CHAPTER 1: INTRO

1) **Explain sniffing attacks, spoofing and hijacking and describe appropriate countermeasures.**

a. Sniffing attacks (or eavesdropping) concern the action of “sniffing packets”, i.e an attacker (which is not part of the communication or which is neither the sender nor the receiver) is able to read packets which are addressed to another network node. Of course this attack is more trivial if we are dealing with broadcast LAN or if the attacker is at the switching nodes. This attack is possible if the attacker puts a network card in promiscuous mode, which can read every packet passing through the device. This attack is also a consequence of the man-in-the-middle attack.

If the packets are not encrypted, then the attacker can read the packets’ content (data, passwords and so on) and then we have a loss of confidentiality.

Countermeasures:

- encryption of the packets in order to prevent the attacker from reading the packets. However, even if confidentiality is ensures, this countermeasure doesn’t keep the attacker away from having actual access to the packets.
- No use of broadcast networks (rarely possible)
- Protection from the statistical analysis by sending continuous stream and continuous data
- Port mirroring (also known as switched port analyzer) in order to protect the ports of routers and switches: strategy of copying and sending network packets transmitted as input from a port to another port of a monitoring computer/switch/device. When enabled, the traffic that comes to and from a specific port number is automatically forwarded to a monitoring device, which is typically part of the monitoring software or security application that analyzes these data packets.

b. Spoofing attacks concern the forge of a source address, typically we deal with IP address spoofing or level 2 address spoofing (other more sophisticated techniques are DNS or DHCP spoofing). More in detail, in source address spoofing the attacker uses the source address of another host to take its place as a client (and hide his own actions) or a server. It is used in attacks where there’s no need of “answer” packets, and if the attacker is in a broadcast network, he is also capable of reading all the replies. A consequence of this attack could be the unauthorized access to some data resources.
Countermeasures:

- No use of address-based authentication, it’s better to use authentication based on certificates or challenges

c. Hijacking (or data spoofing) is an attack in which data are inserted/modified or deleted during their transmission, as a consequence of an attacker which has successfully taken control of a communication channel. This can be achieved in a logical way (by changing the route of the network, i.e the attacker modifies the TCP packets and succeeds to convince the two nodes that are able to talk to each other, but they should pass through him) or in a physical way (the attacker is placed in the communication channel between two nodes). This attack has obviously some important consequence: phishing, DoS, replay attacks, eavesdropping, chosen ciphertext attacks
Countermeasures:

- Packet authentication, in order to be sure of who has sent some packets across the network
- Ensure integrity of packets (maybe through message integrity code or digests) in order to understand If a packet has been modified
- Serialization of each single network packet. In this way the packets come in the same way as they were sent, and this is important to avoid replay attacks.

2) **Explain what is social engineering, make an example of this attack, indicate how a company can prevent this attack.**
Social engineering is the psychological manipulation of people into performing actions of divulging confidential informations. This strategy exploits the "weaknesses" of people by using certain language and topics (e.g "change immediately your password because it's not secure, here's a suggestion → password sniffing). Here are some examples:
   a. Phishing: attack technique based on attracting a network service user to a fake server, using e-mail for instance, in order to acquire user's sensitive informations such as his authentication credentials or other personal informations. Phishing could also persuade the user to install some plugin or software which is actually a trojan horse (i.e a program that is carrying a malware). Phishing has several variants:
      - Spear phishing, when the message used to convince the user includes several personal data to disguise the fake message as a good one
      - Whaling, when the target user has a relevant position inside a company (e.g the CTO or the CEO). In this case the attack can be much more profitable for the attacker
      - Psychological pressure
   b. Pharming: this term embodies a set of techniques used to re-direct a user towards a shadow server by:
      - Changing the "host" file at the client side
      - Changing the nameserver points at client side
      - Changing the nameservers at a DHCP server or poisoning the cache of a nameserver
   c. Fake email: this technique exploits the ingenuity of the user to convince him to do whatever he reads on the fake mail that has been received.

   Countermeasures:

   - Organizations must, at employee level, establish a trust infrastructure where sensitive data should be treated. Of course this trusted infrastructure must be periodically checked.
   - Organizations need to identify what informations are sensitive and discuss their integrity in all forms
   - Establish security protocols for people who manage sensitive informations
   - Employees must be trained in security protocols relevant to their position in the institution, that is make them also aware of security threats

3) **Describe a phishing attack and what are the measures to limit.**
Phishing attack is an attack performed by social engineering technique. More in details, it consists of attracting a network user to a fake/shadow server through messages or e-mail, with the objective of acquire user's sensitive informations such as authentication credentials. Phishing has several variants:
      - Spear phishing, in which the language of the "bait" message is carefully chosen in order to make it indistinguishable from an authentic message, for example by inserting some personal informations
      - Whaling, when the target user is an important figure inside the company. In this way the attack could be a lot more profitable for the attacker
      - Psychological pressure

   Countermeasures:

   - Individual awareness, i.e being careful to visit non secure sites. In case of request for personal information, account numbers, passwords, or credit card number, never send this kind of information it is good practice to delete, forward a copy to the competent authorities and notify the bank or other interested parties, so that they can take further measures against the fake site.

- Check the browser icons to see if a secure connection (SSL/TLS) with the site has been established. This kind of connection guarantees the confidentiality of data

4) **What is “malware food chain”, explain what it is and give an example?**
Malware food chain, as food chain in nature, is composed by a number of individuals who behave on the basis of taking advantage on someone else, in case of malware we talk about economic benefit. The first step of this process is the discovery of a certain vulnerability of a certain asset (i.e the set of goods, data and people needed for an IT service) by some malicious person. While in the past this discovery used to be exploited just to be in a sort of “hall of fame”, nowadays malicious coders prefer sell the information about the vulnerability or a working code that exploits it on the vulnerability marketplace (people buy such things using bitcoins). These vulnerabilities are purchased by the malware toolkit makers, people who create attack programs that needs malware in it. This kind of software is then used by malware distributors (spammers, web owners etc.)

5) **What is the meaning of the following words? Vulnerability, threat, asset, security control, attack**
   - Asset: the set of goods, data and people needed for an IT service. There are four categories of assets: ICT resources (computers, disks, networks), human resources (employees), location (the asset must be placed in a protected room), data (not the disks but something intangible that could be deleted or modified). Before setting up a defense, we must understand which are the risks (risk estimation). To make a risk estimation is good to start from the service. Once we know the service we need to protect, we must identify which are the assets used to provide that service.
   - A vulnerability is a weakness of a certain asset. Assets are exposed to vulnerabilities and those vulnerabilities increase security risks, which is increased also by all the threats that exploit the vulnerabilities.
   - A threat is the deliberate event that can produce the loss of a security property. A threat increases the security risks by exploiting asset vulnerabilities.
   - Security control is an element inside the system that protects against one specific threat and it reduces the security risks to which the system is exposed to. The security risks indicate the security requirements (given the risk we know what are the security requirements) and security requirements are met by security control
   - An attack is the occurrence of a threat.

6) **Explain what is a DOS attack, how they are performed and how to defend against them.**
A DoS attack (Denial-of-Service) is an attack with the aim to keep a service busy for a certain amount of time (for example a web server which is not able to reply to clients’ requests) so that it can’t provide its services. A DoS attack is implemented by saturating the available bandwidth in order to slow the service until it stops working. Examples:
   - Mail/log saturation: for example for a call to a competition, offers can be sent until a date. We can make an offer and stop all the others from sending e-mails. A possible solution is to send tons of email keeping the server busy
   - Ping flooding: the attacker sends a huge amount of ICMP echo requests, keeping the target busy and not able to perform other tasks
   - SYN flooding: an attacker rapidly initiates a connection with a server but without sending ACK packets during the TCP handshake, i.e without finalizing the connection. The malicious client can either simply not send the expected ACK, or by spoofing the source IP address in the SYN, cause the server to send the SYN-ACK to a falsified IP address, which will not send

an ACK because it “knows” that is never sent an ACK. In this way the connection remains “half opened”. The server spends resources waiting for half-opened connections, which can consume enough resources to make the system unresponsive to legitimate traffic

These kind of attacks are difficult to detect because the lack of services could be caused by a huge number of legitimate users trying to connect to the server (for example it’s very common that an online platform which sends concert tickets goes down the first day of sale). Furthermore there are not specific countermeasures against these attack, however we could develop some strategies:

- For SYN flooding we can place a router as SYN interceptor between the client and the server. This router transfers the channel to the server if the handshake with client completes successfully. We can also use a SYN monitor, which kills the pending connection requests
- For ICMP flooding we can place a firewall in the network which limits the ICMP traffic, but this solution could prevent some legitimate users to exchange ICMP traffic

Nevertheless the first countermeasure that comes in mind for almost every security threat is to perform authentication

7) **What is Stuxnet? How does it work and what can we learn from it?**
Stuxnet was an important cyber-attack launched by united states and Israel against Iranian nuclear program in 2010. In a way it was revolutionary because was one of the first attacks with cyber-physical systems as target, i.e it was developed to causa a physical damage.
Stuxnet has three modules:
   1) A worm that contains 4 attack vectors: one known vulnerability with patch available, one with no patch available and two “zero day” vulnerabilities.
   2) A link file that automatically executes the worm and makes it propagate among the systems
   3) A rootkit component responsible for hiding all malicious files and processes, to prevent detection of Stuxnet.

Stuxnet was introduced in the target environment via an infected USB flash drive, thus crossing any air gap. The worm then started propagating across the network, scanning for Siemens Step7 software on computers controlling PLC (programmable logic controllers, which allow the automation of electomechanical processes such as those used to control machinery and industrial processes including gas centrifuges for separating nuclear material). Stuxnet infected PLCs by subverting the Step7 software application that was used to reprogram these devices.

For its targets, Stuxnet contained also a code for a man-in-the-middle attack that faked industrial process control sensor signals so that an infected system doesn’t shut down due to detection of an abnormal behavior. The Stuxnet worm consists of a layered attack against three different systems:

1) The windows operating system of the target computers, which has been infected by USB flash drives (which contain windows shortcut files to initiate executable malicious code). The malware has both user and kernel mode under Windows, thanks to the fact that these digital drives have been digitally signed with the private keys of two public key certificates that were stolen from separate companies.
2) Siemens industrial software applications that run on Windows: Stuxnet infected project files belonging to SCADA control systems, making the malware able to install itself on PLC devices unnoticed
3) PLCs: Stuxnet attacked only those PLCs with a specific motor frequency, in order to damage gas centrifuges used to extract nuclear material.

There are some important things that we can learn from Stuxnet:

- Systems protected with physical separation (air gap) should also be protected with other standard protections, such as anti-virus, patches, firewalls.
- There are often many unnecessary active services running on machines, which could facilitate the propagation of malwares.
- A good idea is to develop a list of legitimate software to be installed

8) What is a window of exposure and how can it be reduced?

A window of exposure is the time interval between the discovery of a security vulnerability and the installation of the patch against that vulnerability. There are several steps:

a. Discovery: at the very beginning there's a low level of security risks. At some point a new vulnerability is discovered and risk goes higher until this vulnerability is made public
b. Publication: the vulnerability is made public. At this point three categories are informed: the bad guys (that will eventually perform an attack), the good guys, that try to protect the system, and the vendor, who will eventually inform his customers of this vulnerability. While the vendor is working to fix the vulnerability, users of that product should not try to fix it, but they should update their security tools at least to detect if the vulnerability has been used actively for an attack.
c. Protection: a patch is finally published by the vendor, but until it is not widespread the risk cannot go lower

The window of exposure can be reduced by performing periodic security check and by trying to develop a patch for a certain vulnerability before the publication phase, i.e before it is made public.

9) Explain what is the principle "defense-in depth", why it is needed in ICT systems and make a concrete example of its implementation.

"defense-in-depth" means that a security system has a double line of defense. Namely, if an attacker is able to defeat the first line, there must be a second line to stop the attacker. It is better to have more defense types, so if the attackers get in it they will get always harder to keep get in. We can find an application of this concept in firewalls, in particular in the dual-homed gateway architecture.

In this architecture we have a screening router which acts as a packet filter connected to the external (insecure) network. There's also a dual-homed gateway, i.e a system with two network cards which acts as a bridge (basically it is the front end of the whole internal network) and is connected to both the external and the internal network. In this way, once a packet is permitted by the screening router, it doesn't go directly inside the internal network but has to be checked also by the gateway. Hence a double-line of defense, i.e defense-in-depth is provided.

10) What is integrity? Mention some possible attacks related to it

Integrity is one of the main security properties. Integrity means that if data have been modified we are able to detect it. However it doesn't mean that we are able to avoid the modification of data, but still it is important to recognize If a message has been modified or not. Integrity concerns also the detection of deleted data, although this mechanism is more complex than detecting modification of data. The main attack related to integrity is the replay attack: data sent on the network are encrypted and no more modifiable, but an attacker could record the message sent to the network and replay it multiple times, thus passing authentication. A countermeasure to this attack could be serializing and enumerating the sent packets. Another attack that targets integrity is connection hijacking/data spoofing, in which the attacker can place himself inside the network channel, thus intercepting and eventually modify (if he is an active attacker) the packets. A countermeasure to this attack is authentication.

11) **Bruce Schneier wrote "security is a process, not a product", explain what does he mean with this statement and make an example of its implementation.**

“Security is a process, not a product” means that, since computer security flaws are inevitable, we mustn’t rely just on products, because they provide some protection, but the only way to achieve a good level (100% is not possible) in an insecure world is to put processes in place that recognize the inherent insecurity in the products. The strategy is to reduce the risk of exposure regardless of the products or patches.

To make this thing possible it is needed to follow some security principles:

- Security in depth: we have a double line of defense, that is if an attacker can beat a security product then he will face another line of defense
- Security by default: security has to be mandatory in the development of a service, users should not have the choice to activate security or not → security should be on by default
- Least privilege: any element which is operating inside the system should be assigned with the minimum amount of privileges that permits it to perform its tasks
- Security by design: security should be put inside the development process of a software, not at the end
- Need-to-know: we should give to any component of the system access only to data that are required to execute a certain task.

An example is the firewall: it cannot be bought but must be designed by assembling its components (which you buy) in the right way in order to obtain the suitable architecture for protecting a certain system

CHAPTER 2: BASICS

1) **Describe what H-MAC is indicating its merits and drawbacks**

H-MAC is a keyed digest technique used to provide message authentication, i.e is a hardly invertible algorithm. It overcomes the standard keyed-digest in which we have md = h(x||k) or md = h(k||x) [x is the original message, md is the message digest, h is the hardly invertible hash function and k is the key. Respectively they are secret suffix MAC and secret prefix MAC], because H-MAC consists of two hashes (one inner and one outer hash) performed on the message. More in details:

- Let B be the block length (bytes), L the output length (bytes), with B > L
- Let K be the key used in the algorithm. K should never be smaller than L. if |K| > B then we define another key K' = H(B) [H is the hash function], otherwise K' = K. If |K'| < B then it is padded with null bytes up to B
- Let ipad = 0x36 repeated B times and opad = x05c repeated B times

Having said that, the H-MAC is computed in this way:

HMAC(M) = H((K' XOR opad) || H(K' XOR ipad || M)).

Advantages:

- HMAC is the best solution to the possible calculation errors of the standard keyed-digest due to the hash calculation (secret prefix and secret suffix), then it also prevents the attack against standard keyed-digest
- HMAC is more resistant because it applies the hash 2 times: with a double digest it compensates for the loss of safety resulting from the reduction of the size of the digest

Drawbacks:

- Since it applies 2 times the hash function, HMAC is necessarily slower than keyed-digest

2) **Describe the properties of authentication and integrity H-MAC in contrast with the electronic signature**

Both HMAC and digital signature provide authentication and integrity, but they are two different techniques in terms of implementation.

HMAC is a keyed-digest technique which consists of applying a hash function 2 times to the original message. In particular HMAC(M) = H((K' XOR opad) || H(K' XOR ipad ||M)) where ipad = 0x36 and opad = 0x5c and K' = H(K) if K (i.e the key) is less than B (message block length).

Meanwhile digital signature is the encryption of a message by using asymmetric cryptography: the message is signed i.e encrypted with the private key (actually the digest is encrypted). The message is then decrypted by the receiver using the public key of the sender. Digital signature offers in addition non repudiation (not guaranteed by HMAC), since it uses public certificates for the keys.

On the other hand, the generation of a keyed-digest (such as HMAC) is based on the knowledge of a shared secret between the parties, hence it can only guarantee integrity and authentication.

3) **What do the following acronyms mean : DSA, RSA, RC-4, RIPEMD? Briefly describe them**

1) DSA: digital signature algorithm. It is one of the main asymmetric cryptography algorithms (together with RSA). It exploits the discrete logarithm problem in order to encrypt asymmetrically the data with the private key of the sender. The good guy has the base and the exponent hence it can obtain the key, while the bad guy has to find the logarithm. The problem of DSA is that it can be used only to perform digital signature, because inside the algorithm there's a one way lossy compression function, so original plaintext cannot be recovered.

2) RSA: Rivest-Shamir-Adleman: asymmetric algorithm based on integer-factorization scheme. It provides secrecy and digital signature. In this algorithm the good guy has to perform multiplication of 2 prime numbers, while the bad guy has to find the two factor. More in detail:
    a. Public key: starting from two prime numbers p,q we obtain n = pq (p and q are secret). The public key is e such that e is relative prime to (p-1)(q-1)
    b. Private key d = (e^-1)mod((p-1)(q-1))
    c. Encryption Enc(m) → c = (m^e)modn
    d. Decryption Dec(c) → m = (c^d)modn
3) RC4: Rivest's code. stream symmetric algorithm with variable key length and that can work with different types of data. The problem of RC4 is that it is secret, so it provides security-through-obscurity. This could be a problem because if the algorithm is unknown then it is difficult to improve it and correct its bugs.
4) RIPEMD = RACE Integrity Primitives Evaluation Message Digest: cryptographic hash algorithm with block of 512 bits.

4) Differences between Stream and block algorithm

Both are symmetric ciphers.

Stream ciphers encrypt the plaintext one bit at a time (actually in practical implementations they encrypt one byte at a time). Some stream algorithms are RC4, Salsa20 and ChaCha20. Stream ciphers are faster than block ciphers, especially on hardware implementations, because they require basic binary operations (such as ARX, addition, rotation, xor). Furthermore, stream ciphers provide confusion, i.e if an attacker is able to change one bit of the ciphertext then he cannot recover the key → confusion makes hard to find a relation between the key and the ciphertext. In these algorithm the sender and the receiver share a secred seed, which is used to generate the keystream used to encrypt the plaintext.

Moreover, stream ciphers are based on the Vernan cipher, i.e on the logical XOR operation. In conclusion, the operation modes used in stream ciphers are CFB (cipher feedback) and OFB (output feedback)

Block ciphers, on the other side, encrypt the plaintext one block of fixed size at a time and each block is encrypted with the same key. For example AES uses blocks of 16 bytes. Block ciphers are better for software implementations and provide both confusion and diffusion (diffusion means that if an attacker changes one bit of the ciphertext than it is difficult to recover the plaintext → diffusion makes hard to find the relation between plaintext and ciphertext). Block ciphers are based on the Feistel networks, in which there are permutations of the block of plaintext before and after the encryption process. The operation modes used in block ciphers are ECB (electronic code block) and CBC (cipher block chaining). The latters are useful in order to avoid replay attacks and to prevent an attacker to find the plaintext from the ciphertext without knowing the key

5) Two companies have to exchange files using a password of max 12 characters : indicate how the system should be designed to perform such exchange

There are two main problems:

1) How to generate a strong and secure key, since 12 characters are not enough for ensure security.

    A key is generally generated by a KDF (Key derivation function), which uses cryptographic hash functions. We define a random key as k = KDF(P, S, I) where:

    - P is the password
    - S is the salt (increases the difficulty to guess P), it's like an IV

- I is the number of iterations performed by the function

The function KDF Is based on cryptographic hash functions such as:

- PBKDF2, which uses SHA-1, |S| >= 64bit and I >= 1000.
  In this function the key is obtained in this way: K = PBKDF2(PRF, P, S, I, KLEN) where PRF is a pseudo-random function (such as SHA-1) and KLEN is the desired key length
- HKDF, which uses HMAC

2) Since the two companies have to share the secret key, the problem is how to exchange it between the two parties
   Basically there are two methods to exchange securely the symmetric key just generated from the KDF:
   1) A VPN is set-up between the gateways of the companies using IPsec in tunnel mode so that a secure channel is created. After the end points have negotiated policies and the cipher suite they can exchange the key and proceed to share files.
   2) The two parties can exchange the key using asymmetric cryptography, in which the public key is authenticated by public key certificates. In this protocol confidentiality without shared secrets is used to send the secret key chosen for a symmetric algorithm. The sender encrypts the secret key with its public key and the receiver decrypts it with the private key. In order to decide in a democratic way who has to decide the key there's the Diffie-Hellman algorithm, that solves the key agreement problem

6) **A message of 28 bytes is encrypted with DES-CBC : how many bytes will be sent in total?**
   DES uses blocks of 8 bytes so a padding of 4 bytes is required → 32 bytes of ciphertext. In addition an initialization vector IV of 8 bytes (block size) is required, then 40 bytes will be sent
7) **A message of 84 byte is encrypted with DES-CBC : how many bytes will be sent in total?**
   #blocks = 84/8 = 10 with remainder of 4 (padding) → 88 bytes + 8 bytes (IV) = 96 bytes in total
8) **A 1500 bytes file is encrypted with 3DES-CBC : what will be the size after the encryption?**
   3DES uses blocks of 8 bytes, then we have 1500/8 = 187 blocks with remainder of 4 (padding) → 1504+8 = 1512 bytes in total
9) **Two peers want to exchange some documents in ASCII format. Advice about what operations it is recommended to perform on this documents to make them attack resistant on the network. Assume that this documents do not contain sensible information. Write down the solution as a formula using appropriate terms**
   Since the documents don't contain sensible information, we may don't worry about confidentiality, i.e encryption. In this case the the security properties we have to guarantee are authentication, integrity and non-repudiation:
   - A computes the keyed-digest of the message in order to provide authentication and integrity, using for example SHA-256: md = HMAC_SHA-256(K, P ||<name_of_algorithm>) where k is the secret key shared by the two parties (→authentication) and P Is the message. The keyed-digest is appended to the original message as a tag.
   - A sends the message concatenated with the tag: P||md
   - B receives the message and checks its integrity by computing the keyed-digest using the shared secret → md'
   - If md == md' then the message has not been modified during the transmission

10) **A wants to send a secret and authenticated message to B and C, they have each one a RSA key pair. Describe the data block format for sending the message addressing the requested security properties**

In this case we need confidentiality (i.e encryption of the message), authentication and integrity (i.e digital signature) If the message length L is not too big (i.e L < (RSA_keyLen/8 - 11 bytes)), then the message could be encrypted using asymmetric cryptography → A can encrypt the message with B's and C's public keys and send it along with its digital signature:

- A computes Cb = ENC_RSA-2048(KpubB, M)
- A computes Cc = ENC_RSA-2048(KpubC, M)
- A computes the digital signature over the digest of the message: S = ENC_RSA-2048(KprA, SHA-256(M))
- A sends (Cb || S) to B and (Cc || S) to C
- B computes M = DEC_RSA-2048(KprB, Cb)
- B computes V = DEC_RSA-2048(KpubA, SHA-256(M))
- B checks: if V == S then everything went good, otherwise the message has been modified
- C computes M = DEC_RSA-2048(KprC, Cc)
- C computes P = DEC_RSA-2048(KpubA, SHA-256(M))
- C checks: if P == S then everything went good

If the message length is big we need symmetric encryption instead of asymmetric encryption. Since for symmetric encryption we need a secret key for each pair, in this case we need n(n-1)/2 = 3 symmetric keys Kab, Kbc, Kac. They can be exchanged securely between the parties using asymmetric cryptography: Let's see how it works between A and be (the same will be for B and C and for A and C)

- A chooses its key (generated with a KDF) and encrypts it with its public key: Cab = ENC_RSA-2048(KpubA, Kab)
- A sends the encrypted symmetric key to B
- B decrypts it using its private key: Kab = DEC_RSA-2048(KprB, Cab)
- Now that A and B have their shared symmetric key they can start exchange messages encrypting them with the symmetric key, using for example AES-128 in CBC_MODE (i.e the key size must be 32 bytes): C = ENC_AES-128_CBC-MODE(Kab, M)
- the scheme for digital signature is the same as before

11) **Speak about keyed digest mechanism : what is it? How is it used? What are its advantages and drawbacks?**

A normal digest is not enough to ensure integrity, in fact the attacker could change both the original message and the digest (which is appended as a tag to the original message) keeping its modification un-detectable. A keyed digest is a digest computed not only on the message but also on a secret key, shared among the parties. Kd = H(K, M) where H is the used hash function, K is the secret key and M is the original message. In this way only who has the secret key can verify the received digest. In this way we achieve not only integrity but also authentication. To protect against length extension attacks the digest is computed as follows: kd = H(K || M ||K). to obtain more protection we can also add the message length to the digest. In this way it's impossible to change the message adding bytes at the end or at the beginning. There are also several sophisticated techniques such as HMAC, which uses a double hash, and CBC-MAC, which provides also confidentiality.

Advantages: fast to compute because it is based on a hash function called once (a hash function is fast by definition); there are few additional data (often 160 bits) needed to be transmitted along with the message

Disadvantages: if implemented in the wrong way it could led to length extension attacks

12) **Keyed-digest and digital signature : what properties do they offer? Which are their advantages and disadvantages?**

They both provide integrity and authentication. Keyed digest is based on symmetric cryptography: kd = H(K || M || K), while digital signature is based on public key cryptography, i.e using RSA or DSA algorithm. The message digest is signed, i.e encrypted with the private key of the sender: ds = ENC_RSA-2048(Kpr, H(M)) and the verification is made using its public key: v = DEC_RSA-2048(Kpub, H(M)). Since it uses asymmetric cryptography, DS is slower than keyed-digest because for example in RSA there's the search of two large prime numbers, and in addition everyone can check the authenticity of the message because of the public key. However, if digital signature is associated with a public key certificate released by a certification authority, then it provides also non repudiation. On the other hand, since keyed digest uses symmetric cryptography, only who has the secret key can check the validity of the message.

13) **Why SHA-1 is deprecated? What's a valid alternative? Motivate why the alternative hash function proposed is better than SHA-1**

SHA-1 is a hash algorithm that works on blocks of 512 bits and produces a digest of 160 bits. This algorithm is deprecated because of the birthday paradox: using a brute force attack, an attacker can guess the hash value in at most 2^160 trials, which is unfeasible. However, due to the birthday paradox, the actual number of trials in order to find a collision, i.e two different messages that have the same hash value, is 2^N/2 where N is the length of the hash value. Hence, for a security level of x bit, the hash function needs to have an output length of 2x bit. Having said that, a collision in SHA-1 is found in 2^40 trials, which is feasible for a computer. The first alternative to SHA-1 was the SHA-2 family, which produce a longer digest (from 224 to 512 bit). However nowadays the most used hash algorithm is the SHA-3 family, which produces digest from 244 to 512 bit:

We have SHA3-224, SHA3-256, SHA3-384 and SHA3-512 and more.

14) **Sort from the fastest to the slowest one the following algorithms : 3DES,DES,AES,RC4**

RC4, AES, DES, 3DES

15) **Sort from the slowest to the fastest one the following algorithms : SHA,AESECB,AES-CBC,RC4**

AES-CBC, AES-ECB, RC4, SHA

16) **Describe elliptic curve cryptography**

Elliptic curve cryptography is an asymmetric cryptography technique which is based on a curve on a 2D surface. The main advantage of such a technique is that it reaches the same security level of other cryptographic algorithms using shorter keys. More in detail:

Instead of modular arithmetic (like in RSA or DH), the operations are executed on the surface of a 2D curve, so we are talking about points on this curve. Then we perform the discrete logarithm problem on the curve. Elliptic curve cryptography can be used for digital signature (ECDSA), key agreement (ECDH), authenticated key agreement (ECMQV) and key distribution (EC integrated encryption scheme).

Definition: The elliptic curve over Zp (p > 3) is the set of all pairs (x, y) in Zp such that y^2 = x^3 + ax + b mod p, with a,b in Zp, and such that 4a^3 + 27b^2 != 0 modp.

On this curve we can compute addition of two points and multiplication of a point by a scalar

Example: R = (x, y) = P + Q where P, Q belong to a certain elliptic curve E

X = l^2 – xP – xQ and y = l(xP – x) – yP where l = (yP-yQ)/(xP-xQ) if P != Q and l = (3xP^2 + a)/2yP if P = Q.

The EC-Diffie Hellman works this way: A and B select the same elliptic curve and a point G of its, then A chooses a random value x and computes X = xG. Meanwhile B chooses a random value y and computes Y = yG → A and B don't agree on the key but on the curve.

A and B exchange X and Y, a computes K = xY and B computes K' = yX but K = K' = xyG

Note that we used scalar multiplication instead of exponentiation, but still this is not a trivial job for an attacker who has to find the keys.

17) **Describe CBC-MAC algorithm**

CBC-MAC (cipher block chaining message authentication code) is a technique for constructing a message authentication code from a block cipher.
CBC-MAC exploits a block-oriented symmetric encryption algorithm in CBC mode with null IV, taking as MAC the last encrypted block. More in details:

- The message M is splitted in N blocks M1, …, Mn
- Iterations → V0 = 0
    - For (k = 1; k < N; k++) do { Vk = ENC(K, Mk xor Vk-1)
    - CBC-MAC = Vn

DES based CBC-MAC is the Data Authentication Algorithm. This technique is secure only for message of fixed length.

This interdependence created by CBC-MAC ensures that a change to any of the plaintext bits whill cause the final encrypted block to change in a way that cannot be predicted without knowing the key of the block cipher.

18) **Describe one of the first methods used for encrypting data**

One of the first encryption algorithm was DES. DES is a symmetric block cipher which works on blocks of 8 bytes and uses a 64-bit key. Actually the key is just 56 bit long because every 7 bits there's one parity bit. DES is really fast (it works very well especially in hardware applications) because it uses basic operations such as XOR, shift and bit permutations (it is based on the feistel network, in which the plaintext bits are permutated before and after the encryption). More in detail:

- First of all the 64 bit blocks and the 56-bit key are permuted
- There are 16 left circular key shift and 16 key permutations
- There are 16 permutation key rounds in which the key encrypts the permuted plaintext
- There's a final permutation

Nowadays DES is deprecated because the key length is too small and a brute force attack is really feasible

19) **Indicate what can be done to attack the integrity of a message protected with MAC computed as MAC=H(K||D)**

If MAC is computed as MAC = H(K || M) then it is called secret prefix MAC because we have the key concatenated with the original message. For the attack we assume that the cryptographic checksum m is computed hashing the original message x, which is composed by n blocks → x = x1x2…xn. Then the MAC is computed as m = MACk(x) = h(k || x1,x2,…,xn). The problem is that if the attacker adds an arbitrary block xn+1 the message becomes x' = x1,x2,…xn,xn+1, the digest can be constructed without knowing the secret key → m' = h(m||xn+1). As a consequence, the receiver will accept the message with the additional block (which could cause some damages) as valid, even though the sender only authenticated x1…xn. This attack is possible since the MAC of additional message block only needs the previous hash output, which is equal to the sender's digest.

20) **Advantages and disadvantages of ciphering a text with enc(K2, enc(K1, P)) where K1 and K2 are different keys with the same length**

This encryption scheme presents no advantages and it's deprecated (an example is the lack of double DES in the transition from DES to 3DES). In fact, encryption with two different keys is the same as a single encryption with another key: Enc(k2, Enc(k1, P)) = Enc(k3, P). Suppose that we have the scheme explained above with two different keys of n bits. A naïve brute force attack could be trying all the possibilities for both the keys, resulting in 2^2n trials. However, there's the meet-in-the-middle attack which reduces the complexity of this computation to just 2^(n+1). Meet-in-the middle attack is based on the following hypothesis:

1) N bit keys
2) The attacker knows P and C such that C = enc(k2, enc(k1, P)). Hence should exist an intermediate artifact M such that Enc(k1, P) = M and Dec(k2, C) = M.

The attacker computes all the 2^n ciphertexts Yi = Enc(Ki, P), then he computes all the 2^n decrypted ciphertexts Yj = Dec(Kj, C) (table computation phase), then he search those values Ki and Kj such that Yj = Yi (matching phase). The attacker has found the two keys.

21) **Describe the ECB mode of operation identifying advantages and drawbacks**

ECB (electronic code block) is the easiest implementation mode for block ciphers. Each block is encrypted with the same key, i.e if two blocks of plaintext are equal then also the two related ciphertext blocks are equal.

Advantages:

- Easy to implement
- There is no security risk in encrypting multiple messages with the same key. In fact, each block can be looked as a separate message encrypted with the same key
- Bit errors in the ciphertext, when decrypted, will cause the entire plaintext block to decrypt incorrectly but will not affect the rest of the plaintext

Disadvantages:

- Since each plaintext block will always produce the same ciphertext block, an attacker who knows the plaintext knows that, whatever is the plaintext's position, it will always generate the same ciphertext. Hence the attacker could pre-compile all the combinations of one specific known plaintext (for example the header or the footer, the so called stereotypes beginnings and endings, in which often there are precious informations).
- ECB is not recommended for long messages, because if the attacker is intercepting the ciphertext and wants to exchange the position of two blocks (block swapping), then the receiver will not detect this manipulation, i.e the attacker can change data inside the encryption.
- If the encrypted message has a lot of redundancies, and these tend to show up in the same places in different messages, a cryptanalyst can get a lot of information through statistical attacks.

22) **Which one of the following techniques is better for protecting integrity of a 1 Gbps flow? Digital signature, symmetric encryption, keyed-digest.**

Since the task is to provide integrity, then symmetric encryption is not required. The best technique is keyed-digest because we have a lot of data and this technique, which exploits symmetric cryptography, is faster than digital signature, which uses asymmetric cryptography

23) **Which one of the following techniques is better for protecting data from replay? Digital signature, symmetric encryption, keyed-digest, others?**

Since replay attack concerns integrity of the message, symmetric encryption is not required. The best solution is to perform a keyed-digest with the addition of a MID (message identifier) in order to detect if a packet has been duplicated or not (duplicated packets would eventually have the same MID). An aleternative to MID is including the timestamp or the sequence number of the packet in the digest. Assuming that the secret key has been already negotiated between the two peers (e.g with ECDH key agreement), the sender computes the digest d = HMAC_SHA-256(K,P,TimeStamp) and sends it along with the message.

24) **If you want to encrypt with AES a password of 12 characters with 2 bytes salt, it is better to use ECB, CBC or CTR?**

ECB is not recommended because it is not secure (the same plaintext results in the same ciphertext). The size of the data is 12 + 2 = 14 bytes < AES.block_size (which is 16 bytes), so the best way to encrypt the password is using CTR (counter mode), which doesn't require padding (in fact this method is suggested if the data to be encrypted are smaller than the block size).

CTR uses a block cipher to cipher N bits at a time. We have a register with the same size as a block initially filled with a nonce and a counter. Since the size is exactly one block we can directly encrypt it with a key, then we take the leftmost group (1, byte, 2 bytes, whatever) and xor it with the plaintext, obtaining the corresponding ciphertext group. After each block encryption, the counter increments by one. The ciphertext can be then sent through the network. The receiver must have another register initialized in the same way and it has just to apply again the xor to the leftmost group of the ciphertext, obtaining the plaintext.

25) **What's the size of a digital signature created on a 10 Mb file with SHA-256 and RSA-2048?**

SHA-256 produces a digest of 256 bits, that must be padded up to 2048 bits for being encrypted with RSA. The output size of RSA is always equal to its modulus size. Hence the total size is 2048 bits.

26) **How can two peers that share a key exchange a file?**

Since the two peers share a secret key, they can exchange file using symmetric encryption in order to provide confidentiality. The most used symmetric algorithm is AES-128-CBC, which encrypts data divided in 16 byte blocks and uses a 16 byte key. However confidentiality is not enough because if a file is exchanged between two peers also integrity and authentication are required. For this purpose HMAC with a chosen algorithm such that the digest size is the double of the key length used by the encryption algorithm to have the cryptographic system balanced. Hence, is AES-128-CBC is used, then HMAC-SHA256 must be used. If the key is shared only between the two parties, this guarantees also authentication.

- The sender computes C = ENC_AES-128-CBC(k, M)
- The sender computes S = HMAC_SHA256(k,M)
- The sender sends C || S
- The receiver decrypts: M = DEC_AES-128-CBC(k,C)
- The receiver checks: S' = HMAC_SHA256(k,M) → if S' == S' then everything went well

Another way could be using asymmetric cryptography to perform a digital signature, in order to obtain also non repudiation. However, in this case a shared key is not enough, the two parties must have a pair (pK, sK) with pK = public key, sK = private key.

27) **What's the size of a digital signature created on a 10 Mb file with SHA-1 and RSA-2048?**

Since SHA-1 produces a digest of 160 bits, which has to be padded up to 2048 bits in order to be encrypted with RSA, the size will be 2048 bits because RSA output is equal to its modulus size.

28) **Which algorithms can be used to exchange secret keys?**

In general secret keys are exchanged using asymmetric cryptography because keys are smaller than normal data, then asymmetric cryptography is good. Since the public key is known, it has to be distributed by means of a public key certificate, released by a certification authority.

There are several techniques to exchange symmetric keys using asymmetric cryptography:

- RSA: A and B have two pairs (pk, sk) = (public key, private key)
    - The public keys pkA, pkB are exchanged between A and B
    - A computes C = ENC_RSA-2048(pkB, K)
    - A computes the digital signature S to provide integrity and authentication, S = ENC_RSA-2048(skA, HMAC_SHA256(K))
    - A sends to B C || S

    - B computes K = DEC_RSA-2048(skB,C)
    - B computes S' = DEC_RSA-2048(pkA, S)
    - If S' == S then OK
- Diffie-Hellman: exploits the discrete logarithm problem
    - A and B choose two public integers p (prime, large) and g (the generator) with 1 < g < p
    - A chooses an integer x and computes X = g^x modp (x is the private key)
    - B chooses an integer y and computes Y = g^y modp (y is the private key)
    - A and B exchange X and Y
    - A computes Ka = xY
    - B computes Kb = yX
    - The public key is Kab = Ka = Kb = g^(xy)modp
- ECDH (elliptic curve Diffie Hellman)
    - A and B agree on a certain elliptic curve E and a point G which is on the curve
    - A chooses a number x and computes X = xG
    - B chooses a number y and computes Y = yG
    - A and B exchange X and Y
    - A computes Ka = xY
    - B computes Kb = yX
    - The public key is Kab = Ka = Kb = xyG

29) How can you use a hash function together with a seed to generate a key?

Cryptographic hash functions could be used to create a key, because a cryptographic key must be random and typically users use password (not random, even guessable) instead of keys.
We define a random key as K = KDF(P, S, I) where:

- P is the password
- S is the seed (increases the randomness), it's like a IV
- I is the number of iteration the function has to do
- KDF is the key derivation function. It's based upon different hash functions:
    - PKDF2, which uses SHA-1, |S| >= 64 bit, I >= 1000 (limited to key <= 160 bit)
    - HKDF, which uses HMAC

30) For which application has CTS mode of operation being developed?

CTS (ciphertext stealing) permits to use block ciphers without padding. it works in the following way:

- the last (partial) block of the plaintext is filled with bytes taken from the second-to-last ciphertext block (tail) → the bits of the tail are encrypted twice because they were part of the encryption of the second-to-last block
- these bytes are then removed from the second-to-last block
- after encryption the last and the second-to-last block positions are swapped

advantages: we can avoid padding, this technique is useful when we cannot increase the size of the data after encryption

disadvantages: we need more computational time

31) Compute how many data have to be stored on disk to decrypt and authenticate a 1 kB file protected with digest SHA-1 and encrypted with RSA-1024

It is not possible to encrypt 1kB file with RSA because RSA-1024 can encrypt a maximum size of 1024/8 - 11 = 117 bytes (11 bytes standard PKCS#7 padding). However RSA can be used to encrypt a symmetric key for symmetrically encrypting the file with for example AES-128-CBC-CTS.

Moreover, the SHA-1 digest can be encrypted with the sender private key so that becomes a digital signature. In this case to perform decryption, on the disk should be stored:

- 1kBytes space for the future plaintext
- 1kBytes for ciphertext (CTS mode guarantees that ciphertext and plaintext have the same size)
- 1024 bits for the sender's public key, needed to decrypt the digital signature
- 1024 bits for the receiver's private key, needed to decrypt the symmetric key
- 16 bytes for the AES symmetric key
- 16 bytes for the IV
- 1024 bits for the digital signature (RSA output has the size of the modulus)
- 160 bits for the SHA-1 digest to be computed
- 160 bits for the decrypted digest

TOT = 1024 + 1024 + 128 + 128 + 16 + 16 + 128 + 20 + 20 = 2504 Bytes

32) Explain what security through obscurity means and its advantages and disadvantages

Security through obscurity means that the algorithm used to encrypt some kind of data remains hidden, so that its implementation is not known. In this way an attacker who wants to break a system has first to find out how it works, but once he has discovered it he could be able to find vulnerabilities. An example is the stream cipher RC4. This concept doesn't have a real advantage even if it seems the better way to achieve security, because if the algorithm details are not known, then it will be extremely difficult to improve it and eventually fix its bugs. In fact security is achieved not by keeping the algorithm secret (on the contrary, it is better if the algorithm is world wide known), but keeping the key secret.

33) Explain what authenticated encryption is

Authenticated encryption is a technique exploited to provide both confidentiality (encryption) and integrity. While at the beginning the two operations were splitted (authentication and encryption, authenticate then encrypt, encrypt then authenticate, using two different keys), authenticated encryption consists of one algorithm which uses one single key. This solution is obviously faster than the previous because there's just one algorithm. Authenticated encryption is needed by applications such as e-mail messages and network packets. However in network we cannot encrypt the whole packet because in this way routers can't know the destination (placed in the packet header), so it's possible to encrypt just the payload. The attacker can exploit error messages in order to take informations about data, thus the normal encryption modes are subject to chosen-ciphertext attacks when used online (the attacker modifies the ciphertext, then observes if the receiver signals an error or not, using for example a padding oracle).

A way to solve this problem Is Authenticated encryption with associated data (AEAD), which makes a distinction between the part which needs privacy (payload) and the part which needs integrity (header, associated data). It's z variant of AE that allows a recipient to check the integrity of both the encrypted and unencrypted information in a message. Ciphertext is placed inside the packet as encrypted payload, and inside it there's one part that provides integrity not only for the payload but also for the header (tag). In front of the ciphertext Is then placed a the nonce used in encryption and the header.
Some examples of authenticated encryption:

- GCM (galois counter mode): like the normal counter mode, blocks are numbered sequentially and this block number is combined with an initialization vector IV and encrypted with a block cipher (usually AES). The result of this encryption is then XORed with the plaintext to produce the ciphertext. It computes the Ciphertext and the Tag in the following way: (C, T) = algo_GCM_enc(K, IV, P, A) where IV is typically 96 bits, K, is the key, P is the plaintext and A is the associated data (up to 2^64 bits). The tag has a size up to 128 bits.

GCM is the most popular technique because it's online, single-pass and parallelizable. It is used by TLS

- CCM (counter mode with cbc-mac): this mode of operation is suitable with 128-bit blocks. It consists of an authentication tag (plaintext + associated data) computed by CBC-MAC, then the plaintext and the tag are separately encrypted in CTR mode. Hence CCM-MAC combines the CBC-MAC and CTR. These two primitives are applied in an "authenticate-then-encrypt" manner. The main insight is that the same encryption key can be used for both, provided that the counter values used in the encryption do not collide with the pre-initialization vector used in authentication. CCM is slower than GCM

Other examples are OCB 2.0 (the fastest one, online single pass AEAD), EAX (online double-pass AEAD, slow but small)

34) Someone sent to you a file digitally signed with RSA and SHA-..., along with the signer certificate and with the certificate of his CA. Describe how validate the signature, identify the information requested and gain access to them

There are several steps to do in order to validate the signature:

- Check if the certification authority in the signer's certificate is a Top-Level CA or at least is certified by one of them
- Check if the signer's certificate is still valid and has not been revoked
- Recover the signer's public key Kpub from the digital certificate.
- Recover from the digital certificate the hash algorithm used to perform the digital signature
- Compute H = dec(Kpub, file)
- Compute H' = h(file)
- Check if H' == H to validate the signature

35) Describe DH, what problems are related to it and how to solve them

Diffie-Hellman is a protocol for key agreement based on asymmetric cryptography and discrete logarithm problem. These are the steps A and B have to perform in order to agree on a secret key:

- A and B agree on an integer p (large prime) and another integer g (the generator), with 1 < g < p
- A arbitrarly chooses an integer x and computes X = g^xmod p
- B arbitrarly chooses an integer y and computes Y = g^ymod p
- A and B exchange X and Y
- A computes Ka = xY
- B computes Kb = yX
- Ka = Kb = K which is the session key

Diffie-Hellman algorithm is resistant to the sniffing attack and man-in-the-middle attack if the attacker is a listener, i.e a passive attacker. The problem of Diffie-Hellman is that it is vulnerable to active attackers. Man-in-the-middle attack is possible if the attacker can manipulate data. The attacker intercepts the communication between A and B, creating his own number z and computing M = g^z modp. Afterwards the attacker sends its number to B, which computes yM and so on. The two peers A and B are not computing the shared key between A & B but the shared key between the peer and MITM. Hence A will encrypt data which will be decrypted by MITM, who will encrypt them again and send to B.

The manipulation can be detected if the values being exchanged are authenticated, so we need certificates or another version of this algorithm which is Authenticated DH = MQV (Menezes-Qu-Vanstone).

36) **Who's the intruder between RSA, RC4 and SHA-1? Why?**
The intruder is SHA-1 because RSA and RC4, even if one is asymmetric and the other is symmetric, provide confidentiality, while SHA-1 is a digest algorithm used to provide integrity.

37) **Explain briefly what symmetric and asymmetric encryption are and their advantages and disadvantages**
Symmetric encryption:

- Based on a shared secret (key) between two parties that need to exchange some data. The same key is used for both encryption and decryption
- The encryption and decryption functions are very similar. C = Enc(K, P, IV) (eventually there's an initialization vector). P = Dec(K, C, IV) = Enc^-1(K, C, IV)
- Advantages:
    - Faster than asymmetric cryptography because it doesn't exploit complex mathematical problems
    - Provides confidentiality
    - It can be used to encrypt big files
- Disadvantages:
    - Key distribution problem: since the key must be shared among the parties, it must be sent through the network, i.e it is necessary to protect the communication between the parties. This problem is solved using asymmetric cryptography to encrypt the secret key
    - Number of keys: if we have a complex topology in which there are N peers connected to each other, we must have a different key for each pair of peers, hence N(N-1)/2 keys are required, and every peer must store N-1 keys
    - Symmetric cryptography doesn't provide integrity and authentication

Asymmetric cryptography:

- Typically based on 3 different mathematical schemes:
    1) Integer-factorization schemes (RSA): problem of factoring large integers.
    2) Discrete logarithm problem (DH, DSA): problem of finding the logarithm in finite fields
    3) Elliptic curve schemes: generalization of discrete logarithm problem exported on a curve
- The encryption and decryption algorithm use different keys, i.e the public key pK and the private key sK. If public key is used to encrypt data, then the private key is used to decrypt data and viceversa. For example in RSA the encryption is done by the public key, meanwhile in digital signature the encryption of the document is done with the private key. The private key is never sent and the public key, since it's visible to everyone, must be binded to a public key certificate, released by a certification authority.
- Advantages:
    - Provides integrity, authentication and non-repudiation through digital signatures and digital certificates.
    - It can be used to exchange securely symmetric keys between parties
- Disadvantages:
    - Slower than symmetric cryptography
    - Since it's slow, it should be used to encrypt just data of small size (like symmetric keys)

38) **Explain what is a KDF, tell which security problem it addresses, which kind of cryptographic function it is normally built upon, which input parameters it takes and explain the role/importance of each parameter.**

KDF stands for Key Derivation Function, and it is used to generate a symmetric key which is the most random possible, i.e it addresses the problem of randomness of symmetric keys. The key is generated in the following way:

K = KDF(P, S, I, K_LEN) where:

- P is a passphrase used to generate the key. Since passphrases usually are not random but guessable, other parameters are needed
- S is salt, i.e a random number to increase randomness
- I is the number of iterations that the KDF has to perform
- KDF is the function used, typically a hash function. Some examples:
  - PKDF2 uses SHA-1 with |S| >= 64 bits and I >= 1000
  - HKDF uses HMAC

39) **A user must decide if to protect a file with a mac or with a digital signature, for each case describe the operations to be performed, the security properties obtained, and the relative advantages and disadvantages.**

MAC (message authentication code) is a key-dependent one-way hash function. It has the same properties as the one-way hash functions, but they also include a key. MACs are used to provide integrity and authentication. MACs append an authentication tag to a message. The most important difference with digital signature is that MAC uses a symmetric key k for both generating the authentication tag and verifying it. These are the step if someone wants to protect a file with a MAC:

- Assuming that the symmetric key has been exchanged securely (for example via DH protocol or asymmetric cryptography)
- The sender computes the MAC md = H(K || M || K) where M is the message, H is the used hash function (for example SHA-1). It's better to hash the key with the message concatenating it at the beginning and at the end because if the sender computes just H(K || M) or H(M || K) then a digest length attack can be performed. To increase the protection the sender could also insert the message length in the digest
- The sender appends the MAC to the message and sends M || md
- The receiver computes the digest md' = H(K || M || K) and verifies
- If md' == md then everything went good

Advantages of MAC:

- there's only one operation
- there are few additional data

Disadvantages of MAC:

- if the key is not correctly hashed with the message (the best way is K || M || K), then the tag is subject of attacks in which the original message could be modified without detection
- it doesn't provide non repudiation

On the other hand, digital signature exploits asymmetric encryption in order to provide integrity and authentication of messages, i.e it uses 2 different keys. Digital signature is the asymmetric encryption of data made with the private key of the author. Another important difference from MAC is that digital signature encrypts the message digest and not the message in clear (because asymmetric cryptography cannot be used in long messages).These are the steps:

- Assuming that the public key of the sender is bound to a public key certificate released by a certification authority, the sender encrypts the digest of the message using its private key. S = ENC_RSA-2048(Kpr, H(M))
- the sender sends M | | S
- the receiver decrypts the digital signature using the public key of the sender: V = DEC_RSA-2048(Kpub, S).
- the receiver computes the digest of the message H' = H(M)
- if H' == V then everything went well

Advantages of digital signature:

- provides non repudiation in addition to authentication and integrity
- each user have a private key which can be used to generate an infinite number of digital signatures

Disadvantages:

- since it exploits asymmetric cryptography, it is slow, hence it can be applied only to small size data, such as message digests

40) What is a X.509 certificate? List its attributes

X.509 is a public key certificate format. A public key certificate is a data structure used to securely bind a public key to some attributes. There could be a public key certificate associated to an identity or IP address. The issuer (certification authority CA) digitalizes it by digitally signing it, providing integrity. The X.509 format is a certificate with several fields:

- Version
- Serial number: provides a short form, unique identifier for each certificate generated by a CA
- Sign algorithm: in which way the digital signature of the certificate has been done
- Issuer: the Certification authority
- Validity: explains the limited lifetime of the certificate. A public key certificate can be revoked before the expiration
- Subject: C (country), O (organization), CN (common name) and email (because emails are identifiers related to applications)
- Subject public key info: information about the asymmetric algorithm (often RSA)

CHAPTER 3: AUTH

1) **Symmetric challenge-response authentication mechanism : definition, advantages and drawbacks**

In symmetric challenge-response authentication the server (i.e the verifier) sends to the claimant (i.e the client) a challenge. The claimant replies with the solution computed using a shared key and the challenge: response = f(challenge, K).The verifier finally compares the response with a solution computed the key → response = f(challenge, K). if the two responses are the same, then the user is authenticated. There are two main key points:
1) the challenge must be non repeatable to avoid replay attack, in fact is it usually a nonce
2) the function f must be non invertible, otherwise a listener can record the traffic and easily find the secret.
In symmetric CRA the function f is SHA-2 (one-way hash function). In this scenario the verifier has to store the user key in clear together with its ID in the database. In this way the database is vulnerable to attacks. The solution is SCRAM (salted challenge response authentication mechanism), which stores hashed passwords and offers also mutual authentication. More in details:

If A and B want to authenticate each other, the SCRAM protocol provides two challenge for both the peers, Ca and Bc. A sends its identity, its challenge, B computes enc(k, Ca) and sends it together with its challenge Cb and its identity. A performs enc(k, Cb) and mutual authentication with encryption is provided, protecting against man-in-the-middle attack.
The SCRAM protocol protects also against replay and sniffing attack because the challenge is usually a nonce, while the challenge is sent in clear.
However an attacker C, even if he doesn't know the secret key, could start another conversation with B, sending him Ca. then B will reply with enc(k, Ca) and send it together with Cb. The attacker, always pretending to be A, can open another connection with B and send to him Cb: B will answer with enc(k, Cb) and send it together with Cc, but enc(k, Cb) is the answer to the first challenge Cb.
This weakness could be overcome if B forces A to open just one connection with the server.

2) **Explain what a dictionary attack is, when it is possible to perform it and what are possible countermeasures**
A dictionary attack is a brute-force attack which aims to guess the password used for a password based authentication. This attack is particularly effective when a large number of passwords are to be cracked. It is based on the hypothesis that the cryptanalyst knows the hash function used to store the password (due to a leakage of information). The attack consists of two phases:
   1) Pre-computation: the attacker takes a "special" dictionary (in which there are not just common words but also country names, people names and so on, since the naïve user tends to use common words as password) and computes the hash value of each word. The advantage of pre-computation is that it is done just once. It is possible to achieve a time-space trade-off by storing the dictionary words in a database using the hash value as the key. Of course this phase is based on the hypothesis that the user has chosen one word of the dictionary as its password
   2) Attack phase: the attacker can obtain a hash value HP of an unknown password. Then he performs a lookup W = lookup(DB, HP) in the database to check if there's one row with the desired value. If yes, then the attack succeeds

   The countermeasures to this attack are:

   - Use of salt, which increases the randomness of the hash value. For each user/UID:
     - A password is asked
     - A salt is being generated, which is random and long
     - The hash value is computed: HP = hash(pwd || salt), i.e the HP depends on the salt
     - The verifier stores in the DB the triples {UID, HP_UID, salt_UID}
   - Don't use common words as password

3) **What is a rainbow table?**
A rainbow table is a technique used in dictionary attacks in order to reduce storage requirements at the cost of slightly longer lookup-times, because the attacker stores less passwords and takes more time to discover the correct one. To do so a rainbow table uses a hash function h and a reduction function r. The reduction function simply takes N characters from the previous computed hash. Each password is hashed and reduced. This process is iterated for an arbitrarly number of times until a final hash is computed. In this way only the couples plaintext-final hash are stored and each one represents a chain "containing" all the possible hashes relative to the starting password. After the table has been built, the algorithm works as follow: given a hash H of an unknown plaintext:
   - For each (final hash FH in table) check if HF == H
   - If no match is found reduce H and start again from 1
   - If the hash matches one of the final hashes, that chain contains the original hash
   - Once the chain's hash has been found, take the plaintext relative to that chain and start hashing and reducing it until it comes the known hash

Example: we have a rainbow table of 10^9 rows, each one representing 1000 passwords:

- Pre-computation: the attacker selects 10^9 distinct passwords P, then:
    - For (p = P, n = 0; n < 1000; n++) { k = h(p); p = r(k); } //r is the reduction function
    - The attacker stores {DB, P, p}, i.e the chain head and tail. This entry implicitly represents all 1000 passwords that the attacker have tried.
- Attack: let HP be the hash of a password
    - For (k = HP, n = 0; n < 1000; n++) {p = r(k); //reduction function to go from the value k to a possible password
    - If the value is found in the DB row, then the chain has been found → k = h(p)

4) **Explain briefly the OTP concept and what security problem it solves. Explain, with proper formulas, initialization and verification phases in S/KEY protocol**

OTP (one-time password), also known as session authentication, is an authentication technique in which a password is random generated and used just once. Since there are not reusable password, this avoids guessing attacks. However, these password must be protected to reduce their being intercepted if they are sent in clear. In other words, OTP is resistant to sniffing attack but it could be subject of active MITM attacks. The first system implemented for OTP was S/KEY, which is based on MD4 and MD5 hash algorithms. The S/KEY protocol phases are the following:

- Initialization: the user chooses a secret passphrase PP which is minimum 8 chars long (i.e 64 bit). PP is then concatenated with a server provided seed in order to generate a list of OTP. This allows to use the same passphrase for multiple access because the seed is always random.
- The user computes N OTP: Pn = h(PP, seed), P2 = h(P1, seed) = h^2(PP, seed) … Pn = h^n(PP, seed).
- The server stores the n-th password h^n(PP, seed), which will not be directly used for authentication, but indirectly
- For authentication, the server asks for the Pn-1 password. The user sends its Pn-1 = X.
- Verification: the server checks if h(X) == h^n(PP, seed). In this case authentication is complete.

5) **Explain how FIDO works and why its use is increasing**

FIDO (Fast identity online) is an authentication method which uses biometric authentication and 2-factor authentication. The main concept of FIDO is that it's up to the user to exploit any personal device which is capable of performing asymmetric cryptography. This device will then be used to respond to an asymmetric challenge and for creating a digital signature of data, providing authentication and non repudiation. FIDO usage is increasing because it avoids to rely on password-based authentication mechanisms, which present several vulnerabilities. Moreover, biometric data never leave the user's device and no secrets are stored by the server, just the pair public key-private key, for which only integrity needs to be provided. Here the details:

- Registration: the web server sends to the device chosen by the user a request to create a pair of public key and private key. The user chooses how to protect them (biometric systems, PIN,…). Then a third key is generated and is associated with user's ID. The private key remains on the user's device, while the public key is sent to the web server and it's associated with the name (decided by the user) that will be used during authentication.
- Login: although it's not based on username-password authentication, the web server asks for username and password (they are used to bind the right key pair to the right service). The server will then send a challenge to the user, encrypted with the retrieved public key. The user will decrypt the challenge using its private key and send it back to the web server in order to authenticate.

Since FIDO doesn't use X.509 certificate, there's no need for a third party in the registration process, i.e there's direct trust between the user and the web server

6) **Explain what is an asymmetric challenge-response authentication system, describe which data are exchanged between the client and the server, and discuss the advantages and disadvantages of this approach.**

In asymmetric CRA the claimant sends to the server its x.509 public key certificate instead of its ID. The verifier chooses a random nonce R and encrypts it with the claimant's public key → C = enc(Kpub, R). the claimant decrypts with its private key → response = dec(Kpr, C). finally the verifier checks if response == C and if the claimant has a valid certificate. In this way it's proven that the claimant is the owner of Kpr.

Pro:

- The server doesn't contain any critical informations about the users but the list of valid users
- This is the strongest mechanism to provide CRA

Cons:

- Slow because it uses asymmetric cryptography
- If designed inaccurately may lead to an involuntary signature by the claimant
- It's important to protect the public key in the right way

CHAPTER 4: NETWORK

1) **Describe at least two possible attacks with ICMP and the possible countermeasures**

Due to lack of authentication in ICMP traffic, there are several attacks that can be launched against it:

- Ping bombing/flooding: the attacker sends as much as ping echo-request he can, occupying the victim's bandwidth and CPU time. It's not possible to completely avoid ping flooding, since it is based on a basic feature of the internet protocol. Nevertheless, with a firewall it is possible to detect that ping flooding attack is happening, and eventually block the attacker's IP address.
- DoS attacks: a fake node sends fake "destination unreachable" messages (this attack is possible after a shadow server attack) thanks to lack of authentication
- Smurf attack

  The attacker spoofs the IP address of the victim and uses it as the source of a large number of echo-request ICMP packets (i.e broadcast) and sends them to various subnets. If all the subnets reply to the echo-request with the echo-repy, then the victim is overloaded of ICMP packets, causing source quence which slows down the victim. The possible countermeasures to this attack are:

  - For external attacks: reject IP broadcast packets at the border router
  - For internal attackers: use of management tools in order to identify the attacker, because inside the network it's not possible to forbid broadcast traffic

2) **Explain what the ICMP redirect attack is**

the ICMP redirect is a feature of the internet protocol that is used for keeping the hosts using the best available gateway for a certain network or to switch it in case of failure. When an ICMP redirect is received, a new route is added to the host's routing table. The ICMP redirect attack exploits the ICMP redirect to force the victim to update its routing table in order to use the compromised system as the gateway, allowing the attacker to perform a man-in-the-middle attack. A solution to avoid the ICMP redirect attack is to disable the ICMP redirect messages on the hosts or on the network nodes.

3) **List DHCP vulnerabilities**

DHCP is a non authenticated and broadcast protocol which provides IP address, netmask, default gateway, local nameserver and local DNS suffix. Due to lack of authentication, DHCP is vulnerable to several attacks:

- Unauthorized DHCP servers: because the client has no way to validate the identity of a DHCP server, unauthorized DHCP servers (commonly called rogue DHCP) can be operated on networks, providing wrong information to DHCP clients. This can result in a DoS attack or a man-in-the-middle attack. An example is a "logical" MITM attack in which the attacker provides a configuration with a netmask in the form 255.255.255.252 so that the victim is trapped in a network where only the attacker's address is reachable.Because the DHCP server provides the DHCP client with server IP addresses, such as the IP address of one or more DNS servers, an attacker can convince a DHCP client to do its DNS lookup through its own DNS server, and can therefore provide its own answers to DNS queries from the client. This in turn allows the attacker to redirect network traffic through itself, allowing to eavesdrop on connections between the client and network servers it contacts.
- Unauthorized DHCP clients: because DHCP server has no secure mechanism for authenticating the client, clients can gain unauthorized access to IP addresses by presenting credentials that belong to other DHCP clients. This also allows DHCP clients to exhaust the DHCP server's store of IP addresses by presenting new credentials each time

The best solution to improve DHCP security is the authentication for DHCP messages that uses HMAC-MD5: it is rarely used because it needs a symmetric key, that has to be installed on all the machines running DHCP server.

4) **How can a VPN be built without using tunnels? What are its advantages and drawbacks?**
a VPN is a private network built upon a public infrastructure, such as Internet, by adding security procedures over the unsecure communication channels. If encryption of packets is not required, then the VPN can be built without using tunnels. In this case the VPN is built using private addresses: the networks to be part of the VPN use non public addresses so that they are unreachable from other networks. In this way packets are not globally routed but they travel only between trusted destinations. The advantage is that these packets don't require authorization since all the networks are private. On the other hand there are some drawbacks: since packets are not protected, if an attacker can guess the private addresses or has access to the communication devices, he can then sniff packet content

5) **Discuss the end-to-end security with basic VPN architecture**
End-to-end security means that IPsec is installed to the end points of the network we want to protect. End to end security is provided by IPsec in transport mode (in which just the payload of the packet is encrypted). More in detail, end-to-end security with basic VPN architecture (site to site) tries to address the security-in-depth principle, creating a double defense line, and it's structured this way:

- IPsec in transport mode is used between the peers, guaranteeing a secure virtual channel. In this way it doesn't matter if LAN, gateway or WAN are insecure because there's the secure channel. In this way integrity and authentication of packets is achieved.
- IPsec in tunnel mode among gateways in order to guarantee confidentiality using the ESP protocol, in which the entire packet (payload + header) is encrypted

In this way it is possible to balance the workload of the network, since encryption and authentication operations can be separated. Unfortunately it is difficult to manage, since requires a VPN to be configured not only on gateways but also on the end points.

6) **What is RADIUS? What are its security features and what's its role in the security architecture?**

RADIUS (random access dial-in user service) is a network authentication protocol based on a client-server model between NAS (Network access server) and AS (authentication server). It provides authentication (using port 1812/UDP), authorization and accounting (using port 1813/UDP) (AAA) to control network access to both physical and virtual ports and circuital ports.
When a RADIUS server receives an AAA request for a username containing a realm, the server will reference a table of configured realms. If the realm searched is known, then the server will proxy the request to the configured home server for that domain. In this way RADIUS acts as a proxy server and proxy chaining is possible.

The authentication and authorization mechanism works this way:

- The user sends a request to the NAS to gain access to a particular network resource using access credentials. The credentials are passed to the NAS via the link layer protocol, using for example PPP (Point-to-point protocol)
- The NAS sends a RADIUS access request to the RADIUS server, requesting authorization to grant access via the RADIUS protocol
- The RADIUS server checks if the information is correct using authentication schemes such as PAP (password-based), CHAP (challenge-based) or EAP.
- After this verification, the RADIUS server can reply in different ways to the NAS:
    - Access-reject: the user is unconditionally denied to access to all requested network resources
    - Access-challenge: the RADIUS server requests additional information from the user such as a secondary password, PIN…
    - Access-accept: the user is granted access

RADIUS provides packet integrity and authentication via keyed-MD5 digest, where the key is shared between the NAS and the RADIUS server. In this way the password is transmitted "encrypted" (not really encrypted) with MD5, after padding with null bytes up to 128 bits: password || padding XOR MD5(key + authenticator), where the (request) authenticator is placed inside the access request packet and consists of a random string of 16 bytes obtained by a MD5 digest computed over other fields of the packet.

7) What is a ARP spoofing attack? How can it be prevented?

ARP spoofing exploits the lack of authentication in ARP (address resolution protocol, which is used to discover the L2 address of a node knowing its L3 address) protocol to send spoofed responses to client on a LAN. ARP spoofing attack can be run from a compromised machine inside the LAN or from the attacker's machine that is connected directly to the target LAN. Generally the goal of the attack is to associate the attacker's host MAC address with the IP address of a target host, so that any traffic meant for the target host will be sent to the attacker's host. The attacker may choose to inspect the packets and eventually modify them before forwarding (man-in-the-middle attack), or he can perform a DoS attack by simply drop some packets.
Possible countermeasures:

- Firewall (packet filter) to filter out packets with conflicting source address information
- Static ARP entries: the simplest form of certification is the use of static, read-only entries for the ARP cache of a host. IP-MAC mapping may be statically entered in the ARP cache. In this way ARP spoofing is avoided but there's a management problem because address mappings for all systems in the network must be generated and distributed. This does no scale on a large network since the mapping

has to be set for each pair of machines resulting in n(n-1) ARP entries that have to be configures when n machines are present. On each machine there must be an ARP entry for every other machine on the network, i.e n-1 ARP entries for each ARP cache in the network.

- OS security, i.e develop security mechanisms for operating systems

8) **What is DNS cache poisoning?**

DNS cache poisoning is an attack in which the resolution of DNS queries is subverted. This can be achieved by a malware that overrides a computer's TCP/IP configuration to point at a rogue DNS server under the control of the attacker, otherwise through modifying the behavior of a trusted DNS server so that it does not comply with internet standards.

The purpose of DNS cache poisoning is thus to write a wrong entry in the DNS cache of the victim, so that whenever the victim performs a DNS query for a certain domain, a response with the attacker's nameserver is provided. There are two possible modes of implementation:

1) The attacker sends to the victim answers to queries never done to overwrite the victim DNS cache, since the most recent entries have higher priority with respect to the present ones
2) The attacker acts as a client and sends a DNS query to the DNS server of the victim and self-provides the wrong answer before the real DNS server, overwriting the victim's cache. The real answer will arrive later and will be discarded

There is no real countermeasure against DNS poisoning, since there's no way to know if the DNS response received are legitimate traffic or not. The only possible solution is DNSsec, which aims to provide a digital signature of DNS queries.

9) **Which are the security problems of the ICMP protocol?**

Since ICMP packets are not protected, some of its messages can be used to perform several attacks:

- Echo-request/echo-reply: these messages can be used to perform a ping flooding or a smurf attack
- Destination unreachable: sent by a shadow server to perform a DoS attack
- ICMP source quence: this is a message sent by the network nodes to inform the sender that the packets cannot be forwarded due to buffer overload of the network nodes → DoS attack
- ICMP redirect: used to perform an ICMP redirect attack, in which the attacker cheats the victim to use a wrong gateway that corresponds to the attacker's system → man-in-the-middle
- ICMP time exceeded for a datagram: sent by network nodes when they have to drop a packet because its TTL (time to leave) has expired → DoS attack

10) **DNS flash crowd : explanation and countermeasures**

DNS flash crowd is a DDoS attack in which the DNS cache of the victim is flooded with DNS queries. It is really hard to detect such an attack because legitimate and illegitimate requests are indistinguishable. A solution to the DNS flash crowd is the use of firewalls and IDS (intrusion detection systems)

11) **What is a S-VPN? Why has this name?**

s-VPN stands for secure-VPN, i.e a VPN in which packet encryption is provided. The traffic is encrypted at the edge of one network or at the originating computer, moved over the internet like any other data, and then decrypted when it reaches the corporate network or a receiving computer. These are the requirements for a s-VPN:

- All the traffic must be encrypted and authenticated. The data is encrypted at the sending edge and decrypted at the receiving edge.

- The security properties of the VPN must be agreed to by all parties in the VPN. Every tunnel in a secure VPN connects two endpoints who must agree on the security properties before the start of data transmission
- No one outside the VPN can affect the security properties of the VPN.

In s-VPN, before the encapsulation, the packets are protected with encryption (→confidentiality), a keyed digest (→ integrity + authentication) and numbering to avoid replay attacks.

In this architecture along with the router, which manages encapsulation and decapsulation, a TAP (tunnel access point) is deployed, which manages cryptographic operations on the packets. If cryptographic algorithms are strong, then the only possible attack to this VPN is DoS.

12) A company wants to create a VPN between two sites, which infrastructure can it use? Explain how it is configured and what are its advantages and drawbacks

A solution could be site-to-site VPN (also called basic VPN), in which IPsec in tunnel mode is established between the two gateways (tunnel mode is not created between routers but between gateway), which are the contact points between the insecure network. The packets can be encapsulated using ESP (encapsulating security payload) protocol, which adds an ESP header after the IP header of the packet and an ESP trailer near the TCP/UDP payload, and then encrypts the entire packet, postponing a new IP header. In this way integrity, authentication and confidentiality are guaranteed.

The entire, original, inner backet travels through a tunnel from one point of an IP network to another, and no routers along the way are able to examine the inner IP header; since the original packet is encapsulated, the new larger packet may have totally different source and destination addresses.

Advantages:

- The company can inspect the inner network
- IPsec has to be configured only on gateways
- Confidentiality is guaranteed, and if the cryptographic algorithms are strong then the only possible attack is DoS

Drawbacks:

- The internal network should be trusted
- The gateway could be overloaded by the huge computation required to manage cryptographic operations

13) EAP : what is it? List differences between EAP, PAP and CHAP

EAP (extensible authentication protocol), PAP (password authentication protocol) and CHAP (challenge handshake authentication protocol) are three different techniques for authenticate network access:

- PAP requires the applicant to repeatedly send to the server authentication request messages, consisting of a username and a password, until a response is provided by the server (this information is sent in clear)
- CHAP uses symmetric challenge response, based on the user password. The server sends a challenge, the client adds on a secret message, hash them both and sends the result back to the server. The server also adds a secret message to the challenge, hashes with an agreed-upon algorithm and compares the two results. This solution is better than PAP in terms of security but the channel is still insecure
- EAP provides a set of protocol messages that can encapsulate various authentication methods to be used between a client and an authentication server. It is a flexible authentication framework for point-to-point channels that allows the usage of external

authentication mechanisms at level 2. EAP is independent from IP because it's for authentication before creating layer 3 connections. In order to provide authentication EAP could use MD5-challennge (similar to CHAP), OTP or generic token card. Since the link is not assumed to be physically secure, EAP methods must provide security on their own. Some possible methods are EAP-TLS, EAP-SRP (secure remote password), EAP-MD5 (which provides only EAP peer authentication, no mutual authentication).

14) **Explain what 802.1x is identifying its architectural components**

IEEE 802.1x (port based Access control) is a layer-2 authentication architecture. It is useful in both wired network to block undesired access and wireless network. More precisely IEEE 802.1x is a framework (i.e it supports many authentication methods) which performs two things:

1) Authentication
2) Key management (optional), especially in wireless networks in which the traffic must be encrypted

802.1x architecture consists of 3 main structures:

1) semi-public network/enterprise edge: it's the network where the supplicant, i.e the user who wants to authenticate, is placed

2) authenticator/etherNAS: the network where is placed the authenticator, i.e the network access point in wireless network or a switch in wired network. It acts as a pass-through orchestrating communication between the supplicant and the application server

3) enterprise/ISP network: where is placed the authentication server (RADIUS).

The supplicant communicates with the authenticator through EAP over wireless (EAPOW) in wireless network and through EAP over lan (EAPOL) in wired network

15) **What is IPsec? what security features does it offer?**

IPsec is a suite of authentication and encryption protocols designed to provide security at IP level.

IPsec sets out to offer protection by providing the following services at network level:

- Access control to prevent unauthorized access to the resources
- Connectionless integrity to give an assurance that the traffic received has not been modified in any way
- Confidentiality to ensure that internet traffic is not eavesdropped by unauthorized parties. This requires all IP datagrams to have their data field, TCP, UDP, ICMP or any other datagram data field segment encrypted.
- (source) authentication so that when a destination host receives an IP datagram, with a particular IP source address, it is possible to be sure that the IP datagram was indeed generated by the host with the source IP address. This prevents IP address spoofing
- Replay protection to guarantee that each packet exchanged between two parties is different

Integrity, authentication and no replay are guaranteed by AH (authentication header) protocol, while confidentiality is provided by ESP (encapsulating security payload)

16) **IPsec transport mode with ESP : what are its security properties? Which ones are optional and which are compulsory?**

In IPsec transport mode with ESP the encrypted part of the packet is the payload, which has the ESP header at the beginning and the ESP trailer at the end. The security properties offered to this traffic are:

- Data authentication and integrity, obtained by computing keyed-digest
- Partial protection against replay attacks, obtained by the sequence number field of the ESP packet
- Confidentiality, obtained through symmetric encryption

It is possible to perform only encryption leaving optional authentication and integrity and viceversa. The compulsory thing is to have at least one of these security properties, hence either confidentiality or authentication/integrity.

17) Explain what an IPsec SA is and how it is established

SA (security association) is a key concept in IPsec. It's a unidirectional logical channel established between two IPsec systems. If a bidirectional packet flow is needed, then also two security associations are needed. A SA is identified by three parameters:

1) SPI (Security Parameters Index): a 32-bit unsigned integer carried by AH and ESP headers to enable the receiving system to select the SA under which a received packet will be processed.
2) IP destination address
3) Security Protocol Identifier: indicates whether the association is an AH or ESP security association

A security association is established through the ISAKMP (internet security association and key management protocol) protocol. ISAKMP describes the procedures needed to negotiate, set-up, modify and delete a SA, but key exchange method is not fixed: it can be OOB (Out-of-Band) or done by OAKLEY (protocol for authenticated exchange of symmetric keys).

Before the creation of an IPsec SA, an ISAKMP SA has to be created in order to protect the negotiation of a IPsec SA. The same ISAKMP SA may be reused several times to negotiate other IPsec SAs. In this SA key exchange Is performed by OAKLEY. The union of ISAKMP and OAKLEY is IKE (internet key exchange) protocol. IKE key determination is a refinement of the Diffie-Hellman algorithm, hence secret keys are created only when needed and there's no need to store secret keys for a long period. There are basically two phases:

- Phase 1: negotiation of a bidirectional ISAKMP SA in "main mode" (slower because it consists of more messages exchanged, but protects the parties identities) or "aggressive mode" (faster, less messages but no protection of identity)
- Phase 2: negotiation of the IPsec SA in "quick mode"

Creating an IPsec SA means defining common policies and crypto suites to be used during the communication. All the policies relative to a certain SA traffic are stored in the SPD (security policy database), while all its parameters like symmetric algorithm, hash functions etc. are saved in the SAD (SA database).

18) Describe SYN cookies and what's their role in security

SYN cookies are the only really effective solution to SYN flooding attacks. A SYN cookie is a TCP 32 bits sequence number containing:

- The timestamp t
- The maximum TCP segment size m
- The digest computed over IP source address/source port, IP destination address/destination port and t

If no SYN cookie is used, whenever a server asks to open a TCP connection it stores in the TCP stack the couple SYN-ACK relative to a specific connection: the purpose of SYN flooding is to saturate the TCP stack. With TCP cookies the state of the TCP connection is no more stored by the server, but it is encoded in the SYN-ACK that contains a sequence number generated as above that will be used by the server to reconstruct the data stream, allowing the recognition of clients that have already sent the SYN without storing any information about them on the TCP stack.

CHAPTER 5: NETWORK 2

1) **List TLS's security properties**

TLS is an internet standard that evolved from SSL (Secure Socket Layer). It allows to create a secure communication channel between a client and a server. The TLS protocol provides basic security services to various higher-layer protocols, especially HTTP. These are TLS's security properties:

- Server authentication (compulsory): the server authenticates itself by sending its public key certificate X.509 and by responding to an implicit asymmetric challenge
- Client authentication (optional): the client sends its own certificate to the server to verify that the client's certificate was signed by a trusted CA. this is optional because few customers are willing, know how, or care to get digital certificates, requiring them to do this would amount to locking a huge number of customers out of the system which would not make business sense.
- Data integrity and authentication through HMAC
- Data confidentiality (optional) through symmetric algorithms such as 3DES, RC4, IDEA, AES etc.
- Perfect forward secrecy if ECDHE (elliptic curve Diffie-hellman) is used for key exchange

2) **How are data transformed when they are sent in a TLS channel with ciphersuite TLS_DHE_DSS_WITH_AES_128_CBC_SHA_1 and compressed with ZIP ?**

The TLS record protocol, which provides confidentiality (the TLS handshake protocol defines a symmetric key to encrypt TLS payloads) and message integrity (TLS handshake defines also a key for keyed-digest) takes an application message to be transmitted across the TLS secure channel and transforms it in the following way:

- The message D is fragmented into D0, D1,…Dn because TLS can send only specific length fragments
- Each Di is compressed with a proper compression function
- The MAC is computed over the compressed data. HMAC is computed for each compressed data fragmend: HMAC(D) = SHA-1(K1 xor opad, SHA-1(K1 xor ipad, D)). After this step every block is in the form Di||HMACi, where K1 is the key used for keyed-digest
- The data block is encrypted: EncBlock = ENC_AES_128_CBC(K2, IV, Di||HMACi||padding)
- A TLS record header is appended to the encrypted message: TLSh || EncBlocki and the message Is sent. The header consists of several fields:
    - Content type: the higher-layer protocol used to process the enclosed fragment
    - Major version
    - Minor version
    - Compressed length: the length in bytes of the plaintext fragment

3) **Describe how does the client authentication work in TLS. Discuss advantages and disadvantages**

client authentication is optional in TLS. If required, it is performed during the TLS handshake protocol. The handshake protocol allows server and client to authenticate each other and to negotiate an encryption and MAC algorithm and cryptographic keys to be used to protect data sent in TLS channel. During this protocol the server may request client authentication. In this case it sends a "certificate request" message to the client, which replies with its certificate and eventually certificate verification. Along with its certificate, the client sends the hash of all the previous messages exchanged with the server encrypted with its private key, the server will decrypt the challenge with the client's public key, looking for the hashes of the previous messages to correspond.

Advantage: TLS authentication uses asymmetric challenge response system, which is the strongest authentication mechanism available and does not require to store any information on the server

Drawbacks: slower than symmetric authentication, unpopular because it requires security awareness of users, in fact client authentication is optional.

4) **Explain what the downgrade problem is and how can it be prevented**

TLS downgrade problem concerns the handshake protocol, in which client and server authenticate each other and choose MAC and encryption algorithms to be used during client-server TLS communication, In addition to key exchange (one key for keyed-digest, one key for symmetric encryption). the client sends a "client-hello" message, which contains the highest supported protocol version, to the server, and the server replies with a "server hello" message, which contains the protocol version to use. In general the two message contain the same version, but the server could not support the version sent by the client, and now comes to play the TLS downgrade problem: the server replies with an older version of the protocol (i.e less secure) or forces the client to restart the communication sending a "client hello" with an older version. There may be also some attacks: the attacker sends fake server responses, to force repeated downgrade until reaching a vulnerable TLS version, then execute a suitable attack.

The TLS downgrade problem can be prevented if TLS fallback signaling cipher suite value (SCSV) is implemented.

SCSV prevents protocol downgrade through the creation of a new (dummy) ciphersuite, without new algorithms, which contains just the TLS_FALLBACK_SCSV message, which should be sent by the client when opening a downgrade connection. There's also a new fatal alert value in the TLS alert protocol: inappropriate_fallback. This message must be sent by the server when receiving TLS_FALLBACK_SCSV and a version lower than the highest one supported, then the channel is closed and the client should retry with its highest protocol version.

5) **TLS and HTML form-based authentication : advantages and drawbacks**

TLS authentication is performed by using the client authentication feature of TLS, that is the client authenticates itself using an X.509 certificate. In this case authentication happens at the level of TLS library.

HTML form-based authentication happens at the application level: the HTML page will contain a form requesting the authentication parameters, sent along a POST request to the web-server and processed by the proper application logic server side.

Advantages of TLS authentication:

- Since the TLS library is the first piece of code the user enters in contact with, an attacker could only exploit vulnerabilities related to TLS: the attack surface is the smallest possible
- Some servers support also a semi-automatic mapping between the credentials extracted from the X.509 certificate and the users of the HTTP service

Drawbacks of TLS authentication:

- It requires the user to have a X.509 certificate, that is not very common and requires the user itself to have some knowledge about public certificates and PKI

Advantages of HTML form-based authentication:

- Fine-grained control over the presentation and behavior of the solicitation for user credentials

Drawbacks of HTML form-based authentication:

- It exposes the web server to the attacks: the attack surface is the biggest possible
- Unless this last request uses a secure protocol (like HTTPS), the data are sent in clear, so no security is provided automatically
- If the page is not secured, the user could be exposed to phishing, because other people could create a fake web page with similar URL address, but with another action performed in the form and only few users have the technical knowledge to verify the URL of the HTTP method to send user/password

6) **Explain differences between HTTP authentication mechanisms and TLS authentication mechanisms, enlightening advantages and drawbacks for both**

TLS authentication system is performed in the TLS handshake protocol and it's based on asymmetric challenge authentication. Server and user (optional) authentication requires both to have an X.509 public key certificate.
Advantages of TLS authentication:

- Since the TLS library is the first piece of code the user enters in contact with, an attacker could only exploit vulnerabilities related to TLS: the attack surface is the smallest possible
- Some servers support also a semi-automatic mapping between the credentials extracted from the X.509 certificate and the users of the HTTP service

Drawbacks of TLS authentication:

- It requires the user to have a X.509 certificate, that is not very common and requires the user itself to have some knowledge about public certificates and PKI

On the other hand, HTTP supports two types of authentication:

1) Basic HTTP authentication: it's the simplest technique for enforcing access controls to web resources because it doesn't require cookies, session identifiers or login pages. In basic HTTP authentication, a request contains a header field in the form of authentication: Basic <credentials> where credentials is the Base64 encoding of ID and password. Since the message is encoded, not encrypted, it is easily recoverable with openSSL.
2) Digest HTTP authentication

7) **TLS-then-protocol vs protocol-then-TLS : explain how does it work, advantages and drawbacks for both**

- TLS-then-protocol: first the TLS connection is established, then the application level protocol uses the secure channel to communicate. In this solution security features are all upon the system manager, so the web developer doesn't have to be worried about security. In this case all the pages are protected, even the ones that doo not require security.
- Protocol-then-TLS: in this solution security has to be managed by the web developer, who has to decide if a page requires to be protected with the activation of a TLS channel or not.

From a firewall/IDS point of view, the use of different ports by TLS-then-protocol and by protocol-then-TLS can impact on complexity. Moreover TLS-then-protocol implies that the traffic Is encrypted: hence no check can be performed on packets. Using protocol-then-TLS at least traffic which do not require protection can be checked before activating the TLS channel.

CHAPTER 6: FIREWALL

1) **Describe the firewall screened host architecture enlightening the entities involved and listing advantages and drawbacks**

Screened host architecture is composed by a packet filter, which filters traffic at network level, and a bastion host, which acts as a gateway. In this configuration the packet filter is connected to both the external (insecure) and the internal network, while the bastion host is connected to the packet filter. When a packet arrives, the packet filter can make three decisions:

1. It can drop the packet, because it doesn't match with the rules of the packet filter
2. It can forward the packet directly to the internal network (e.g it can send an email straight to the email server).
3. If the packet needs additional check, the packet filter can forward it to the gateway, which will perform the final decision, i.e either drop the packet or forward it to the internal network.

Moreover the router (packet filter) blocks all the ingress traffic unless it is meant to go to the bastion and blocks all the egress traffic unless it comes from the bastion.

Advantages:

- This solution is more flexible than the dual-homed gateway, in which is the router the one connected to both the external and internal network, because in screened host architecture the traffic doesn't necessarily have to pass through the gateway
- If the router decides to forward an incoming packet to the gateway, than a double defense line is deployed, satisfying the security-in-depth principle

Disadvantages:

- Solution complex and expensive because we need additional components
- If the router doesn't forward the packet to the gateway then there's no double defence line, furthermore there's a single point of failure
- Only the hosts/protocols passing through the bastion can be masked (unless the oruter uses a NAT)

2) **With reference to the network firewall, explain the whitelist and blacklist configurations**

A firewall can be configured using different policies, based on which network traffic is filtered.
Whitelist: policy based on the concept "everything which is not explicitly permitted is forbidden", hence the default rule is to drop incoming and outgoing packets unless they match the specific firewall rule
Blacklisting: policy based on the concept "everything which is not explicitly forbidden is permitted", hence the default rule is to accept incoming and outgoing packets unless they are in the firewall's "blacklist"

3) **Describe the packet filter technology, enlightening advantages and limitations**

Packet filter technology is a configuration in which a screened router is connected to both the external (insecure) and the internal network. It performs verifications at network leve, i.e it analyzes the information embedded in IP packet header. The packet filter is typically set up as a set of rules based on matches to fields in the IP or TCP header. If there's a match to one of the rules, that rule is invoked to determine whether to forward or discard the packet. If there's no match to any rule, then a default action is taken. Two default policies are possible: whitelisting or blacklisting.

Advantages:

- Easy to implement because no additional components are needed
- Cheap because it consists of just one component

Disadvantages:

- Single point of failure
- No double defense line

- Since it relies just on network level we are not able to provide security at higher levels

4) **Describe the dual-homed gateway architecture, illustrating the role of each component and listing advantages and drawbacks**

Dual-homed gateway consists of a packet filter connected to the external (insecure) network and a gateway with two network cards, the first connected to the external network and the second connected to the internal network, acting as a bridge. The gateway performs control at a higher level and all the traffic passes through the packet filter at the beginning and then through the gateway, before entering the internal network.

Advantages:

- Still simple implementation, because it needs small additional hardware requirements
- Provides double defense line, hence satisfying the security-in-depth principle
- No single point of failure: if the screening router has a bug, then there's the gateway yet to perform controls.

Disadvantages:

- Very conservative solution, because all the traffic must pass necessarily through both the devices, even if additional controls are not needed. Hence this solution is unflexible
- Since this solution is not flexible, it could led to a large work overhead because packets have to be re-assembled, resulting in worse performances.

5) **Packet filter and circuit-level gateway : similarities, differences, advantages and drawbacks for both**

- Packet filter: works at level 3 (network), so it considers IP headers processing one packet at a time. When a TCP connection has to be established, It is performed between the external client and the internal server. Packet filter can be static (stateless, i.e every packet is checked independently from the historu of packets passed through the packet filter, hence it has no memory) or dynamic (stateful, i.e the packet filter keeps information about packets belonging to a connection considered valid, avoiding to check them multiple times).

  Advantages:
    - Cheap
    - Easy to implement
    - Can process packets at very high speed
    - Can easily check most fields in L3 and L4 headers, allowing a lot of flexibility in defining security policies

  Drawbacks:
    - Does not support authentication since it is able to process only information available at L3
    - Single point of failure
    - Requires a NAT implementation for masquerading the internal network
    - Security is not provided at higher levels
- Circuit level gateway: firewall that inspects traffic at level 4 (transport), so It considers TCP/UDP datagrams. It creates a transport-level circuit between client and server: the client is talking to the circuit level gateway for example by opening a TCP channel, the gateway takes the TCP segments and then it opens another TCP channel to forward the packets to the server without understanding the application protocol. It doesn't understand or manipulate in any way the payload data, it just copies between its two interfacs the TCP segments or UDP datagrams (if they match the access control rules), but, in doing this, it will re-assemble the IP packets and hence it will provide protection against some L3/L4 attacks.

Circuit-level gateway breaks the TCP/UDP client-server model during the connection: the client performs handshake with the gateway and the gateway performs handshake with the server, i.e there's no direct communication between client and server.
Advantages:

- Does not require any specific application implementation
- Protects the internal server for any kind of attacks against the TCP handshake or against the IP fragmentation. for example the attacker sends the attack fragmented in many different packets: if a packet filter is implemented then it will process one packet at time, without being able to detect the attack. On the other hand, a circuit-level gateway allows to re-assemble packets, considering the whole segment and hence being able to detect the attack
- May authenticate the client, but this requires modification to the application

Disadvantages:

- Slower than packet filter
- Still has some limitations of the packet filter technology, for example it doesn't understand the application commands and data

6) **Packet filter and application-level gateway : similarities, differences, advantages and drawbacks for both**

- Packet filter: works at level 3 (network), so it considers IP headers processing one packet at a time. When a TCP connection has to be established, It is performed between the external client and the internal server. Packet filter can be static (stateless, i.e every packet is checked independently from the historu of packets passed through the packet filter, hence it has no memory) or dynamic (stateful, i.e the packet filter keeps information about packets belonging to a connection considered valid, avoiding to check them multiple times).
  Advantages:
  - Cheap
  - Easy to implement
  - Can process packets at very high speed
  - Can easily check most fields in L3 and L4 headers, allowing a lot of flexibility in defining security policies

  Drawbacks:

  - Does not support authentication since it is able to process only information available at L3
  - Single point of failure
  - Requires a NAT implementation for masquerading the internal network
  - Security is not provided at higher levels
- Application level gateway: composed by a set of proxies inspecting the packet payload at application level. An application proxy is a program which receives requests from someone (client) and sends back the answer, i.e it breaks the client-server model. Application level proxies automate the filtering and forwarding processes for the client. The client application initiates the process by contacting the firewall. The daemon proxy on the firewall picks up the request, processes it, and if it is acceptable, connects it to the server in the "bad network". If there's not response, it then waits and returns the data to the client application. Usually proxies are used for caching, but in security thy have additional features, such as masquerading/renumbering the internal IP addresses or, when used as part of a firewall, can perform peer authentication.
  Advantages:

- Top security because it understands application level commands. They offer a higher level security because in handling all the communications, they can log every detail of the process, including all URLs visited and files downloaded
- Since users do not have direct access to the server, it makes it harder for an intruder to install a backdoor around the security system.
- Rules are more fine-grained and simpler than those of a packet filter

Disadvantages:

- Every application needs a specific proxy. Not having proxy for certain protocol means that if an application requires that protocol it can be delayed until a proper proxy is available. For this reason performance may be lowered
- Computationally heavy since a new process is started for each application
- The firewall's operating system may be exposed to attacks
- Traffic on encrypted channels cannot be inspected

7) Describe the screened sub-net architecture enlightening the role of each component involved. List also advantages and drawbacks of this architecture

Screened subnet architecture consists of 4 main components: DMZ, two screening router and an application gateway

- DMZ (de-militarized zone), i.e a network which is decoupled from both the external and the internal network. The DMZ is home not only to the gateway but also to other hosts (typically the public servers such as web, remote access etc.)
- The two packet filters are set up this way:
  - The first packet filter performs a first check on packets at network level. It is directly connected only with the DMZ and the external network. When a packet is accepted, it is forwarded to the second packet filter
  - The second packet filter performs a second check on packets at network level, it is connected to the DMZ and the internal network. When a packet is accepted it is then forwarded directly to the internal network, otherwise if checks at upper levels are required the packet is forwarded to the application gateway

  It is important to buy the two routers from different providers (differentiation is a crucial point in security), because otherwise there could be a single point of failure (if one router is bugged, then the other is bugged too because they both come from the same vendor)

- The application gateway is connected to the DMZ and performs additional higher level checks on packets coming from the second packet filter

In order to reduce costs and simplify management there's an alternative screened-subnet architecture, called "three legged firewall": in this solution the firewall is a general-purpose computer equipped with 3 network interfaces, connected to the DMZ, the external and the internal network. From a financial point of view the functionality is the same, but there's single point of failure.

Advantages:

- Screened subnet architecture is the most secure among firewall architectures
- No single point of failure
- Only packets that do not pass the second packet filter are checked by the application level gateway

Disadvantages:

- Expensive
- No double defense line if bad configurations are set

8) **A company wants to make an http service available to its customers on the port 8080. Explain what network/server configuration should be adopted to distribute the service in a secure way. Justify your answer**

The company should first of all put in place a firewall at its network boundaries: the best solution is to set up a screened-subnet firewall architecture and place the server providing the service on the DMZ. Adopting this configuration we have different advantages:

- Incoming requests to the server are first checked by the two packet filters at least, and also by the application gateway if required, protecting the server from some attacks at L3 or L4 (e.g SYN flooding attack)
- If the software running on the server has a bug, it cannot be exploited to gain access to the internal network, since the server is placed on the DMZ

9) **What are IDS and IPS? Which their differences?**

- Intrusion Detection System (IDS): system to identify actors using a computer or a network service without authorization. An IDS differs from a firewall in that a traditional firewall uses a static set of rules to permit or deny network connections. It implicitly prevents intrusions, assuming an appropriate set of rules has been defined. On the other hand, an IDS describes a suspected intrusion once it has taken place and signals an alarm. An IDS can be:
    - Passive: tries to identify attacks using pattern matching techniques and periodically checking the hash of files stored on the server. Passive IDS look for specific packets which are typical of a certain attack
    - Active: based on:
        - Learning: very accurate statistical analysis of the system behavior
        - Monitoring: active statistical collection of traffic, data, sequences, actions
        - Reaction: comparison against statistical parameters, i.e reaction when a threshold is exceeded
    - Host-based IDS (HIDS): the source of the statistics is the OS itself because the IDS is installed on the host
    - Network-based IDS (NIDS): monitor the traffic on the network to detect intrusions. A NIDS also captures and inspects every packet that is destined to the network regardless of whether it is permitted or not.
- Intrusion Prevention System (IPS): in a way, IDS are passive components which only detect and report without preventing. Meanwhile, IPS is a system used to speed-up and automate the reaction to intrusions, i.e it is the “fusion” of and IDS and a distributed dynamic firewall: The IDS sends an alarm if an intrusion has been detected, the distributed firewall receives the alarm and can block all the traffic. However, this solution can be dangerous because it may take the wrong decision and block innocent traffic
  Also IPSs can be host-based or network-based.

CHAPTER 7: E-MAIL

1) **Identify possible security problems related with the Mail Handling System (HMS) and list what techniques can be applied to mitigate/solve them**

Emails are connection-less since peers exchanging emails do not speak directly each other, so end-to-end solutions are not suitable for mail systems. Moreover, the destination for a mail could be a mail list: in this case encryption system can result very complex and hard to implement and manage, since a symmetric key is required for each mail sent. MHS is a store-and-forward system which is based on several parts, each one having different possible vulnerabilities that can be exploited:

- MUA (message user agent): is the software component that the user is using in order to send email. Typically it is housed in the user's computer and is referred to as a client email program or a local network email server. The author MUA formats a message and performs initial submission into the MHS via the MSA. The recipient MUA processes received mail for storage and/or display it to the recipient user.
- MSA (Message submission agent): after the mail has been sent it travels from the MUA to the MSA, which is another software component that has the task to inject the e-mail in the MTA. The MSA accepts the message submitted by a MUA and enforces the policies of the hosting domain and the requirements of internet standards. This function may be located together with the MUA or as a separate functional model. In the latter case, SMTP (simple mail transfer protocol) is used between the MUA and the MSA.
- MTA (message transfer agent): arranged as a chain. MTA relays mail for one application-level hop. It is like a packet switch or IP router in that its job is to make routing assessments and to move the message closer to the recipients.
- MS (message storage): it's the incoming mailbox. It can be located on a remote server or on the same machine as the MUA. Typically a MUA retrieves messages from a remote server using POP (post office protocol) or IMAP (internet message access protocol).

The main problems we have to face dealing with email services are:

- Untrusted MTA: you don't know who has a copy of your email
- Security of MS
- Mailing-list encryption
- Compatibility with what is already installed
- Authentication

The solution is to use S/MIME format, which provides authentication, integrity, non repudiation and optionally privacy be means of encryption, protecting the message while in transit and while resting on the mail server.

2) **Why does APOP represent an enhanced version of POP? How is it implemented?**
Post Office Protocol (POP) provides access via an internet protocol (IP) network for a user client application to a mailbox (maildrop) maintained on a mail server. The protocol supports download and delete operations for messages. POP has different versions, each one performing a different authentication scheme:
    - POP2 is obsolete because it performs authentication just by means of username
    - POP3 is obsolete too because it uses a password sent in clear
    - APOP is an enhanced version because it performs user authentication by means of a challenge

3) **Between MSA and MTA, where is it better to perform authentication in a mail system?**
It is better to perform authentication at MSA, before the message is injected in the mail transport system. There are many solutions:
    - ESMTP (extended SMTP): the ESMTP clients must identify themselves to the communicating parties with EHLO hostname. If the receiving server speaks ESMTP, it must declare the extensions that it supports, one per line, in its response to EHLO. Then the client chooses one authentication mechanism, thus the authentication protocol is executed

- Symmetric challenge-response methods, like CRAM-MD5 or DIGEST-MD5

4) **Explain how PGP works, its advantages and drawbacks**

Created by phil Zimmermann, PGP (pretty good privacy) is a software which provides authentication, integrity and confidentiality for data communication. It is based on public key cryptography but do not relay on the public key hierarchy: it uses a chain of trusted “friends” based on propagation algebra (transitivity property). Since there’s no certification authority, every public key can gain different level of trust depending on how many friends sign it (completely trusted, partially trusted, untrusted, unknown).

Public keys are stored individually by each user in his “key-ring” and are distributed by the owners at a PGP party or by a key-server where the public key along with all the related signatures are stored.

Whenever a peer wants to send a message, he generates a random key and symmetrically encrypts the message, then uses the receiver’s public key to encrypt the symmetric key and sends message + encrypted symmetric key. When the message is received, the receiver decrypts the symmetric key using his private key and decrypts the message with the symmetric key.

Advantages:

- PGP in its earlier version is unbreakable, since it uses very strong cryptography
- It does not need to relay on certification authorities

Drawbacks:

- It is not user friendly, since some technical knowledge is required
- If one key is compromised or corrupted, all the related keys are exposed
- PGP users can communicate with other PGP users only

5) **List anti-spam techniques from incoming mails (MTA) identifying advantages and drawbacks for each one**

There are several strategies:

1) DNSBL (DNS-based blacklist), which performs a check via DNS query (nslookup -p a.b.c.d.dnsbl.antispam.net) whether a sending host’s IP address a.b.c.d is blacklisted for email spam. If the answer is NXDOMAIN (no such domain) then it is not a spammer. Else the query returns 127.0.0.X where X is a code providing the reason for being black-listed.
   Since spammers tend to change MTA quite frequently, maybe those listed in DNSBL system are often “late”, so another strategy is not to look at the address of the MTA but to look at the content of the message and if the message contains an URI then a lookup about the repudiation of the URI is performed: if that URI is well-known to host phishing or malware distribution, then the whole mail is identified as spam
2) Greylisting: technique based on the hypothesis that spammers have scarce time. Usually spammers send messages to a huge number of users, so they need to send messages as fast as possible. Greylisting is implemented in two steps:
   a. When you are reached by a MTA you gibe out a temporary error (“try later”) and keep note of the MTA address
   b. If the same MTA comes back within a certain time interval T, then you accept the connection
3) Nolisting: also based on the scarce time hypothesis. This technique is used when the server cannot be loaded. If more than one mail exchanger MX (i.e the node which is designed inside the DNS as the contact point to receive mail) for a certain domain have been defined, then a set of priorities can be added.
   Hypothesis: spammers typically tend to send spam emails to MX with higher priority
   Strategy: at least 3 MXs are defined, each one with a priority level (high, medium, low), the first one and the last one don’t answer, while the middle one is the correct one.

4) DKIM (DomainKeys identified mail): email authentication method designed to detect forged sender addresses in email (email spoofing). DKIM allows the receiver to check that an email claimed to have come from a specific domain was indeed authorized by the owner of that domain. In this case the sender is using a signature to guarantee two things:
   a. The identity of the sender
   b. The partial integrity of the message

   This is done via digital signature created by the MSA or outgoing MTA. This digital signature covers some headers and part of the body and it is verifiable via a public key.

5) SPF (sender policy framework): technique in which a mail domain declares which are its outgoing MTA, via a specific record in the DNS. SPF allows the receiving mail server to check during mail delivery that a mail claiming to come from a specific domain is submitted by an IP address authorized by that domain's administrators.

6) With reference to formats for electronic documents digitally signed, describe the available solutions and discuss advantages and drawbacks

- Clear-signed: the message is sent in clear (so that anybody is able to read it, this could be a drawback if confidentiality is required) + digital signature (as an attachment or inside the message). Only the one who has a secure MUA can verify the signature. There could be some problems since the RFC 822 supports only ASCII characters on 7 bits (the 8th is the parity bit), so the MSB of any ASCII character can be cleared when passing through an MTA. This type allows viewers without S/MIME capabilities to read the message
- Signed: the message and the digital signature are base64 encoded. Only who has a secure MUA can decode and verify the signature. Encoding solves the ASCII problem described above.
- Encrypted/enveloped: base64(symmetrically-enc(message)): provides confidentiality, but application require to be modified.
- Signed and enveloped: base64(symmetrically-enc(message), pk-encryption(symmetric-key)). Provides confidentiality, authentication and integrity, also here application requires to be modified. Moreover, the message needs to be decrypted before performing authentication and integrity check

7) What is ESMTP? Explain how the AUTH command can be used

ESMPT is the extension of the Simple Mail Transfer Protocol. With ESMPT the client must authenticate with the mail server using an authentication mechanism. Whenever a client wants to send an email, he sends the message EHLO <hostname> to the mail server, which will respond with the possible authentication mechanisms available. The client replies with the AUTH command, specifying the authentication mechanism chosen. If the method chosen by the client is not supported by the server, the latter replies with error 504 (unrecognized authentication type). There are several authentication mechanisms:

- AUTH LOGIN: username and password are base64 encoded and sent separately to the mail server
- AUTH PLAIN: same as AUTH LOGIN but username and password are sent in a single message
- Symmetric challenge-response methods:
  - CRAM-MD5: the mail server sends a challenge which is a base64 encoded nonce to the client, which replies with base64(username + HMAC-MD5(password, nonce) in lower hexadecimal characters

Advantages:
→ client authentication by means of password
→ sniffing resistant because hash is not invertible
→ resistant to reply attack because the challenge is composed by a random number plus a timestamp plus FQDN (fully qualified domain name)
Drawbacks:
→ no server authentication
→ cleartext storage of the password, unless the intermediate steps of HMAC are stored
→ vulnerable to dictionary attacks
→ vulnerable to possible MITM attack after authentication because authentication is performed just when the channel is opened and not before and after every message

- DIGEST-MD5: similar to HTTP/1.1 digest authentication but it's obsolete

8) **What is S/MIME? What security properties does it offer and using which algorithms?**
S/MIME (secure multipurpose internet mail exchange) defines a secure format for messages, creating new parts in MIME multipart message, namely PKCS objects, always base64 encoded:
- Enveloped data: encrypted content of any type and encrypted symmetric keys
- Signed data: a digital signature is formed by taking the message digest of the message content and encrypting it with the private key of the signer. A signed data message can only be viewed by a recipient with S/MIME capabilities
- Clear-signed data: same as signed data, but here only the digital signature is encoded in base64, allowing recipients without S/MIME capabilities to view the message, although they cannot verify the signature.
- Signed and enveloped data: signed-only and encrypted-only entities may be nested, so that data which require only confidentiality are encrypted and data requiring only a signature are only signed. However both encryption and signature can be applied on the same message

Depending on the context, a PKCS object can be a well known one, like PKCS10 which is the standard for requiring a digital certificate or like the ones above, that belong to PKCS7 standard. Thanks to these functionalities, S/MIME is able to provide authentication, integrity, non repudiation and confidentiality.

Authentication, integrity and non repudiation are obtained thanks to digital certificates, while confidentiality is obtained thanks to symmetric encryption, possible thanks to key exchange. If we consider S/MIME v4.0 the following algorithms are implemented:

- For digital signature:
  - ECDSA with curve P-256 and SHA-256
  - EdDSA
  - RSA with SHA-256 is NOT used (while it is used in v3.2)
- For key exchange:
  - ECDH
  - NO RSA (mandatory in v3.2)
- For confidentiality:
  - AES-128-GCM and AES-128-CCM (AES-128-CBC is used in v3.2)

9) **Describe with a formula the structure of a S/MIME for a text T**
The text T can be protected with S/MIME in different ways:
- Encrypted: base64_encode(PKCS7envelope(T))

- Signed: base64_encode(PKCS7signed(T))
- Clear signed: MIME(T) + base64_encode(PKCS7signed(T))
- Signed and encrypted: base64_encode(PKCS7envelope(PKCS7signed(T)))