Redes de Computadores (RCOMP)

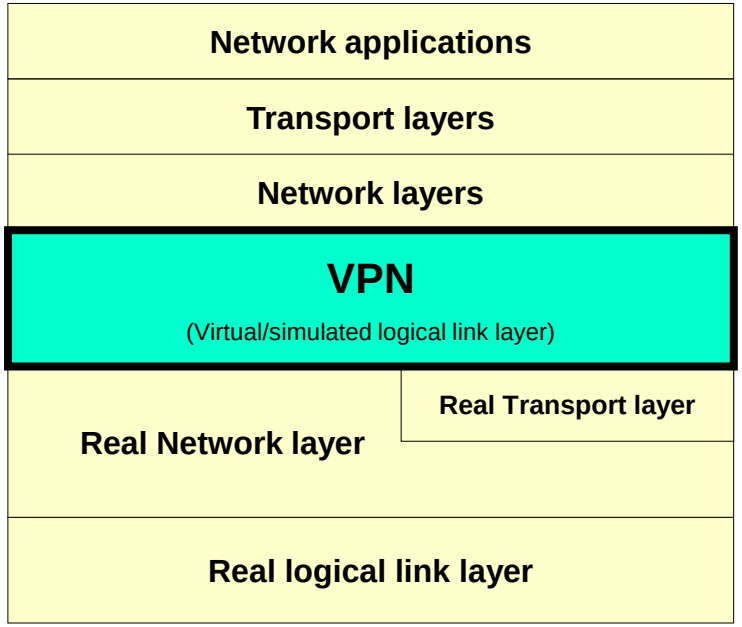
Lecture 12

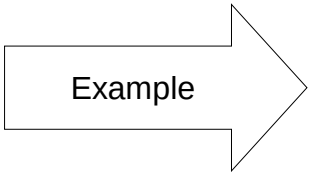
2017/2018

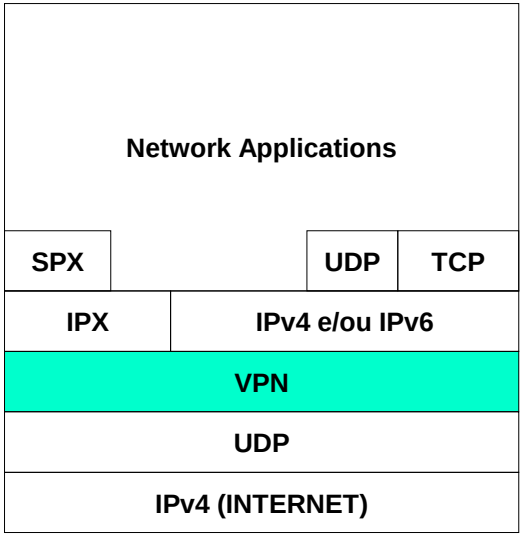
- Virtual Private Networks.
- Types and technologies.
- Security and security layers. TLS and IPsec.
- PPP, L2TP, PPTP and SSTP.

Virtual Private Network (VPN)

A VPN is a layer two communication link (logical link layer) simulated over one already existing infrastructure, normally a layer three infrastructure (network layer), typically the internet.







Virtual stand for it not being a real physical infrastructure, its just a point-to-point tunnel created over an already existing network. Because information sent through the tunnel is encrypted it is also called **private**. Even though every VPN uses the same principles, they can be classified into types with several criterions.

Network tunnel concept

The main concept around a VPN is the tunnel concept. What a VPN adds to this concept are security related features, namely authentication and data privacy.

Network tunnels are created by application level tunnelling protocols and are based on encapsulation, here this means packets being transported by other packets (as payload).

Any pair of network applications able to communicate with each other can establish a tunnel. The tunnel is essentially a logical communication channel between two applications, we call it a network tunnel not so much for what it is, but more due to what we do with it.

Instead of using the network tunnel to transport application's data it's used to transport layer two or layer three packets.

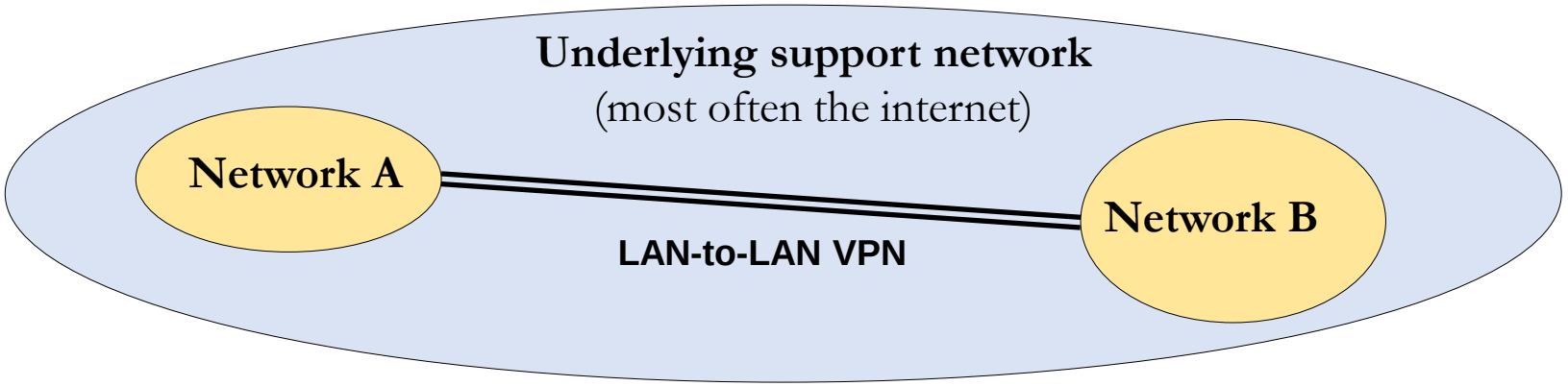
A humble TCP connection may be regarded as a simple and ready to be used network tunnel. Instead of being used for application's data transfers it is used for packets transfer. For instance, ethernet frames or IP datagrams.

Network tunnels have two end points, one application at each end. Both these applications receive packets from local networks and forward them through the tunnel to the other end application. They can be therefore regarded as switches or routers, depending on forwarding layer two or layer three packets.

LAN-to-LAN VPN (or Site-to-Site VPN)

One important VPN use is creating a permanent direct link between two remote networks. Of course, this is a task for network administrators, the same way it would be if we were talking about a physical link. This VPN usage is called LAN-to-LAN.

Network users' traffic passes through the VPN without users being aware it exists, again, as it would be if it was a physical link.

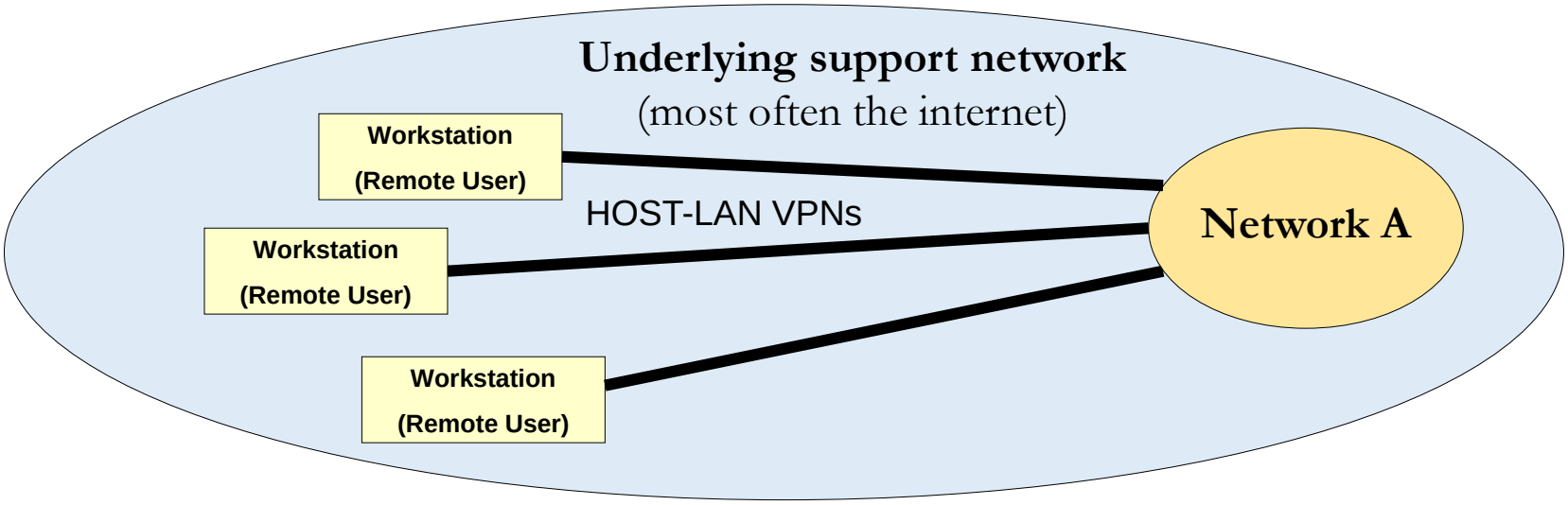


There are several reasons why network administrators may want to do this:

- Networks are private, thus they can't use the internet to talk to each other.
- Networks are using a layer two or layer three protocol not supported by the internet. For instance creating an IPv6 tunnel over an IPv4 only network.
- Just for the sake of security regrading all traffic between those networks.

Host-LAN VPN (or Remote-Access VPN)

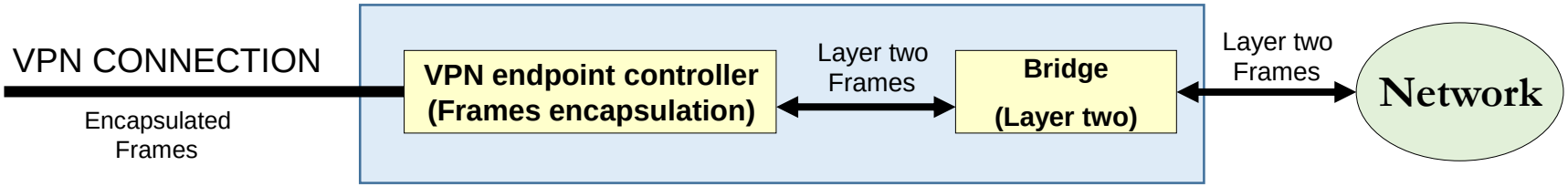
Network administrators may offer their users a VPN service. This will allow users to create personal VPN connections to the network whenever they need. VPN connections are created on user demand, this is known as Host-LAN VPN.
The remote user endpoint workstation is equipped with a VPN client software. On the opposite endpoint there's a VPN server shared by all users.



Users authentication and access control are enforced by the VPN server. Once the access is granted, the user´s workstation is provided with an IP address belonging to the remote network. The workstation operates as if it was directly connected to that network, this allows the VPN client to bypass firewalls around the remote network and also guarantees privacy on transactions with the network.

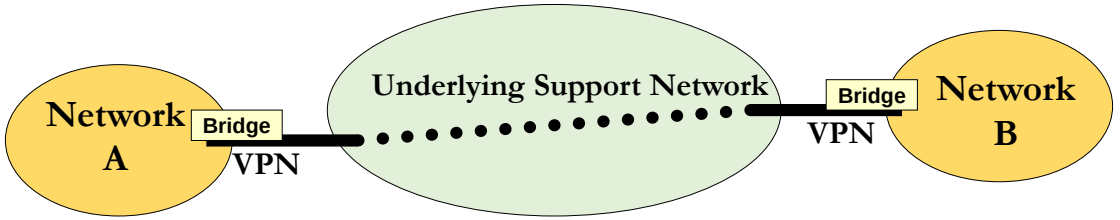
Layer two VPN

A VPN is a point-to-point physical link equivalent. As with a physical link, it can be used for several purposes, one way to connect two networks or a node to a remote network is by forwarding **layer two frames**. This means the VPN connection will behave like a **layer two bridge or switch**.



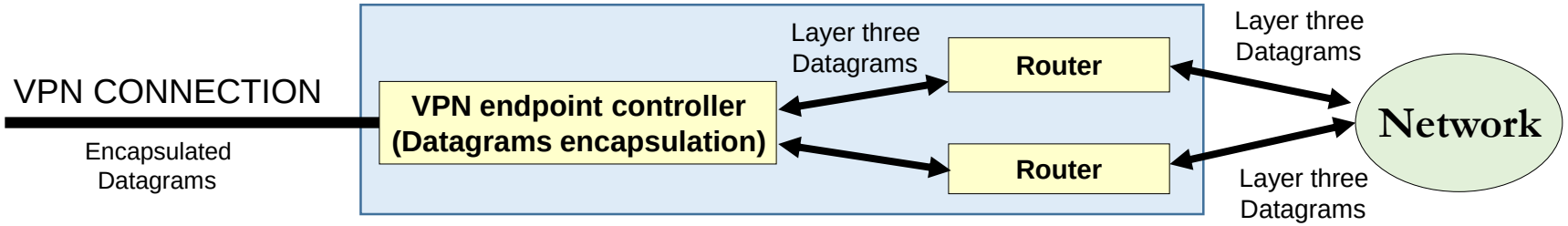
This has some pros and cons. Every layer two frame is forwarded, therefore the VPN is effective for whatever layer three and upper levels protocols are being used. Broadcast layer two traffic is forwarded as well, thus protocols like ARP and DHCP can operate normally, both sides of the VPN connection become a single network. Nevertheless, sending broadcast traffic through the VPN may represent a significate traffic burden for the VPN. Also, both VPN sides must use the same layer two technology, otherwise frame formats will be incompatible.

The image show how Network A and Network B become two segments of the same layer two network, interconnected by a layer two bridge. For instance, under IP point of view they are a single network.



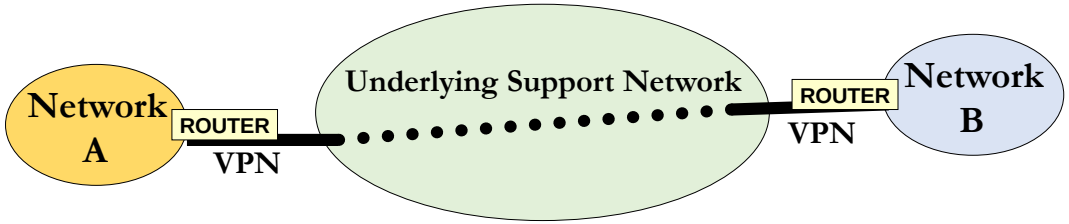
Layer three VPN

The VPN can also be used to forward layer three packets (datagrams), now VPN endpoints operate like routers. If different layer three protocols are to be used, then, for each, a router is required (multiprotocol router).



Because endpoints are routers, broadcast traffic is not forwarded. For a layer three VPN, routeing configuration is required because its not a single network anymore.

The image shown two networks connected by a layer three VPN. In fact three network addresses are required because there are three different networks.

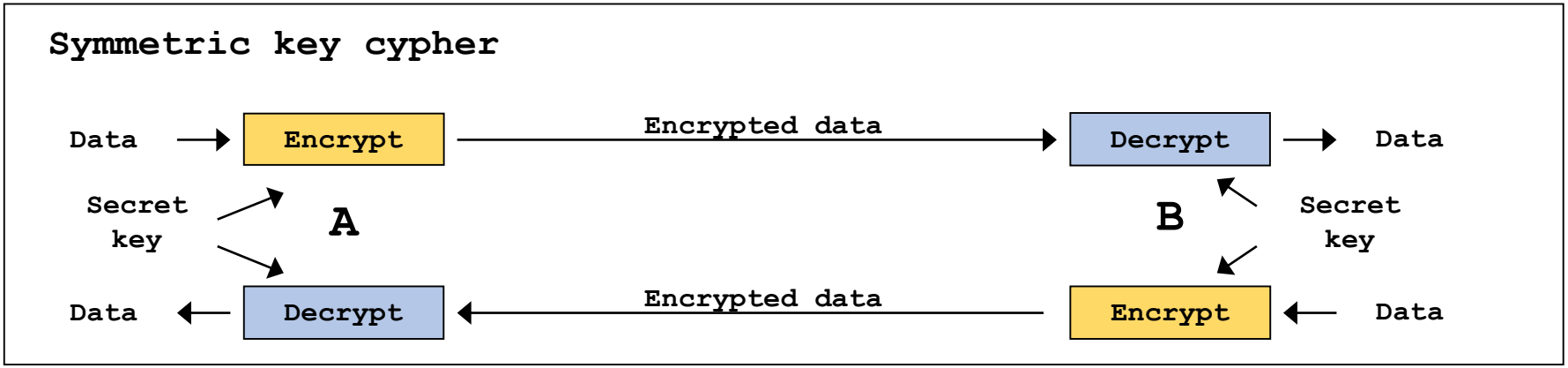


Network A, Network B and VPN connection, they all are different layer three networks, thus, each must have its own valid network address. Layer three routeing must also be settled in all routers to guarantee all networks are reachable.

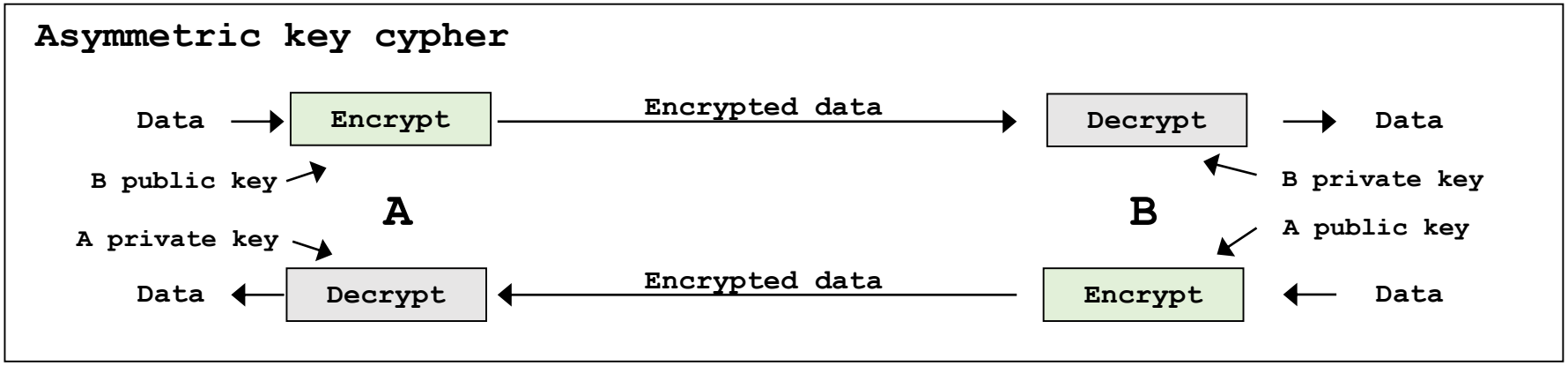
Due to the need of having different layer three networks and routeing, generally speaking a layer three VPN is more complex to configure.

Data encryption

If data is passing through untrusted networks, the way to protect it is encrypting. With symmetric key cyphers, there is **a single secret key**, known only by the two parties.



Asymmetric key cyphers, require **one key pair for each direction**. Each party has it's own key pair public/private. Public keys are used for encryption, private keys for decryption. One major advantage is that public keys are not required to be secret, and thus they can be sent to the counterpart.



VPN security

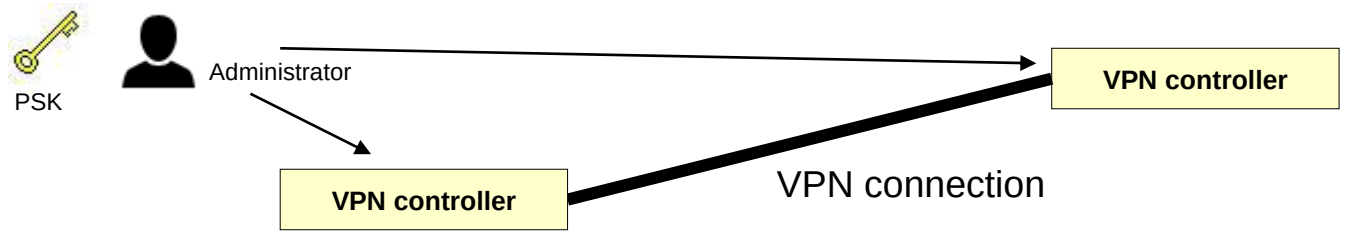
In simple terms, a VPN can be seen as a secure network tunnel, security is therefore a key factor for an VPN. Because the underlying support network is untrusted it must be regarded as exposed to malicious attacks. Two crucial issues are encompassed:

- **Authentication** – Guaranteeing one VPN endpoint is talking with the authentic another endpoint, thus, avoiding attacks known as the man-in-the-middle (MITM). In addition, for a host-LAN VPN, this may also comprise user's authentication.
- **Privacy –** Guaranteeing packets transferred through the VPN are not readable by attackers (sniffers). Since the infrastructure is untrusted the way to solve this issue is data encryption.

One approach to solving both issues at once is by using a symmetric key cypher. This requires a secret shared key to be known only by both endpoints (Pre-Shared Key - PSK).

All data is encrypted with the same secret key on both sides, thus attackers not knowing the secret key will be unable to either impersonate one endpoint or read data being sent.

Safely placing the secret key to both endpoints may, however, be tricky. In the case of a LAN-to-LAN VPN, the administrator of both endpoints may find a safe manual way to do this.



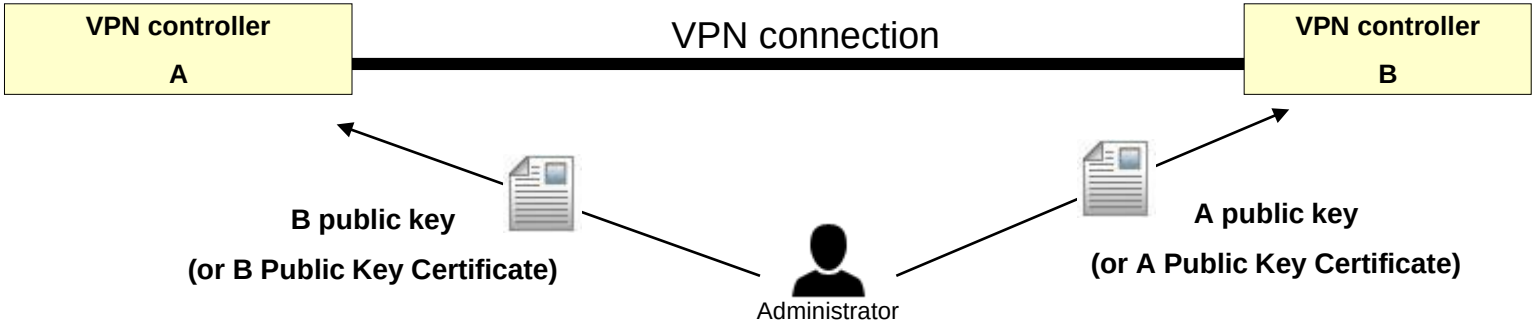
VPN security – public keys authentication

Asymmetric key cyphers brought to an end the key distribution issue, simply because the key being distributed is not secret anymore. Under privacy point of view this is very pleasant, but under authentication point of view, it raises an issue. Because keys used to encrypt are not secret, receiving encrypted data doesn't authenticate the sender anymore.

Strictly speaking, this is true for a unidirectional data transfer. Nevertheless, most network applications (including VPN endpoints) use bidirectional dialogues to establish sessions, this may help here.

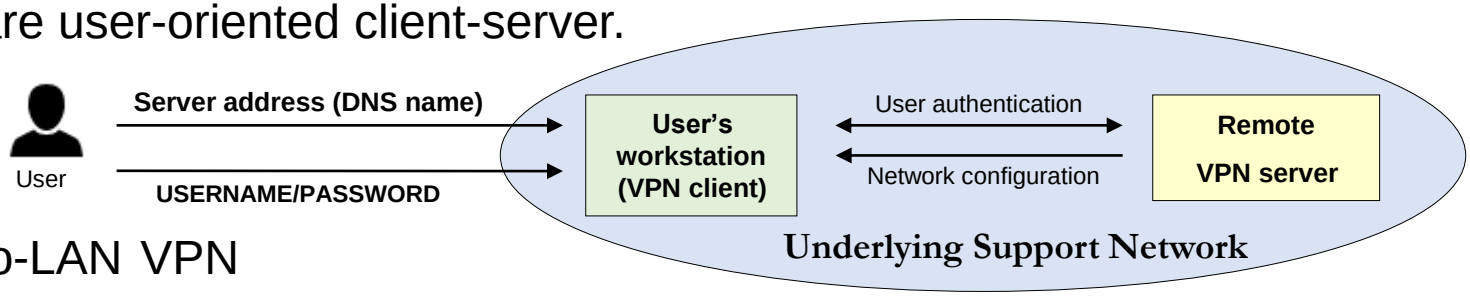
It helps because the session will be established only if both parties are holding the correct private keys, otherwise they can't decrypt what they are receiving from the partner. All the parties need to be sure is they are using the correct public key for encryption, by doing so they have the guarantee only the licit partner will be able to decrypt.

Public key certificates are digital documents testifying a public key belongs to an identified entity, they are digitally signed by a certification authority. Nevertheless, the administrator of a LAN-to-LAN VPN may manually place on each VPN endpoint the partner's public key or public key certificate.



Host-LAN VPN and user authentication

A host-LAN VPN, has some specific characteristics, many are security related. Most implementations are user-oriented client-server.



Unlike in a LAN-to-LAN VPN there are clearly two roles.

One VPN server accepts sessions from several VPN clients. VPN clients are supposed to receive IP configuration information from the server, either by DHCP or another equivalent protocol.

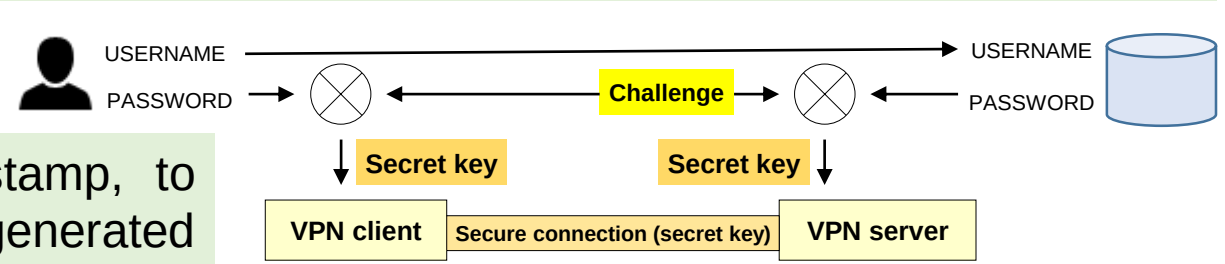
The VPN server demands each client's authentication. Because each client is under a user's control, one simple solution is enforcing user authentication, most often by username/password. Before sending the user password, the client must be sure the server is authentic and transmitted data is being safely encrypted. Regarding the server authenticity, one option is providing to the client a public key certificate. Anyway, data privacy must be coordinated with user authentication.

The direct use of a symmetric key cypher with a pre-shared secret key is not realistic because one key would be required for each user. However, the user's password can be used to generate the same secret key on both sides.

Host-LAN VPN – secret key authentication and privacy

If both the user and the VPN server possess the user password, it´s a secret nobody else knows, therefore it can be used to generate a secret key know only to the VPN client and the VPN server.

Challenge is a long server generated, random number.



It can also include a timestamp, to avoid replay attacks. The generated secret key is unique for each session.

At first glance, this looks magnificent. The server is authenticated, the user is authenticated and data privacy is assured, also the user password is never sent through the network. As far as the password is known only to the user and the VPN server, both authenticity is guaranteed. This results from being impossible to generate the secret key without knowing the password.

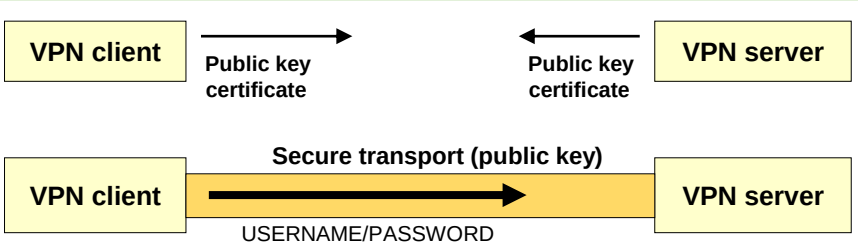
Bear in mind that some users databases store password hashes and not passwords, but that's no obstacle, on those cases the stored hash is used on the server side and the same hash is generated on the client side.

This is a really all-in-one technique, but there is one major drawback. It all depends on the user's password quality, and that is very hard to warrant. Notice the challenge is public and passwords are highly susceptible to dictionary attacks, with some effort, the password may be guessed.

Host-LAN VPN – public key authentication and privacy

One better way to protect the user password is by using a strongly secured transport in the first place. This can be achieved with public key encryption and public key certificates.

Public key certificates are exchanged between client and server, this will guarantee both authenticities. Though, the server may not be too keen on the client certificate as it will rely on the user's authentication.



If the server public key certificate is valid, the client can safely send anything to it by encrypting data with the server's public key. So the client may now securely send the username and password, and thus, authenticate the user.

This technique is widely used, though normally the public key secure transport is switched to a much faster symmetric key cypher.

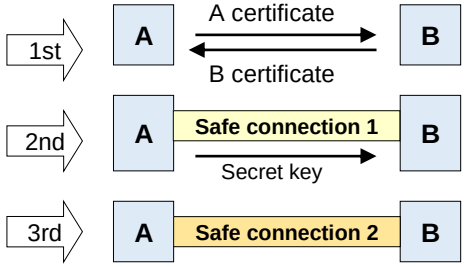
Public keys are used as an intermediate step. Once the public key secure transport is established, the client generates a random secret key and safely sends it to the server. Afterwards, the asymmetric cypher is abandoned and a symmetric cypher is used instead.

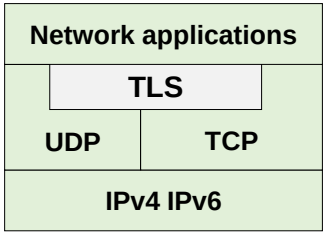
Regarding user authentication, it works the same, but username and password are now encrypted using the symmetric cypher.

Security layers

VPN implementations may take advantage of existing ready-to-use security layers. **Transport Layer Security (TLS)** was originally developed by Netscape under Secure Sockets Layer (SSL) name. TLS mission is adding security to UDP and TCP transactions, pre-shared key authentication is available, but most popular is by public key certificates.

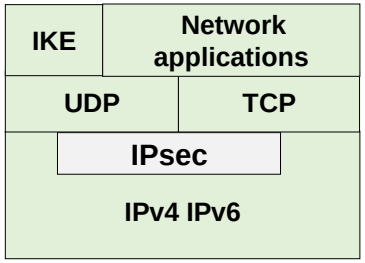
TLS client and server start by negotiating a cypher-suite, if public key authentication is settled they exchange certificates (1st), a faster symmetric cypher secret key is exchanged (2nd), this end up with a secret key secured session (3rd).





Many application protocols use the TLS layer instead the TCP layer to create safe versions of themselves, for instance HTTP runs over TCP, HTTPS is the same protocol running over TLS.

IPsec (IP security) is another security layer, but it runs at a lower layer, over IPv4 and IPV6. IPsec only works once cyphers and keys are settled (security associations), IKE (Internet Key Exchange) is the most widely used protocol to automatically create the required IPsec security associations. Regarding the protocol stack shown at the right, bear in mind IPsec is an additional protocol to IPv4 but is already included in IPv6 as an extension header.



PPP – Point to Point Protocol

PPP is pretty old, it comes from the primordial HDLC (High-Level Data Link Control). Despite its age, it's widely used whenever packets are to be transported over a point-to-point layer two link. Many VPN implementations do use PPP. The image bellow represents a PPP frame:

Frames start and end with an 8-bits flag, between two consecutive frames, only one flag is required. The sender must ensure the flag does not appear elsewhere on the transmitted frame, a technique called bit stuffing is used to guarantee that.

In a point-to-point connection, addresses are implicit and not really required. The 8-bits address field, inherited from HDLC has fixed value 0xFF, meaning broadcast. The 8-bits control field comes as well from HDLC, fixed 0x03 value means unnumbered frame.

The 16-bits protocol field is used to identify the payload type, some payload types will be upper level data like IPv4 or IPv6 datagrams, yet several others are used by PPP itself. One of those protocols is LCP (Link Control Protocol - 0xC021).

LCP defines several commands, including Configure-Request, Terminate-Request, Echo-Request and Echo-Reply. Configure-Request commands are used to negotiate the link configuration (LCP options), they include MRU (Maximum-Receive-Unit), **authentication protocol to be used** and header compression.

The original PPP RFC1661 defines two authentication protocols: PAP (Password Authentication Protocol) with number 0xC023 and CHAP (Challenge Handshake Authentication Protocol) with number 0xC223. By default PPP does not require authentication.

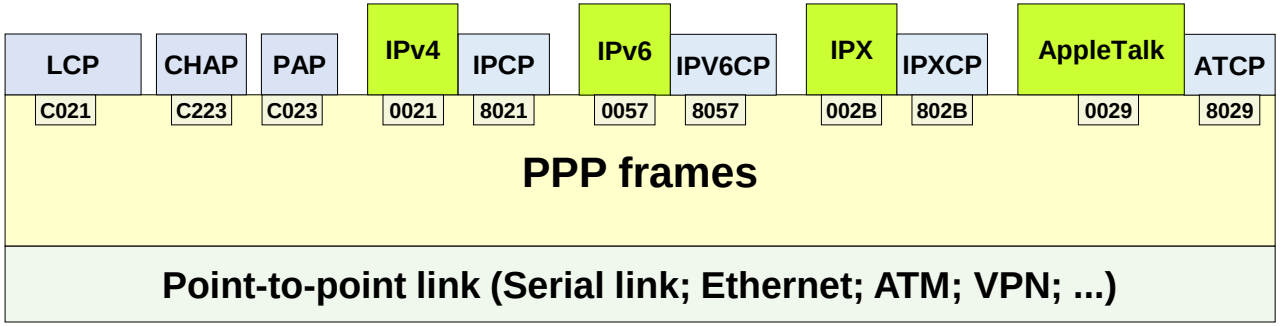
PPP – NCP (Network Control Protocol)

Once LCP negotiation phase is finished, including and optional authentication phase, upper layer protocols can be used, for each there will also be a dedicated NCP (Network Control Protocol).

Network layer protocols are identified by numbers ranging from 0x0000 up to 0x3FFF, for each, there is a corresponding same offset NCP identifier on the 0x8000 to 0xBFFF range. NCP handles the corresponding layer three protocol issues.

For instance IPv4 has is number 0x0021, the corresponding NCP has number 0x8021 and is known as IPCP (Internet Protocol Control Protocol). Among other things, IPCP may handle automatic IP node configuration in a DHCP style.

The image below presents a sample global usage scenario for PPP:



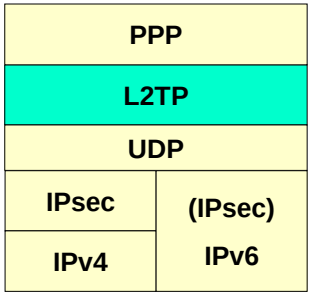
L2TP - Layer 2 Tunneling Protocol

L2TP is the name by which a popular VPN type is known, though the name does not tell the whole story because L2TP is solely a tunnelling protocol.

L2TP creates and maintains network tunnels by encapsulating data into UDP packets. Tunnel endpoints are called LAC (L2TP Access Concentrator) and LNS (L2TP Network Server). Usually the LAC acts as client and establishes a tunnel control connection with the LNS on UDP port 1701. Over the L2TP tunnel one or several L2TP sessions can be created, each L2TP session is used to transport PPP frames from one PPP session.

L2TP has an optional CHAP authentication procedure during the tunnel creation, PPP, running over L2TP sessions may enforce some security features. But beyond this, L2TP by itself offer no guarantees regarding authentication or privacy. To achieve a viable VPN, L2TP is run over IPsec.

On what is called an L2TP VPN, before L2TP entering in action, IPsec must already be settled, this usually requires IKE. IKE negotiates authentication and encryption by sending UDP datagrams to port number 500.



Two main techniques are available:

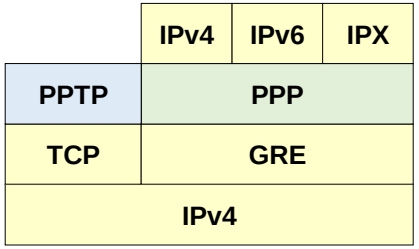
- A secret pre-shared-key (PSK) manually placed on both endpoints.
- Public key certificates (required to be valid for both endpoints).

User authentication can be imposed within the PPP protocol running over a L2TP tunnel session.

PPTP - Point-to-point tunnelling protocol

PPTP is an L2TP predecessor but is still widely used by Microsoft and also by Cisco. Since it does not use IPsec or TLS, it turns out to be simpler for end users because it does not require pre-shared keys or public key certificates.
PPTP uses a TCP control connection (server port number 1723) to control a data tunnel creation. The tunnel encapsulates PPP packets into GRE (Generic Routing Encapsulation) packets. GRE was developed by Cisco with the specific purpose of tunnels creation, it runs directly over IPv4 or IPv6 with protocol identifier 47.

One issue with GRE is it hasn't port numbers, thus NAPT devices will have a difficult time handling GRE packets, many NAT routers may require a specific option, usually called **PPTP pass-through**.
Neither GRE or PPTP have security features, both authentication and privacy must be guaranteed by PPP.



PPP is rather flexible on supported authentication and privacy protocols, currently Microsoft PPTP uses MSCHAPv2 user authentication, during user authentication a secret key is generated for MPPE (Microsoft Point-to-Point Encryption) protocol, this is based on RC4 (Rivest Cipher 4). Along with MPPE, MPPC (Microsoft Point-to-Point Compression) can also be used.

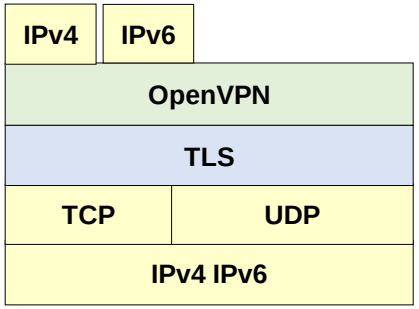
TLS based VPNs

TLS is a security layer under constant scrutiny and thus regarded as very solid. Several VPN implementations use it.

OpenVPN is open source and rather popular, it creates TLS protected tunnels either over UDP or TCP. Authentication/privacy may use a secret shared key or public key certificates. Both a secret key or public key certificates are suitable for a LAN-to-LAN VPN. On a Host-LAN VPN, however, public key certificates are the best option, the server certificate is especially relevant to the client, also user authentication will be required.

OpenVPN does not use PPP, so it must handle user authentication on its own, also client automatic configuration in DHCP style must be performed by OpenVPN on its own.

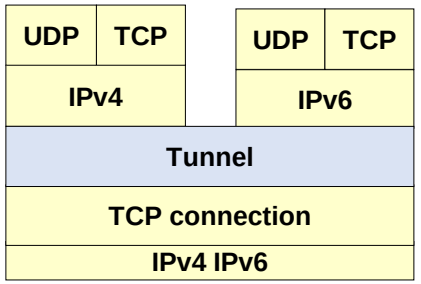
Either running over UDP or TCP, the OpenVPN server listens on port number 1194.



Secure Socket Tunneling Protocol (SSTP) uses a TLS protected TCP connection to port number 443. Therefore, firewalls will regard SSTP traffic as being HTTPS traffic, though internally SSTP has no relation to HTTP. Authentication is established with public key certificates, a **PPP** session is established and user authentication is demanded by the server.

The TCP meltdown issue

TCP based tunnels, like for instance in SSTP and possibly in OpenVPN, suffer from one common drawback called TCP meltdown.



The origin of this issue comes from a reliable TCP connection (the tunnel) being regarded as unreliable by protocols using it, they see it as an unreliable layer two physical link.

Whilst the underlying physical network is not congested and presents no significant error rate, everything runs smoothly, possibly better than over a UDP based tunnel.

Yet, when significant delays and packet loss start occurring on the physical network, problems will come to the surface.

For protocols using the tunnel, data loss is rather normal because they assume they are using an unreliable layer two physical link, so they immediately forget about it and send data again. But the TCP tunnel connection doesn't forget, it will insist in retransmitting every byte until success is achieved, even though the protocol above has long ago forgotten that byte even exists. This will eventually overload and stall the TCP tunnel.

Imagine for instance a bunch of TCP connections running over the tunnel, on failure they will all start retransmitting, and for each retransmission, the tunnel TCP connection will fail to deliver and also starts retransmitting.