its authenticity. (You will learn more about authentication in Section 8.) Of course, this assumes that the key has not been compromised in any way.

Modern encryption systems are derived from one of two basic systems: symmetric key (sometimes called shared key) systems, and asymmetric key (often called public key) systems.

4.2 An overview of symmetric key systems

We can think of **symmetric key systems** as sharing a single secret key between the two communicating entities – this key is used for both encryption and decryption. (In practice, the encryption and decryption keys are often different but it is relatively straightforward to calculate one key from the other.) It is common to refer to these two entities as Alice and Bob because this simplifies the descriptions of the transactions, but you should be aware that these entities are just as likely to be software applications or hardware devices as individuals.

Symmetric key systems rely on using some secure method whereby Alice and Bob can first agree on a secret key that is known only to them. When Alice wants to send a private message to some other entity, say Charlie, another secret key must first be shared. If Bob then wishes to communicate privately with Charlie himself, he and Charlie require a separate secret key to share. Figure 5 is a graphical representation of the keys Alice, Bob and Charlie would each need if they were to send private messages to each other. As you can see from this, for a group of three separate entities to send each other private messages, three separate shared keys are required.

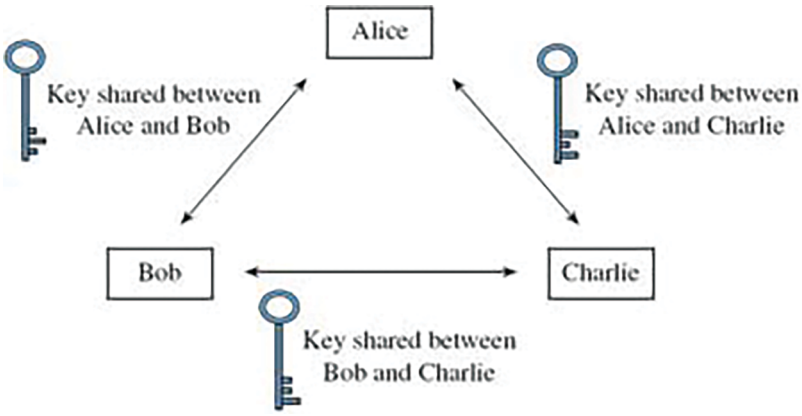

Figure 5 Keys needed by Alice, Bob and Charlie for privately communicating with each other

SAQ 4

Derive a formula for the number of shared keys needed in a system of n communicating entities.

Answer

Each entity in the network of n entities requires a separate key to use for communications with every other entity in the network, so the number of keys required by each entity is:

$n - 1$

But each entity shares a key with another entity, so the number of shared keys for each entity is:

$(n - 1)/2$

In a system of n communicating entities the number of shared keys required is:

$n(n - 1)/2$

SAQ 5

How many shared keys are required for a company of 50 employees who all need to communicate securely with each other? How many shared keys would be needed if the company doubles in size?

Answer

50 people would require (50 x 49)/2 = 1225 shared secret keys.

100 people would require (100 × 99)/2 = 4950 shared secret keys.

4.3 The components of a symmetric key system

I shall now explain the components of a symmetric key system in more detail.

A **block cipher** operates on groups of bits – typically groups of 64. If the final block of the plaintext message is shorter than 64 bits, it is padded with some regular pattern of 1s and 0s to make a complete block. Block ciphers encrypt each block independently, so the plaintext does not have to be processed in a sequential manner. This means that as well as allowing parallel processing for faster throughput, a block cipher also enables specific portions of the message (e.g. specific records in a database) to be extracted and manipulated. A block of plaintext will always encrypt to the same block of ciphertext provided that the same algorithm and key are used.

A **stream cipher** generally operates on one bit of plaintext at a time, although some stream ciphers operate on bytes. A component called a keystream generator generates a sequence of bits, usually known as a **keystream**. In the simplest form of stream cipher, a modulo-2 adder (exclusive-OR or XOR gate) combines each bit in the plaintext with each bit in the keystream to produce the ciphertext. At the receiving end, another modulo-2 adder combines the ciphertext with the keystream to recover the plaintext. This is illustrated in Figure 6. The encryption of a unit of plain text is dependent on its position in the data stream, so identical units of plaintext will not always encrypt to identical units of ciphertext when using the same algorithm and key.

Plaintext	1	1	0	1	0	0	0	1	1	0	1	0	1	0	0	1
Keystream	0	1	1	1	1	0	0	0	0	1	1	0	1	1	1	0
Ciphertext	1	0	1	0	1	0	0	1	1	1	0	0	0	1	1	1

(a) Encryption

Ciphertext	1	0	1	0	1	0	0	1	1	1	0	0	0	1	1	1
Keystream	0	1	1	1	1	0	0	0	0	1	1	0	1	1	1	0
Plaintext	1	1	0	1	0	0	0	1	1	0	1	0	1	0	0	1

(b) Decryption using an identical keystream

Figure 6 Encryption and decryption using a modulo-2 adder

Stream ciphers can be classified as either synchronous or self-synchronising. In a synchronous stream cipher, depicted in Figure 7, the keystream output is a function of a key, and is generated independently of the plaintext and the ciphertext. A single bit error in the ciphertext will result in only a single bit error in the decrypted plaintext – a useful property when the transmission error rate is high.

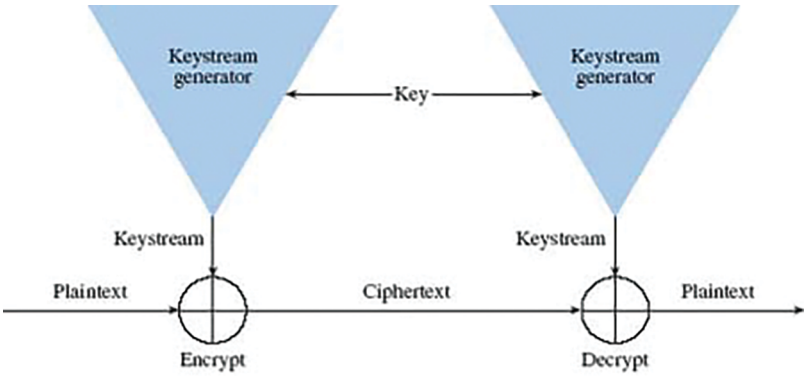

Figure 7 Synchronous stream cipher (Source: based on Schneier, 1996, Figure 9.6)

In a self-synchronising cipher, depicted in Figure 8, the keystream is a function of the key and several bits of the cipher output. Because the keystream outputs depend on the previous *n* bits of the plaintext or the ciphertext, the encryption and decryption keystream generators are automatically synchronised after *n* bits. However, a single bit error in the ciphertext results in an error burst with a length dependent on the number of cipher output bits used to compute the keystream.

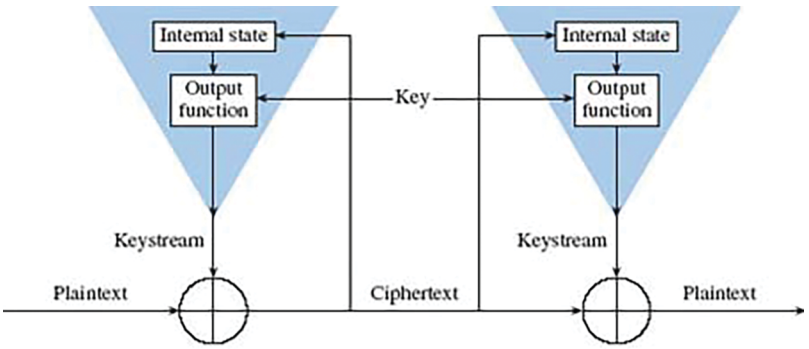

Figure 8 Self-synchronising stream cipher (Source: based on Schneier, 1996, Figure 9.8)

A selection of some symmetric key systems used in popular software products is given in Table 2.

Table 2 Examples of commercial symmetric key systems

Algorithm	Description
DES (Data Encryption Standard)	A block cipher with a 56-bit key. Adopted in 1977 by the US National Security Agency (NSA) as the US Federal standard, it has been one of the most widely used encryption algorithms but, as computers have become more powerful, it is now considered to have become too weak.
Triple-DES (or 3DES)	A variant of DES developed to increase its security. It has several forms; each operates on a block three times using the DES algorithm, thus effectively increasing the key length. Some variants can use three different keys, the same key three times, or use an encryption–decryption–encryption mode.
IDEA(International Data Encryption Algorithm)	A block cipher with a 128-bit key published in 1990. It encrypts data faster than DES and is considered to be a more secure algorithm.
Blowfish	A compact and simple block cipher with a variable-length key of up to 448 bits.
RC2 (Rivest cipher no. 2)	A block cipher with a variable-length key of up to 2048 bits. The details of the algorithm used have not been officially published.
RC4 (Rivest cipher no. 4)	A stream cipher with a variable-length key of up to 2048 bits.

Often the key length for RC2 and RC4 is limited to 40 bits because of the US export approval process. A shorter key reduces the strength of an encryption algorithm.

4.4 Asymmetric key systems

Asymmetric or **public key systems** are based on encryption techniques whereby data that has been encrypted by one key can be decrypted by a different, seemingly unrelated, key. One of the keys is known as the **public key** and the other is known as the **private key**. The keys are, in fact, related to each other mathematically but this relationship is complex, so that it is computationally infeasible to calculate one key from the other. Thus, anyone possessing only the public key is unable to derive the private key. They are able to encrypt messages that can be decrypted with the private key, but are unable to decrypt any messages already encrypted with the public key.

I shall not explain the mathematical techniques used in asymmetric key systems, as you do not need to understand the mathematics in order to appreciate the important features of such systems.

Each communicating entity will have its own key pair; the private key will be kept secret but the public key will be made freely available. For example, Bob, the owner of a key pair, could send a copy of his public key to everyone he knows, he could enter it into a public database, or he could respond to individual requests from entities wishing to communicate by sending his public key to them. But he would keep his private key secret. For Alice to send a private message to Bob, she first encrypts it using Bob's easily accessible public key. On receipt, Bob decrypts the ciphertext with his secret private key and recovers the original message. No one other than Bob can decrypt the ciphertext because only Bob has the private key and it is computationally infeasible to derive the private key from the public key. Thus, the message can be sent secretly from Alice to Bob without the need for the prior exchange of a secret key.

Using asymmetric key systems with *n* communicating entities, the number of key pairs required is *n*. Compare this with the number of shared keys required for symmetric key systems (see SAQs 4 and 5) where the number of keys is related to the square of the number of communicating entities. Asymmetric key systems are therefore more scalable.

Public key algorithms can allow either the public key or the private key to be used for encryption with the remaining key used for decryption. This allows these particular public key algorithms to be used for authentication, as you will see later.

Public key algorithms place higher demands on processing resources than symmetric key algorithms and so tend to be slower. Public key encryption is therefore often used just to exchange a temporary key for a symmetric encryption algorithm. This is discussed further in Section 4.6.

As with symmetric key systems, there are many public key algorithms available for use, although most of them are block ciphers. Two used in popular commercial software products are listed in Table 3.

Table 3 Examples of commercial asymmetric key systems

Algorithm	Description
RSA (named after its creators–Rivest, Shamir and Adleman)	A block cipher first published in 1978 and used for both encryption and authentication. Its security is based on the problem of factoring large integers, so any advances in the mathematical methods of achieving this will affect the algorithm's vulnerability.
DSS (Digital Signature Standard [1])	Developed by the US National Security Agency (NSA). Can be used only for digital signatures and not for encryption or key distribution.

Digital signatures are explained in Section 8.

SAQ 6

Construct a table to compare the features of symmetric and asymmetric key systems.

Answer

Symmetric key and asymmetric key systems are compared in Table 4.

Table 4

Symmetric key systems	Asymmetric key systems
The same key is used for encryption and decryption.	One key is used for encryption and a different but mathematically related key is used for decryption.
Relies on the sender and the receiver sharing a secret key.	Shared secret key exchange is not needed.
The key must be kept secret.	One key (the secret key) must be kept secret, but the other key (the public key) is published.
It should be computationally infeasible to derive the key or the plaintext given the algorithm and a sample of ciphertext.	It should be computationally infeasible to derive the decryption key given the algorithm, the encryption key and a sample of ciphertext.
Faster and computationally less demanding than public key encryption.	Slower and computationally more demanding than symmetric key encryption.

4.5 Vulnerability to attack

All the symmetric and public key algorithms listed in Table 2 and Table 3 share the fundamental property that their secrecy lies in the key and not in the algorithm. (This is generally known as Kerchoff's principle after the Dutchman who first proposed it in the nineteenth century.) This means that the security of any system using encryption should not be compromised by knowledge of the algorithm used. In fact, the use of a well-known and well-tested algorithm is preferred, since such methods have been subjected to intense scrutiny by practitioners in the field. If practitioners with detailed knowledge of an algorithm have not found messages encrypted with it vulnerable to attack and have been unable to break it, then it is safe to assume that others, without that knowledge, will also be unable to do so. However, the strength of a cryptographic algorithm is difficult if not impossible to prove, as it can only be shown that the algorithm has resisted specific known attacks. (An attack in this context is an attempt to discover the plaintext of an encrypted message without knowledge of the decryption key.) New and more sophisticated mathematical tools may emerge that substantially weaken algorithms previously considered to be immune from attack.

Cryptanalysis is the science of breaking a cipher without knowledge of the key (and often the algorithm) used. Its goal is either to recover the plaintext of the message or to deduce the decryption key so that other messages encrypted with the same key can be decrypted.

One of the more obvious attacks is to try every possible key (i.e. the finite set of possible keys, known as the **keyspace**) until the result yields some intelligible data. This kind of attack is known as a **brute force attack**. Clearly, the greater the keyspace, the greater the immunity to a brute force attack.

SAQ 7

Assuming you could process 10^{12} key attempts per second, calculate how long it would take to search the keyspace of a 56-bit key. Compare this with the time needed to search the keyspace of a 128-bit key.

Answer

A keyspace of 56 bits provides $2^{56} \approx 7.2 \times 10^{16}$ possible keys. At a rate of 10^{12} keys per second it would take approximately 7.2×10^{4} seconds or about 20 hours to try every key. A keyspace of 128 bits provides $2^{128} \approx 3.4 \times 10^{38}$ possible keys. This would take approximately 3.4×10^{26} seconds or about 10^{19} years. (*Note:* the lifetime to date of the universe is thought to be of the order of 10^{10} years.)

In practice it is unlikely that an attacker would need to try every possible key before finding the correct one. The correct key could be found to a 50 per cent probability by searching only half of the keyspace. Even allowing for this, the time taken to break a 128-bit key is still impossibly long.

From the answer to SAQ 7 you may conclude that all that is needed for true data security is to apply an encryption system with an appropriate length key. Unfortunately, key length is only one of the factors that determine the effectiveness of a cipher. Cryptanalysts have a variety of tools, which they select according to the amount of information they have about a cryptosystem. In each of the cases below, a knowledge of the encryption algorithm but not the key is assumed:

- **Ciphertext only.** The attacker has only a sample of ciphertext. The speed and success of such an attack increases as the size of the ciphertext sample increases, provided that each portion of the sample has been encrypted with the same algorithm and key.
- **Known plaintext.** The attacker has a sample of plaintext and a corresponding sample of ciphertext. The purpose of this attack is to deduce the encryption key so that it can be used to decrypt other portions of ciphertext encrypted with the same algorithm and key.
- **Chosen text.** The attacker usually has a sample of chosen plaintext and a corresponding sample of ciphertext. This attack is more effective than known plaintext attacks since the attacker can select particular blocks of plaintext that can yield more information about the key. The term may also refer to cases where the attacker has a stream of chosen ciphertext and a corresponding stream of plaintext.

Activity 6

From the list above how would you classify a brute force attack?

Answer

To mount a brute force attack, the attacker would need a sample of ciphertext and knowledge of the algorithm used, so this would be classified as a ciphertext-only attack.

A ciphertext-only attack is one of the most difficult to mount successfully (and therefore the easiest to defend against) because the attacker possesses such limited information. In some cases even the encryption algorithm is also unknown. However, the attacker may still be able to use statistical analysis to reveal patterns in the ciphertext, which can be

used to identify naturally occurring language patterns in the corresponding plaintext. This method relies on exploiting the relative frequencies of letters. In the English language, for example, E is the most frequently occurring letter with a probability of about 0.12. This is followed by the letter T (probability 0.06) then A, O, I, N, S and R. Common letter sequences in natural language (e.g. TH, HE, IN, ER and THE, ING, AND and HER) may also be detected in the corresponding ciphertext.

These letters and their ordering may differ slightly according to the type and length of the sampled text. All authors have their own style and vocabulary and this can lead to statistical differences, as can the subject matter and spelling, e.g. English or American.

The only truly secure encryption scheme is one known as a **one-time pad**, introduced in 1918 by Gilbert Vernam, an AT&T engineer. Vernam's cipher used for its key a truly random and non-repeating stream of bits, each bit being used only once in the encryption process. Each bit in the plaintext message is XORed with each bit of the keystream to produce the ciphertext. After encryption the key is destroyed. Because of the random properties of the keystream, the resulting ciphertext bears no statistical relationship with the plaintext and so is truly unbreakable. The disadvantage of such a scheme, however, is that it requires the key to be at least the same length as the message and each key can be used only once (hence the name one-time pad). Since both sender and recipient require a copy of the key and a fresh key is needed for each message, this presents somewhat of a problem for key management. Despite these practical difficulties, use of the one-time pad has proved effective for high-level government and military security applications.

4.6 Hybrid systems

As you have seen from earlier sections, a major advantage of asymmetric key systems over symmetric key systems is that no exchange of a secret key is required between communicating entities. However, in practice public key cryptography is rarely used for encrypting messages for the following reasons:

- Security: it is vulnerable to chosen plaintext attacks.
- Speed: encrypting data with public key algorithms generally takes about 1000 times longer than with symmetric key algorithms.

Instead, a combination of symmetric and asymmetric key systems is often used. This system is based on the use of a **session key** – a temporary key used only for a single transaction or for a limited number of transactions before being discarded. The following sequence between Alice and Bob demonstrates the use of a session key.

1. Alice chooses a secret symmetric key that will be used as a session key.
2. Alice uses the session key to encrypt her message to Bob.
3. Alice uses Bob's public key to encrypt the session key.
4. Alice sends the encrypted message and the encrypted session key to Bob.
5. On receipt, Bob decrypts the session key using his own private key.
6. Bob uses the session key to decrypt Alice's message.

Activity 7

Why might a session key be preferable to the use of a recipient's public key?

Answer

I can think of a couple of reasons:

1. The more often a key is used and the more ciphertext produced by that key, the more likely it is to come under attack. A session key can simply be discarded after use.
2. Encryption and decryption can be performed much faster using symmetric keys than asymmetric keys.

5 Implementing encryption in networks

5.1 Overview

Confidentiality between two communicating nodes is achieved by using an appropriate encryption scheme: data is encrypted at the sending node and decrypted at the receiving node. Encryption will also protect the traffic between the two nodes from eavesdropping to some extent. However, for encryption to be used effectively in networks, it is necessary to define what will be encrypted, where this takes place in the network, and the layers that are involved in a reference model.

Activity 8

What are the implications of applying encryption to whole protocol data units including the headers at any particular layer of a reference model?

Answer

The protocol data unit headers include addressing information; if this is obscured, it will prevent the effective routing of protocol data units to their destination. In a packet-switched environment each switch must be able to read the address information in the packet headers. Encrypting all the data including the headers of each packet at the sending node would render the switches at intermediate nodes unable to read the source or destination address without first decrypting the data.

The implementation of encryption in packet-switched networks must ensure that essential addressing information can be accessed by the relevant network devices such as switches, bridges and routers. Encryption is broadly termed **link layer encryption** or **end-to-end encryption** depending on whether it is applied and re-applied at each end of each link in a communication path, or whether it is applied over the whole path between end systems. It is useful to identify the various implementations of encryption with the appropriate OSI layer, as indicated in Figure 9.

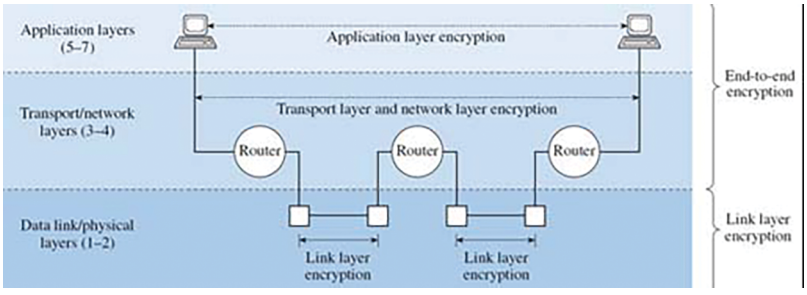

Figure 9 Encryption in relation to the protocol layers (Source: based on King and Newson, 1999, p. 104)

End-to-end encryption is implemented at or above layer 3, the network layer of the OSI reference model, while link layer encryption is applied at the data link and physical layers. When encryption is applied at the transport or network layers, end-to-end refers to hosts identified by IP (internet protocol) addresses and, in the case of TCP (transmission control

protocol) connections, port numbers. In the context of application layer encryption, however, end-to-end is more correctly interpreted as **process-to-process**. Figure 10 identifies the extent of encryption (unshaded areas) applied at each layer.

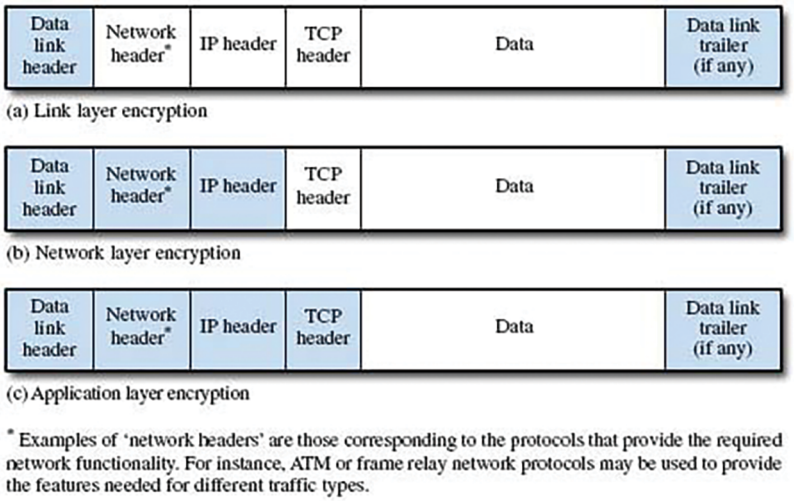

Figure 10 Alternative strategies for encryption (Source: based on Stallings, 1995, p. 139)

5.2 Link layer encryption

Link layer encryption has been available for some time and can be applied by bulk encryptors, which encrypt all the traffic on a given link. Packets are encrypted when they leave a node and decrypted when they enter a node. As Figure 10 (a) shows, data link layer headers are not encrypted. Because network layer information, in the form of layer headers, is embedded in the link data stream, link layer encryption is independent of network protocols. However, each link will typically use a separate key to encrypt all traffic. This makes the encryption devices specific to a given medium or interface type. In a large network, where many individual links may be used in a connection, traffic will need to be repeatedly encrypted and decrypted. One disadvantage is that while data is held at a node it will be in the clear (unencrypted) and vulnerable. Another is the need for a large number of keys along any path comprising many links. Hardware-based encryption devices are required to give high-speed performance and to ensure acceptable delays at data link layer interfaces. The effectiveness of link layer encryption depends on the relative security of nodes in the path, some of which may be within the internet. The question of who can access nodes in the internet then becomes a significant concern.

When applied to terrestrial networks, link layer encryption creates problems of delay and expense, but it is particularly useful in satellite links, because of their vulnerability to eavesdropping. In this case the satellite service provider takes responsibility for providing encryption between any two earth stations.

5.3 End-to-end encryption

I shall consider end-to-end encryption at the network layer and the application layer separately.

5.3.1 Network layer encryption

Network layer encryption is normally implemented between specific source and destination nodes as identified, for example, by IP addresses. As Figure 10 (b) indicates, the network layer headers remain unencrypted.

SAQ 8

What threats that you have previously encountered in this unit are still present with network layer encryption?

Answer

As information contained in IP packet headers is not concealed, eavesdroppers could perform traffic analysis based on IP addresses, and information in the headers could also be modified for malicious purposes.

Network layer encryption may be applied to sections of a network rather than end-to-end; in this case the network layer packets are encapsulated within IP packets. A major advantage of network layer encryption is that it need not normally be concerned with the details of the transmission medium.

A feature of encryption up to and including the network layer is that it is generally transparent to the user. This means that users may be unaware of security breaches, and a single breach could have implications for many users. This is not the case for application layer encryption. As with link layer encryption, delays associated with encryption and decryption processes need to be kept to an acceptable level, but hardware-based devices capable of carrying out these processes have become increasingly available.

An important set of standards that has been introduced to provide network layer encryption, as well as other security services such as authentication, integrity and access control in IP networks, is IPSec from the IP Security Working Group of the Internet Engineering Task Force. You should refer to RFC 2401 if you need further details on these standards.

5.3.2 Application layer encryption

In **application layer encryption**, end-to-end security is provided at a user level by encryption applications at client workstations and server hosts. Of necessity, encryption will be as close to the source, and decryption as close to the destination, as is possible. As Figure 10 (c) shows, in application layer encryption only the data is encrypted.

Examples of application layer encryption are S/MIME (secure/multipurpose internet mail extensions), S-HTTP (secure hypertext transfer protocol), PGP (Pretty Good Privacy) and MSP (message security protocol). Another example is SET (secure electronic transactions), which is used for bank card transactions over public networks. ‘Host layer encryption’ is a term sometimes used to refer to programs that perform encryption and decryption on behalf of the applications that access them. An example is secure socket layer.

5.4 Link layer encryption and end-to-end encryption compared and combined

Activity 9

Comparing end-to-end encryption with link layer encryption, which do you think is better?

Answer

It would be tempting to believe that end-to-end encryption is the more secure method since the user data is encrypted for the entire journey of the data packets. However, the addressing information is transmitted in the clear and this allows, at the least, traffic analysis to take place.

Much useful information can be gleaned by learning where messages come from and go to, when they occur, and for what duration and frequency, as described in Section 3.3.

In contrast, with a link layer encryption system the data is at risk in each node since that is where the unencrypted data is processed. Furthermore, link layer encryption is expensive because each node has to be equipped with the means to carry out encryption and decryption.

An effective way of securing a network is to combine end-to-end with link layer encryption. The user data portion of a packet is encrypted at the host using an end-to-end encryption key. The packet is then transported across the nodes using link layer encryption, allowing each node to read the header information but not the user data. The user data is secure for the entire journey and only the packet headers are in the clear during the time the packet is processed by any node.

SAQ 9

A network security manager in an organisation has overall responsibility for ensuring that networks are operated in a secure manner. From the manager's perspective, what level of encryption would be most suitable and why?

Answer

Link layer encryption may be viewed as disadvantageous because of the possible vulnerability of nodes outside the organisation. Application layer encryption can be implemented directly and individually by users of applications, but is not necessarily under the control of a network manager. A network layer approach, however, allows implementation of organisational security policies in terms of IP addressing for example, and is also transparent to users.

In considering the application of any encryption scheme, the cost in terms of network delay, increased overheads and finance must be weighed against the need for protection. As always, there is a need to balance the advantages of a more secure network against the disadvantages of implementing security measures and the potential costs of data interception and network attack.

6 Integrity

6.1 Encryption and integrity

You should recall from Section 3.2 that integrity relates to assurance that there has been no unauthorised modification of a message and that the version received is the same as the version sent.

Activity 10

Pause here for a while and consider whether encryption can be used as an effective assurance of the integrity of a message.

Answer

Encryption does provide some assurance about the integrity of a message. After all, if we are confident that the message has been immune from eavesdropping then, with the use of an appropriate encryption scheme, we might also be reasonably confident that it has not been altered in any way. You should recall, though, that in the discussion about block ciphers, I said that they allowed specific portions of a message to be extracted and manipulated. If an attacker knew which portions of the message to target, it would be possible to extract one portion and substitute another. Imagine, for example, a bank that uses a block cipher to encrypt information about certain transactions. One block may contain details of the account to be debited, another the account to be credited, and another the amount to be transferred. It might not be too difficult to substitute any of these blocks with data that had been extracted and recorded from some earlier transaction.

There are other reasons why encryption alone does not provide a completely workable solution. As you have already seen, the encryption process carries overheads in terms of resources and for some applications it is preferable to send data in the clear. Also some network management protocols separate the confidentiality and integrity functions, so encryption is not always appropriate.

6.2 Other ways of providing assurance of integrity

Some other method of providing assurance of the integrity of a message is therefore needed – some kind of concise identity of the original message that can be checked against the received message to reveal any possible discrepancies between the two. This is the purpose of a **message digest**. It consists of a small, fixed-length block of data, also known as a **hash value**, which is a function of the original message. (The term ‘hash value’ can be used in other contexts.) The hash value is dependent on all the original data (in other words, it will change even if only one bit of the data changes) and is calculated by applying a mathematical function, known as a hash function, which converts a variable-length string to a fixed-length string. A simple example is a function that XORs together each byte of an input string to produce a single output byte.

A common use of a hash value is the storage of passwords on a computer system. If the passwords are stored in the clear, anyone gaining unlawful access to the computer files could discover and use them. This can be avoided if a hash of the password is stored instead. When a user enters a password at log-in the hash value is recalculated and compared with the stored value. The security of this method relies on the hash function being computationally irreversible. In other words, it is easy to compute a hash value for a given input string, but extremely difficult to deduce the input string from the hash value. Hash functions with this characteristic are known as **one-way hash functions**.

For a hash value to give an effective assurance about the integrity of data, it should also be computationally infeasible to generate another message that hashes to the same value. Hash functions that provide this characteristic are said to be **collision-free**. The example of the XOR function given earlier is not collision-free, since it would be simple to generate messages that would produce an identical hash.

The following very simple method gives an insight into how a one-way hash could be derived. (This example is not a practical method of producing hash values but does serve to demonstrate their function.)

1. Concatenate the message by removing all the spaces.
2. Arrange the message in blocks of five characters.
3. Pad the final block if it contains less than five characters. (For example, if the final block has only two characters it could be padded by adding AAA.)
4. Assign each block a numerical code from one of 26^5 possible values according to the arrangement of letters. (See the example in the box below.)
5. Derive a value that is the modulo-26^5 sum of all the codes.

At the receiving end the hash value is recalculated using the same algorithm and is compared with the appended hash value received with the message. Any alterations in the original message should be revealed by a different hash value.

Box 3 : A method of block coding

This is a worked example of a method of block coding the text VALUE

1. Code each letter according to its position in the alphabet (A=0, B=1, etc.), giving the number sequence 21, 0, 11, 20, 4.
2. Multiply each coded number by a power of 26 depending on its position in the sequence, giving: 21×26^4, 0×26^3, 11×26^2, 20×26^1, 4×26^0
3. Add together the resulting numbers: 9 596 496 + 0 + 7436 + 520 + 4 = 9 604 456

In practice, of course, message digest algorithms in common use are very much more complex than the method described above. Two are briefly described in Table 5.

Table 5 Examples of common message digest algorithms

Algorithm	Description
MD5	Takes any arbitrary length input string and produces a fixed 128-bit value. This is done by a method of blocking and padding and then performing four rounds of processing based on a combination of logical functions. Considered to be reasonably secure although potential weaknesses have been reported.

SHA (secure hash algorithm)	Similar to MD5 but produces a 160-bit hash value so is more resistant to brute force attacks [1].

[1] A brute force attack on a hash value can be either an attempt to find another message that hashes to the same value or an attempt to find two messages that hash to the same value.

A **message authentication code** is similar to a one-way hash function and has the same properties, but the algorithm uses the additional ingredient of a secret key, and therefore possession of the key to perform the check is necessary.

7 Freshness

7.1 Introduction

A message replay attack was introduced briefly in Section 3.4. In this attack a message, or a portion of a message, is recorded and replayed at some later date. For example, an instruction to a bank to transfer a sum of money from account A to account B could be recorded and replayed some time later to fool the bank into making a second payment to account B. The incorporation of a freshness indicator in the message is a means of thwarting attacks of this kind. In this section I introduce three methods for indicating freshness: time stamps, sequence numbers and nonces.

7.2 Time stamps

A digital **time stamp** is analogous to a conventional postmark on an envelope: it provides some check of when a message was sent. Returning to the example of Alice and Bob, Alice could add the time and date to her communication to Bob. If she encrypts this with her own private key, or with a key that is known only to Alice and Bob, then Bob may feel reassured that Alice's message is not an old one that has been recorded and replayed.

Activity 11

Look back to Section 3.4, which introduced some types of active attack. If the encrypted message and the encrypted time stamp were sent together, could Bob be truly sure of the freshness of the message?

Answer

No. The exchange could be subject to a message replay attack. An eavesdropper could separate the encrypted message from the encrypted time stamp, and substitute a different message in place of the original one. (This could be a previously recorded encrypted message sent from Alice to Bob.)

To prevent this kind of message replay attack, the message and the time stamp need to be bound together in some way. One method of doing this is to encrypt them together. Only those in possession of the decryption key can then separate the two elements.

7.3 Sequence numbers

Sequence numbers are an alternative way of indicating freshness. If Alice is sending a stream of messages to Bob she can bind each one to a sequential serial number, and encryption will prevent an eavesdropper from altering any sequence number. If Bob is suspicious he can check that the numbers in Alice's messages are incremented sequentially. It would be a straightforward matter for him to spot a replayed message since the sequence order would be incorrect and its number would duplicate that of an earlier message.

SAQ 10

In a connectionless packet-switched network, would sequence numbers provide effective freshness indicators?

Answer

In a packet-switched network, messages between two points could take different routes and might arrive out of sequence. It would be impossible for Bob to determine whether this was a result of network delays or some malicious intent. However, the sequence numbers would still provide a means of identifying duplicated messages.

7.4 Nonces

This third method of freshness indication uses an unpredictable value in a challenge–response sequence. The sequence of events is illustrated in Figure 11. Bob wants to communicate with Alice but she needs reassurance that his message is not an old one that is simply being replayed. She generates some random number, which she encrypts and sends to Bob. He then binds the decrypted version of the random number to his message to Alice. On receipt she checks that the returned number is indeed the one she recently issued and sent to Bob. This number, which is used only once by Alice, is called a **nonce** (derived from 'number used once'). The term 'nonce' is also often used in a wider sense to indicate any freshness indicator.

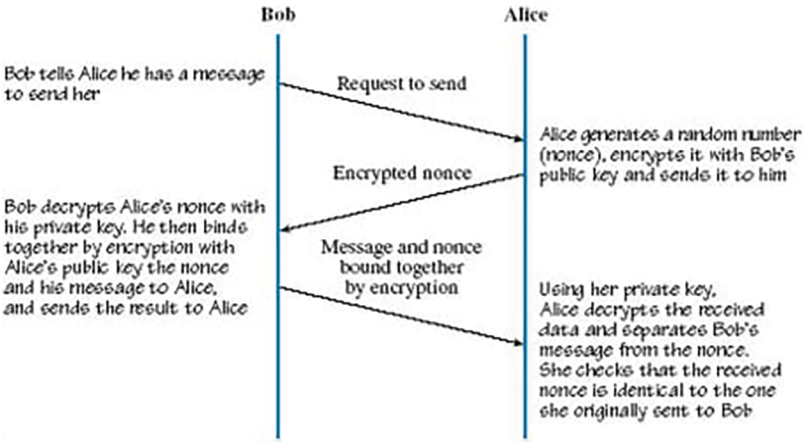

Figure 11 Using a nonce as a freshness indicator

8 Authentication

8.1 Overview of authentication methods

Authentication is needed to provide some assurance about the source of a message: did it originate from the location it appears to have originated from? One of the simplest authentication methods is the use of a shared secret such as a password. Assume that Alice and Bob share a password. Alice may challenge Bob to provide the shared password and if he does so correctly and Alice is confident that the password has not been compromised in any way, then she may be reassured that she is indeed communicating with Bob. (The use of passwords is examined in more detail in Section 9.2.)

Using the following steps, public key encryption can be used to provide an alternative challenge–response protocol between communicating entities who do not share a secret key:

1. Alice challenges Bob by sending him some random number.
2. Bob encrypts the random number using his own private key and sends the result to Alice.
3. Alice decrypts the message using Bob's public key. If the result matches her original random value and if she has confidence that the public key does indeed belong to Bob, then she may be assured that it is Bob who has sent the message to her.

In effect, when a message is encrypted with a private key, the key acts like the signature of the owner. As long as the key has not been compromised in any way it will act as an assurance of the authenticity of the message. However, Bob would be ill-advised to sign a document unless he was very sure about its contents. What if the value sent by Alice was not, after all, some random number but instead was an encrypted message giving instructions to Bob's bank to transfer funds into Alice's account? A better way for Bob to provide authentication when sending messages to Alice would be for him to create a digest of his message (message digests were discussed in Section 6.2) encrypted with his private key and to append this to the message he sends to Alice. On receipt Alice could create a new digest using an identical algorithm and compare this with the decrypted digest sent by Bob. If the two match and she is confident that Bob's private key has not been compromised in any way she may feel reasonably confident that the message did originate with Bob. Such an encrypted message digest is known as a **digital signature**.

8.2 Certification authorities and digital certificates

There are snags to this procedure, however: for example, Charlie could generate a key pair for himself and publish the public key using Bob's name. Some additional assurance is required that irrevocably binds together the true identity of a person with a public key. This assurance can be provided by a trusted third party, known as a **certification authority**, which is able to vouch for Bob. Certification authorities can be independent organisations, system administrators, or companies (such as Verisign) that specialise in

validating the identity of an entity and issue a digital certificate that binds the identity with a public key. The certification authority knows only the public key of the entity and not the private key, which should of course be kept secret at all times. The entity may not be a person – it could also, for example, be a computer, a website, or a network resource such as a router. Once the digital certificate has been issued, the entity can append it to messages it sends in order to provide assurance about its identity.

Activity 12

Can you think of a problem that might arise with this arrangement?

Answer

An entity, say Charlie, could create his own digital certificate, which he claims has been issued by a certification authority and which allows him to masquerade as Bob. Alternatively, he could modify Bob's authentic certificate by substituting his own public key in place of his.

So a digital certificate itself needs some form of authentication to provide assurance that it is valid.

Activity 13

How could a certification authority provide assurance about the validity of a digital certificate?

Answer

The certification authority could include its own identity and digital signature in the digital certificate.

Typically, a digital certificate includes the information illustrated in Figure 12. It may also include the level of trust that the certification authority is prepared to recommend. The emerging standard for digital certificates is ITU-T X.509.

Digital certificate
Subject's identity (e.g. name, address, organisation)
Subject's public key
Serial number of certificate
Validity dates (e.g. issue date, expiry date)
Certification authority's identity
Certification authority's digital signature

Figure 12 Format of a typical digital certificate

A user will need to obtain the certification authority's public key in order to validate its signature. In turn, the binding of the certification authority's identity to a public key will

itself need to be the subject of validity assurances, and thus the system of authentication depends on an extended structure and often relies on a chain of certificates.

Certification authorities form part of what is known as a **public key infrastructure** – a combination of services and encryption techniques that together are used to protect the security of data over networks. At the time of writing, the definition of a public key infrastructure is rather loose, but it is generally accepted that it will include:

- a **registration authority**, which checks and verifies the credentials of a user before a digital certificate can be issued
- a certification authority that issues and verifies digital certificates
- directory services for the publishing of public keys and certificates
- certificate management and key management services.

9 Access control

9.1 Introduction

In this section I shall discuss two major approaches used to restrict access to networks – passwords and firewalls.

9.2 Passwords

I have introduced encryption keys in previous sections. A **password** can also be thought of as a type of key in as much as it enables the keyholder to gain access to a particular resource. In Section 2.3, I described the process of starting up my computer at The Open University. I referred to the need to enter several sets of user identities and passwords to access various services or software using my computer. Given the frequent use of passwords, it is reasonable to consider what constitutes an effective password.

A major issue here is human behaviour. It is tempting, for instance, to make a record of passwords that are used but not always remembered, or to make all one's passwords identical, or to make them short or highly memorable in some personal way by linking them to personal information, which is unlikely to be highly secure. Alternatively, names, places or normal words may be used as passwords. There are security concerns with all these strategies. For example, electronic dictionaries could be used to probe passwords that are based on known words in all languages. Where passwords are restricted to a small number of characters, brute force methods may quickly find the one correct combination out of many that may be possible.

An effective password, technically speaking, is one that can resist both dictionary and brute force attacks. (For the purposes of network security a dictionary is a compilation of combinations of characters that find use in any field of activity. It is not restricted to words commonly used for general human communication.) A dictionary attack seeks to identify any predictable structure within the string of characters included in the password: for example, a name, a word, or a sequence of numbers, such as in a date format. A brute force attack relies on the power of computers to cycle through combinations of characters on a trial-and-error basis in the absence of predictable structure, until a successful conclusion is reached. If a password contains any partial structure then the processing needed to discover it is reduced.

SAQ 11

Based on the above, how would you specify how a password should be constructed?

Answer

To avoid a dictionary attack, it is wise to ensure that strings of characters do not produce a recognisable dictionary word. Ensuring that each password is made up of a minimum number of characters reduces the likelihood of a successful attack over a given time period. Including special characters (non-alphanumeric), numbers and upper and lower case letters helps to increase the range of combinations that would need to be tested, and also helps to remove any recognisable structure.

The security of an encrypted password used to access a remote station over a network depends on the form of encryption used and whether it is applied over the whole path from sender to receiver. A variety of means can be used to collect or bypass password protection systems. For instance, **password crackers** are programs specifically designed to capture password sequences, and decrypt or disable them. I referred earlier to the use of protocol analysers, which may be used to 'sniff' traffic for password sequences. In addition, Trojans can be hidden in programs that an attacker expects the legitimate user to run, and will contain a hidden routine to bypass the user's system's password protection. Hence encryption does not prevent capture and there is a danger that message replay can lead to successful access even when passwords cannot be decrypted by an attacker.

Despite the problems associated with passwords, they remain a first line of defence to intruder access. There are several examples of internet sites offering a consolidation service for an individual's multiplicity of passwords. The idea is that a single encrypted password can be used to release the collection of passwords – a potential 'winner-takes-all' situation.

Activity 14

What alternatives to passwords could be used to allow or bar the use of facilities by a person (not necessarily restricted to data networks)?

Answer

Any characteristic that is unique to a person can in principle be used to allow or bar access. For example, voice, face, hand, finger and iris recognition are candidates for authenticating an individual seeking access to some facility and so can be considered a key, like passwords. Magnetic strip cards too are often used to allow access to facilities such as workplaces, libraries and photocopiers.

Activity 15

In the answer to the question above, what advantages do the examples referred to have over a password?

Answer

With the exception of magnetic cards, the examples are normally inseparable from the individual being authenticated. All the examples dispense with the need to remember or record a password.

In general, combining two components, such as something you know (a password) and something you possess (e.g. a physical device or attribute, whether separable from the

legitimate user or not) gives a higher level of security than either component alone. This is a valuable concept that is put to use in many practical security systems.

Activity 16

Think of examples where this principle is in common use.

Answer

I thought of my cash point card and my mobile telephone. Both require me to enter a number sequence (what I know) in addition to possessing an artefact (the card in one case and the mobile telephone containing the SIM card in the other). (SIM stands for subscriber identity module.) This assumes that the user has enabled the SIM card key, although it seems that many choose not to.

9.3 Firewalls – an overview

Firewalls play an important role in restricting and controlling access to networks. A **firewall** is normally implemented within a router or gateway, and will monitor incoming and outgoing traffic at the boundary of the protected zone. It is a device that denies external hosts access to selected insecure services within the protected zone (e.g. denial of dial-in services), while also denying internal hosts access to insecure services outside the protected zone. Figure 13 shows a firewall protecting the only access to network A. There may be further control within the protected zone, for example to limit access from one internal LAN segment to another. A firewall provides the means to implement some of an organisation's network security policies and may be transparent to users of the network in terms of its presence and the level of inconvenience caused. This depends on the type of firewall and the policies that are implemented.

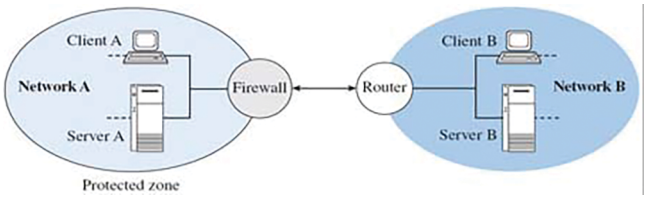

Figure 13 Firewall protection at point of access

Activity 17

In the previous section I found it useful to regard a password as a type of key that would allow legitimate users access to particular services. Would you consider a firewall to be another type of key?

Answer

A firewall also allows or bars access to services, but its role is more selective in that users may be allowed access to some services but barred from others. To that extent it may be helpful to consider a firewall as performing a gatekeeping role, i.e. allowing access to some but not others.

I shall now look at three different types of firewall – packet-filtering routers, application level gateways, and circuit level gateways – concluding with examples of firewall implementation.

9.4 Packet-filtering router

A packet-filtering router either blocks or passes packets presented to it according to a set of **filtering rules**. Figure 14 shows this arrangement.

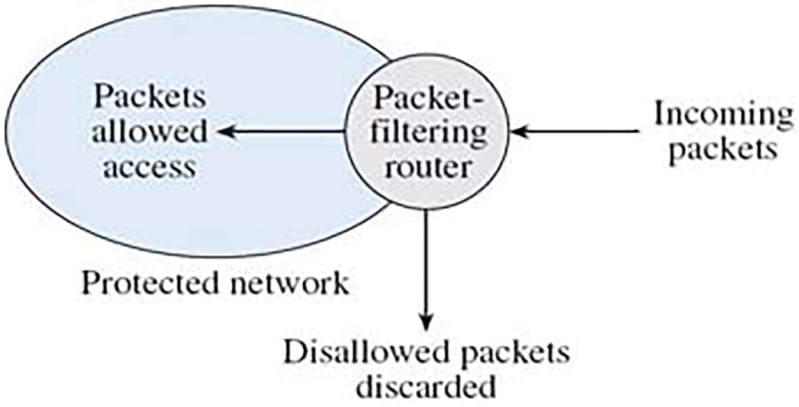

Figure 14 Packet-filtering router

Filtering rules are based on various features of the service or protocols involved, including:

- the packet header information, e.g. IP source and destination addresses
- the encapsulated protocol being used, e.g. TCP or UDP, ICMP or IP tunnel (see Box 4 below)
- the transport layer source and destination ports
- the ICMP message type
- the incoming and outgoing interfaces for the packet.

UDP (user datagram protocol) is a transport layer protocol in the internet reference model. It is used for traffic that does not need the services of a TCP connection. ICMP (internet control message protocol) is used to communicate problems from routers and hosts in the network. It supports, for example, the widely used PING command referred to in Section 3.6.

Box 4 : IP tunnel

In **IP tunnel** an extra IP header is added to a packet to avoid revealing the originating source and final destination IP addresses when a message is being sent across an intermediate network. This idea of concealment of header details by tunnelling is employed

in some important security protocols that have been developed specifically for use in IP networks over the internet.

The choice of rules and the way in which they are implemented will allow a router to admit or bar specific types of user traffic. Services that may be called up by users at their computers are generally identified in terms of TCP or UDP port numbers. Figure 15 shows where TCP port numbers are identified in the header and the box on TCP port numbers explains briefly the concept of TCP ports.

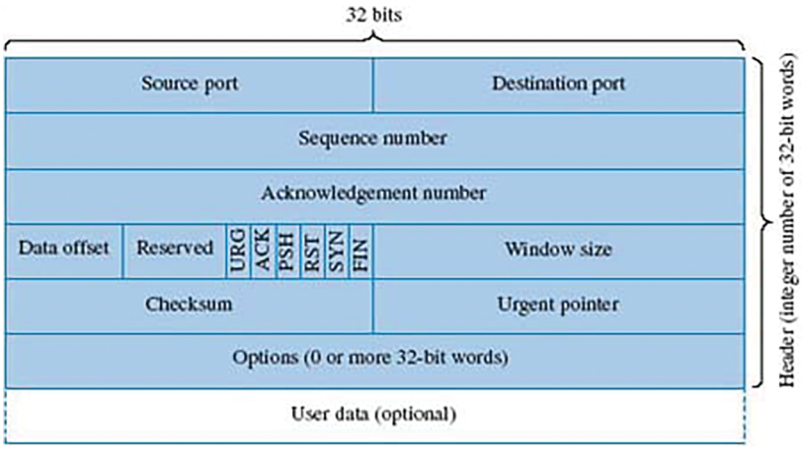

Figure 15 TCP segment format

Box 5 : TCP port numbers

TCP is a connection-oriented protocol and provides services by creating end points called sockets at client and server machines. A socket uses the IP address of the host and a 16-bit port number. (A port is sometimes referred to as a transport service access point.) Port numbers below 1024 are reserved for standard services and are referred to as **well-known ports**.

For example, Telnet is an application protocol associated with the TCP/IP family. It enables a user at one computer to log in to another computer, issue commands to the operating system, and run programs. A Telnet server will listen for remote connections on TCP port 23, while an email service based on another application protocol, SMTP (simple mail transfer protocol), will listen for incoming connections on TCP port 25.

The Internet Assigned Numbers Authority (IANA) has a list of well-known ports on its website.

From the description in the box, Telnet presents a significant security risk in terms of its ability to exercise remote control of a workstation, server or other network device. However, Telnet can be disallowed by including its TCP port number in the filtering rules. Given that a TCP port number specifies a type of service and an IP address specifies a host address, you should be able to see how combinations of the two can be used to restrict access to certain services to certain hosts. Consider this further by attempting SAQ 12.

SAQ 12

How could a packet-filtering router:

1. restrict incoming traffic from a specified external network?
2. restrict access to a Telnet service to selected hosts behind the firewall?
3. combat an attempt by an outside source to masquerade as an internal host?

Answer

1. The router could disallow all packets from the specified network by referring to the relevant IP addresses in the filtering rules.
2. In the filtering rules TCP port 23 could be disallowed for all hosts except those with certain internal IP addresses.
3. An external attack could be based on knowledge of one of the target network's internal IP addresses. However, a packet arriving at a network interface from an external circuit, but having an internal source IP address, would be highly suspect. This could be reflected in the packet-filtering rules by specifying that, for all interfaces from external circuits, packets presenting source addresses that are internal network IP addresses would be barred. This type of attack is termed ‘IP address spoofing’.

There are many types of attack that can be resisted using packet-filtering rules, but I shall consider just one other as an example.

Figure 16 shows the structure of an IPv4 packet. The fragment offset field and the three flag bits that precede it in the packet header allow an IP packet to be split into two or more fragments, if it would otherwise exceed the maximum size set by the lower layer frame limit. However, this feature could be misused in a number of ways. For example, a fragment could be forced to be so small that the encapsulated packet header information would be split between fragments. Using this tactic, an attacker could circumvent filtering rules that checked the header information of encapsulated packets (e.g. the port identities of an encapsulated TCP segment as in Figure 15) only in the first fragment that is received.

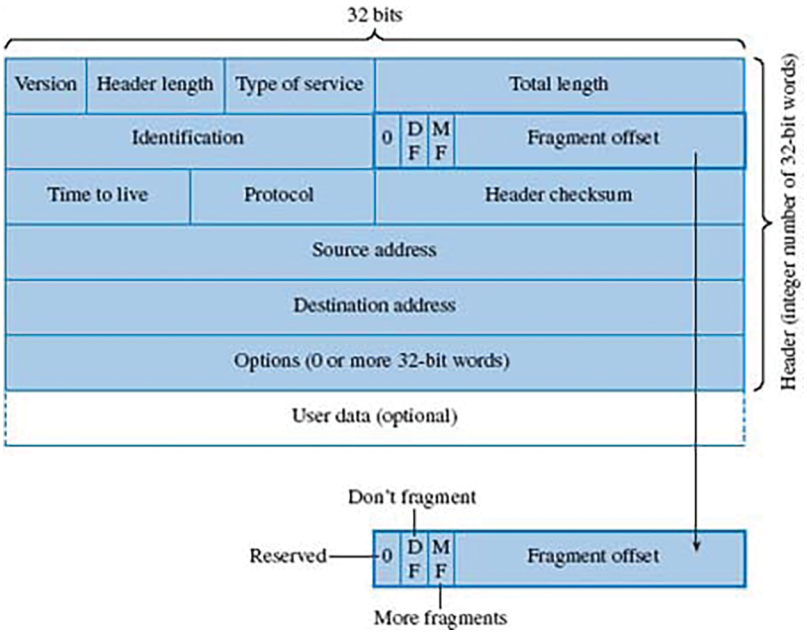

Figure 16 IP packet format

SAQ 13

Figure 15 defines the format of a TCP segment that is to be transported using IP. What would be the minimum number of bytes that should be included in a fragmented IP packet to ensure that the TCP port identification was included?

Answer

The IP header shown in Figure 16 includes 5 rows of 32 bits, assuming no options are present. This equates to 20 bytes. The TCP destination port in Figure 15 is at the end of the first row of 32 bits, so another 4 bytes would ensure that both TCP ports were included. Therefore a minimum of 24 bytes would be required.

A certain way to avoid a restricted service being accessed through the misuse of IP fragmentation would be to reject any IP packet in which fragmentation was allowed. In practice, when fragmentation is allowed, packet-filtering routers are usually set to reject IP fragments that are less than 20 bytes greater than the IP header length.

Defining the rules under which packets are filtered demands a wide knowledge of internet service types. Filtering rules need to be detailed and can become complex. When packets are filtered using complex rules, the time for each packet to be processed by the router may increase significantly and degrade system performance. When traffic must be restricted because of its likely topic content, a packet-filtering approach that works on the basis of addresses (at the IP and TCP levels) will not be able to meet requirements. A higher layer approach is needed and this can be provided by the application level gateways that I shall describe next.

Despite the limitations of packet-filtering routers, they are widely deployed as they are economical and can be implemented on standard routers, although additional software may need to be installed. Users behind a packet-filtering firewall generally find the degree of restriction involved acceptable and relatively unobtrusive.

9.5 Application level gateways

An **application level gateway** is implemented through a **proxy server**, which acts as an intermediary between a client and a server. A client application from within the protected network may request services originating from less secure networks such as the internet. After the client's authentication has been confirmed, the requests for services are relayed onwards by the proxy server, provided that they are allowed by the security policies in force. All subsequent data exchanges in relation to the service request are handled by the proxy server.

An application level gateway relays requests for services at the application level. Policy decisions to block or permit traffic are based on features identified in the application. For instance, electronic mail will be associated with a variety of mail applications and an application level gateway will act on criteria such as the message size, header fields or likely content, as indicated by key words. Application level gateways typically provide proxy services for email, Telnet and the World Wide Web.

Normally, each supported service is rigorously defined so that any undefined services are not available to users. Each internal host allowed to use or provide the specified services must also be defined. The term ‘application level gateway’ is appropriate because, from the view of both the clients within the protected network and the remote servers, the proxy server is seen as the end user. The originating client and the remote server are hidden from each other.

Because an application level gateway is exposed to greater risk than the hosts it protects, the proxy server normally takes the form of a specially secured host, referred to as a **bastion host**. This is specifically designed to be more resistant to attacks than other hosts on the protected network. For instance, a bastion host will run a secure version of the operating system, and may allow only essential services to be installed with a restricted set of Telnet, DNS, FTP and SMTP protocols. (DNS is the domain name system used on the internet to convert between the names of devices and their IP addresses. FTP is file transfer protocol, an application protocol in the TCP/IP family used, for example, to connect file servers.) In addition, a strong user authentication process is employed along with audit facilities that record any attempts to intrude.

Code specifically designed to enhance regular checking for software bugs is used, and each proxy service is designed to operate independently of others so that installation or removal of a service can be undertaken without affecting other services. Viruses and worms may also be screened.

Access to memory drives on the gateway is severely restricted to minimise threats from Trojans, and user log-on is not allowed. Other threats that could be countered using this type of firewall include those arising from importing macros (a software macro defines how a sequence of operations can be condensed into a single command), or inbound packets that include executable files (containing EXE or COM extensions), because of the possibility of introducing virus and worm files into a network.

SAQ 14

What do you think could be the disadvantages of the application level gateway approach compared with the packet-filtering approach?

Answer

An application level gateway is more demanding in terms of the necessary hardware and software because of the burden of acting as a proxy. It is therefore likely to be more expensive than packet filtering and also to incur longer processing delays. The enforcement of strict policies may also be seen as restricting the options of users behind the firewall or of legitimate ones outside. This type of firewall is less user-friendly and less transparent than a packet-filtering firewall.

9.6 Circuit level gateways

A **circuit level gateway** operates at the transport layer of the OSI or internet reference models and, as the name implies, implements circuit level filtering rather than packet level filtering. It checks the validity of connections (i.e. circuits) at the transport layer (typically TCP connections) against a table of allowed connections, before a session can be opened and data exchanged. The rules defining a valid session prescribe, for example, the destination and source addresses and ports, the time of day, the protocol being used, the user and the password. Once a session is allowed, no further checks, for example at the level of individual packets, are performed.

A circuit level gateway acts as a proxy and has the same advantage as an application level gateway in hiding the internal host from the serving host, but it incurs less processing than an application level gateway.

Disadvantages of circuit level gateways include the absence of content filtering and the requirement for software modifications relating to the transport function.

Circuit level gateways can be implemented within application level gateways or as stand-alone systems. Implementation within an application level gateway allows screening to be asymmetric, with a circuit level gateway in one direction and an application level gateway in the other.

SAQ 15

What advantages could arise from the asymmetry of the arrangement just described?

Answer

Firewall asymmetry could complement the different levels of risk relating to incoming and outgoing traffic on the protected network. For example, user-friendly outgoing services could be maintained to hosts behind the firewall by allowing circuit level functionality on outbound traffic. This is appropriate where internal users' requests are relatively trustworthy. By contrast, inbound traffic could be subjected to the full scrutiny of application level content. Application level examination of traffic involves a considerable processing overhead, but this would be performed on incoming traffic only.

9.7 Examples of firewall implementation

In practice, firewalls are likely to be combinations of the types that I have described. For example, a **screened sub-network** is commonly incorporated in a firewall scheme, as

shown in Figure 17. In this configuration an application level gateway implemented in a bastion host is used in combination with two packet-filtering routers. The screened sub-network that is formed is termed a **demilitarised zone** (DMZ). Placing servers and dial-in modems that are accessed by external users in a DMZ is a way of separating these higher-risk components from the protected internal network. Both external hosts and hosts within the internal network have access to services provided on the DMZ, but traffic across it is blocked, preventing external users from gaining direct access to the protected internal network.

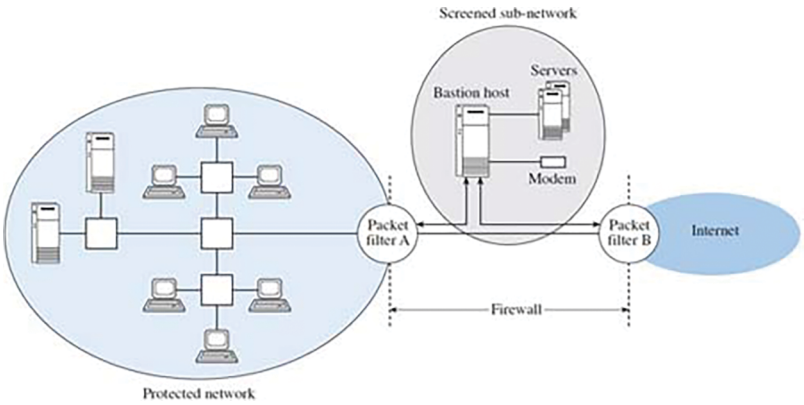

Figure 17 A combined firewall

To end this unit I shall very briefly indicate the way in which the Open University's network is protected by its firewall.

Figure 18 represents the Open University's firewall arrangement, which needs to accommodate the diverse networking needs of many people: for example, students, administrators, academics, whether on site or working from remote sites such as conference venues, home or summer school locations. The Open University has its headquarters at Walton Hall, Milton Keynes. Thirteen regional centres and warehousing facilities each have LANs linked to the Walton Hall LAN to create the Open University's wide area network.

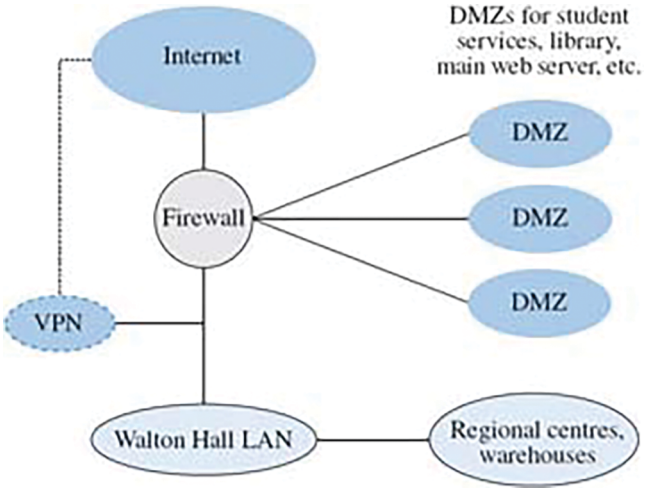

Figure 18 The Open University firewall

Figure 18 shows the firewall protecting the Walton Hall / internet interface. The services that students need to access are located within the DMZs. Students typically connect from their homes using dial-up modems or ADSL links to access the internet through their

internet service providers, or they gain access from their workplaces. Web browsers are used to access services such as the library, the main web server or student services, and electronic conferencing software is used to access the servers that support the various course conferences.

In general, the firewall allows traffic to and from the DMZs but only traffic that can be identified as being initiated by internal users on the Open University's LANs is permitted to cross the firewall.

An additional feature of the Open University's arrangements allows authorised staff access to appropriate areas of the Walton Hall LAN from external locations. To do this a virtual private network (VPN) provides a logical bypass to the firewall, but access is secured by the use of ‘one-time’ password generators in ‘key fobs’ allocated to authorised users. These generate a frequent supply of different passwords. Before any request for services using the VPN is granted, the user requesting the service must respond with a valid password to a challenge from the VPN security system.

Conclusion

10.1 Summary of Sections 1–5

There are many terms and abbreviations relating to this topic, and it is important to understand them.

Looking at the background to network security can help to put its more technical aspects in context.

Communication networks and the data they carry are vulnerable to a range of attacks. These can be categorised as either passive or active attacks. In a passive attack, communication across a network is observed but data within messages is not interfered with and messages may not even be readable by the attacker. Traffic analysis is strongly associated with passive attacks, but may also be a legitimate process for effective network management. An active attack typically involves, for example, the modification of messages or their replay, or access to data and networks through the assumption of a false identity, leading to either the misuse of data or disruption of network services.

There are many opportunities for an attacker to gain physical access to networks and these need to be resisted by both electronic and mechanical means. The sharing of networks through, for example, the internet, provides opportunities for attack without the need for the attacker to be physically close to the target networks. Wireless LAN technology presents further opportunities for illicit network access.

Encryption is one of the fundamental building blocks of network security. Encryption transforms plaintext into ciphertext, while decryption reverses the process. Encryption systems are based on one of two basic methods. The first is a symmetric key system in which a single secret key is shared between the two communicating entities. The second is an asymmetric system which uses two mathematically related keys known as the public key and the private key. One key is used for encryption and the other for decryption. Asymmetric key systems place higher demands on processing resources than symmetric key systems and tend to be slower, but they are more scalable and they do not rely on any prior secret key exchange. Hybrid systems are a combination of symmetric and asymmetric key systems.

The implementation of encryption systems can be related to network protocol layers. Encryption can be applied in link layer or end-to-end mode. In link layer encryption, the encryption and decryption processes take place at each node along a path, but this can be expensive and slow. End-to-end encryption involves applying encryption at higher layers in the protocol stack. Network layer encryption and application layer encryption are examples. Security vulnerabilities arise when protocol header information is exposed during message transmission or at nodes where data is processed in unencrypted form.

10.2 Summary of Sections 6–9

Integrity relates to assurances that a message has not been tampered with in any unauthorised way. A method of providing this assurance is to create a message digest, which gives a concise identity of the original message, and append it to the message. The message digest of the received message can then be calculated and checked for

discrepancies against the digest sent. A message digest takes the form of a small fixed-length block of data known as a hash value. A hash value created by a one-way hash function is relatively easy to compute but difficult to reverse.

Time stamps, sequence numbers and nonces are used to provide assurances about the freshness of a message and help to prevent replay attacks.

Message authentication can be provided by including a digest of the message encrypted by the sender's private key. The encrypted digest is known as a digital signature. The recipient decrypts the digest using the sender's public key, computes a new digest of the received message and compares the results. A certification authority is a trusted third party that is able to validate public keys by issuing a digital certificate that binds the identity of the user with the key.

The most common ways of controlling access to communication networks are restricting mechanical access and implementing password schemes and firewalls. Strong passwords can be generated provided that they contain no recognisable structure. Such passwords should be capable of withstanding, at least for a useful period of time, brute force and other computer-assisted discovery techniques. However, such passwords are difficult to remember and human factors become critical.

Firewalls are implemented to control traffic at the borders of protected networks. Three approaches are based on packet-filtering rules (packet-filtering router), application type and content (application level gateway) and validity of transport connection (circuit level gateway). These approaches in varying combinations can provide firewalls appropriate to the level of perceived threat, but sufficiently non-restricting to legitimate users of the protected networks.

References

Halsall, F. (2001) *Multimedia Communications*, Addison Wesley.

ITU-T X.509 (2000) *Information Technology – Open Systems Interconnection – The Directory: Public-Key and Attribute Certificate Frameworks* , International Telecommunication Union.

King, T. and Newson, D. (1999) *Data Network Engineering*, Kluwer.

Peterson, L. L. and Davie, B. S. (1996) *Computer Networks: A Systems Approach*, Morgan Kaufmann.

RFC 2401 (1998) *Security Architecture for the Internet Protocol*, Kent, S., Atkinson, R.

Schneier, B. (1996) *Applied Cryptography*, 2nd edn, Wiley.

Stallings, W (1999) *Cryptography and Network Security*, Prentice Hall.

Stallings, W (2001) *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*, 3rd edn, Addison Wesley.

Further reading

Anderson, R. (2001) *Security Engineering: A Guide to Building Dependable Distributed Systems* , Wiley.

BS 7799-2 (2002) *Information Security Management Systems – Specification with Guidance for Use* , British Standards Institution.

Ellis, J. and Speed, T. (2001) *The Internet Security Guidebook*, Academic Press.

ISO/IEC 17799 (2000) *Information Technology – Code of Practice for Information Security Management* , International Organization for Standardization.

Tanenbaum, A. S. (1996) *Computer Networks*, 3rd edn, Prentice Hall.

Websites

British Standards Institution

Communications-Electronics Security Group

Internet Assigned Numbers Authority

Acknowledgements

Grateful acknowledgement is made to the following sources for permission to reproduce material within this unit:

Figure 1, accessed 8 February 2007. Figure 2 based on Stallings, W., *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*, 3rd edition, May 2001, Addison Wesley; Figures 7 and 8 based on Schneier, B. (1996) *Applied Cryptography*, 2nd edition, John Wiley & Sons; Figure 9 based on King, T. and Newson, D. (1999) *Data Network Engineering*, Kluwer; Figure 10 based on Stallings, W (1995) *Cryptography and Network Security*, Prentice Hall.

Every effort has been made to contact copyright owners. If any have been inadvertently overlooked, the publishers will be pleased to make the necessary arrangements at the first opportunity.

Don't miss out:

If reading this text has inspired you to learn more, you may be interested in joining the millions of people who discover our free learning resources and qualifications by visiting The Open University - www.open.edu/openlearn/free-courses