Chapter 27 Network Security

Content

Syllabus Learning Outcomes

4.3 Network Security

Understand computer network security in terms of threats, protection mechanisms, and secure access.

- 4.3.1 Understand how malware (e.g. worms and viruses) and denial of service (DOS) attacks can compromise computer systems.
- 4.3.2 Understand how firewall (filtering function), intrusion detection system (IDS), and intrusion prevention system (IPS) can be used to restrict network access and their limitations.
- 4.3.3 Understand how encryption, digital signature, and authentication can ensure the security of network applications.

1 Introduction to Network Security

In today's digital world, our reliance on computer networks is undeniable. We use them for everything from online banking to social media, making them a prime target for malicious actors. This is where network security comes in.

Network security is the shield that protects the **confidentiality**, **integrity**, and **availability** (CIA triad) of information and resources within a network.

Confidentiality: This principle ensures that only authorised users can access sensitive information within the network. Think of it as keeping your secret documents locked away in a safe, accessible only to those with the proper key (authorization). Encryption and access controls are crucial for maintaining confidentiality.

Integrity: This principle ensures that data within the network is accurate and hasn't been tampered with. Imagine working on a crucial report and someone changing the data without your knowledge. Intrusion detection systems (IDS) and digital signatures help guarantee the integrity of data by identifying unauthorized modifications.

Availability: This principle ensures that authorised users can access the information and resources they need whenever they require them. A network that's constantly under denial-of-service attacks or experiences frequent outages fails to meet the availability criteria. Firewalls and network redundancy strategies play a vital role in keeping resources accessible.

Imagine your network as a castle. Network security acts like the guards, firewalls, and secure gates, ensuring only authorised users can enter, and your valuables (data) are safe from theft or destruction.

This introduction provides an overview of network security, highlighting its importance in protecting our increasingly digital lives. We'll delve deeper into specific threats, protection mechanisms, and secure access methods in the following sections.

2 Threats

2.1 Malware

Malware (malicious software) is the general term for software that is specifically designed to disrupt, damage, or gain unauthorised access to a computer system. The site https://threatmap.checkpoint.com/ allows us to visualise some types of live cyberattacks worldwide.

Malware	Description
Ransomware	A type of malware that blocks access to the victim's computer system until a certain amount of money, usually in bitcoin or other cryptocurrency is paid. The most recent category of malware is ransomware, which garnered headlines in 2016 and 2017 when ransomware infections encrypted the computer systems of major organizations and thousands of individual users around the globe.
Scareware	It is a program that attempts to frighten the victim into buying unnecessary software or providing their financial data. Scareware pops up on a user's desktop with flashing images or loud alarms, announcing that the computer has been infected. It usually urges the victim to quickly enter their credit card data and download a fake antivirus program.

Spyware	A hidden program that secretly collects personal information about users and sends the information to attackers without the user's knowledge, without causing data corruption or data loss. Spyware may record the websites the user visits, information about the user's computer system and vulnerabilities for a future attack, or the user's keystrokes. Spyware that records keystrokes is called a keylogger. Keyloggers steal credit card numbers, passwords, account numbers, and other sensitive data simply by logging what the user types.
Adware	Adware pushes unwanted advertisements at users and spyware secretly collects information about the user.
Fileless malware	Unlike traditional malware, fileless malware does not download code onto a computer, so there is no malware signature for a virus scanner to detect. Instead, fileless malware operates in the computer's memory and may evade detection by hiding in a trusted utility, productivity tool, or security application. An example is Operation RogueRobin, which was uncovered in July 2018. RogueRobin is spread through Microsoft Excel Web Query files that are attached to an email. It causes the computer to run PowerShell command scripts, providing an attacker access to the

	system. As PowerShell is a trusted part of the Microsoft platform, this attack typically does not trigger a security alert. Some fileless malware is also clickless, so a victim does not need to click on the file to activate it.
Cookies	A small piece of data used by websites to store personal information on a user's web browser. It is misused by attackers to collect personal information about users.
Pharming	The interception of requests sent from a computer to a legitimate website and redirection to a fake website to steal personal data or credit card details. The attacker can use the personal details to access the victims' bank account in the bank's actual website.
Phishing	The use of emails and fake websites that appear to be from reputable companies. It is used to steal personal information such as passwords and credit card numbers from users.

Spamming	The mass distribution of unwanted messages or advertising sent to email addresses collected from sources such as public mailing lists, social networking sites, company websites and blogs. Emails are usually easily sent to users and the emails sent are used to lure users to enter their personal information and steal their personal data.
Trojan horse	A computer program that pretends to be a harmless file or useful application. When the trojan horse is run, it does something harmful such as giving intruders unauthorised access to the computer instead.
Virus	A computer program that attaches itself to a normally harmless program and modifies it. When the modified program is run by a user, the virus attaches copies of itself to any other programs it can find, thus infecting it.
Worm	A standalone computer program that runs automatically and attempts to spread copies of itself over a network. Unlike a virus, it does not attach itself to a program and it spreads by exploiting a vulnerability in the infected system or through email as an attachment masquerading as a legitimate file. Worms consume bandwidth and overload web servers, harming the host network.

	A graduate student created the first worm (the Morris worm) in 1988 as an intellectual exercise. Unfortunately, it replicated itself quickly and soon spread across the internet.

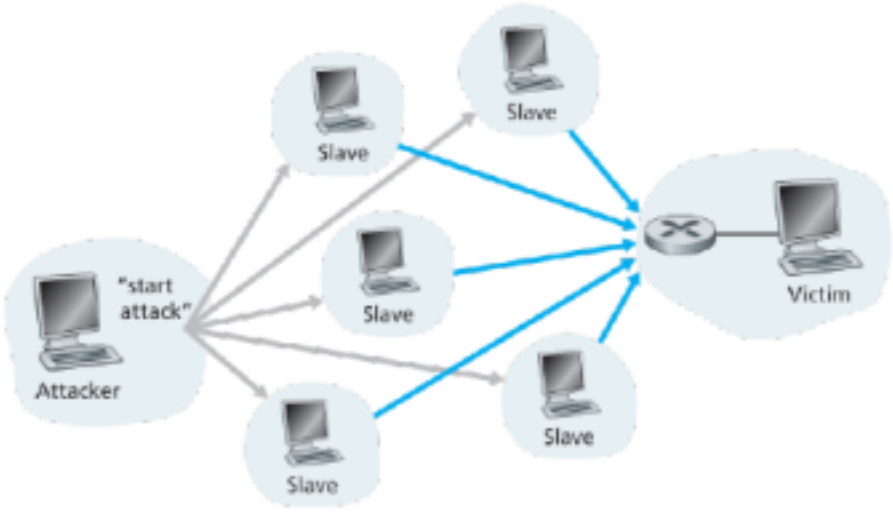
2.2 Denial of Service (DoS)

A denial of service (DoS) attack is an attempt, by a malicious actor, to overload a website or network, with the aim of degrading its performance or even making it completely inaccessible to its intended users. DoS attacks typically function by overwhelming or flooding a targeted machine with requests until normal traffic is unable to be processed, resulting in denial-of-service to addition users. A DoS attack is characterized by using a single computer to launch the attack.

A distributed denial-of-service (DDoS) attack is a malicious attempt to disrupt the normal traffic of a targeted server, service or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic.

DDoS attacks achieve effectiveness by utilizing multiple compromised computer systems as sources of attack traffic. Exploited machines can include computers and other networked resources such as IoT devices.

DDoS attacks are carried out with networks of Internet-connected machines.



A distributed denial-of-service attack

Source: Kurose, J., & Ross, K. (2016). Computer Networking: A Top-Down Approach (7th ed.). Pearson.

These networks consist of computers and other devices (such as Internet of Things (IoT) devices) which have been infected with malware, allowing them to be controlled remotely by an attacker. These individual devices are referred to as bots (or zombies), and a group of bots is called a botnet.

Once a botnet has been established, the attacker is able to direct an attack by sending remote instructions to each bot.

When a victim's server or network is targeted by the botnet, each bot sends requests to the target's IP address, potentially causing the server or network to become overwhelmed, resulting in a denial-of-service to normal traffic.

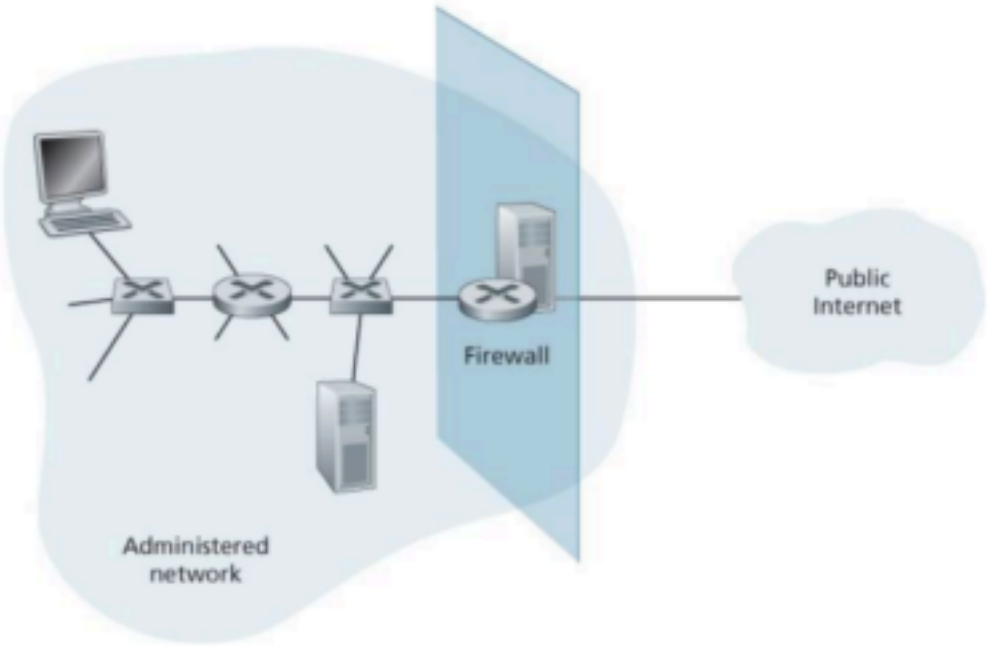
3 Protection Mechanisms

Network security devices, like firewalls, intrusion detection systems (IDS), and intrusion prevention systems (IPS), play a crucial role in safeguarding our networks. These devices constantly monitor incoming and outgoing traffic, analyzing its security risk. Based on pre-defined rules, they can either allow safe traffic to pass through (forward), log suspicious activity, or completely block malicious traffic (drop).

3.1 Firewalls

A firewall acts as a filter that monitors access between an organisation's internal network and the Internet at large, allowing some packets to pass and blocking others. A firewall allows a network administrator to control access between the outside world and resources within the

administered network by managing the traffic flow to and from these resources.



3.1.1 Software and hardware firewalls

There are software and hardware firewalls. Each format serves a different but important purpose.

A hardware firewall is physical, like a broadband router – it connects the network and gateway through hardware like wires. A software firewall is internal – a program on your computer that works through port numbers and applications.

There also are cloud-based firewalls, known as firewall as a service (FaaS). One benefit of cloud-based firewalls is that they can grow with your organisation and, similar to hardware firewalls, do well with perimeter security (preventing unauthorised users from accessing a network).

3.1.2 Host-based vs Network-based firewalls

A host-based firewall is installed on an individual computer to protect it from activity occurring on its network. The policy may affect what traffic the computer accepts from the Internet, from the

local network, or even from itself.

A network-based firewall is implemented at a specified point in the network path and protects all computers on the “internal” side of the firewall from all computers on the “external” side of the firewall. Network-based firewalls may be installed at the perimeter, or edge, of a network to protect a corporation from hosts on the Internet, or internally to protect one segment of the community from another, such as separating corporate and residential systems, or research systems from martketing systems. A network-based firewall cannot protect one computer from another on the same network, or any computer from itself.

3.1.3 Functionality and Structure of firewalls

A firewall has three goals:

1. **Traffic Control**: All incoming and outgoing communication must pass through the firewall, which sits squarely at the boundary between the administered network and the rest of the Internet. While large organisations may use multiple levels of firewalls or distributed firewalls, locating a firewall at a single access point to the network makes it easier to manage and enforce a security-access policy, creating a single choke point for inspection.
2. **Authorised Traffic**: Only authorised traffic, as defined by the local security policy, will be allowed to pass. Unauthorised access attempts are blocked, protecting valuable network resources.
3. **Mainting security**: While connected to the network, the firewall itself is designed to be resistant to attacks. It acts as a robust barrier, preventing external threats from infiltrating the internal network.

There are several different types of firewalls based on their structure and functionality. Here are some firewalls implementation, depending on the size of network and the level of security needed:

- **Packet Filters**: Traditional packet filters occur at a gateway router that connects the internal network to its ISP. It examines each datagram based on the administrator-specific rules.

- **Stateful Packet Filters**: Stateful packet filters track TCP connections and use this information to make filtering decisions.

- **Application Gateways**: These are application-specific servers through which all application data (inbound and outbound) must pass. They look beyond the IP/TCP/UDP headers and make policy decisions based on application data.

You may refer to this resource for more in-depth discussion on firewalls:

https://www.ciscopress.com/articles/printerfriendly/1823359

3.1.4 Limitations to firewalls

They cannot protect against attacks from a source if a user has explicitly allowed it to bypass the firewall. For example, some applications may ask the user to allow adding an exception in the firewall software so that the application can send or receive data without being blocked by the firewall. However, that prevents firewalls from stopping any malicious traffic into the application.

They also cannot protect against internal attacks since the malicious traffic may not need to pass through a firewall.

As the firewall is a single point of failure, the system becomes vulnerable when the firewall is compromised.

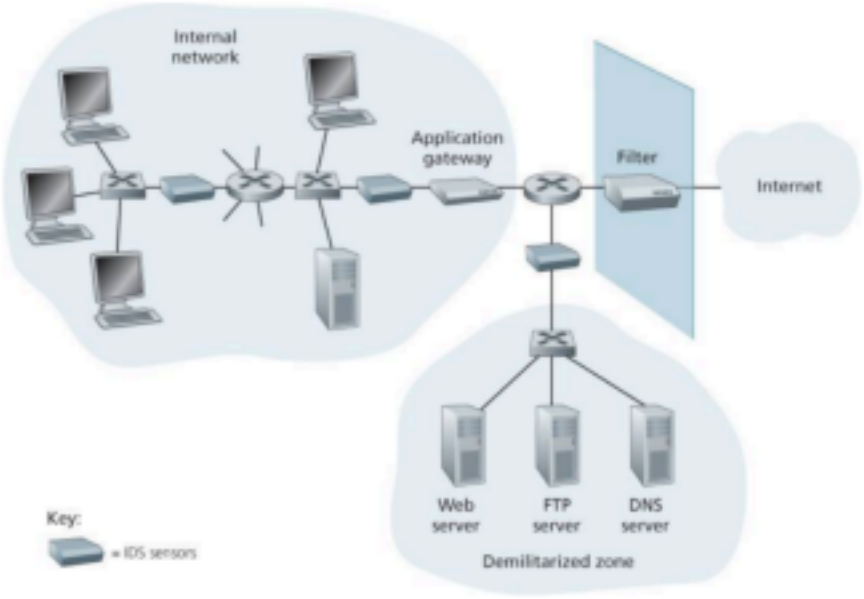
3.2 Intrusion Detection System and Intrusion Prevention System

To detect many attack types, we need to perform deep packet inspection, that is, look beyond the header fields and into the actual application data that the packets carry. Application gateways often do deep packet inspection but this is only possible for a specific application.

Thus, Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) are needed. IDS is a device that generates alerts when it observes potentially malicious traffic. IPS is a device that filters out suspicious traffic. In this section, we study both systems together, hence, we will collectively refer to IDS systems and IPS systems as IDS systems.

An IDS can be used to detect a wide range of attacks, including network mapping (emanating, for example, from nmap), port scans, TCP stack scans, DoS bandwidth-flooding attacks, worms, and viruses, OS vulnerability attacks, and application vulnerability attacks.

An organisation may deploy one or more IDS sensors in its organizational network. When multiple sensors are deployed, they typically work in concert, sending information about suspicious traffic activity to a central IDS processor, which collects and integrates the information and sends alarms to network administrators when deemed appropriate.



In the above diagram, the organisation has partitioned its network into two regions - a high-security region, protected by a packet filter and an application gateway and monitored by IDS sensors; and a lower-security region also known as a demilitarized zone (DMZ), which is protected by the packet filter, and monitored by IDS sensors. Note that the DMZ includes the organisation's servers that need to communicate with the outside world, such as its public Web server and its authoritative DNS server.

IDS not only needs to do deep packet inspection but must also compare each passing packet with tens of thousands of "signatures". By placing the IDS sensors further downstream, each sensor sees only a fraction of the organisation's traffic, and can more easily keep up. Nevertheless, high-performance IDS and IPS systems are available today, and many organisations can actually get by with just one sensor located near their access router.

IDS systems are broadly classified as either **signature-based systems** or **anomaly-based systems**. A signature-based IDS maintains an extensive database of attack signatures. Each signature is a set of rules pertaining to an intrusion activity. A signature may simply be a list of characteristics about a single packet (e.g., source and destination port numbers, protocol type, and a specific string of bits in the packet payload), or may relate to a series of packets. The signatures are normally created by skilled network security engineers who research known attacks. An organisation's network administrator can customize the signatures or add its own to the database.

Operationally, a signature-based IDS sniffs every packet passing by it, comparing each sniffed packet with the signatures in its database. If a packet (or series of packets) matches a signature in the database, the IDS generates an alert. The alert could be sent to the network administrator in an e-mail message, could be sent to the network management system, or could simply be logged for future inspection.

Signature-based IDS systems, although widely deployed, have a number of limitations. Most importantly, they require previous knowledge of the attack to generate an accurate signature. In

other words, a signature-based IDS is completely blind to new attacks that have yet to be recorded.

Another disadvantage is that even if a signature is matched, it may not be the result of an attack, so that a false alarm is generated. Finally, because every packet must be compared with an extensive collection of signatures, the IDS can become overwhelmed with processing and actually fail to detect many malicious packets.

An anomaly-based IDS creates a traffic profile as it observes traffic in normal operation. It then looks for packet streams that are statistically unusual, for example, an inordinate percentage of ICMP packets or a sudden exponential growth in port scans and ping sweeps. The great thing about anomaly-based IDS systems is that they don't rely on previous knowledge about existing attacks—that is, they can potentially detect new, undocumented attacks. On the other hand, it is an extremely challenging problem to distinguish between normal traffic and statistically unusual traffic. To date, most IDS deployments are primarily signature-based, although some include some anomaly-based features.

You may refer to this resource for a more in-depth discussion on IDS/IPS:

https://www.socinvestigation.com/ids-vs-ips-key-differences-rule-structure-pros-and-cons/

https://www.giac.org/paper/gsec/235/limitations-network-intrusion-detection/100739 **4**

Secure access method

Network Applications handle data. How do we ensure data is safely transmitted? We can use encryption, digital signature and authentication.

4.1 Encryption

Encryption is a process that uses an algorithm and a key to code a message written in plain text into ciphertext, which is transmitted to the recipient. Decryption is decoding the ciphertext back into the original plain text using a decryption algorithm and a key.

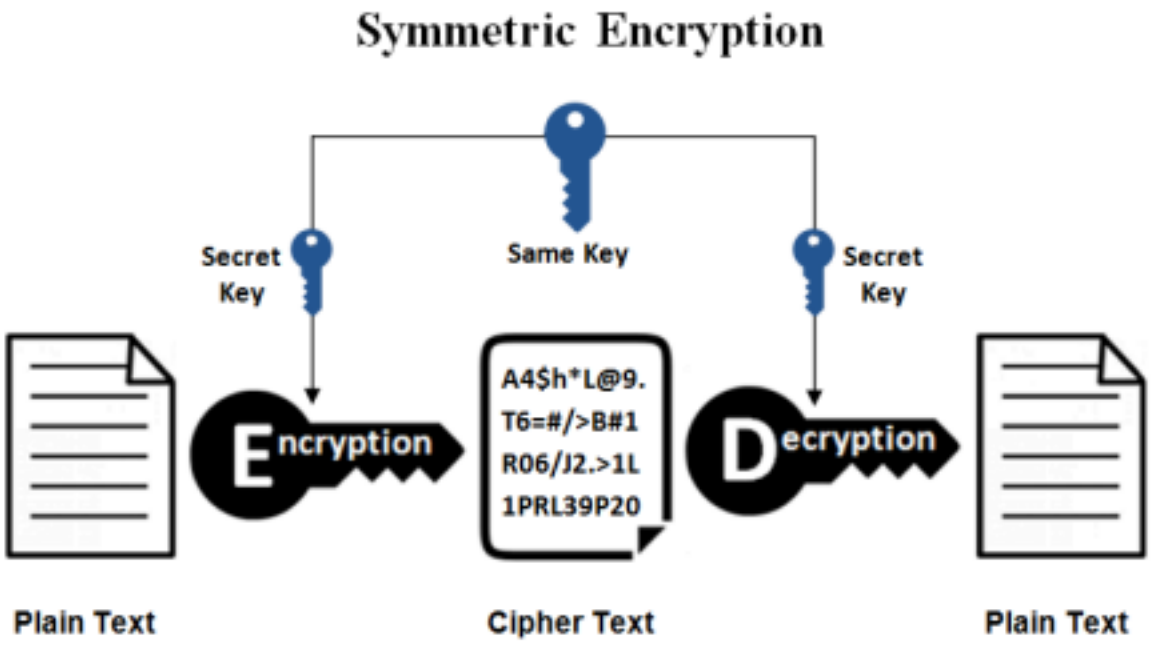
Encryption is often used to protect data from unauthorised access by allowing only authorised users to have the secret key. It can also be used in combination with file permissions so that an unauthorised user who is able to bypass file permissions would still be unable to use the accessed data without knowing the secret key.

Although cryptography is used for confidentiality, cryptographic techniques are inextricably woven into authentication, message integrity, and non-repudiation. Non-repudiation is the assurance that someone cannot deny the validity of something.

There are typically two types of encryption - symmetric key encryption and asymmetric key encryption also known as public key encryption.

4.2 Symmetric key encryption

In symmetric key encryption, there is just one key. This key is a secret shared by the sender and the receiver of a message. The sender uses the encryption algorithm together with the key to encrypt some plaintext. The receiver decrypts the ciphertext using the same key.

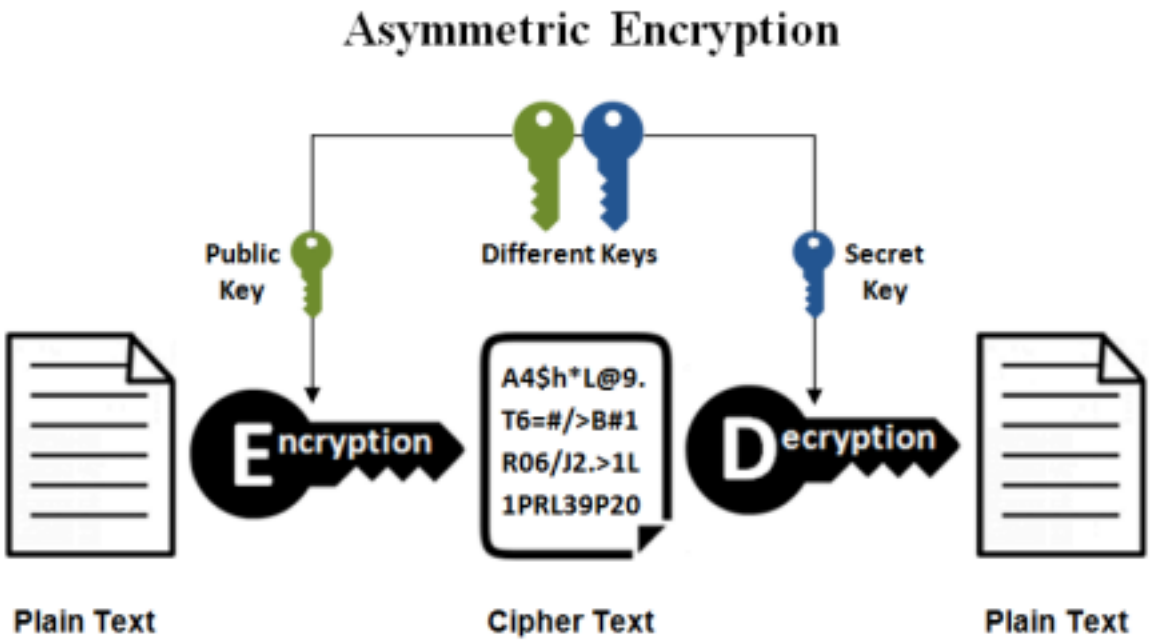


The issue with symmetric key encryption is delivery of the secret key. The sender needs the key to encrypt but it is difficult to securely deliver the key to the receiver to allow decryption.

Examples of old symmetric key cryptography are Caesar cipher, monoalphabetic cipher, polyalphabetic encryption. Modern symmetric key cryptography is categorised into two broad areas - stream ciphers and block ciphers. The latter typically, uses a technique called Cipher Block Chaining (CBC).

4.3 Public key encryption

Asymmetric encryption, also known as public-key encryption, is a type of encryption that uses a pair of keys to encrypt and decrypt data. The pair of keys includes a public key, which can be shared with anyone, and a private key, which is kept secret by the owner. In asymmetric encryption, the sender uses the recipient's public key to encrypt the data. The recipient then uses their private key to decrypt the data. This approach allows for secure communication between two parties without the need for both parties to have the same secret key.



Asymmetric encryption has several advantages over symmetric encryption, which uses the same key for both encryption and decryption. One of the main advantages is that it eliminates the need to exchange secret keys, which can be a challenging process, especially when communicating with multiple parties.

If asymmetric encryption is to be used, the process is initiated by someone in possession of two keys. One of these is a public key, which is sent to anyone who is going to partake in an encrypted communication. The other is a secret private key, which is never sent to anyone. Having a means of secure transmission of a secret key is no longer an issue.

Examples of asymmetric encryption algorithms include Rivest-Shamir-Adleman (RSA), Diffie-Hellman, and Elliptic Curve Cryptography (ECC).

4.4 Digital signature

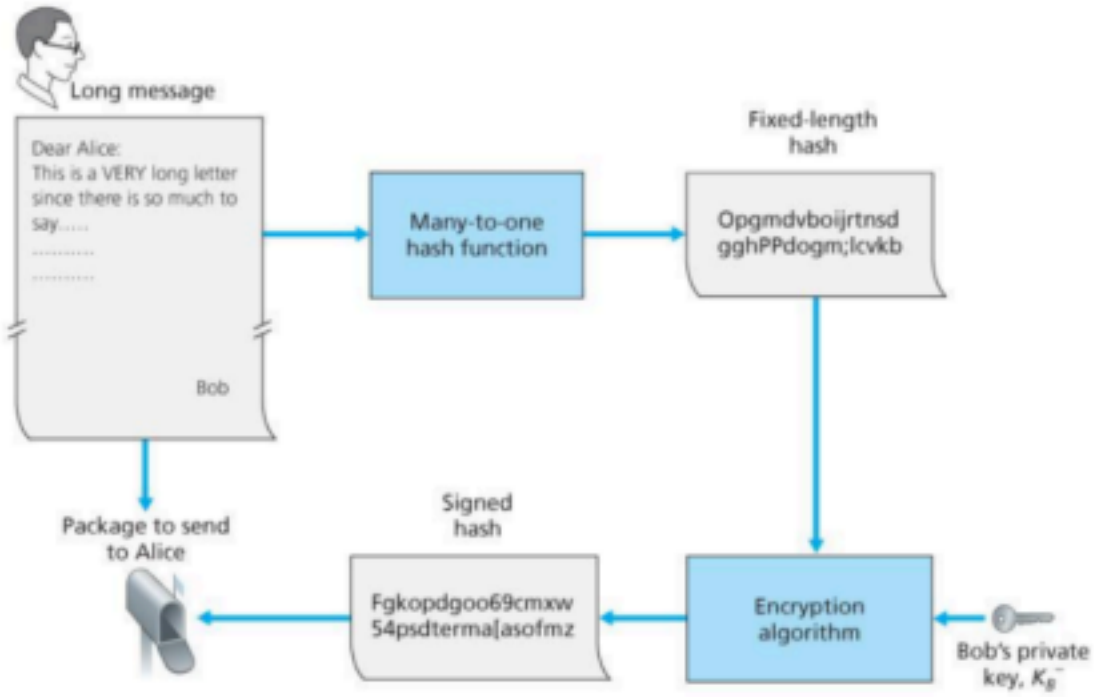
A digital signature authenticates the sender. It is a cryptographic technique to indicate the owner or creator of a resource or to signify one's agreement with a document's content in a digital world. Therefore, it must be possible to prove that a document signed by an individual was indeed signed by that individual (the signature must be verifiable) and that only that individual could have signed the document (the signature cannot be forged).

Asymmetric encryption allows for the creation of digital signatures, which can be used to verify the authenticity of data. Using asymmetric encryption, the decryption-encryption works if the keys are used the other way round. An individual can encrypt a message with a private key and send this to a recipient who has the corresponding public key, who can then use the public key to decrypt the received ciphertext. This approach would not be used if the content of a message was confidential because anyone might be in possession of the public key. However, it could be used to verify who the sender was. Only the sender has the private key and the public key only works with that one specific private key.

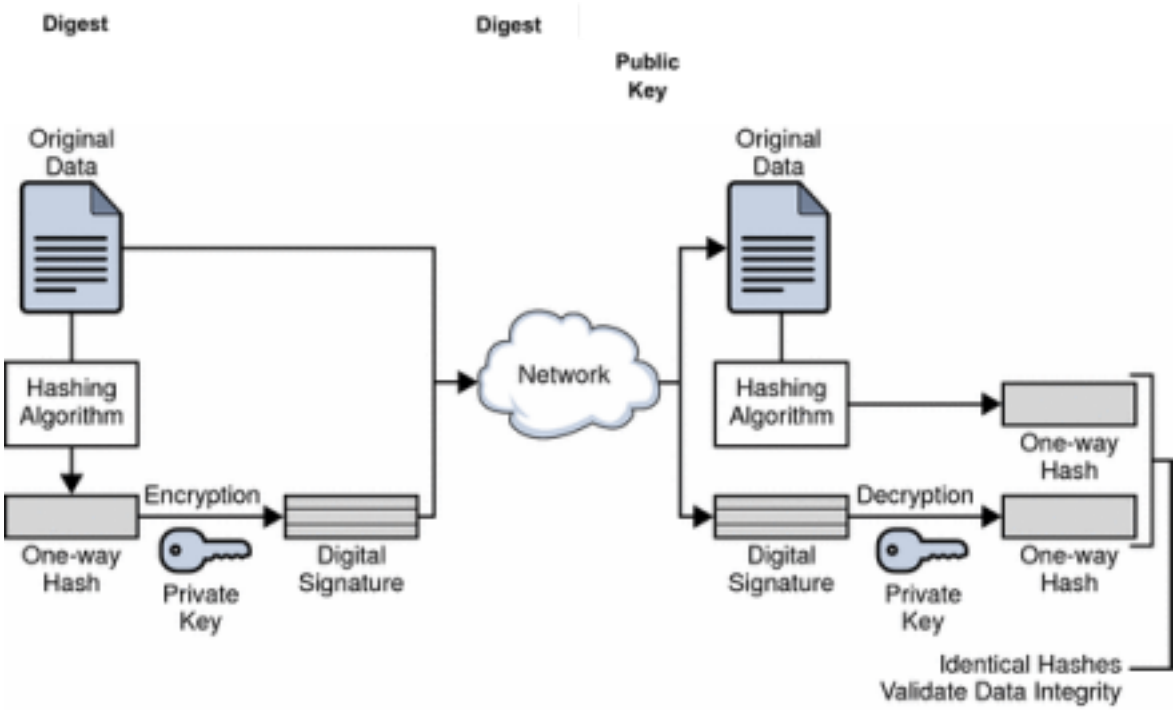
One may suggest that digital signatures can be generated via encryption techniques. However, one concern with signing data by encryption is that encryption and decryption are computationally expensive. Given the overheads of encryption and decryption, signing data via complete encryption/decryption can be overkill.

An alternative is for the sender to use a public cryptographic one-way hash function which creates a fixed-length hash that is uniquely defined for the particular message, called a 'digest'. The private key is used to encrypt the digest. The encrypted digest is the digital signature.

The message can be transmitted as plaintext together with the encrypted digest as a separate file. As the digest is much smaller than the whole message, the encryption and the transmission are faster processes than if the whole message were encrypted.



At the receiver end, the same public one-way hash function is used to create a digest from the received message. The encrypted version of the original digest is decrypted using the public key. If the two digests are identical, the receiver can be confident that the message is authentic and has been transmitted unaltered.



In summary,

1. The sender uses a hash algorithm to create a hashed version of the message 2. The sender uses its private key to encrypt the hash to the digital signature 3. Both the message (encrypted or not) and the digital signature are sent to the receiver

4. The receiver uses the sender's public key to decrypt the digital signature back to the sender's version of hash
5. The receiver uses the same hash algorithm to create a new hash from the received message. If the two hashes match, it means the data is not altered and is sent by the known sender

An important application of digital signatures is public key certification, that is, certifying that a public key belongs to a specific entity. Public key certification is used in many popular secure networking protocols, including IPsec and SSL.

4.5 Authentication

End-point authentication is the process of one entity proving its identity to another entity over a computer network, for example, a user proving its identity to an e-mail server. A concrete example is a user authenticating him or herself to an e-mail server. This is a subtly different problem from proving that a message received at some point in the past did indeed come from that claimed sender.

Often, network elements such as routers and client/server processes must authenticate each other. Here, authentication must be done solely on the basis of messages and data exchanged

as part of an authentication protocol. Typically, an authentication protocol would run before the two communicating parties run some other protocol. The authentication protocol first establishes the identities of the parties to each other's satisfaction; only after authentication do the parties get down to the work at hand.

Common ways of authentication include:

- passwords
- biometrics, for example: fingerprints, facial recognition, iris scans
- token values, such as from a physical device, a mobile phone or a software application

Some applications use two-factor authentication (2FA), which uses two different ways of authentication for better security.

Annex 1 - What is the Internet of Things (IoT)?

Internet of Things (IoT) is a catchall phrase for all the various Internet-connected devices that are not traditional computers. This includes everything from fitness trackers and smart watches to smart refrigerators, headphones, cameras, washing machines, cars, traffic lights, airplane engines, and home security systems.

As access to broadband Internet service grows and processors become more affordable, more and more gadgets with Wi-Fi capabilities are being created. Today there are billions of IoT

devices in existence.

This network of devices produces great benefits and convenience for users, but IoT devices can also be targeted by attackers as well as used to carry out cyber attacks. As with Internet-connected computers, these devices are usually safe to use, but precautions should be taken to ensure they are not compromised.

Annex 2 - Packet Sniffer

The convenience and ease of accessing the Internet nowadays creates a major security vulnerability—by placing a passive receiver in the vicinity of the wireless transmitter, that receiver can obtain a copy of every packet that is transmitted. These packets can contain all kinds of sensitive information, including passwords, social security numbers, trade secrets, and private personal messages. A passive receiver that records a copy of every packet that flies by is called a packet sniffer.

In wired broadcast environments such as Ethernet LANs, the cable access technologies also broadcast packets and are thus, vulnerable to sniffing. A packet sniffer can obtain copies of broadcast packets sent over the LAN.

A malicious attacker who gains access to an institution's access router or access link to the Internet may be able to plant a sniffer that makes a copy of every packet going to/from the organization. Sniffed packets can then be analysed offline for sensitive information.

Packet-sniffing software is freely available on various Web sites and as commercial products. An example is the Wireshark lab which is usually used in networking courses. Because packet sniffers are passive—that is, they do not inject packets into the channel—they are difficult to detect. So, when we send packets into a wireless channel, we must accept the possibility that some bad guy may be recording copies of our packets. As you may have guessed, some of the best defences against packet sniffing involve cryptography.

Annex 3 - IP Spoofing

IP spoofing is the ability to inject packets into the Internet with a false source address. It is surprisingly easy to create a packet with an arbitrary source address, packet content, and destination address and then transmit this hand-crafted packet into the Internet, which will dutifully forward the packet to its destination. Imagine the unsuspecting receiver (e.g. an Internet router) who receives such a packet, takes the (false) source address as being truthful, and then performs some command embedded in the packet's contents (e.g modifies its forwarding table).

To solve this problem, we will need end-point authentication, that is, a mechanism that will allow us to determine with certainty if a message originates from where we think it does.

Annex 4 - Security Tokens (Two-Factor Authentication)

More stringent authentication systems often require evidence from more than one authentication factor. For instance, banks typically issue a device called a security token to users who wish to access their bank accounts online.

To access his or her bank account online, the user has to confirm his or her identity by providing a secret password or personal identification number (PIN), followed by a one-time password (OTP) generated from the security token or a mobile phone that the user owns. This kind of authentication that uses evidence from both something the user knows and something the user owns is called two-factor authentication.

An OTP is usually a string of seemingly random numeric or alphanumeric characters. For security reasons, an OTP is usually valid only for a short period of time, after which a new one has to be requested. This reduces the possibility that the OTP is obtained by an intruder between the time it is generated and when it is accepted by the bank. OTPs may either be generated by the bank and sent securely to the user's device, or generated directly on the device using a secret algorithm that only the bank knows.

Two-factor authentication is stronger than using only a password as it is much more difficult for an intruder to both guess a password and steal the user's security token. This is why it is important to keep the security token in a secure location at all times and to report a missing security token as soon as possible.

However, if an OTP is sent wirelessly to the user's mobile phone, it may be intercepted and used by an intruder during the transmission process. Alternatively, if the secret algorithm used to generate OTPs is poorly chosen or accidentally revealed, an intruder may find out how to generate OTPs without needing the security token at all. Unfortunately, there is not much that a user can do to prevent such intrusion attempts.

References

1. https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences

2. https://docs.oracle.com/cd/E19693-01/819-0997/auto26/index.html

3. Kurose, J., & Ross, K. (2016). Computer Networking: A Top-Down Approach (7th ed.). Pearson.

4. O-level Computing Textbook (3rd ed.). CPDD, MOE

5. https://cheapsslsecurity.com/blog/digital-signature-vs-digital-certificate-the-difference explained/

6. https://www.docusign.com/how-it-works/electronic-signature/digital-signature/digital-signature-faq#:~:text=Digital%20signatures%20are%20like%20electronic,document%20in%20a%20recorded%20transaction.

7. https://medium.com/@Mailfence/digitally-signed-emails-what-is-it-and-how-do-digital signatures-work-a8c789e62c4a

8. https://brilliant.org/wiki/rsa-encryption/