**Content**

[1. Computer security principles - Základné princípy počítačovej bezpečnosti 3](#_Toc40264465)

[1. Confidentiality – Utajenost 4](#_Toc40264466)

[2. Integrity - Integrita 5](#_Toc40264467)

[3. Availability - Dostupnost 6](#_Toc40264468)

[4. Further concepts 6](#_Toc40264469)

[2. Cyptography 8](#_Toc40264470)

[1. Symmetric Key Cryptography 10](#_Toc40264471)

[2.1.1. DES 10](#_Toc40264472)

[2.1.2. SSL 11](#_Toc40264473)

[2.1.3. AES 11](#_Toc40264474)

[2. Asymmetric Key Encryption 11](#_Toc40264475)

[2.2.1. Diffie – Helman 11](#_Toc40264476)

[2.2.2. RSA 12](#_Toc40264477)

[3. Use of cryptography to provide integrity 13](#_Toc40264478)

[2.3.1. Hashing algorithms 13](#_Toc40264479)

[2.3.2. Message Authentication Code (MAC) 13](#_Toc40264480)

[2.3.3. Digital signatures 14](#_Toc40264481)

[2.3.4. PKI – Public Key Infrastructure 14](#_Toc40264482)

[4. Comparison of symmetric and assymetric keys 14](#_Toc40264483)

[3. Identification and autentification 16](#_Toc40264484)

[1. Autentification 16](#_Toc40264485)

[2. Ideas what to talk about 17](#_Toc40264486)

[3. Summary 19](#_Toc40264487)

[4. Access control and security models 20](#_Toc40264488)

[1. DAC, MAC and RBAC 20](#_Toc40264489)

[2. Access control implementation 21](#_Toc40264490)

[4.2.1. Access control matrix. 21](#_Toc40264491)

[4.2.2. List based access control 21](#_Toc40264492)

[3. Security models 22](#_Toc40264493)

[5. Web Security 23](#_Toc40264494)

1. Computer security principles - Základné princípy počítačovej bezpečnosti

**Computer security** is the protection of the items you value, called the assets of a computer or computer system. There are many types of assets, involving hardware, software, data, people, processes, or combinations of these. To determine what to protect, we must first identify what has value and to whom. [hardware, software, and data].

**Value of the asstest:** after identifying the assets to protect, we next determine their value. We make valuebased decisions frequently, even when we are not aware of them.

A **vulnerability** is a weakness in the system, for example, in procedures, design, or implementation, that might be exploited to cause loss or harm.

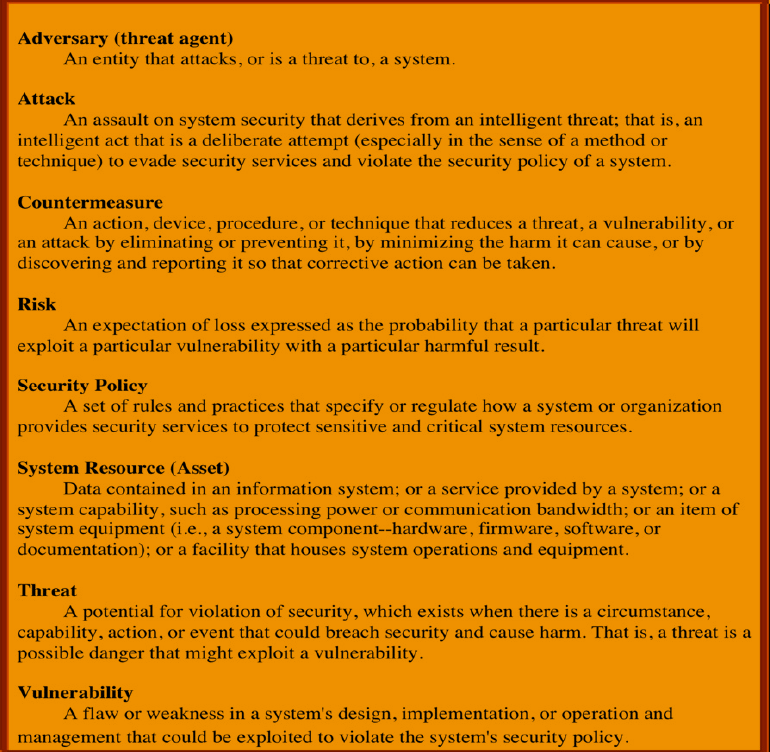
A **threat** to a computing system is a set of circumstances that has the potential to cause loss or harm.

A human who exploits a vulnerability perpetrates an **attack** on the system.

**Controls** prevent **threats** from exercising **vulnerabilities**.

Harm can also be characterized by four acts: **interception**, **interruption**, **modification**, and **fabrication**. **Confidentiality** can suffer if someone **intercepts** data, **availability** is lost if someone or something **interrupts** a flow of data or access to a computer, and **integrity** can fail if someone or something **modifies** data or **fabricates** false data.

**Computer security seeks to prevent unauthorized viewing (confidentiality) or modification (integrity) of data while preserving access (availability).**



* 1. Confidentiality – Utajenost

*the ability of a system to ensure that an asset is viewed only by authorized parties*

**Data** **confidentiality** : Assures that private or confidential information is not made available or disclosed to unauthorized individuals. **Privacy** : Assures that individuals control or influence what Information related to them may be collected and stored and by whom and to whom that information may be disclosed.

Nastroje pre zabezpecenia utajenia:

* + Sifrovanie
  + Kontrola Pristupu
  + Autentifikacia
  + Autorizacia
  + Fyzicka kontrola pristupu

A failure of data confidentiality:

* An unauthorized person accesses a data item.
* An unauthorized process or program accesses a data item.
* A person authorized to access certain data accesses other data not authorized (which is a specialized version of “an unauthorized person accesses a data item”).
* An unauthorized person accesses an approximate data value (for example, not knowing someone’s exact salary but knowing that the salary falls in a particular range or exceeds a particular amount).
* An unauthorized person learns the existence of a piece of data (for example, knowing that a company is developing a certain new product or that talks are underway about the merger of two companies).
  1. Integrity - Integrita

*the ability of a system to ensure that an asset is modified only by authorized parties*

**Data** **integrity** : Assures that information and programs are changed only in a specified and authorized manner. **System** **integrity** : Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.

Nastroje pre zabezpecenia integrity

* + Zalohovanie
  + Kontrolne Sumy

we say that we have preserved the integrity of an item, we may mean that the item is:

* precise
* accurate
* unmodified
* modified only in acceptable ways
* modified only by authorized people
* modified only by authorized processes
* consistent
* internally consistent
* meaningful and usable
  1. Availability - Dostupnost

*the ability of a system to ensure that an asset can be used by any authorized parties*

Nastroje:

* + Redundancie
  + Fyzicka ochrana

an object or service is thought to be available if the following are true:

* It is present in a usable form.
* It has enough capacity to meet the service’s needs.
* It is making clear progress, and, if in wait mode, it has a bounded waiting time.
* The service is completed in an acceptable period of time.

We can construct an overall description of availability by combining these goals. Following are some criteria to define availability:

* There is a timely response to our request.
* Resources are allocated fairly so that some requesters are not favored over others.
* Concurrency is controlled; that is, simultaneous access, deadlock management, and exclusive access are supported as required.
* The service or system involved follows a philosophy of fault tolerance, whereby hardware or software faults lead to graceful cessation of service or to work-arounds rather than to crashes and abrupt loss of information. (Cessation does mean end; whether it is graceful or not, ultimately the system is unavailable. However, with fair warning of the system’s stopping, the user may be able to move to another system and continue work.)
* The service or system can be used easily and in the way it was intended to be used. (This is a characteristic of usability, but an unusable system may also cause an availability failure.)
  1. Further concepts

**Authentication**: the ability of a system to confirm the identity of a sender

**Auditability**: the ability of a system to trace all actions related to a given asset.

**Anonymity**: It is a property that certain records or transactions do not belong to any individual.

The tools for ensuring anonymity are: **Aggregation** - a combination of data from multiple users. **Mixing** - aggregating information from multiple pages and combining it into folders that cannot be broken down. **Proxy** - trusted agents that replace the real identity of the user. **Pseudonym** - a fictitious identity of a user who pretends to be an identity.

**Authenticity**: The property of being genuine and being able to be verified and trusted; confidence in the validity of a transmission, a message, or message originator. This means verifying that users are who they say they are and that each input arriving at the system came from a trusted source. Proves, that data, process and rights of the system are true. Under Integrity. Main tool is Digital Signature.

**Accountability**: The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity. This supports nonrepudiation, deterrence, fault isolation, intrusion detection and prevention, and afteraction recovery and legal action. Because truly secure systems aren’t yet an achievable goal, we must be able to trace a security breach to a responsible party. Systems must keep records of their activities to permit later forensic analysis to trace security breaches or to aid in transaction disputes.

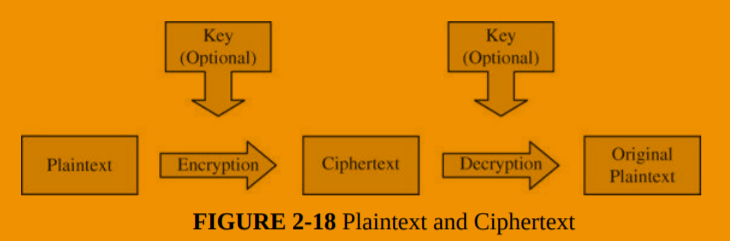
1. Cyptography

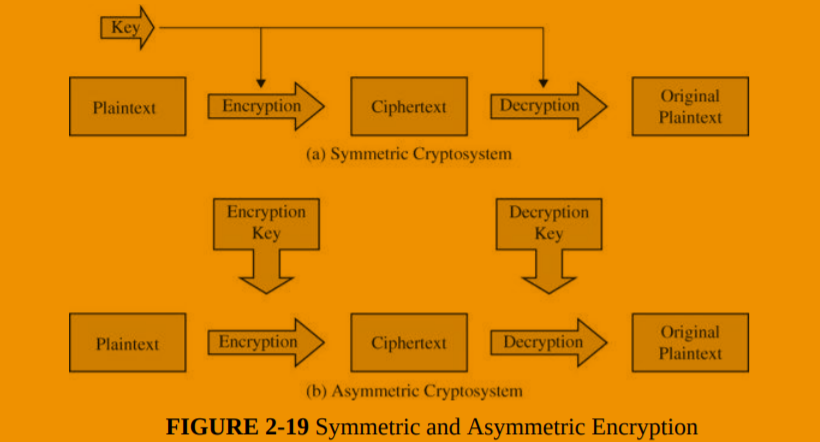
*Cryptography conceals data against unauthorized access.*

An important element in computer security is the use of cryptographic algorithms.

**Cryptography** - is a scientific discipline that deals mainly with the creation of ciphers, the aim of which is to hide sensitive data from unauthorized persons. Its goal is information systems security with a focus on:

* confidentiality - during data transfer, storage on media
* integrity (data integrity) - correctness of the content of the transmitted message
* authentication - confirmation of the sender's identity





**Work factor**: amount of effort needed to break an encryption (or mount a successful attack)

Applications:

* SSL and TLS (https://)
* VPN
* Hash on a downloaded file
* Digital signatures

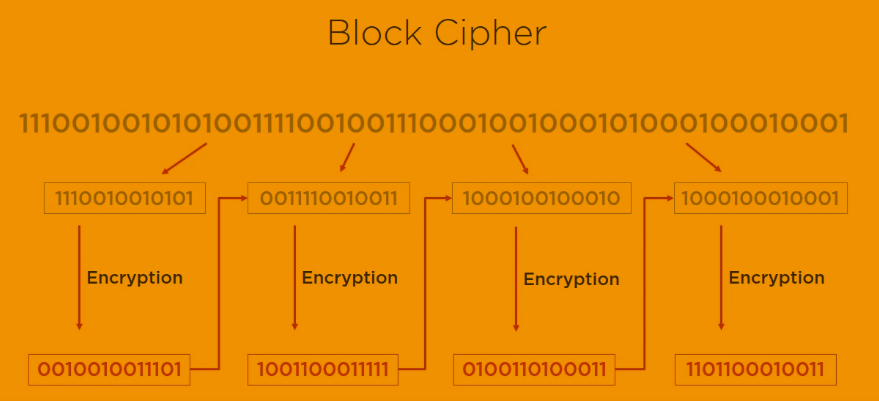
Cipher – a system used to create an encoded or secret message (not key)

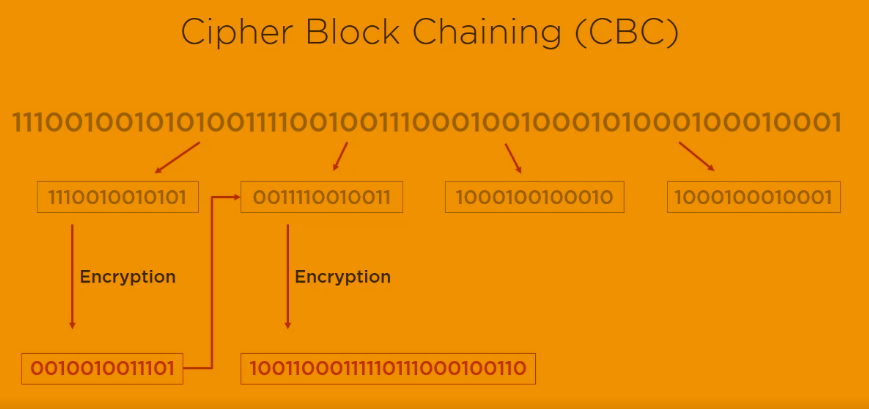
Cryptography and CIA:

* Confidentiality: file, message and link encryption
* Integrity: digital signatures, hashes

Stream cypher vs block cypher:

Stream cypher performs opearion on each bit, not on the whole string at once, or a proportion of it.





* 1. Symmetric Key Cryptography

This encryption system is called symmetric because the encryption key can be derived from the decryption key and vice versa. Most symmetric algorithms have the same encryption and decryption key. These algorithms, known as secret key algorithms, require both the recipient and the sender to agree in advance on the key they will use. Security is based on the secrecy of the key. Two basic conditions:

1. the need for a strong encryption algorithm

2. both the sender and the recipient must own a copy of the secret key and must keep it secret, otherwise this method does not make sense

This method has 5 basic parts: text, encryption, algorithm, secret key, encrypted text and decoding algorithm

* File encryption for transmision – AES256
* Encryption of files in storage
* Financial transactions – triple DES
* VPC encryption – AES256

Weaknesses:

* Method for secure transfer is needed
* Does not provide non-repudiation
* Key management can be difficult
* Not recommended: digital signatures, key transfer, web security, message security
  + 1. DES

A block cipher is an encryption/decryption scheme in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length.

Many block ciphers have a Feistel structure. Such a structure consists of a number of identical rounds of processing. In each round, a substitution is performed on one half of the data being processed, followed by a permutation that interchanges the two halves. The original key is expanded so that a different key is used for each round.

The Data Encryption Standard (DES) has been the most widely used encryption algorithm until recently. It exhibits the classic Feistel structure. DES uses a 64-bit block and a 56-bit key.

Two important methods of cryptanalysis are differential cryptanalysis and linear cryptanalysis. DES has been shown to be highly resistant to these two types of attack.

DES – 64bit key, 16 rounds, cracked in 1998

Tripe DES (quickfix) – 168bit key, 48 rounds. The changes made encryption time 3 times longer as well.

Blowfish and Twofish – free algorithms

* + 1. SSL
    2. AES

128, 192, 256 bit keys, 10;12;14 rounds. With fewer round and longer keys better perofrmance and higher security is achieved.

* 1. Asymmetric Key Encryption

Pros:

* Secure key exchange
* Authentication + Encryption
* Digital Signature
* Key exchange; Message encryption; Part of web security; Message authenication

Contra

* Slower than symmetric key cryptography
* Not a competition, use both types together
  + 1. Diffie – Helman

Uses a public and private key to generate symmetric key. The original algorithm was found vulnerable against man in the midle attack. If the attacker intercepts the public key he can send the client its own and read the messages. Can be solved by digital sigantures and an another level of authentification.

* + 1. RSA



Other: ECC (mbe even better than RSE, less vulnerable against quantum computing, google uses it in its certificate) , el gamal (usually the slowest, extendion of Diffie-Helman)

RSA (Rivest Shamir Adleman) - very widespread and used for asymmetric encryption

The public key is used for encryption and the private key is used for decryption

the encryption principle is that there is no inverse function to the encryption and decryption function that would allow you to decrypt the message with the same key that was used to encrypt the message

the public and secret key is generated by the addressee of the message to be transmitted in encrypted form, the public key is provided by the addressee to the sender of the message

* two large prime numbers p and q are generated
* let n = p.q - the number n is called the module
* let m = lcm [(p-1); (Q-1)] - lowest common multiple
* we find such a number e that the greatest common divisor of the numbers m and e is the number 1
* find another number d such that d.e mod m = 1
* the number e is called the public exponent and the number d is called the private exponent
* the public key is a pair (n, e)
* private key is a pair (d, e)
* at present it is very difficult to obtain the number d from the pair (n, e)
* if someone were able to factorize the number n to the primes p and q, they could calculate the number d and get to the private key
* RSA encryption security is therefore based on the assumption that factoring large numbers is extremely complex
* The discovery of a simpler factorization method could compromise the security of RSA encryption
* encryption security is therefore based on the great computational complexity of factoring large numbers
* For encryption to be secure, the p and q numbers must have at least 100 decimal digits

The base idea: there are mathematical operations which are easier to do in one way than into another. Simplified scenario:

907 and 773 – 2 primes

n = 701,111 = 907 x 773

It is so much easier to determine the results of 907 x 773, than to determine which 2 primes to multiplicate to get 701,111.

* 1. Use of cryptography to provide integrity
     1. Hashing algorithms

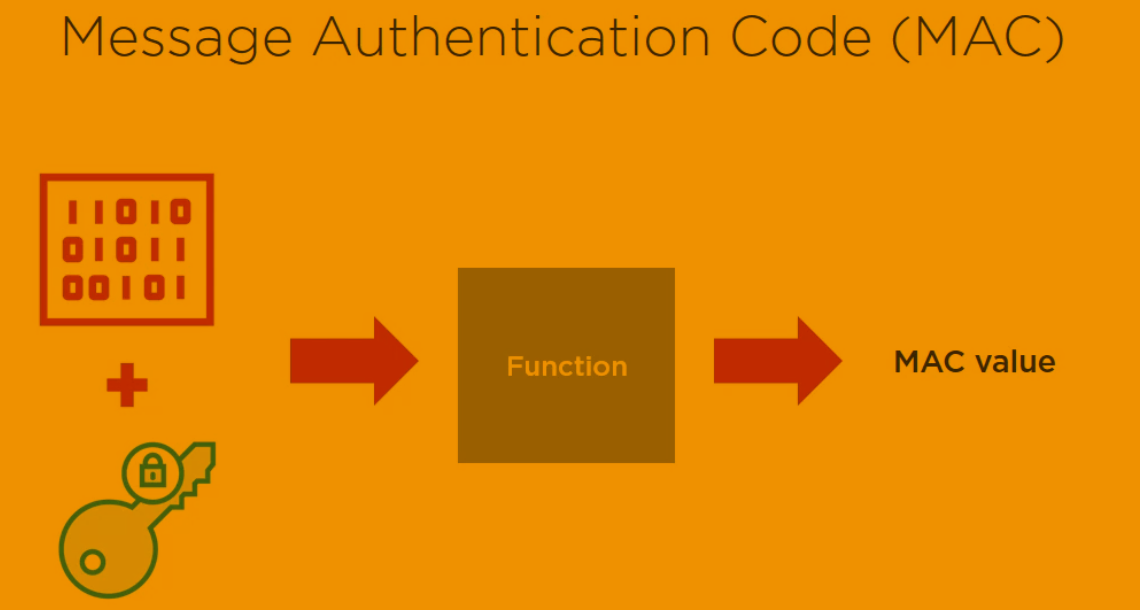
They take data and produce a unique hash value.

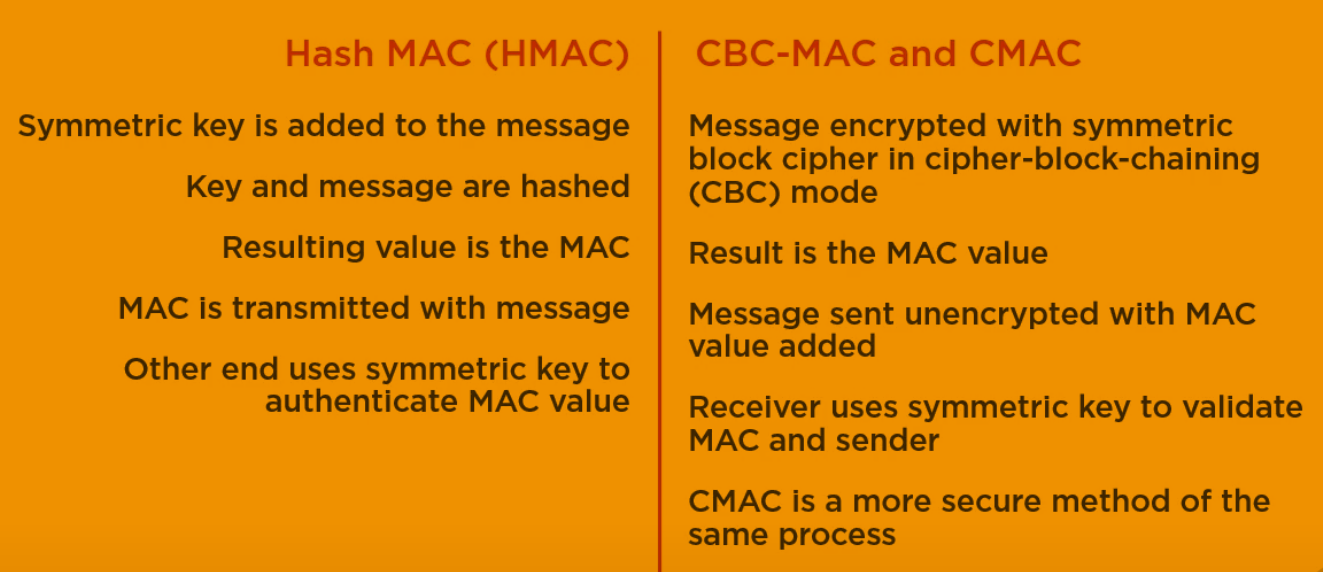
MD5 – vulnerability (different input provides the same hash value), not used anymore in SSL.

SHA-1 and SHA-256 (best) haval, tiger

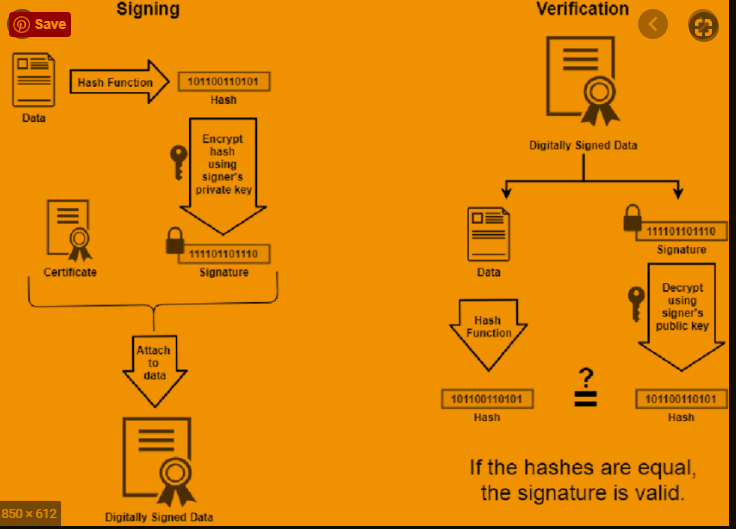
* + 1. Message Authentication Code (MAC)

The principle is to verify that the key / message sent is actually from the sender. (or authentic). It focuses on whether the content of the message has not been grounded, or whether the message is sent really from the source from which it should have been. It also checks the time and correct order of messages. It uses agreed encryption, so only the sender and recipient know the key.





* + 1. Digital signatures



Email: S/MIME – Secure Multipurpose Internet Mail Extention, standard for encrypting and digitally signing email. RDS + SHA (example).

PGP – Pretty Good Privacy, RSA + MD5. Examples of different types types of cryptography methods combined.

* + 1. PKI – Public Key Infrastructure

**Trusted third party** = **Certificate Authority** (internal or external) they provide **Certificate** using Registration Authority.

Similar to: You go to a certificate to verify that you say the truth when you say you know this and this.

* 1. Comparison of symmetric and assymetric keys

The problem with symmetric encryption is in key transmission. The K key must be transmitted through a medium. This has been one of the biggest priorities for international espionage in the past. It was no longer possible to transfer the key via an electronic channel, which is very easy to listen to. Physical transmission, on the other hand, is very slow. Asymmetric encryption solves this problem very effectively. Asymmetric encryption is a series of procedures in which we unambiguously convert the text T1 to the text T2 using the key Kn (n = 1,2). It consists of two parts. The first part (encryption) converts the text M to the text T using the key K1 (usually referred to as the public key). The second part (decryption) converts the text T to the text M, using the key K2 (usually referred to as the private key). In principle, no mathematical procedure can be used to obtain K2 from K1. The K2 private key is a key owned only by the person to whom the message is addressed. K1 is a public key that can be owned by anyone (so that person can provide it for download on the Internet). The text M encrypted with the key K1 can therefore only be decrypted with the key K2, which is only available to the person to whom the message is addressed (it follows that the text T to the text M cannot be decrypted even by the person who encrypted it because he does not have the private key K2 , required for this operation).

1. Identification and autentification

* Identification is the act of asserting who a person is.
* Authentication is the act of proving that asserted identity: that the person is who she says she is.

Identities are typically public or well known. Authentication should be private.

**Authorization** is the allocation of permissions for specific types of access to restricted information. In the real world, **authorization** is conferred on real human beings; in contrast, information technology normally confers authorization on user **identifiers** (IDs). Computer systems need to link specific Ids to particular authorized users of those IDs. Even inanimate components, such as network interface cards, firewalls, and printers, need IDs. Identification is the process of ascribing an ID to a human being or to another computer or network component. Authentication is the process of binding an ID to a specific entity. For example, authentication of a user’s identity generally involves narrowing the range of possible entities claiming to have authorized use of a specific ID down to a single person.

* 1. Autentification

Authentication is based on something you know, are, have or do.Authentication mechanisms use any of three qualities to confirm a user’s identity:

* What only you know (passwords and passphrases)
* What only you have (tokens: physical keys, smart cards)
* What only you are (static biometrics: fingerprint, face, retina, and iris recognition)
* What only you do (dynamic biometrics: voice, handwriting, and typing recognition).
* Where you are (terminals with strategic plases, GPS signals)

Every password can be guessed; password strength is determined by how many guesses are required.

Problems with passwords:

* Forgotten passwords - saving in sealed envelopes in the vault, sending the password to e-mail ...)
* Password guessing - it is necessary to choose a strong password
* Intentional and unintentional disclosure of a password
* They can be stolen without knowledge of the user

**How to store passwords**: An encrypted system file is a good way to prevent unauthorized password browsing. It is best to use a one-way cipher to store the password. When a user enters their password, it is encrypted in the same way and then compared. It is necessary that the selected cipher does not provide the same result for different input words. If two users choose the same password, they will easily find out after looking at the password file, as their encrypted passwords will match. Therefore, in Unix + systems, the so-called "Salt" to expand the password. Salt is a 12-bit number composed of system time and process identifier. Salt is unique to each user, and can be stored unencrypted. The encrypted password then consists of the password itself in conjunction with the "salt". Encrypted in this way, even if the same passwords will certainly be different.

Rainbow Table (database of most frequent passwords) or Brute Force attacks (trying all the possbile combination) are examples of how authentication can be attacked.

Defensive mechanisms on top of password usage: fraud detection (identify unusual activity and take actions when they happen), MFA (multi factor authentication – using mobile application for example) , security questions.

authentication based on **biometrics** (for example fingerprint) – what you are. Costly.

SSO – Single Sign On: the user authenticates once per session, and the system "forwards" this authenticated identity to any process that requires authentication. Of course, such a single-login is not more secure than a single login. The weak point is also that if someone unauthorized authenticates, he can abuse other services. Microsoft Active Directory – a solution for SSO. The advantage of this solution is that reduces the number of passwords required. Secondly, it is easier to deny access for a user to the system (just change in one place, the Active Directory).

* 1. Ideas what to talk about
* Risk of Undetected password theft
* Risk of Undetected password sharing
* Risk of Weakest – by finding out the weakest password other passwords can be revealed
* Dictionary attacks
* Risk of Online Guessing – based on exploiting best practices and personal information
* Risk of Off-Line Dictionary Attacks – checking against hash
* Risk of Password Replay

1. The hash is sufficient for a dictionary attack unless a salt is used and kept secret.
2. The attacker does not even need to recover the password. Instead, the attacker can replay the hash of the password when needed.
3. send passwords from client to server encrypted using the server’s public key !!

Risk of Server Spoofing

Risk of Password Reuse

Authentication Using Recognition of Symbols – pictures the user know

TOKEN-BASED AUTHENTICATION:

1. Card Entry Systems
2. Proximity and Touch Cards
3. Smart Cards and Dongles
4. Soft Tokens
5. One-Time Password Generators
6. Authentication Using Mobile Devices - SMS

BIOMETRIC AUTHENTICATION

CROSS-DOMAIN AUTHENTICATION - SAML

RELATIVE COSTS OF AUTHENTICATION

TECHNOLOGIES

* 1. Summary

Passwords are widely used in practice and will continue to be a dominant form of user authentication. There are many risks in deploying passwords, and a number of widely used password systems have serious vulnerabilities. Nonetheless, technical measures can mitigate the inherent vulnerabilities of passwords. Although it takes great skill and care, with our current understanding it is technically possible to build and deploy strong password-based authentication systems using commercial products. The truly inherent risks of undetected theft and undetected sharing can be largely mitigated by new technologies, such as intrusion detection systems. Undetected sharing may be deterred further by a system that couples high-value secret data, such as credit card account numbers, with passwords. Tokens are available to generate one-time passwords or to communicate directly with authentication systems. Although costs have been dropping, tokens are still not as widely deployed as early predictions suggested they would be. Biometric authentication has been implemented only infrequently and on a small scale but offers great potential, especially for high-security applications. Interesting new research and applications are extending the use of authentication (and authorization) over untrusted networks between federated organizations.

* + - 1. Salting

In cryptography, a salt is random data that is used as an additional input to a one-way function that hashes data, a password or passphrase.



A fixed salt is when a programmer uses the same salt for every hashed password.

While this will make current rainbow tables useless (if the salt is properly chosen), if the salt is hard-coded into a popular product that salt can be extracted and a new rainbow table can be generated using that salt.

Using a single fixed salt also means that every user who inputs the same password will have the same hash (unless the password hash is also dependent on the username). This makes it easier to attack multiple users by cracking only one hash.

* + - 1. Pepper

In cryptography, a pepper is a secret added to an input such as a password prior to being hashed with a cryptographic hash function.

1. Access control and security models

* A **subject** is an active entity, such as a process or a user.
* An **object** is a passive entity, such as a file.
* A **right** describes what a subject is allowed to do to an object; for example, the read right gives permission for a subject to read a file.
* The **protection state** of a system simply refers to the rights held by all subjects on the system.
* **Access control** – its goal is to prevent an unauthorized user (subject) from accessing the resource (s), including preventing the use of resources in an unauthorized manner.
* **Access Control Policy (ACP)** – describes what type of access is allowed and under what circumstances.
* **Audit** - independent review and review of system records and activities to test system controls. The aim is to identify breaches of security policy, changes in management and procedures.
* **Autorization –** process of validation if XY has rights to Z
* **Authentification** – validation of the identity of the subject
  1. DAC, MAC and RBAC

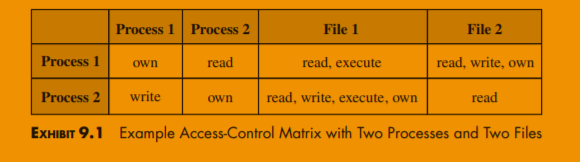
Discretionary access control (DAC) – gives the owner (or anyone authorized to decide on access to the object) the freedom to decide about the object access control. The owner can assign access rights to the object and can decide which rights to assign. Rights can be assigned to the whole group, but also to individual users. Typically, DAC access rights can be changed dynamically. MAC and DAC can also be applied to an object at the same time. In this case, the MAC takes precedence over the DAC. This means that anyone who has permission to access the object through the MAC, and also has permission through the DAC, can actually access the object.

Mandatory access control (MAC) - means that security policy decisions are made outside the property owner. The central authority decides what information is accessible to whom and the user cannot change the access rights in any way. Defines the access of subjects to objects based on the classification hierarchy of labels. Each object and subject in the system has its own designation. Access to objects is based on a comparison of the designations of the accessing entity and the given object. The control is statically secured. The obligation lies in centralized decision-making based on labeling. Entities cannot influence the decision.

Role-based access control (RBAC) - Responding to the problem of a large number of definitions of access rights due to the large number of objects and entities. Simplify administration, increase performance, simpler scalability of the system (adding, removing objects and entities). Users are assigned to roles. Objects are assigned to groups. Roles have defined rights and can be organized hierarchically with the support of inheritance rights

* 1. Access control implementation
     1. Access control matrix.

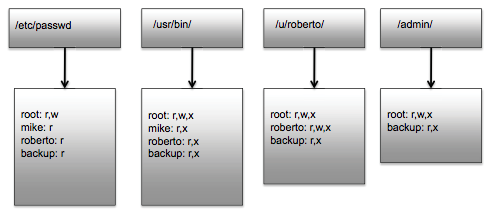
Easy, but not scalable (in case we have large nuber of objects or subjects).



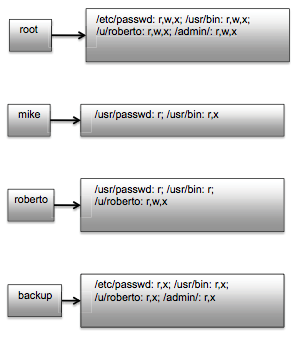
* + 1. List based access control

List based on the objects.

Easy to implement, but it is hard to find which objects belongs to a given subject. (list all files where mike have write access)



The list can be based on the subjects. Harder to implement.



* 1. Security models
* Bell-La Padula (BLP) model – want to preserve **CONFIDENTIALTY, MILLITARY ORIGINS**

„no read up“ - A person is not allowed to read something which has higher security level than he has access to (an officer is not allowed to read the field marshall’s notes).

„no write down“ – A person is not allowed to publish something into security category lower than the information requires. (field marshall is not allowed to talk about the strategic decisions to the officer).

* Biba model – wants to preserve **INTEGRITY**

„no read down“ – a higly secure datacenter cannot rely on the information from untrusted source.

„no write up“ – an untrusted source cannot publish information into secure datacenter.

1. Web Security

From: [file:///C:/Users/nbarto/Desktop/Statne%20Skusky%202020/02-knihy/Stallings\_Cryptography\_and\_Network\_Security.pdf](file:///C:\Users\nbarto\Desktop\Statne%20Skusky%202020\02-knihy\Stallings_Cryptography_and_Network_Security.pdf)

* Secure socket layer (SSL) provides security services between TCP and applications that use TCP. The Internet standard version is called transport layer service (TLS).
* SSL/TLS provides confidentiality using symmetric encryption and message integrity using a message authentication code.
* SSL/TLS includes protocol mechanisms to enable two TCP users to determine the security mechanisms and services they will use.
* Secure electronic transaction (SET) is an open encryption and security specification designed to protect credit card transactions on the Internet.

**Phishing**

fake websites created to fraudulently obtain sensitive information The user usually becomes a victim of phishing via unsolicited email. Most targeted pages:

* Financial services (Citibank, ..)
* Payment services (PayPal, ..)
* Auctions (ebay, ..)

How to prevent the detection of URLs with typos, fogging of URLs, deleted or fake URLs line.

Škodlivý kód

Programová bezpečnosť

Bezpečnosť operačných systémov

Bezpečnosť databázových systémov

Bezpečnosť v počítačových sieťach

Webová bezpečnosť

Forénzna analýza IKT