english

MOVET

Modernisation of VET through
Collaboration with the Industry

Ivan Pravda

Network security

Erasmus+

This project has been funded with support from the European Commission.
This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Title: Network security
Author: Ivan Pravda
Transalted by: Michal Łucki
Published by: Czech Technical University of Prague
Faculty of electrical engineering
Contact address: Technicka 2, Prague 6, Czech Republic
Phone Number: +420 224352084
Print: (only electronic form)
Number of pages: 41
Edition: 1st Edition, 2019

MoVET
Modernisation of VET through
Collaboration with the Industry

https://movet.fel.cvut.cz

This project has been funded with support from the European Commission.
This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Explanatory Notes

	Definition
	Interesting
	Note
	Example
	Summary
	Advantage
	Disadvantage

ANNOTATION

This teaching module deals with network security capabilities with a focus on Virtual Private Networking (VPN). It defines a number of basic concepts, it includes the description of some basic components and concepts of VPNs. In addition, attention is paid to the explanation of the IPSec protocol and to the mechanisms allowing the implementation of private network security, such as the ISAKMP/IKE method and the Diffie-Hellmann key exchange mechanism. Last but not least, the module contains a number of practical examples and solutions for them. The final part of the module describes electronic signature.

OBJECTIVES

By studying the module students will get an overview of the problem of computer networks' security provided by virtual private networks. The topic is very current, as the concept of security is very closely related to cybercrime. Emphasis is placed not only on the clarification of terminology in this field, but also on the explanation of the principle of basic procedures, supplemented by specific examples of implementation. The final part explains electronic signature and its implementation in everyday life.

LITERATURE

[1] Deal, Richard. The Complete Cisco VPN Configuration Guide. Cisco Press, 2005. 1032 pages. ISBN: 978-1-58705-204-0.

[2] Cisco Systems. Clientless SSL VPN (WebVPN) on Cisco IOS with SDM Configuration Example. 2009. https://www.cisco.com/c/en/us/support/docs/security/ssl-vpn-client/70663-webvpn.html [online]

[3] RFC4301 - Security Architecture for the Internet Protocol http://tools.ietf.org/html/rfc4301 [online]

[4] RFC4302 - IP Authentication Header http://tools.ietf.org/html/rfc4302 [online]

[5] RFC4303 - IP Encapsulating Security Payload (ESP) http://tools.ietf.org/html/rfc4303 [online]

[6] RFC4308 - Cryptographic Suites for IPsec http://tools.ietf.org/html/rfc4308 [online]

[7] RFC4364 - BGP/MPLS IP Virtual Private Networks (VPNs). http://tools.ietf.org/html/rfc4364 [online]

[8] RFC4835 - Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH) http://tools.ietf.org/html/rfc4835 [online]

[9] RFC5246 - The Transport Layer Security (TLS) Protocol Version 1.2. http://tools.ietf.org/html/rfc5246 [online]

Index

1 Virtual Private Network - definition of basic concepts

FORMAL DEFINITION

A Virtual Private Network **VPN** (*Virtual Private Network*) is a communication environment in which access to communication between individual entities is controlled. The communication environment is created on the basis of a predefined form of distribution of the common communication medium, which is further able to provide network services on a non-exclusive basis.

NON-FORMAL DEFINITION

A Virtual Private Network **VPN** is a non-public (computer) network built within a public network infrastructure, such as the Internet. Typically, this network provides secure connection of remote branches or subscribers to the parent network.

From the previous definitions, it can be briefly stated that a VPN is essentially a logical network within a shared public infrastructure. It provides the same performance and rules like any private **LAN** (*Local Area Network*).

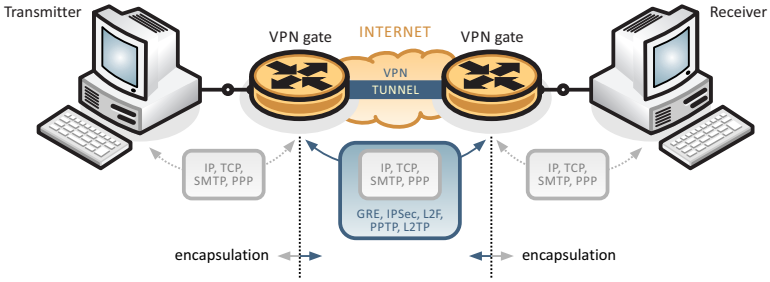
A major problem with the use of VPNs is to ensure their security and service delivery in the required quality and with respect to **QoS** (*Quality of Service*) indicators. Both of these requirements do not address network infrastructure based on **TCP/IP** (*Transmission Control Protocol/Internet Protocol*).

The security requirements are addressed by the VPN design:

- tunneling,
- encrypting,
- authentication and
- access control.

The term "tunneling" is understood as a process of encapsulating an original packet into another. The original packet is unreadable for all intermediate devices during its transmission.

The reason for the implementation of tunneling is to ensure safety and to create a transport mechanism between geographically remote locations. Encapsulation can for example use **GRE** (*Generic Routing Encapsulation*), **IPSec** (*Internet Protocol Security*), **L2F** (*Layer 2 Forwarding*), **PPTP** (*Point-to-Point Tunneling Protocol*), **L2TP** (*Layer 2 Tunneling Protocol*).



Tunneling mechanism in VPN

However, tunneling can also be used to customize incompatible protocols, such as LAN interconnection with **NetBEUI** (*NetBIOS Extended User Interface*) or **IPX** (*Internetwork Packet Exchange*) over the Internet (IP protocol).

In reality, it is possible to implement so-called Split Tunneling, where the client can simultaneously communicate inside the VPN and with the Internet.

The term "encryption" refers to the process of ensuring confidentiality and data integrity. Technically speaking, it means encapsulating data into a secure envelope, i.e. encryption with a secret key.

Authentication within VPNs ensures verification of authenticity. It makes sure that data really come from the source for which it is claimed claim that they are coming.

Shared key-based schemes, such as **CHAP** (*Challenge Handshake Authentication Protocol*), **RSA** (*Rivest-Shamir-Adleman*) signature, and others are used. Beyond security, these systems also provide data integrity.

Access control allows you to restrict access or intrusion of unauthorized users associated with the verification process of particular user's rights.

2 VPN Components

VPNs use security encryption tunneling protocols to provide packet sniffing protection, they guarantee adequate authentication, and declare the integrity of the messages, i.e. their entirety.

The components necessary to build a VPN connection are:

- an existing LAN or a separate terminal (such as a PC, notebook, netbook, etc.)
- available Internet connection,
- VPN Gateways (e.g., routers, firewalls, VPN concentrators), and
- appropriate software (Software) needed to build and manage VPN tunnels.

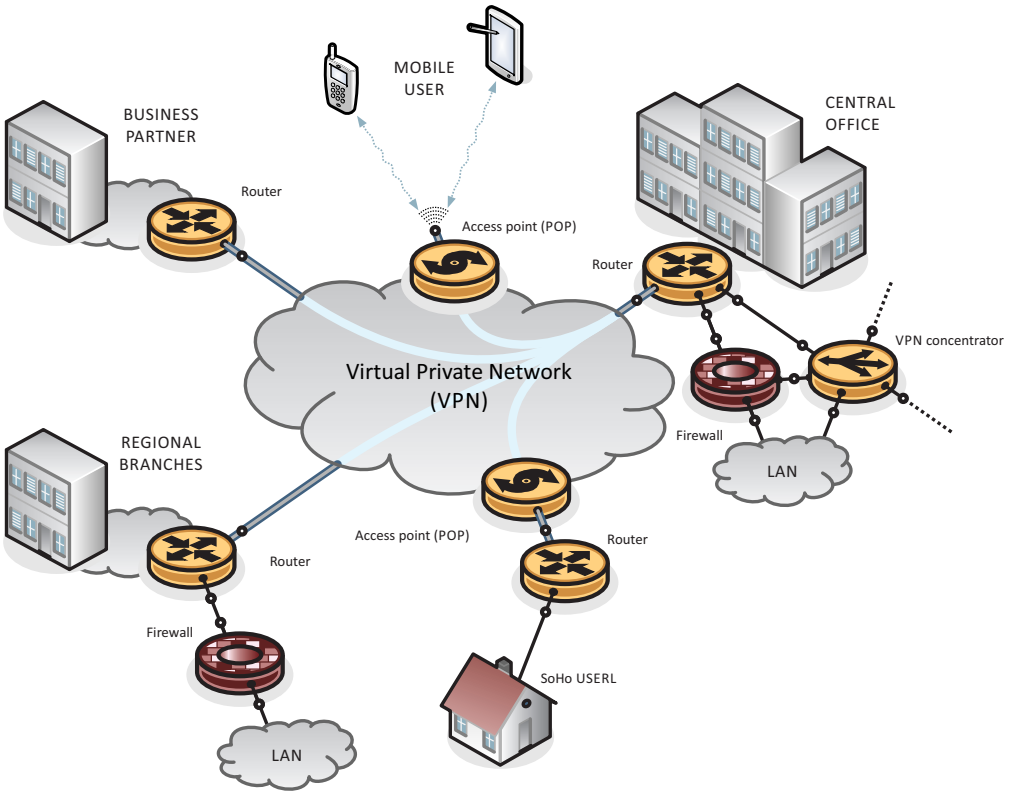
1. Site-to-Site or LAN-to-LAN connectivity

This type of VPN connection is used to connect geographically sparse locations in a similar way as if they were connected by a leased line or another **WAN** (*Wide Area Network*), (e.g. Frame-Relay, **ATM** (*Asynchronous Transfer Mode*)). The advantage of such a link is sharing a corporate intranet or an extranet with a partner. In this topology, users send and receive data via a VPN gateway, usually a router or a server. The VPN gateway is responsible for encrypting outbound traffic and routing it to the VPN tunnel through the Internet to the opposite VPN gateway of a target network. This VPN gateway removes the packet header, decrypts its content, and delivers the packet to a target user within the target network.

2. Remote-Access

Terrain workers or home workers use remote access VPN connections a lot. In the past, these remote workers had a connection by telephone lines, which meant a low transmission rate associated with high operating costs. At present, however, most of them have fast internet access directly from home through broadband technologies and can build high-quality VPN connections.

Each user typically has a VPN client installed, that is, a software that encapsulates and encrypts packets before sending them over the Internet to the destination VPN gateway. This software makes it much easier to connect, because the user only needs basic knowledge to build a high-quality VPN connection.



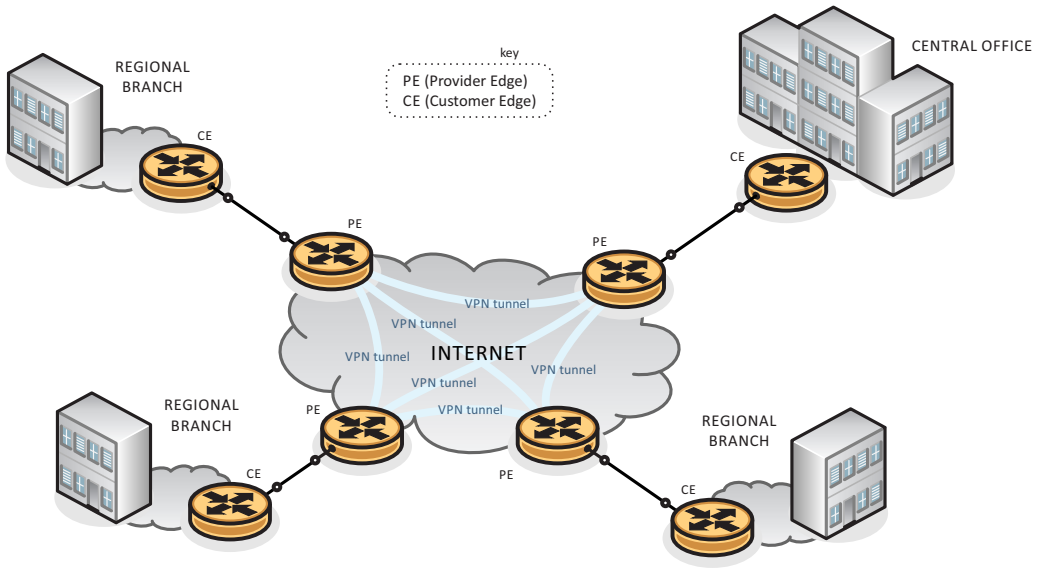
VPN connectivity options

3 VPN Classification according to RM-OSI

1. VPN based on a provider's device (PE-based VPN)

PE (*Provider Edge*) is the boundary device of the **ISP** (*Internet Service Provider*), which include routers, switches or devices that are a combination of both.

The PE device participates in routing and forwarding traffic based on the customer's address range. Data is typically transmitted between PE devices via VPN tunnels created using **MPLS** (*Multi Protocol Layer Switching*), IPSec, L2TPv3 or GRE. In this case, **CE** (*Customer Edge*) devices do not recognize that they are part of a VPN.



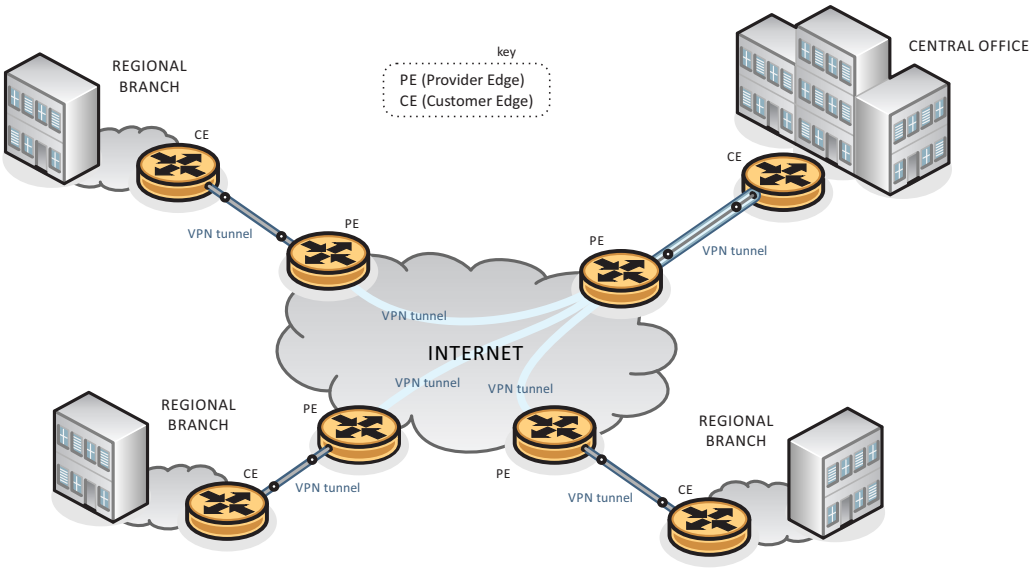
VPN layout based on the device of the provider

VPN tunnels are terminated at the PE boundary router and are usually configured as permanent.

2. VPN based on customer equipment (CE-based VPN)

The CE device is a customer boundary device connected to the PE device.

PE devices in this mode do not distinguish the type of traffic, VPN connections are handled by the CE device that routes and sends user traffic. Tunnels are created between the CE devices based on IPSec or GRE.



VPN layout based on customer equipment

The CE devices (VPN gateway) usually have some other features for VPN clients (such as **DHCP** *(Dynamic Host Configuration Protocol*), **DNS** (*Domain Name Server*)). This solution generally puts higher demands on client authentication, as they connect anytime and anywhere.

4 IPSec protocol - description

IPSec protocol is a comprehensive set of protocols for encryption, authentication, data integrity, and tunneling. The security is implemented on the **OSI** reference network layer (*Open System Interconnection*), and therefore it provides transparent security for any transmission or network application.

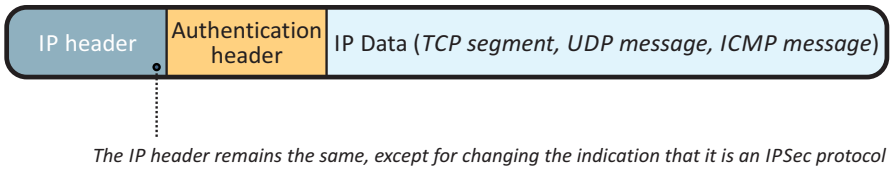
Basic IPSec components include:

- security protocols - **AH** (*Authentication Header*), **ESP** (*Encapsulating Security Payload*),
- protocols for exchange of keys - **ISAKMP** (*Internet Security Association and Key Management Protocol*), **IKE** (*Internet Key Exchange*),
- Assistance Databases - **SPD** (*Security Policy Database*), **SAD** (*Security Association Database*), and
- **DOI** (*Domain of Interpretation*) – it contains different values, such as identifiers and indicators for **SA** (*Security Association*)

IPSec offers two working modes:

1. Transport mode - for Host-to-Host connection

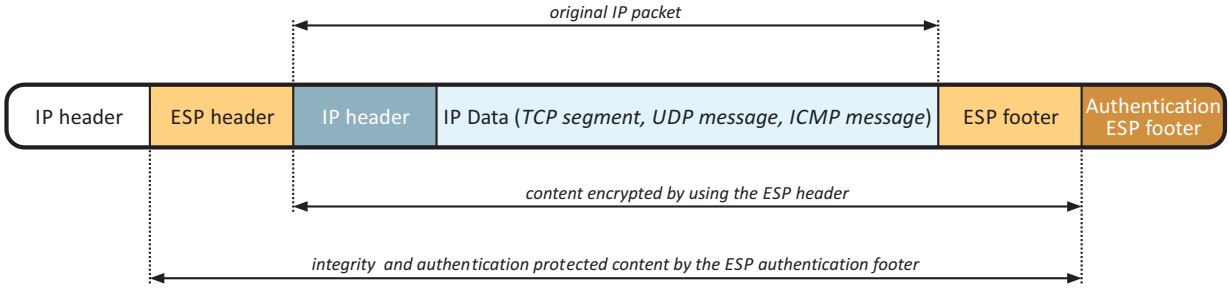
In a transport mode, only the content of a given IP packet is usually encrypted or authenticated. Routing information remains unchanged, unless the IP packet header is modified or encrypted. When an **AH** (*Authentication Header*) is used, IP addresses cannot be translated, because the hash value is always lost. Transport and application layers are always secured by hash function, so they cannot be modified (e.g. by changing the port number).



Structure of the IPSec packet in transport mode using the AH header

2. Tunneling mode - designed primarily for site-to-site connections

In the tunneling mode, the entire IP packet is encrypted or authenticated by **ESP** (*Encapsulating Security Payload*). It is then encapsulated in a new IP packet with a brand new header using the AH authentication header. This mode is used to create VPNs for communication between individual Site-to-Site networks (e.g., between routers linking different networks), Host-to-Site communications (e.g., remote user access) and Host-to-Host communications e.g. private chat).



Structure of the IPSec packet in tunneling mode using ESP

The tunneling mode supports **NAT** (*Network Address Translation*) and **PAT** (*Port Address Translation*).

IPSec does not contain in its header any field for the specification of an operating mode. The operating mode is set based on the value of the Next Header field ("IP" value specifies the tunneling mode; "TCP, UDP, ICMP" values (or other) identify the transport mode).

The benefits of IPSec include its transparency, there is no need to modify higher layer protocols, IPsec can secure any IP protocol, it secures “older” protocols that are unsecured and is widely supported by **HW** (*Hardware*) a **SW** (*Software*).

Disadvantages of IPSec include overhead, necessity of installing the client in case of remote access. It does not deal with user authentication; problematic NAT and PAT (possible to use only in the tunneling mode) and multicast and broadcast traffic.

IPSec protocol:

- provides network layer traffic,
- is universal for securing any TCP/IP traffic,
- protects from Packet Sniffing network layer traffic analysis,
- is suitable for fixed remote users,
- does not support multicast and broadcast transmission,
- exhibits address translation problems (NAT and PAT) - the address field protected by **HMAC-SHA1** (*Hash Message Authentication Code - Secure Hash Algorithm*) is changed; the solution is to pack the IPSec packet into the **UDP**

datagram (*User Datagram Protocol*) → the **NAT-T** method (*NAT-Traversal*), and:

- in case of remote access, client installation is required (but there may be compatibility issues with different implementations).

5 IPSec key exchange - ISAKMP/IKE method

Exchange of keys between clients before starting their own secure communications is important from several points of view. However, the question arises: how to deal with a secure key exchange? For the communication purposes, it is necessary to ensure:

1. Agreement on the type of key and the way it is created, i.e. to establish a shared key - **PSK** (*Pre-Shared Key*)
2. Authentication of participants, i.e. mutual authentication of the participants of the communication
3. Protection of identity of participants, i.e. the passive attacker should not be able to reveal the identity of participants by simply monitoring the communication
4. **DoS** (*Denial of Service*), i.e. a malicious user should not be able to abuse the protocol to force the counterpart to waste resources (**CPU**, (*Central Processing Unit*), memory, storage capacity, ...)

The ISAKMP protocol is defined by RFC 2408. It uses UDP transport protocol on port 500 for operation.

ISAKMP is a general protocol for generating SAs, i.e. it does not address how to replace authenticated keys. This is the task for the IKE protocol. The ISAKMP is used to authenticate communicating parties and exchange data for encryption keys.

This is not a Client-Server communication, but a Call-Response type. The party that wants to create a new SA initiates communication with the ISAKMP protocol.

IKE is a flexible "negotiation" protocol defined by the RFC Recommendation RFC 2409. It allows the negotiation of a specific authentication method, encryption, key lengths and their secure exchange. To do this, it uses the Diffie-Hellman algorithm (D-H algorithm).

The IKE protocol is used to exchange session keys, called Session Keys. IKE messages are encapsulated in ISAKMP packets.

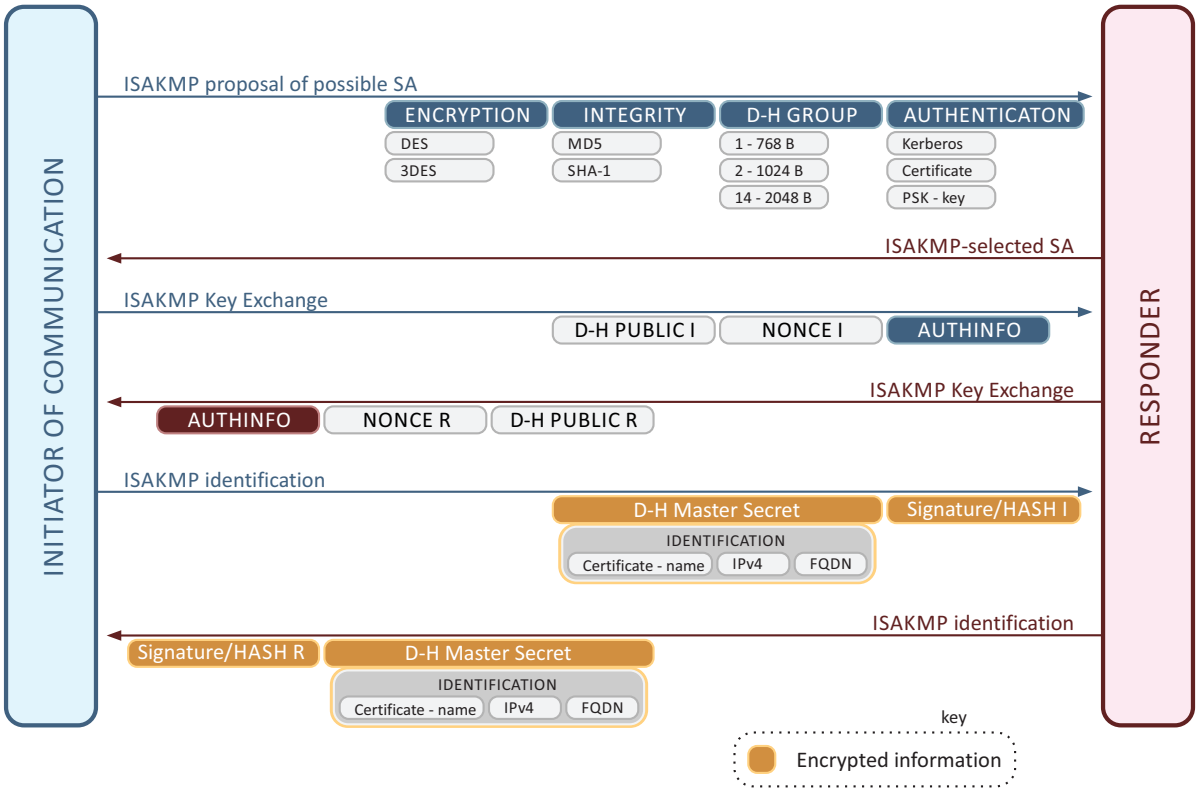
The IKE protocol can be divided into two independent phases. The first phase builds a secure authenticated channel between communicating entities (computers). Within this phase, the identity of the communicating parties is authenticated in a protected way. Both communicating parties agree on the use of SA and make an

authenticated PSK shared key exchange. Subsequently, a safe tunnel for the second phase is established. Two modes are available to create the tunnel:

- Main Mode
  - it arranges algorithms and hashing functions, it generates shared secrecy using the D-H algorithm, and verifies counterparty identity. In total, there are 6 reports.
- Aggressive Mode
  - it shortens negotiation into a smaller number of packets. There are 3 reports in total.

Some advantages of the aggressive mode are the bandwidth savings and time required for message transfer.

One disadvantage of the aggressive mode is the exchange of important information before the encrypted connection is established, which is susceptible to interception, known as Sniffing.



Process diagram IKE protocol stage 1 (main mode)

In the first phase, it is possible to use 4 different ways of exchanging the PSK key:

- asymmetric public key encryption (original version)
- asymmetric public key encryption (enhanced/improved version)
- digital signature
- secret key (symmetric algorithm)

Each key exchange option can be used in the main or aggressive mode, i.e. there are altogether 8 different options for the first stage of the IKE protocol! The main mode must always be implemented, the aggressive mode is optional, i.e. it should be implemented.

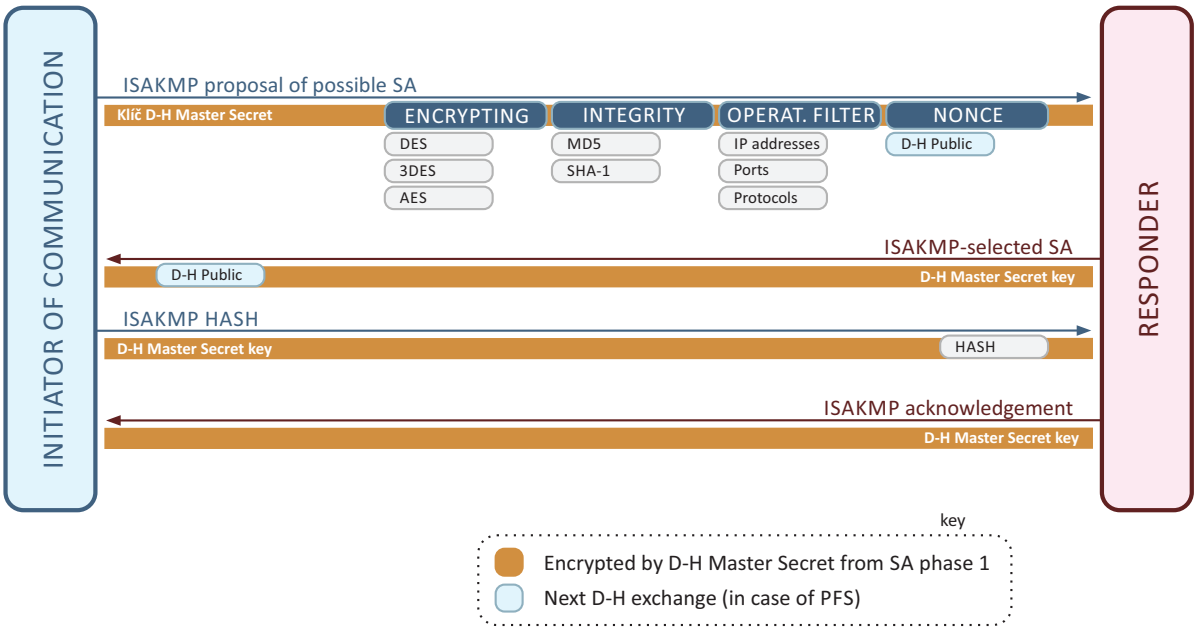
The result of the first stage of the IKE protocol is the mutual authentication of the communicating parties, the exchange of the shared symmetric key PSK and establishing the IKE Security Association (SA).

The second phase (so-called Quick Mode) creates an SA for IPSec session, i.e. SA IPsec connection parameters are established, IPSec SA is established for a specific connection (e.g. FTP, telnet, etc.) Optionally, additional D-H exchanges are made and other material is specified for the communication purposes.

This communication is protected from the very beginning by using algorithms and keys obtained during the first phase.

To encrypt conventional-type communication, a Session Key derived from the D-H Master Key obtained from Main Mode SA and from Nonce given by Quick Mode SA.

PFS (*Perfect Forward Secrecy*) refers to a state, in which the current keys are not used to generate additional keys. If a particular key is accidentally decrypted, that is, revealed, it will not allow the attacker to easily break the other keys. If PFS is used, new Shared Secrets will be generated using D-H in Quick Mode. Using PFS is safer, but a little bit more demanding in terms of performance and time when establishing a connection. The relation key is obtained from the new D-H Secret Key and Nonce, obtained from the Quick Mode SA. By applying PFS, it is ensured that the session key is never generated from the same material.



Process Diagram IKE Phase 2 (Quick Mode)

Comparison with SSL/TLS - SSL session can be compared to the first stage of the IKE protocol; SSL connection corresponds to the second phase of the IKE protocol.

6 Diffie-Hellmann Algorithm

The Diffie-Hellman algorithm (D-H algorithm) is a cryptographic protocol that is used to create an encrypted connection between communicating parties over an unsecured channel, without the need to determine the encryption key in advance. The result of the algorithm is a symmetric encryption key, which can be used to encrypt the rest of the communication.

One advantage is that a potential attacker cannot get this key by eavesdropping (interception). The key is created by all participants in this communication and is never sent in an open form. This algorithm guarantees the exchange of a common key in such a way that if the attacker listens to this communication, it is unable to reconstruct the combined key based on the intercepted information.

One disadvantage of this protocol is the defenselessness of Man in the Middle-type of attack, because it does not allow authentication of participants. This protocol, without any combination with other authentication methods, is therefore only appropriate when an attacker cannot disturb communication in an active way.

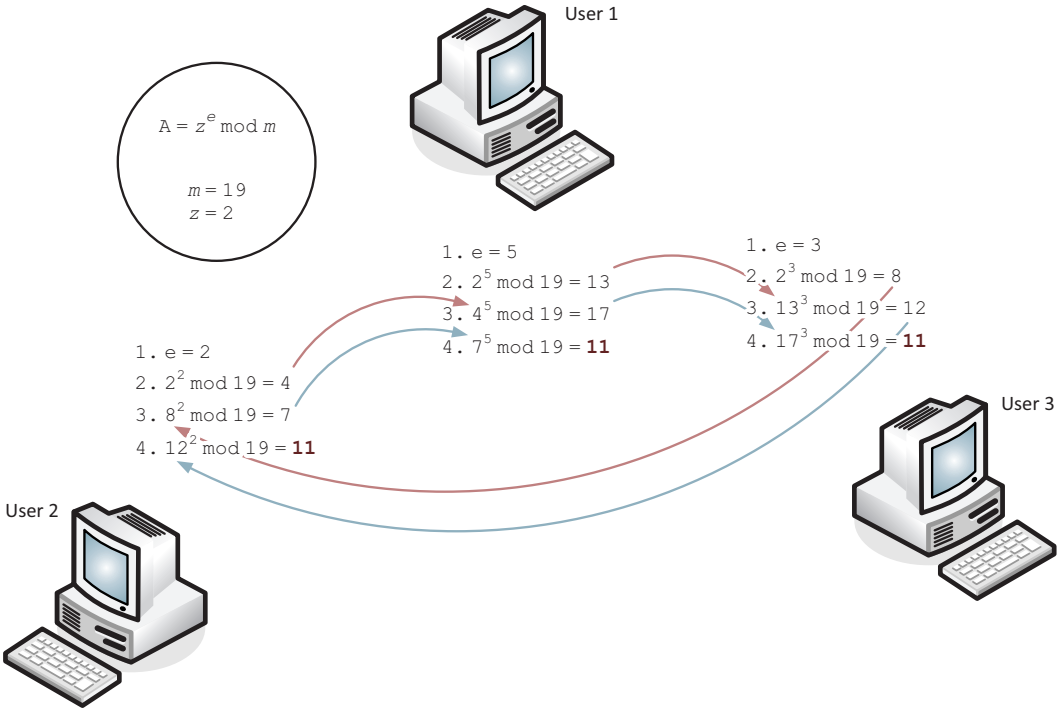
The principle of the D-H algorithm defined by RFC 2409, RFC 3526 a RFC 5114 is based on the exponentiation of numbers $(A^B)^C = (A^C)^B$, respectively on a modular options of this formula $(A^B)^C|m = (A^C)^B|m$.

The module size specifies the type of the group. Usually, there are such groups: 1, 2 and 5. The number of group indicates the length of key - DH-1 (768 bits), DH-2 (1024 bits), DH-5 (1535 bits), DH-14 (2048 bits).

Calculation of the resulting value is very easy (fast), but it is very difficult to find some of the values known only by another participant. This principle, which is the basis of security of this algorithm, is known as the discrete logarithm problem.

Communication using the D-H algorithm proceeds as follows:

- Participants publicly agrees on the used module *m*, (i.e. type of group) and basis *z*.
- Each participant chooses his exponent *e* (disjoint with module *m*).
- Each participant exponentiates the basis (by using its own exponent) and sends the result to the next participant.
- The algorithm ends when each of the original bases is processed by each participant.



Communication principle of three participants using Diffie-Hellman algorithm

7 Attacks on Local Networks - Examples and Solutions

The security of network elements has been underestimated for a long time and pushed backwards by the companies. Recently, however, the trend has changed and many companies are aware of the significance and consequences of potential threats. The number of attacks inside the network quickly outweighs the number of attacks from the outside of the network. That is why we deal with security of Access Switches, to which users have direct access, where high risk of different types of attacks comes to existence.

Examples of possible attacks on switches:

- MAC Address Flooding - the **CAM** (*Content Addressable Memory*) → the switch acts as a simple hub
- DHCP Spoofing - Denial of the DHCP address by an attacking a DHCP server
- Trunk port abuse - the attacker has access to traffic from other transmitted **VLAN** (*Virtual LAN*)
- **CDP** (*Cisco Discovery Protocol*) attacks - CDP messages are not encrypted, sent periodically, and provide detailed information about the device type, the IOS version, among many others
- Additional attacks - such as attacks on remote access passwords, DoS attacks, and others
- Installation of unauthorized wireless access points (Rogue AP) that the employee installs to have an Internet available for their **PDA** (*Portable Digital Assistant*), which can make the company's internal network available (opened up) due to its inadequate security

Potential solutions:

1. Port Security

Port Security is the easiest way to secure ports to check **MAC** addresses (*Medium Access Control*) connected to the ports. In case of a violation of a defined rule, the action is performed according to how the port was set up.

There are three responses to security breaches:

- Protect - enabled MAC addresses can continue to communicate, communication from unauthorized MAC addresses is blocked
- Restrict - the behavior is the same as in Protect mode, but an error message is generated in the device log, and if **SNMP** (*Simple Network Management Protocol*) is configured, the SNMP trap is sent to the SNMP server

- Shutdown - entire communication (even from allowed addresses) is blocked. The port is switched to a special Error-Disable state when administrator intervention is required and the port is turned on manually again.

In this way, a given physical port is concatenated (linked up) with a fixed virtual network (VLAN). This creates a fixed connection of the MAC address group and one VLAN to the given access port.

For large enterprise networks, the previous solution is not sufficient. Complex integrated solutions such as protocol-based solutions resulting from the recommendation IEEE 802.1X.

Port Security is made on the Cisco switch as follows. First, you need to enable Port-Security function on a given port, by using command "switchport port-security". The default value is 1, which means that only one device can be connected to that port. This value, i.e. the allowed number (amount) of MAC addresses that can access the port, can be changed. Addresses with the switch can also be learned either dynamically or manually. The manual configuration is done using the "port-security mac-address MAC-ADDRESS" command. This command can be extended by so-called "sticky" parameter, which ensures that the dynamically learned MAC address is stored in the device configuration. As mentioned above, you need to prepare an action that the switch executes in case of violation of rules using the "switchport port-security violation" command. That is clearly shown in the following figure.

```
Switch(config)#interface fastethernet 0/1
switch(config-if)#switchport mode access //sets the port to the appropriate mode
Switch(config-if)#switchport port-security
Switch(config-if)#switchport port-security maximum ADDRESS_AMOUNT
Switch(config-if)#switchport port-security mac-address MAC_DEVICE_ADDRESS //manual addressing
Switch(config-if)#switchport port-security mac-address sticky //dynamic MAC address learning
Switch(config-if)#switchport port-security violation {shutdown | restrict | protect}
```

Example of the Port Security Configuration on the Cisco switch

2. DHCP Snooping

DHCP Spoofing is a type of network attack where an attacker (in a local network) falsifies DHCP protocol messages (for example, by running a custom DHCP server with altered network parameters) to deceive the victim, for example, by using another default gateway. This allows the attacker to redirect traffic from victim to his computer. Subsequently, they are able to intercept all the outgoing traffic from the victim.

Another type of attack on the DHCP server is the depletion of the DHCP server ranges (DHCP Starvation). In this case, the attacker generates a large number of false requests to assign an address, which results in running out of addresses.

DHCP Snooping is an indication of how to defend against DHCP Spoofing. It is configured on switches that are directly connected to end stations (so-called Access Switches). The essence of the whole process of defending against DHCP spoofing is listening to DHCP queries on switcher ports, and blocking transmission of false responses to queries. This eliminates the attacker's spoofed DHCP server's effect. Sending responses from a DHCP server is only enabled on trusted switch ports. The port that is "trusted" is set manually by the administrator, and usually there is only one port to which the right DHCP server is connected. Cisco switches allow you to set up DHCP Snooping for any number of VLANs, set trusted ports to which DHCP servers are connected, and reduce the number of **PPS** (*Packet Per Second*) queries on the DHCP server to prevent overloading. An example of the configuration of DHCP Snooping is in the following figure.

```
Switch(config)#ip dhcp snooping
switch(config)#no ip dhcp snooping information option
Switch(config)#ip dhcp snooping vlan ONE_VLAN_OR_RANGE
Switch(config)#interface fastethernet NUMBER
Switch(config-if)#ip dhcp snooping trust
Switch(config-if)#ip dhcp snooping limit rate PPS //disables option 82 - used for DHCP Relay
```

An example of a DHCP Snooping configuration on a Cisco switch

By enabling the DHCP Snooping Binding Database (see the figure below), one can also protect against other types of attacks on local networks. After switching on this function, the switch creates a table containing links between the MAC address of the station, the IP address, the IP address lease time, the port from which it communicates, the virtual network (VLAN) where it is located, and the way the item was added to the table (manually or automatically). This information uses **DAI** (*Dynamic ARP Inspection*) to protect against ARP Cache Poisoning.

```
Switch(config)#ip dhcp snooping database flash:/dhcpbind.txt
```

Enabling the Snooping Binding Database DHCP feature on the Cisco switch

3. Dynamic ARP Inspection

ARP Cache Poisoning is an easy-to-implement and hard-to-detect attack based on falsifying message replies of **ARP** (*Address Resolution Protocol*). The ARP protocol provides IP address-MAC address links in the local network. An attacker using fake responses can cause the infected PC communication to be redirected to an attacker. It can then listen to the complete victim's communication with other stations on the network.

This attack can be detected (and prevented) by a switch that supports the DAI function.

The attack can be performed on a PC, for example by using Cain&Abel tool (www.oxid.it) or Ettercap (http://ettercap.sourceforge.net/).

DAI is a way of defending against ARP Cache Poisoning. Tables created using DHCP Snooping are used. If the ARP comes to a packet from a trusted port, it is sent further. However, if the ARP comes to a packet from an untrusted port, it is analyzed. In the case of an ARP Request message, the packet network processor detects whether the MAC and IP address of the requesting computer belong to each other. If so, the packet is forwarded to the network. Otherwise, it is discarded. In the case of the ARP Reply, it also checks whether the MAC and the IP address of the computer corresponding to the ARP Request message are related to each other. The combinations of IP and MAC addresses are taken from a database created by the Snooping DHCP feature. The DAI enable command is presented in the following figure.

```
Switch(config)#ip arp inspection vlan Vlan_ID //enables DAI
Switch#show ip arp inspection vlan Vlan_ID //views monitored VLAN
```

Turning on the DAI function on the Cisco switch

The next figure shows a command to disable DAI control on trusted interfaces.

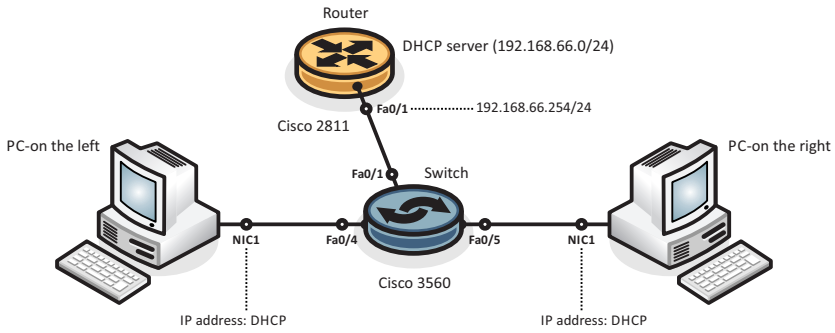
```
Switch(config)#interface fastethernet 0/1
Switch(confid-if)#ip arp inspection trust //tagging the interface as trusted
```

Turning off DAI control on the trusted Cisco switch interface

IP Source Guard has a similar function to DAI, but instead of detecting fake MAC addresses, fake source IP addresses are detected. It allows blocking of unauthorized IP addresses on ports. It is set to a specific port. This feature also uses the DHCP Snooping Binding Database. The command to turn on IP Source Guard is shown in the following figure.

```
Switch(config)#interface INTERFACE_NAME
Switch(confid-if)#ip verify source port-security //filters by source IP and MAC address
```

Enabling IP Source Guard on the Cisco switch



An example of topology to engage tasks for the simulation of attacks in the local network

8 Building VPN with IPSec - Examples and Solutions

Remote access is now an undisputable part of the management of network devices of any vast local area network, especially with regard to the need for a prompt intervention of a network administrator in the event of a sudden situation and in connection with a reduction in the total cost of such an action. Therefore, the network administrator has to be connected to the Internet to remotely monitor and reconfigure individual network elements.

In the past, the Telnet protocol was used for remote access for network element management purposes. However, it did not protect his own communication, so it was relatively easy to intercept and capture login information. Spreading the Internet access, there was a need for a protocol that would secure communications against potential attackers. Thus, **SSH** (*Secure Shell*), which communicates with the TCP transport protocol by default on port 22, provides secure authentication on both sides, ensures their integrity, transparent data encryption, and optionally lossless compression (more information can be found in RFC 4252 [https://www.ietf.org/rfc/rfc4252.txt]).

The needs of large companies to securely interconnect their branches have created virtual private networks that were to provide connection of two or more network devices in an untrusted public Internet environment. Another reason was the price for the interconnection. In the case of dedicated circuits, the costs would be incomparably higher. From the perspective of the OSI reference model, VPNs can generally be divided by the layer they are working on. The most common VPN technologies are listed in the following table.

The most common technologies in VPNs

VPN type	RM-OSI layer	Description
Frame Relay	link	It requires a homogeneous Frame Relay environment. Reliable, safer, but also more expensive compared to IP VPN.
ATM	link	It requires a homogeneous ATM environment. Like FR, it provides virtual channels with agreed parameters.
L2TP/PPTP	link	L2TP as a replacement for PPTP, which derives keys from the user's password (potential weakness). PPTP uses **MPPE** (*Microsoft Point-to-Point Encryption*) and L2TP IPsec for encryption. Defined by RFC 2637 [https://www.ietf.org/rfc/rfc2637.txt] and RFC 2661 [https://www.ietf.org/rfc/rfc2661.txt] .
BGP/MPLS	link/network	It securely exchanges information between **BGP** (*Border Gateway Protocol*) border routers in backbone networks using MPLS tunnels. Defined by RFC 4364 [https://www.ietf.org/rfc/rfc4364.txt] and others.
IPSec	network	It is a security extension of the conventional IP protocol. Encrypting each packet creates a transparent secure transmission (so-called tunnel). Defined by several RFC recommendations.
SSL/TLS	transport and higher	**SSL** (*Secure Sockets Layer*) is a technology that is transparent to technology used on the OSI network layer. SSL is then derived from the **TLS** (*Transport Layer Security*) protocol defined in RFC 5246 [https://www.ietf.org/rfc/rfc5246.txt] .

The most common way to connect branches is to connect them with IPSec VPN when an encrypted unidirectional (virtual) channel called SA is established between routers/firewalls located on the local area network.

For duplex (two-way) communication, two independent unidirectional SAs must be established.

IPSec is a mandatory component of IPv6 and additionally has also been deployed to IPv4.

It allows you to work in two modes (tunneling - the fully encapsulated original IP packet into a new IP packet, and the transport - the IPsec header is inserted between the original IP header and the upper layer header) and uses it to secure two AH and ESP protocols. Both protocols support either zero encryption (NULL), **DES** (*Data Encryption Standard*), **3DES** (*Triple DES*), **AES** (*Advanced Encryption Standard*), and Blowfish. The IPsec protocol is defined in many **RFC** (**Request For**

Comments) recommendations, but RFC 4301 [https://www.ietf.org/rfc/rfc4301.txt] is the most fundamental one. To ensure integrity, HMAC algorithms **MD5** (*Message-Digest 5*) and SHA-1 are used.

8.1 Example of an IPSec VPN Configuration on Cisco Devices

In the first stage, you need to set ISAKMP IKE policies. The IKE policy serves IPsec to build an SA. However, a shared PSK key between the two parties must be created, so that the custom encryption keys will be derived from it. The DH mechanism is usually used to exchange keys. ISAKMP uses the UDP transport protocol on the port 500. An example of a policy configuration (authentication type, encryption and hash algorithm, DH groups, and SA life time in second units) can be seen in the following figure.

```
Router(config)#crypto isakmp policy 1
Router(config-isakmp)#authentication pre-share
Router(config-isakmp)#encryption {des | 3des | aes 128 | aes 192 | aes 256}
Router(config-isakmp)#hash {md5 | sha}
Router(config-isakmp)#group {1 | 2 | 5}
Router(config-isakmp)#lifetime 86400
```

Configuration of ISAKMP IKE Policies on a Cisco Router

Additionally, you need to set up a shared PSK key so that the parties can to authenticate each other. The IP address of the other party is also defined within the command. The example is shown in the following figure.

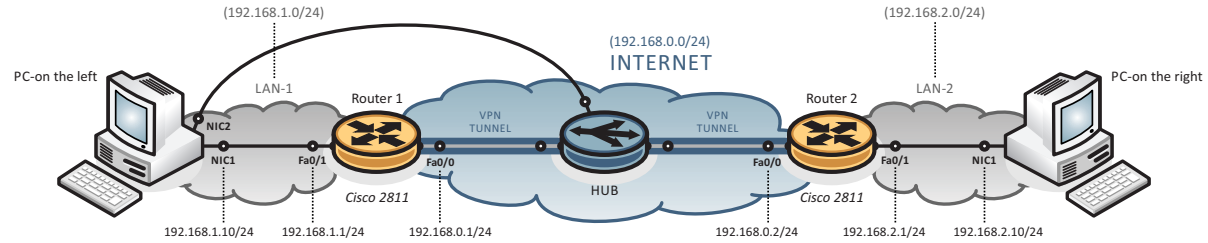
```
Router(config)#crypto isakmp key HEREisTHISsecretKEY address 192.168.0.2
```

Configuration of Shared Key PSK on Cisco Router

The second stage configures the IPsec custom settings. A set of algorithms for encryption and data integrity, known as **TS** (*Transform Set*), is defined. For example, we use the ESP protocol in combination with the HMAC algorithm SHA-1. The router will only be able to encrypt the traffic if it has so-called Interesting Traffic set using the conventional **ACL** (*Access List*) firewall rule. The parameters that are defined in this way combine the Crypto Map object, which - together with other additional parameters, such as the default address of the other side (generally, multiple addresses can be defined) and optional parameters as DH group, the IPsec SA life time (in seconds) – is applied to the appropriate **WAN** (*Wide Area Network*) interface. The above is evident from the example in the following figure.

```
Router(config)#crypto ipsec transform-set ESP-AES esp-aes 256 esp-sha-hmac
Router(config)#ip access-list extended INTERESTING-OPERATION
Router(config-ext-nacl)#permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
Router(config)#crypto map IPSEC-MAP 1 ipsec-isakmp
Router(config-crypto-map)#match address INTERESTING-OPERATION
Router(config-crypto-map)#set peer 192.168.0.2 default
Router(config-crypto-map)#set transform-set ESP-AES
Router(config-crypto-map)#set pfs group2
Router(config-crypto-map)#set security-association lifetime seconds 86400
Router(config)#interface fastethernet 0/0
Router(config-if)#crypto map IPSEC-MAP
```

Configuration of IPsec VPN on a Cisco router



An example of topology to engage tasks in IPSec VPN

9 Building VPN with SSL/TLS - Examples and Solutions

A VPN cannot just interconnect individual branches, but it can also facilitate the client's access to sources located in inaccessible part of the company's network through **HTTPS** (*HyperText Transfer Protocol for Secure*) - **HTTP** (*HyperText Transfer Protocol*) protocol with SSL/TLS support to source clients located in an inaccessible part of the corporate network. The client connects through a standard SSL/TLS-enabled web browser to the inbound web page where he/she enters his/her login information. If they are correct, a shared network resources page becomes available. Every connection is secured by SSL/TLS.

SSL VPN also addresses some drawbacks to the classic IPSec VPN. IPSec VPN has problems passing through NAT. This can be bypassed by the NAT-T mechanism, which consists of packing IPsec packets and / ESP packets into UDP datagrams, however, it increases the protocol overhead. Another disadvantage is the need for special software on the subscriber side in the case of remote access to the VPN. Implementation of IPsec clients of different manufacturers may also not be mutually compatible, the tunnel may not be built due to security rules in foreign networks (e.g. by filtering outbound traffic, using proxy servers), ...

A number of these issues can be avoided by using SSL/TLS VPNs. This VPN access is referred to as SSL VPN or Clientless VPN because the user does not need special software to access VPNs; a common web browser with HTTPS support can be used.

The term SSL VPN is often referred to as a number of mutually incompatible technologies. However, they are all based on the same basic idea, which is the use of asymmetric cryptography and SSL/TLS libraries for secure communication. Nowadays, SSL/TLS technology is widely used for encrypted HTTPS web server access.

The goal of SSL VPN is to create a transparent encrypted tunnel based on SSL/TLS. Due to the presence of SSL in common web browsers, it is not necessary to install any special client software on client computers to achieve most of the functionality offered. SSL VPN solutions are also used for small applications in the form of Java applets or ActiveX components. The abundance of premium equipment significantly influences the value of SSL VPN implementations from different manufacturers.

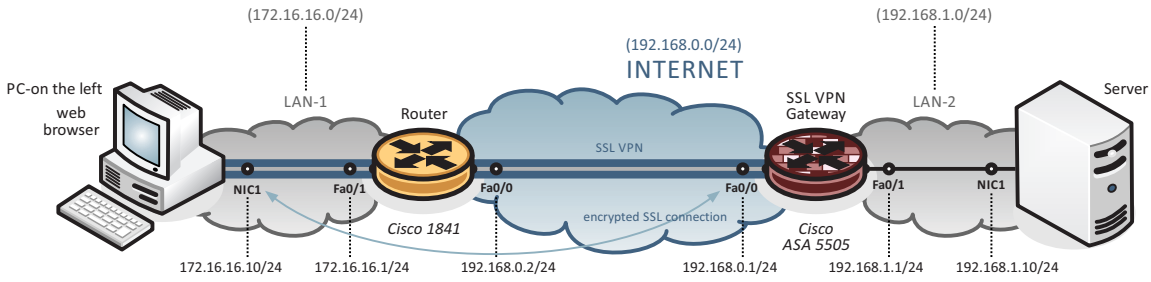
The basic functionality of SSL VPN is secure access to the company's internal information resources. An encrypted SSL tunnel is created between the SSL VPN gateway and the web browser on the client computer. In this form, SSL VPN can serve very well as an easy-to-implement way to securely access the Web portals of the company's information systems within the Internet. Another common feature of the SSL VPN solution is the ability to use shared **CIFS** (*Common Internet File System*) to share files from newer versions of Windows or **NFS** (*Network File System*).

9.1 Types of SSL VPN access

1. Clientless VPN

In this mode, the remote user accesses the internal network using a web browser (FireFox, Chrome, Internet Explorer, Edge, Safari, ...) on the client computer (see figure below). The following applications are available for the remote user:

- Internet browsing (using HTTPS) - a portal provides a list of web servers URLs that a remote user can view
- File sharing (using the CIFS file system) - the portal provides a list of file servers where the remote user can:
  - view and download shared files,
  - rename and delete files,
  - upload and download files, and
  - create and rename new files and directories.



An example of topology to engage a task on IPSec VPN

2. ThinClient

A basic condition is that a remote user's computer must support this way of communication. The remote user downloads Java applet from the portal page. This applet works on the client as a TCP proxy server for the services that are configured on the portal page. This type allows remote access to standard TCP-based applications such as **POP3** (*Post Office Protocol 3*), **SMTP** (*Simple Mail Transfer Protocol*), **IMAP** (*Internet Message Access Protocol*), or Telnet, as well as access to enterprise information systems such as **SAP** (*System Application Products*). The client's applications must be configured to communicate via a TCP connection to a known server and port. The server address is typically a loopback (127.0.0.1), where communication is captured by the TCP proxy server and then routed to the SSL tunnel.

3. Tunnel

This mode exhibits the largest list of options for a remote users. The user downloads (manually or automatically) a full SSL VPN client after logging on to the VPN server. For Cisco, it is "Cisco AnyConnect VPN Client". This program creates a virtual network interface that provides access to the network layer to various applications. This type of SSL VPN provides options comparable to IPsec VPN (Remote Access). When the connection is terminated, the Cisco AnyConnect VPN client will be removed from the client station or may remain installed on the station.

Conventional SSL VPNs cannot be used to create Site-to-Site VPNs, they are mostly used as Remote-Access VPNs. An exception to this rule is the OpenVPN project that allows you to create SSL/TLS secure site-to-site VPNs.

10 Electronic Signature

There are several reasons for introducing an electronic signature. On one hand, there was a need to introduce an equivalent to a conventional signature, and secondly, a large number of documents are produced in electronic form; some data even exist only in digital form, but - above all – its nature limits easy forgery of data.

In general, electronic signatures require identification of the signatory/entity, the integrity of the delivered document (data integrity), the undeniability and legal acceptability.

For an electronically signed document, we may also require to conceal the content of the message (i.e., encryption) and to determine, whether the document existed at a specific time (i.e. time stamp).

What is an electronic signature? The definition of electronic signature is based on Regulation No. 910/2014 of the European Parliament and of the Council on electronic identification and trust services for electronic transactions in the internal market (abbreviated as **eIDAS** (*electronic IDentification, Authentication and trust Services*)).

The regulation for eIDAS in Article 3 (10) defines an electronic signature as data in electronic form that is attached to a data message or is logically associated with this data message. Thus, the electronic signature serves as a method of unambiguous verification of identity of the signatory in relation to the data message.

This very general definition could also refer to a text signature of ordinary e-mail.

The electronic signature represents several possible options:

- electronic signature,
- guaranteed electronic signature,
- recognized electronic signature,
- qualified electronic signature.

The concept of a recognized electronic signature is/was specific for the Czech Republic (until 09/2018). It is a guaranteed electronic signature based on a qualified certificate. It is therefore applicable to communication with public authorities (but only in the Czech Republic!) There is no requirement for a secure HW repository. For a qualified electronic signature, the keys must be stored in a "safe" place.

In relation with the electronic signature, it is also worth mentioning the term digital signature. A digital signature uses the means of asymmetric cryptography. That is a specific technical solution for the electronic signature.

Digital signature is the safest way to implement electronic signatures.

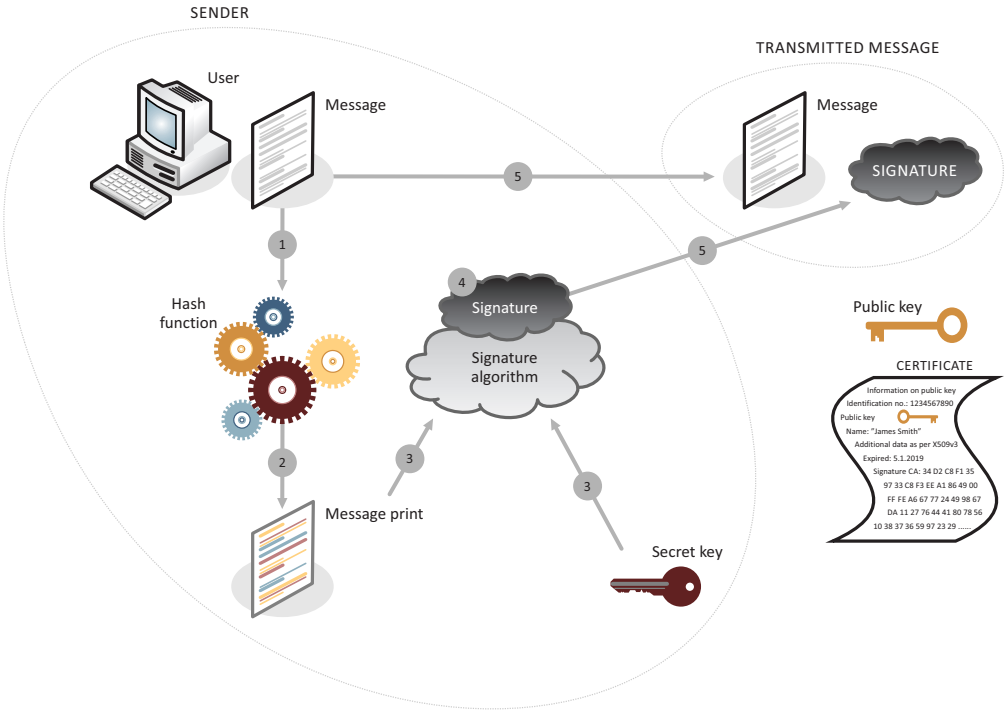
The term electronic signature is more general in the context of digital signature; it is technologically neutral. It includes, in addition to digital signature, all other methods providing the required properties (e.g. biometric methods). For this reason, it is also applicable to legislative documents.

1. Digital signature is a cryptological/mathematical term.
2. Electronic signature is, in particular, a legal and normative term.
3. The definition of electronic signature itself sets the requirements, but does not address how they can be achieved.
4. On the contrary, digital signature tools are fully focused on fulfilling the requirements.

10.1 Guaranteed electronic signature

From a cryptographic point of view, an electronic signature is understood as a set of partial cryptographic functions that ensure identification, authentication, integrity, and undeniability. Mathematically, the electronic signature is just one large number.

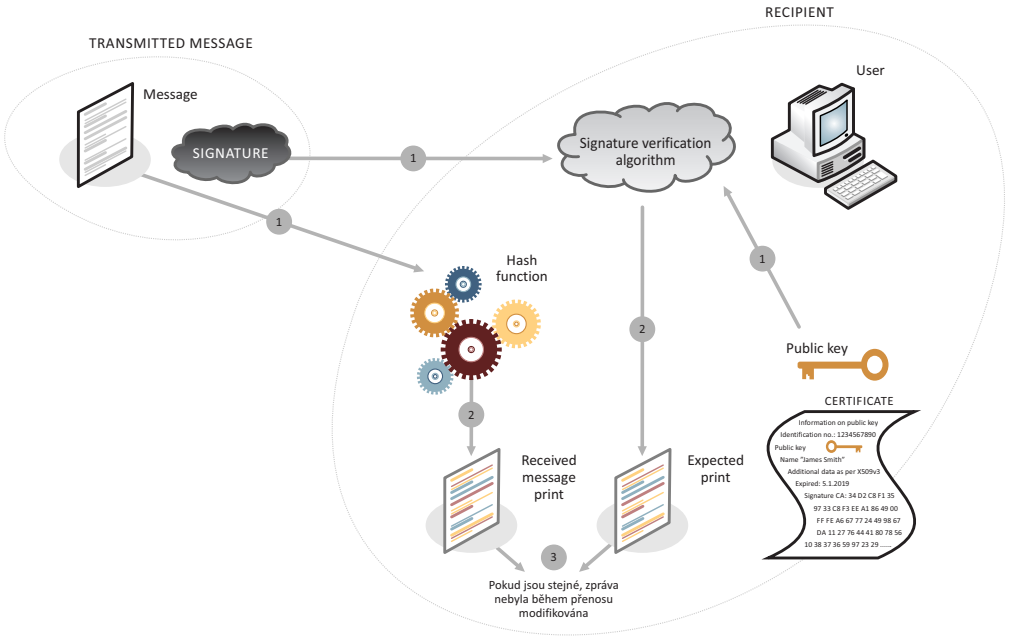
The following figure shows the process of creating a secured electronic signature. The numbers in the figure indicate the steps of the process of creating a guaranteed electronic signature.



The process of creating a guaranteed electronic signature

Any digital data such as text (PDF, TXT, DOCX, RTF, XLSX, ...), image (BMP, JPG, GIF, PNG, ...), audio (WAV, MP3, FLAC, (AVI, MPG, ...), executable files (EXE, COM, ...), and more, can be signed electronically. Essentially, anything can be signed electronically.

The following figure shows the process of verifying the guaranteed electronic signature. The numbers in the figure indicate the steps of the verification process of the guaranteed electronic signature.



Verification process for guaranteed electronic signature

The secured electronic signature ensures integrity of the messages and documents transmitted, identification of the communicating parties, authentication of the communicating parties (i.e. verification of their identification), undeniability and unrefusability.

However, the guaranteed electronic signature does not guarantee the legal acceptability of the signed documents.

The eIDAS Regulation defines the guaranteed electronic signature in Article 3 (11), if it fulfills the conditions in Article 26, then:

1. It is clearly associated with the signatory.
2. It enables identification of the signatory in relation to the data message.
3. It (the guaranteed signature) has been created and attached to the data message by using tools that the signatory can keep under his/her exclusive control.
4. It is attached to the data message to which it relates in such a way that any subsequent data change can be detected.

10.2 Qualified electronic signature

The eIDAS Regulation defines a qualified electronic signature in Article 3 (12) as a guaranteed electronic signature that is created by a qualified means designed to create electronic signatures and that is based on a qualified certificate for electronic signatures.

It has got the same validity as a handwritten signature.

A qualified certificate is defined in Article 3 (15) of the eIDAS as an electronic signature certificate issued by a qualified trust service provider creating trusted certificates and fulfilling the requirements set out in Annex I to that Regulation.

Technically speaking, a qualified certificate is same as any conventional certificate.

10.3 Electronic Seal

Technologically, it is the same thing as in the case of a secure electronic signature. The difference is in the area of its use, which is focused on the legal area.

The electronic signature is used exclusively by a natural person, the electronic seal may be used exclusively by a legal entity or an organizational unit of the state.

Previously, the term electronic tag was used for electronic seal. It is equivalent to an official stamp, which guarantees the integrity and origin of a document.

Qualified seal is based on a qualified electronic signature and is its equivalent with respect to the area of its use (exclusively for legal entities). It also requires a specific **HSM** (*Hardware Security Module*) to store a private (secret) key.

Types of certificates in terms of electronic signature are, as follows:

1. Personal, i.e. intended for natural persons only
   a) commercial - the law does not specify their content; it is used, for example, to log into data boxes
   b) qualified - the law does not specify their content; it is used, for example, to sign messages and documents
2. System, i.e. intended for legal persons, organizational units of the state or public authorities
   a) commercial - for example, to log into a file service
   b) qualified - designed for creating electronic seals

10.4 Time stamp

A time stamp (or timestamp) demonstrates the existence of a document in a given form at a given (specific) time; it does not deal with when the document was created and who owned the document.

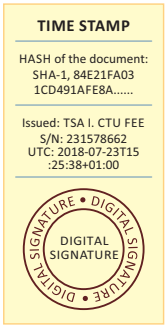
The time stamp's data structure is similar to the structure of certificates. Technically, a time stamp is implemented as just another electronic signature derived from the **TSA** (*Time Stamp Authority*).

There is also a so-called qualified time stamp as the equivalent of a qualified electronic seal or qualified electronic signature. Qualified time stamp is created by a qualified time stamp service provider.

The electronically signed structure of a time stamp includes, among many others:

- name of the publisher,
- unique stamp's serial number,
- HASH (checksum) derived from document
- time

TSA, time stamp authority, proves synchronization of its time source with **UTC** (*Universal Time Coordinated*).



Time stamp example