Table of Contents

Index of Figures

Index of Tables

1 Introduction

1.1 Background

Internet is no doubt the most widely present data communication network on the Earth today. Although it has some globally unresolved issues like guaranteed quality of service or latency, its ubiquitous presence and access costs are making the Internet very practical for road warriors and mobile users as their connection medium to distant corporate central offices. Protocol, which is dominating on the Internet today, IP (Internet Protocol) version four or in short IPv4, unfortunately does not offer any confidentiality or authentication features, which are necessary for passing private traffic over the public networks. To address these problems, as well as the growing lack of IPv4 address space, back in 1992, IETF (Internet Engineering Task Force) has started the development of a new version of IP protocol which today has a name: IP version six or IPv6. As a result of that development, we have got a framework that provides security services for traffic at the IP layer, in both the IPv4 and IPv6 environments, which is commonly referred today as the IP Security protocol, or IPsec. The IPsec framework is mandatory for the IPv6 protocol and optional for the IPv4 protocol. In combination with the IPv4 protocol, the IPsec is nowadays the most widely used standardized solution for providing integrity, authentication and confidentiality on a per packet basis of the layer three OSI (Open System Interconnection) model in IP based networks. However, as in any good and extensively used technology, the IPsec standard framework is developing further to address specific needs and requirements.

In this thesis, I will describe requirements, protocol extension proposals and functionalities, which are currently still undergoing development. I will also address the particular applicability of the IPsec for remote access VPNs (Virtual Private Networks). Some of the protocol extension proposals, which will be described, have already existed for a longer period and have gone through several revisions, proposals and implementations. The others, which are still in the design phase, are described and compared to each other only from the theoretical analysis perspective. I will also analyze remote access VPN solution alternatives such as the L2TP/IPsec ((Layer 2 Tunneling Protocol/IP Security Protocol) combination and compare it to the remote access VPN requirements from the perspective of which functionality it provides and which it is lacking. The practical part of this thesis

includes product requirements influence, a definition of features as well as beta testing of early engineering software images. It also includes a knowledge transfer to field engineers about the particular IPsec extension implementations for remote access VPN on Cisco Systems IOS (Internetwork Operating System) based low-end router platforms. Two practical test bed setup scenarios are explained in Appendix A. The first setup is the software VPN client running on Microsoft Windows operating systems connecting to the IOS based VPN gateway with a demonstration of the VPN tunnel establishment and dynamic internal IP address, primary and secondary DNS (Domain Name System), WINS (Windows Internet Naming Service) servers and domain name assignments. It also shows the IPsec paradigm change from an entirely peer to peer protocol to a client server protocol where centrally controlled resource parameters are dynamically pushed from the VPN gateway to a VPN client, thus enabling a scalable deployment and central management of a large number of the VPN clients. The same concept is then applied to the IOS based low-end VPN router running Easy VPN client code offloading IPsec client functionality from the local hosts connecting from its local network interface. The main purpose of the dynamic configuration, control of the VPN parameters on a central VPN gateway and pushing them from the central site to the remote VPN devices is the scalable deployment of a large number of remote sites.

1.2 Virtual Private Networks

One of the widest practical uses of the IPsec protocol today is in VPN (Virtual Private Network) [6,22,26,32]. VPN as a term is not new or invented by the usage of an IPsec protocol. It represents a wide variety of layer two and layer three technologies for tunneling and separation of the data packets, frames or cells over shared networks. VPN is also commonly referred to as the network, which offers secure connectivity over the shared public network infrastructure such as the Internet. “Virtual” in VPN means that there is a logical, not physical connection between the two end points communicating to each other. “Private” means that the traffic through this connection is separated from other traffic passing over the same, shared infrastructure. It is important to distinguish different technologies, which are used as the building blocks of VPN as traffic separation. Virtual connections could be established using various protocols and on the different layers of OSI model. For the purpose of easier understanding my further research and relation to already existing protocols, I will start by briefly outlining the difference between a layer 2 and a layer 3 VPN technologies.

1.2.1 Layer 2 Virtual Private Networks

Throughout the history of networking technologies, there was always the need to separate a certain type of network traffic from another and this was achieved in a different ways. Some examples of the WAN (Wide Area Network) VPN technologies on layer two of the OSI model are SVCs (Switched Virtual Circuits) or PVCs (Permanent Virtual Circuits) in X.25, Frame Relay or ATM (Asynchronous Transfer Mode) networks. These were the mechanisms for separation of traffic based on the labels or headers of the packet frames or cells. At the end of the last decade, with the explosion of the Internet and the IP protocol, additional tunneling protocols were developed for carrying layer two frames over the IP protocol. Those are the **L2F** (**Layer Two Forwarding protocol**) as described in historical RFC (Request for Comments) 2341, the **PPTP** (**Point To Point Tunneling Protocol**) as described in informational RFC 2637 and the **L2TP** (**Layer 2 Tunneling Protocol**) as defined in RFC 2661 [33]. The first two protocols, L2F and PPTP have resulted in the development of a third merged standardized protocol L2TP. In the following chapters, I will focus only on the resulting tunneling protocol L2TP and its functionality in relation to the IPsec protocol.

1.2.2 Layer 3 Virtual Private Networks

Layer 3 VPNs represent the separation of data packets on the third layer of OSI model, which in TCP/IP (Transport Control Protocol / Internet Protocol) protocol suite means IP layer. IP based VPN networks are commonly referred to as the tunneling or encapsulating protocols and technologies, where the packet of one protocol type is wrapped up or encapsulated within the IP protocol packet type before being carried over the network. There are multiple standard protocols defining the possibilities of how this could be done. The **GRE** (**Generic Routing Encapsulation**) protocol, **MPLS** (**Multi Protocol Label Switching**) protocol or the IPsec protocol are all ways of the IP packet encapsulation or IP packet separation based on the additional header or label in front of the standard IPv4 packet header. In the following chapters, I will focus only on the **IPsec based layer 3 VPN** types. Depending on the type of the VPN gateway, the type of the networks or nodes needed to be connected over the VPN, we could distinguish three major types of IPsec based VPNs: site-to-site, firewall based and remote access VPN. Each type has its specific needs in regard to confidentiality, authentication or addressing schemes, which I will outline in the following sections.

2 Current IPsec standards

2.1 Overview of IPsec

To understand the missing elements of the IPsec framework, it is important to comprehend what is included in the current IPsec protocol suite [12,19,20,21,25,30]. This chapter is by no means an extensive IPsec protocol description but rather an overview, of the elements that are necessary in order to understand its further development.

The IPsec is a framework of open standards for ensuring secure private communications over IP networks. It is based on the standards developed by the IETF to ensure confidentiality, integrity, and authenticity of data communications across an IP network. IPsec provides a necessary component of a flexible solution for deploying a network-wide security policy by combining several different security technologies into a complete system to provide confidentiality, integrity, and authenticity of an IP packet. In particular, IPsec uses:

- Diffie-Hellman key exchange for deriving key material between two peers on a public network.
- Public key cryptography for signing the Diffie-Hellman exchanges to guarantee the identities of the two parties and avoid man-in-the-middle attacks.
- Bulk encryption algorithms, such as DES (Data Encryption Standard), 3DES (Triple DES) or IDEA (International Data Encryption Algorithm) for encrypting the data.
- Keyed hash algorithms, such as HMAC (Hashed Message Authentication Code), combined with traditional hash algorithms such as MD5 (Message Digest 5) or SHA1 (Secure Hashing Algorithm 1) for providing packet authentication.

The IPsec framework, which is described in RFCs 2401-2412, could be basically divided into two major parts:

- IP Security Protocol suite, which defines the information to add to an IP packet to enable confidentiality, integrity, and authenticity controls as well as defining how to encrypt the packet data.
- Internet Key Exchange, which negotiates the security association between two entities and exchanges key material.

2.2 IPsec headers

IPsec is adding two new headers to the IPv4 packet: AH (Authentication header) and ESP (Encapsulation Security Payload) header. **AH header** provides authentication, integrity and replay protection for IPv4 header as well as for all the upper-layer protocols of an IP packet. However, it does not provide any confidentiality to them. Confidentiality is the task of the **ESP header**, besides providing authentication, integrity and replay protection for the packet payload. Both of the headers could be used in two modes: transport and tunnel modes. The **transport mode** is used when both the communicating peers are hosts. It may also be applied when one peer is a host and the other is a gateway, if that gateway is acting as a host or ending point of the communication traffic. The transport mode has the advantage of adding only a few bytes to the header of each packet. With this choice however, the original IP packet header could only be authenticated but not encrypted. The **tunnel mode** is used between two gateway devices, or between a host and a gateway if that gateway is the conduit to the actual source or destination. In the tunnel mode, the entire original IP packet is encrypted and becomes the payload of a new IP packet. The new IP header has the destination address of its IPsec peer. All the information from the original packet, including the headers, is protected. The tunnel mode protects against attacks on the endpoints due to the fact that, although the IPsec tunnel endpoints can be determined, the true source and destination endpoints cannot be determined because the information in the original IP header has been encrypted. The difference between the transport and tunnel mode protection of the original packet is illustrated in Figure 1.

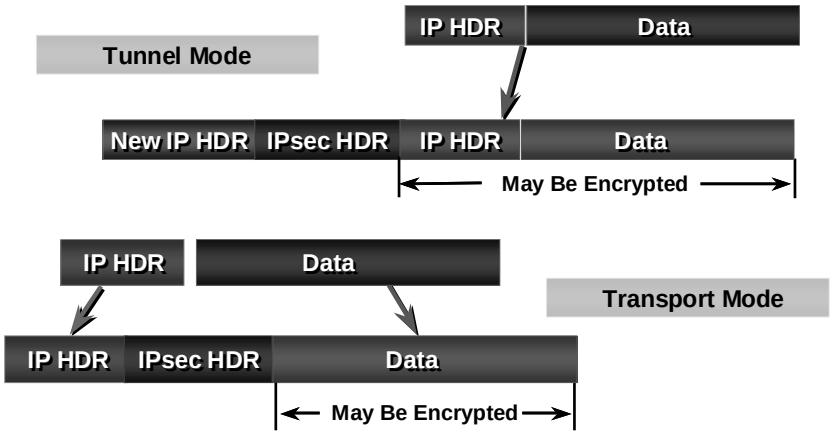


Figure 1. Transport and Tunnel modes of IPsec

Whether the tunnel or transport mode will be used for the VPN, depends on the type of VPN, as we will see in later chapters in more details. I will mention here, that dependency comes from doing VPN on the layer two or on the layer three of OSI model and whether any other tunneling technology is already applied on the packet or not. The transport mode could be used if IPsec is combined with layer two tunneling technologies such as L2TP or L2F, while the tunnel mode will be used if we are using IPsec as the only tunneling mechanism for the IP packets. An exception to that is if the IPsec is combined with another IP tunneling protocol such as GRE tunneling, in which case, it suffices to use it only in the transport mode.

2.3 Authentication Header

The AH (Authentication header) [20], when added to an IP packet, ensures the integrity and authenticity of the data, including the invariant fields in the outer IP header. It zeros mutable[1] fields (such as Type of Service, Flags, Fragment Offset, Time to Live or Header Checksum) of the original header before authenticating it. It does not provide confidentiality protection. Due to the performance issues, AH, which is responsible for the integrity and authenticity of each packet, uses a keyed-hash function such as HMAC-MD5 or HMAC-SHA1. Fields of the AH header and AH placement in the IPv4 packet, depending on the mode of usage, are illustrated in Figure 2.

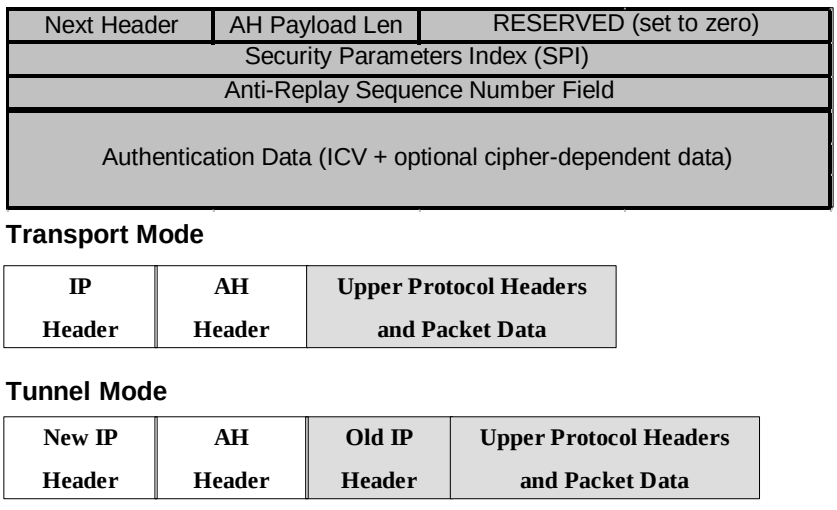


Figure 2. IPsec Authentication Header

[1] Mutable fields are the fields of an IP packet that are changed by the transit routers.

An SPI (Security Parameter Index) field is, together with a destination address of the outer IP header, used to identify the SA (Security Association) for particular authenticated packet. Security Associations are defined in the key exchange chapter. A sequence number is a monolithically increasing counter that is used for the anti-replay function of AH. An authentication data field is a variable length field that contains the result of the integrity checking function.

2.4 Encapsulation Security Payload Header

The ESP (Encapsulating Security Payload) header [21], when added to an IP packet, protects the confidentiality, integrity, and authenticity of the data. If the ESP is used to validate data integrity, it does not (as in the case of the AH header) validate the mutable fields. The confidentiality of the IP packet payload with the ESP is achieved by using a variety of symmetric encryption algorithms. The default algorithm for the IPsec is 56-bit DES. Other possible symmetrical algorithms are 3DES, IDEA, Blowfish, CAST (Charlie Adams and Stafford Tavares crypto protocol) or soon AES (Advanced Encryption Standard). All symmetrical encryption algorithms require both peer devices participating in confidential communication to use the same key for encryption and decryption. The mechanism for key distribution is described in the following chapter. Figure 3 illustrates the details of the ESP header and ESP header position in IPv4 packet for transport and tunnel modes.

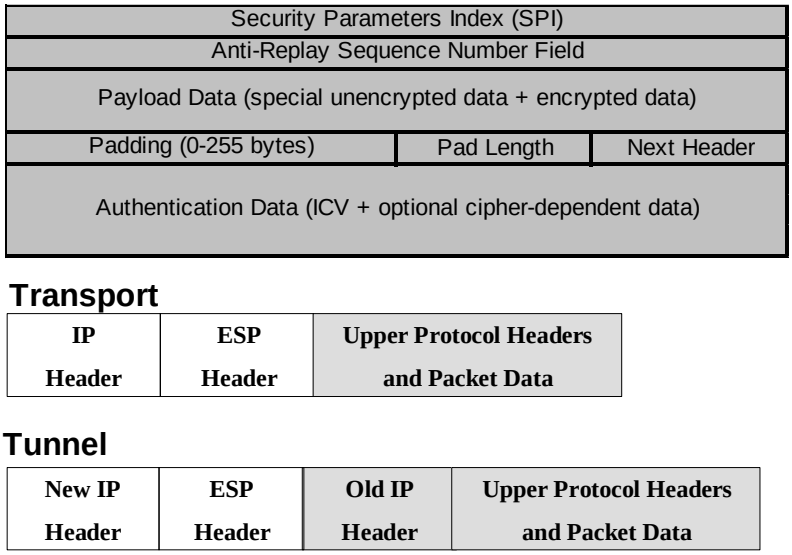


Figure 3. IPsec Encapsulation Security Payload Header

The SPI and anti-replay fields have the same functionality as with the AH but applied only to the payload of the packet without the original header. The authentication data field is the field of the variable size, which contains the result of the integrity checking function of an ESP header plus the new payload data, which may be encrypted in the payload data field.

2.5 Internet Key Exchange

2.5.1 Internet Key Exchange overview

As both of the new headers require a certain set of parameters to be agreed upon before participating in a confidential or authenticated communication, we need to have a management mechanism to distribute parameters such as shared symmetrical secret keys, hashing or encryption algorithm types and agree on what type of traffic need to be protected. The simplest form of management is the manual management, in which we manually configure each system with all the necessary management data relevant to secure communication with other systems. Manual techniques are practical in small, static environments but they do not scale well. Although the IPsec framework allows manual configuration setup as minimal compliance requirement, it also defines extensive suite of protocols for dynamic negotiation of parameters between two peers.

The **IKE** (**Internet Key Exchange**) [12], based on **ISAKMP/Oakley** (**Internet Security Association and Key Management Protocol/Oakley**) [25], is the protocol suite used for dynamic policy negotiation and establishment of authenticated keying material between two IPsec peers. ISAKMP define packet formats, retransmission timers and message construction requirements which in effect represent the language or generic transport mechanism while the whole purpose of IKE is to use the ISAKMP language to establish shared security parameters and authentication keys - in other words, security associations - between the IPsec peers.

Rule #	Src Addr	Dest Addr	Src Port	Dest Port	Prot	Action	IPsec Hdr	Enc Alg	Auth Alg	Mode
1	SG1	SG2	500	500	Any	Accept	--	--	--	--
2	SG1	SG2	Any	Any	Any	IPsec	AH	--	HMAC-SHA-1	Tunnel
3	H1-1	Any	Any	Any	Any	IPsec	ESP	DES	HMAC-SHA-1	Tunnel
4	N1	N2	Any	Any	Any	IPsec	ESP	3DES	HMAC-SHA-1	Tunnel

Figure 4. Example of Security Association Policy Database

Negotiated policy is, in IPsec terminology, referred to as **Security Association** (**SA**), which is in essence, an agreed way of handling the data that will be exchanged between two peer devices. An example of a policy item is the algorithm used to encrypt data. SA policy is, once when negotiated, stored and maintained in the **SPD** (**SA Policy Database**). Each entry of the SPD defines the traffic to be protected, how to protect it and with whom the protection is shared as a shown example SPD in Figure 4.

2.5.2 Internet Key Exchange authentication methods

The IKE protocol is very flexible and supports multiple peer authentication methods. The two entities using IKE, must agree on a common authentication protocol through a negotiation process. In the current set of standards, the following authentication mechanisms are defined: pre-shared key, digital signature and authentication with public key encryption.

- **Pre-shared keys** are the same keys preinstalled on each host. IKE peers authenticate each other by computing and sending a keyed hash of data that include the pre-shared key. If the receiving peer is able to independently create the same hash using its pre-shared key, then it knows that both parties must share the same secret, thus authenticating the other party. Pre-shared keys are a nonpublic key option. As with manual keys, each peer shares a secret key, which has been exchanged out-of-band and configured into the device.
- **Public key cryptography** requires that each party generates a pseudo-random number (a nonce) and encrypt it with the other party's public key. Authentication occurs when each party decrypts the other party's nonce with a local private key (and other publicly and privately available information) and then uses the decrypted nonce to compute a keyed hash.
- With the **digital signature**, each device digitally signs a set of data and sends it to the other party. Currently both the RSA (Rivest, Shamir, Adelman) public key algorithm and the DSS (Digital Signature Standard) are supported.

To summarize the IKE authentication mechanisms, we could say that IKE provides strong peer device authentication mechanisms, however it does not provide a mechanism for

authenticating remote users connecting from authenticated IPsec device. The only exception to that is in the case when an IPsec device is storing a digital signature to the external storage such as a SmartCard[2] and when the user is prompted to type in a digital code to access the stored signature.

2.5.3 Internet Key Exchange modes

Current IKE mechanism, as defined in RFC 2409 [12], provides **three modes** and **two phases** for exchanging the key information and setting up SAs. The creation of IKE SA is referred to as **phase 1** exchange. In phase 1 exchange, peers also authenticate each other using one of the previously mentioned authentication mechanisms. Once the phase 1 exchange is completed, **phase 2** exchange - creation of IPsec SAs - may commence. There are two exchanges that can be performed for phase 1 exchange: **Main Mode** and **Aggressive Mode**. Aggressive Mode is faster as it has just three message exchanges, but Main Mode, although more complex, is also more flexible. An example of both phase 1 exchanges with pre-shared key authentication method is illustrated in Figures 5 and 6.

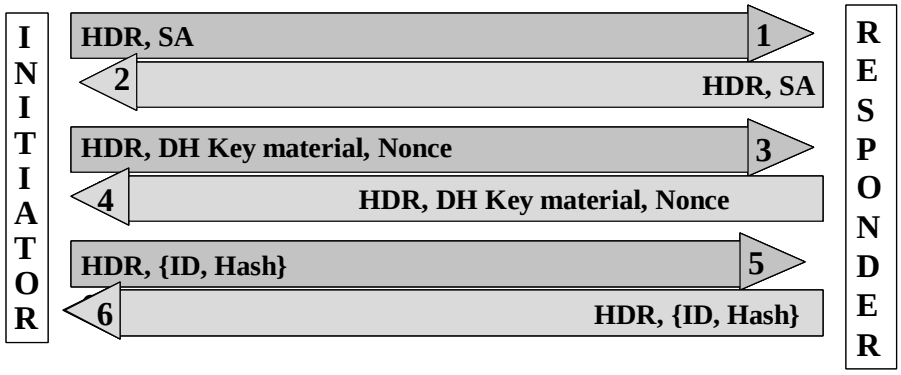


Figure 5. IKE phase 1, Main Mode exchange with pre-shared key

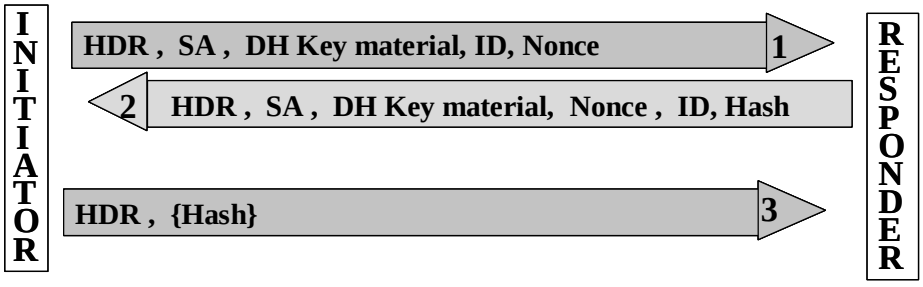


Figure 6. IKE phase 1, Aggressive Mode exchange with pre-shared key

[2] A modern SmartCard technology could also provide other cryptographic functions than just storage.

Phase 1 modes provide a mechanism for establishing IKE SA and protected[3] communication channel for negotiating following phase 2 communication parameters. The phase 2 exchange has only one mode, which is also, referred to as **Quick Mode** exchange. Quick Mode exchange is responsible for negotiating one or multiple IPsec SAs as illustrated in Figure 7.

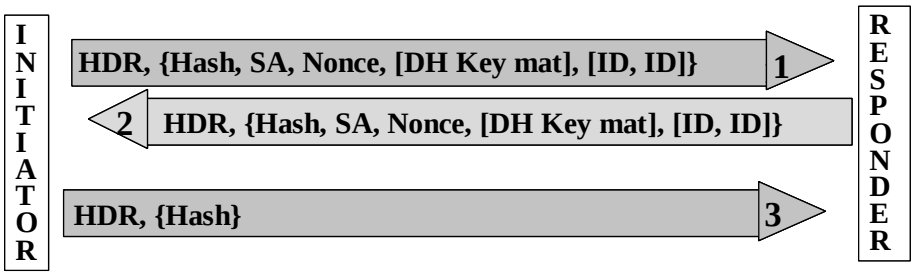


Figure 7. IKE phase 2, Quick Mode establishment of IPsec SAs

3 IPsec usage in Virtual Private Networks

3.1 Site-to-Site Virtual Private Networks

Site-to-Site VPN is a tunnel between two devices acting as security gateways [6,22]. IPsec can either directly encapsulate the IP traffic that it will secure, or it can secure another tunneling protocol e.g. GRE tunnel running between the two security gateways, where that tunnel has already encapsulated the data. The security gateways may be responsible for securing a wide variety of traffic like interactive (Telnet), file transfer and web traffic from one host or a group of hosts connecting from one site to another. The specifics of site-to-site VPN are that in most cases, security gateways will have static IP addresses so there is no need for dynamic address assignments of addresses. Typically there is also no need to do an IP address translation between the connecting sites as both sites are most of the time within a single VPN administrative domain belonging to same address space. For security gateways authentication mechanism it is sufficient to use the machine authentication only.

In many cases, there will be a need for a separate handling of different types of traffic between sites, which requires security gateways to support the QoS (Quality of Service)

[3] Quick mode exchange traffic in {} is protected by the negotiated IKE symmetric encryption session key, while parameters in [ ] are optional.

classification and marking for various types of traffic and even maybe negotiates different QoS characteristics for the various tunnels between the sites.

3.2 Remote Access Virtual Private Networks

Remote access VPN [6,17,18,22] is the most convenient method for mobile users, home workers or day extenders, having most of the time the only single machine connecting to the central corporate network. When the mobile user is outside the protected network that it needs to access, it will have to interact with a VPN gateway in order to access that network. In many cases, this type of VPN is replacing a remote access scenario where dial access was once utilized to reach an internal protected network. In this type of VPN, before the IPsec tunnel is established between the client and the VPN gateway, the client must first authenticate itself to an authentication server. The IPsec connection will most likely operate in a tunnel mode, and will most likely always be initiated by the client. Another specific of the remote access VPN is that it requires, besides a particular user authentication, also a machine authentication. In the case where there is a NAT (Network Address Translation) present on the connection path, NAT awareness is also required. In most cases where remote user is connecting to a VPN gateway, there is a need for a dynamic address assignment. Most of the time, the remote user uses only one VPN tunnel towards the VPN gateway, so it might seldom be required to use the QoS functionality in this type of VPN. Detailed requirements of remote access VPN are discussed in the remote access VPN requirements chapter.

3.3 Firewall based Virtual Private Networks

In its latest generation, a firewall[4] represents a stateful inspection system, which separates segments of the network. It recognizes and maintains traffic session flows on almost all OSI layers and securely forwards or block packets between less secure and more secure network segments. If the choice of security gateway is to use the firewall device for the VPN termination point, then we are talking about a firewall based VPN [22,32]. From the VPN technology perspective, the firewall based VPN is a hybrid solution and could be used for either site-to-site or for remote access VPN. Its specifics are only in the integration of functionality, single place of security control and cost savings while using the same device

[4] The most generic understanding of a modern firewall system.

for multiple functions. All other requirements are the same as for the above two types of VPNs.

4 Remote Access Virtual Private Networks

4.1 Comparison to legacy dial environment

Until recently, remote access has typically been characterized by dial-up users accessing the target network via the PSTN (Public Switched Telephone Network), with the dial-up connection terminating at a **NAS** (**Network Access Server**[5]**)** within the corporate domain. The protocols facilitating this have usually been **PPP** (**Point-to-Point Protocol**) based. Access control, authorization, and accounting functions have typically been provided using one or more of a number of available mechanisms, such as RADIUS (Remote Authentication Dial-In User Service) or TACAS+ (Terminal Access Controller Access Control System). PPP has also built-in mechanisms for the authentication and configuration of the remote access devices and users.

Authentication protocols, which are part of the PPP, are **PAP** (**Password Authentication Protocol**), **CHAP** (**Challenge Handshake Authentication Protocol**) or the newest one, **EAP** (**Extensible Authentication Protocol**). All of the above-mentioned protocols are capable of authenticating a remote device as well as remote users. PAP protocol is the oldest one and has been, due to its weakness of passing the username and password in clear, replaced with a newer CHAP protocol. CHAP uses a challenge response mechanism for authenticating the remote peer. The challenge is based on MD5 hashing of sequence number, shared secret phrase and the identity of a peer and as such, never passes a shared secret phrase (equivalent to password) in clear over the wire. EAP is the newest authentication protocol, which offers multiple authentication mechanisms including the challenge response mode and OTP (One-Time Password) mechanisms.

For remote device configuration running IP, PPP uses IPCP (IP control protocol). IPCP is responsible for negotiating necessary IP parameters for configuring the remote peer connecting over the PPP link, such as the IP address of a remote peer or additional servers or services.

[5] Under NAS, in general, we can think of a router with a large number of asynchronous and/or synchronous serial interfaces.

4.2 Layer 2 Tunneling Protocol Overview

Dynamic authentication and configuration possibilities within PPP have encouraged the development of several PPP-based tunneling mechanisms: **L2TP** (Layer 2 Tunneling Protocol), **L2F** (Layer 2 Forwarding) and **PPTP** (Point to Point Tunneling Protocol). They have all been developed to provide remote access by allowing the user to first dial into a local ISP (Internet Service Provider) POP (Point of Presence), and then tunnel an additional PPP connection over a shared network into the target network. While the first two PPP tunneling protocols offer similar though complementary architecture (PPTP and L2F), the third one (L2TP) has, as a compromised solution, the best capabilities of both. For scalable remote access solutions, it is very important to understand what features we have offered by the L2TP protocol as well as what features we are in lack off.

L2TP is the IETF standard track protocol [33] that encapsulates layer two PPP frames to be sent over IP, X.25, Frame Relay, or ATM networks. When configured to use IP as its transport, L2TP can be used as a VPN tunneling protocol over the Internet. L2TP over IP uses UDP port 1701 and includes a series of L2TP control messages for tunnel maintenance. L2TP also uses UDP to send L2TP-encapsulated PPP frames as the tunneled data. The encapsulated PPP frames can be encrypted or compressed by using PPP encryption and PPP compression. L2TP was specifically designed for client connections to network access servers, but can be also used for gateway-to-gateway connections. Through its use of PPP, L2TP gains multiprotocol support and provides a wide range of user authentication options as well as remote node configuration options.

4.2.1 Layer 2 Tunneling Protocol Modes

L2TP extends the PPP model by allowing the PPP endpoints to reside on different devices interconnected by a packet-switched network. With L2TP, a remote user has a layer two connection to an access concentrator and the concentrator then tunnels individual PPP frames to the NAS. This allows the actual processing of PPP packets to be divorced from the termination of the layer two circuit. In L2TP terminology, we are distinguishing two end points of the L2TP tunnel: **LAC** (**L2TP Access Concentrator)** and **LNS** (**L2TP Network Server**). Both, LAC and LNS are in essence layer 3 routers with additional functionality.

The **LAC** is a system that accepts the remote client connection, sits between LNS and a remote client and forwards PPP frames to and from each other. The **LNS** is a system that logically terminates the PPP session that is being L2TP tunneled for the remote client by the LAC. Depending on whether the L2TP connection from the remote client is local or PPP link, we distinguish two modes of L2TP protocol: **voluntary** and **compulsory**.

- In the **compulsory tunneling** mode, a tunnel is created without any action from the remote client and without allowing the remote client any choice. The remote client sends PPP packets to the LAC, which encapsulates them into L2TP and tunnels them to the LNS as illustrated in Figure 8. The LNS is the termination point of the PPP frames.
- In **voluntary tunneling**, a tunnel is created by the remote client, by using L2TP client software. The remote client sends L2TP packets to the NAS, which forwards them to the LNS. In voluntary tunneling, the NAS does not need to support L2TP as the LAC resides on the same machine as the remote client. This is illustrated in Figure 9.

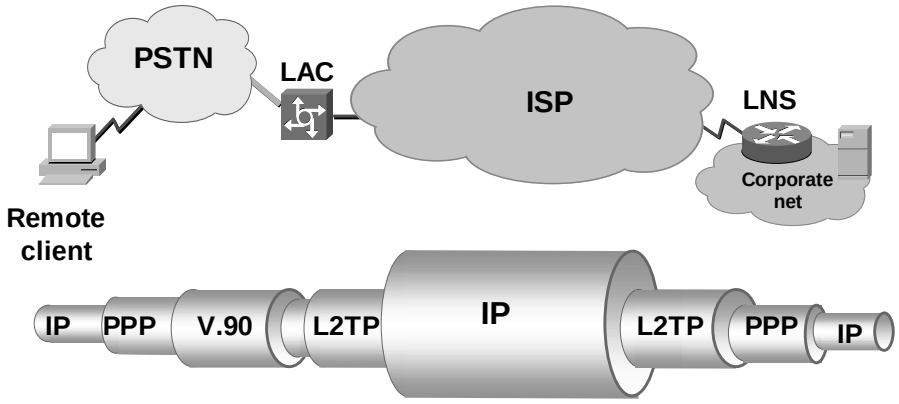


Figure 8. Compulsory mode of L2TP protocol

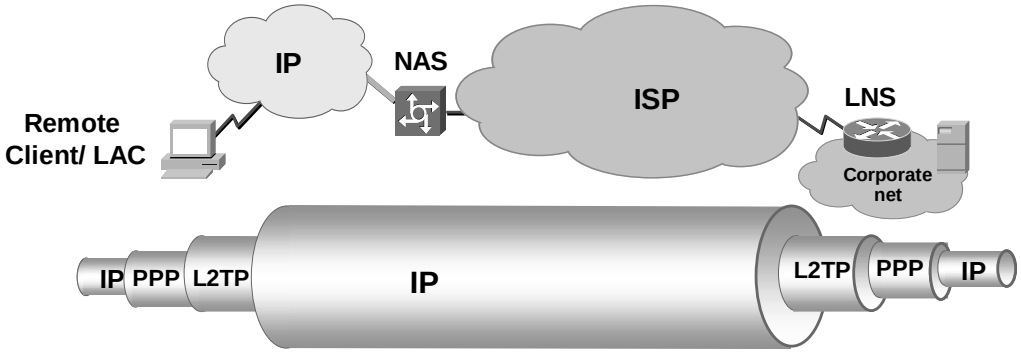


Figure 9. Voluntary mode of L2TP protocol

.

4.3 Remote Access Virtual Private Network specific requirements

There are several basic categories of requirements relevant to remote access VPN [17,18,26], some of which were already briefly outlined in the remote access VPN chapter. These are endpoint authentication, remote system configuration, security policy configuration, auditing, and NAT traversal. Here, we will explain each of them in more details. Endpoint authentication refers to the verification of the identities of the communication peers. Remote node configuration refers to the device configuration parameters of the remote system. Security policy configuration refers to the IPsec policy configuration of both the security gateway and the remote host, and might also be termed as access control and authorization configuration. Auditing refers to the generation and collection of connection status information, which is required for the purpose of statistics or maintaining the overall security and integrity of the connected networks. NAT traversal refers to the ability to pass secured traffic across intermediary nodes, some of which may modify the packets in some manner. Such intermediary nodes include NAPT (Network Address or Port Translation) devices and firewall devices. Here is the option we have for each category:

- In the context of remote access VPN, the authenticated entity may be a machine, a user (application), or both. For remote access VPN it is necessary to have the capability of authenticating either or both the end user and the device, alone or together.
- Remote node configuration refers to the network-related device configuration of the client system. This configuration may be fixed or dynamic. It may be completely provided by the administrator of the network upon which the remote user currently resides (e.g. the ISP). It may also be partially provided by that administrator, with the balance provided by an entity on the remote corporate network, where the remote VPN client is accessing. In general, this configuration may include these parameters:
  - IP address(es)
  - Subnet mask(s)
  - Domain name
  - Servers (e.g. DNS, WINS, NTP[6] etc.)
  - Default gateway(s)
  - Static routes

[6] Network Time Protocol

- NetBIOS options
- Vendor-specific options
- other options

For scalable remote access VPN deployment, it is vital to have the possibility to assign dynamically configurable parameters to remote end node.

- Security policy configuration refers to IPsec access policies for both the remote access client and the security gateway. It may be desirable to dynamically configure and enforce access policies on connecting VPN remote access client systems, which will protect the target network. For example, since a client has access to the Internet (via its routable address), other systems on the Internet also have some level of reciprocal access to the client. In some cases, it may be desirable to block the Internet access (or force it to pass only through the tunnel) while the client has a tunneled connection to the target network. This is sometimes referred to as a split tunneling policy and is a matter of client security policy configuration.

- Auditing is used to refer to the collection and reporting of connection status information by the VPN gateway, for the purpose of statistics or maintaining the security and integrity of the network.

- NAT traversal requirement refers to the passing of an IPsec secured data stream through an intermediary node such as a firewall or NAPT device. In the case of firewalls, numerous deployed products do not recognize the IPsec protocol suite, making it difficult or even impossible to configure them to pass it through. In such cases, a mechanism is required for making the data stream appear to be of a type, which the firewall is capable of managing. In the case of NAPT devices, there are a number of issues attempting to pass an encrypted or authenticated data stream. These issues will be discussed further in the IPsec and Network Address Translation chapter.

4.4 Layer 2 Tunneling Protocol protected with IPsec

The PPP protocol has the capability, via ECP (Encryption Control Protocol), to negotiate and encrypt the data on the layer 2. The L2TP protocol has built in mechanisms for tunnel end nodes authentication. In spite of both, it has been decided from the early days of development that the L2TP protocol itself or PPP authentication and encryption schemes do not meet the security requirements for L2TP tunneling for several reasons as follows:

- L2TP tunnel authentication provides mutual authentication between the LAC and the LNS only at tunnel initiation. Therefore, it does not protect control nor data traffic on a per packet basis. Thus, L2TP tunnel authentication leaves the L2TP tunnel vulnerable to attacks.
- PPP authenticates the client to the LNS, but does not provide per-packet authentication, integrity, or replay protection.
- PPP encryption meets confidentiality requirements for PPP traffic but does not address authentication, integrity, replay protection and key management requirements.
- PPP ECP negotiation does not provide for a protected ciphersuite negotiation. Therefore, PPP encryption provides a weak security solution, and in addition does not assist in securing a L2TP control channel.
- Key management facilities are not provided by the L2TP protocol.

To meet the above requirements, all L2TP security compliant implementations for securing both L2TP control and data packets **must also implement IPsec** protocol. By placing L2TP as a payload within an IPsec packet, L2TP benefits from the standards-based encryption and authentication mechanisms from IPsec, while the remote VPN node is (from PPP) getting ways to accomplish user authentication, IP address assignment, multiprotocol and multicast support.

However, L2TP/IPsec just partially solves the problem for remote access VPN, because of the lack of complete scalable solution. While the PPP (within L2TP) offers dynamic configuration of remote IP based device parameters, we still do need to configure IPsec parameters of the remote VPN peers manually or rely on some other distribution mechanism, like directory services. On top of that, this solution adds complexity, which might also have

performance impact on the protocol stack that needs to be implemented on the local desktop operating systems. Original IP packet carried by PPP is encapsulated in L2TP packets which itself runs on top of UDP (User Datagram Protocol) protocol that is than protected by the IPsec (ESP). IPsec is running directly on top of carrier IP that runs over certain media (could be again PPP or Ethernet, Token-ring etc.). This is illustrated in Figure 10.

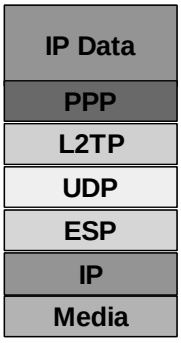


Figure 10. L2TP protected with IPsec Protocol Stack

In the L2TP/IPsec combination, L2TP is providing tunneling capability, so it is sufficient to use the IPsec in the transport mode. However the two protocols, IPsec and L2TP, have complete independent signaling messages, and as such, could initiate or terminate the connections independently. While the initiation of the L2TP tunnel will, by default, trigger an IPsec connection as well, termination of the IPsec connection might not be signaled back to the L2TP tunnel and could potentially leave the L2TP tunnel unprotected.

Due to all of the above issues with L2TP/IPsec, we are facing a further development of multiple other protocol proposals, which try to resolve remote access VPN requirements.

5 IPsec extensions for Remote Access Virtual Private Networks

5.1 The ISAKMP Configuration Method

One of the first proposals, that offered resolution of the missing features in IKE and IPsec protocols, is the so called **ISAKMP Configuration Method** [9], which is most of the time, referred to in the abbreviated form as **IKE mode config**. IKE mode config protocol relies only on the IPsec protocol suite to enable a dynamic configuration push method towards the remote VPN peer. As we have already seen in the previous chapters, IPsec key exchange

protocol provides a framework to negotiate and generate the Security Associations. While negotiating IKE SAs, it is quite useful to push certain information to the other peer before the non-IKE SA can be established. Luckily, IKE is also flexible enough to be able to be extended in such a way as to provide configuration information and to do it securely. The IKE configuration method is an extension of the ISAKMP to provide such functionality.

The ISAKMP Configuration Method extends the IKE protocol in such a way that it adds an additional configuration phase immediately after the IKE phase 1 (Main or Aggressive Mode) and before phase 2, as illustrated in Figure 11.

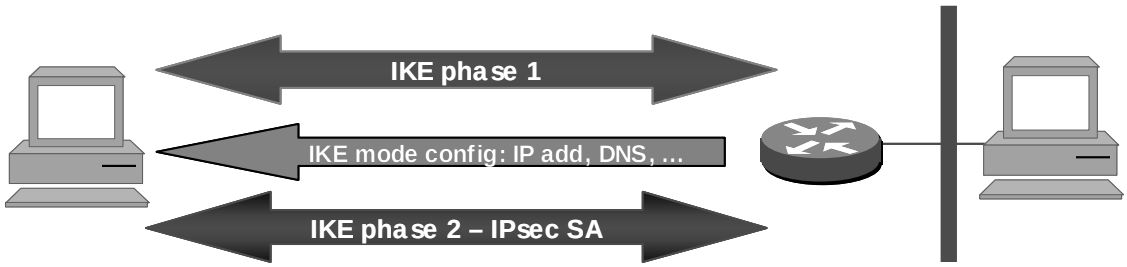


Figure 11. ISAKMP Configuration Mode

As it happens immediately after the phase 1 exchange and before the phase 2 exchange, the IKE mode config is also sometimes called **phase 1-1/2**. It is not a phase 2 exchange, because it does not result in the negotiation of an IPsec SA, and it is also independent from phase 1 SA. Each message sent via the IKE mode config protocol from the VPN gateway towards the remote access VPN peer consists of the ISAKMP header, a hash payload and the characteristic attributes payload. Attributes which could be exchanged with a remote peer are parameters that were already listed in the remote access requirements chapter, such as an internal IP address for the remote access peer, network mask, network static routes, other services that need to be dynamically configured (DNS, WINS…) or any other informational or policy parameters. Examples of informational parameters are a verification of the protocol version, vendor identification or a banner to be showed on the screen of the remote access peer.

Attribute	Value	Octets
INTERNAL_IP4_ADDRESS	1	0 or 4
INTERNAL_IP4_NETMASK	2	0 or 4
INTERNAL_IP4_DNS	3	0 or 4
INTERNAL_IP4_NBNS	4	0 or 4
INTERNAL_ADDRESS_EXPIRY	5	0 or 4
INTERNAL_IP4_DHCP	6	0 or 4
APPLICATION_VERSION	7	0 or more
INTERNAL_IP6_ADDRESS	8	0 or 16
INTERNAL_IP6_NETMASK	9	0 or 16
INTERNAL_IP6_DNS	10	0 or 16
INTERNAL_IP6_NBNS	11	0 or 16
INTERNAL_IP6_DHCP	12	0 or 16
INTERNAL_IP4_SUBNET	13	0 or 8
SUPPORTED_ATRIBUTES	14	0 or multiples of 2
INTERNAL_IP6_SUBNET	15	0 or 17
Reserved for future use	16-16383	
Reserved for private use	16384-32767	

Table 1. IKE mode config attributes

An example of policy parameters, as already mentioned in the requirements chapter is split tunneling, which could be allowed or disallowed for the remote peer. A list of the configuration parameter attributes, which could be exchanged with a remote peer via IKE mode config, is given in Table 1. Although the IKE mode config protocol supports a wide number of parameters to be exchanged within negotiation, authors of the protocol do not recommend it to be used for wide scale management, but preferably only for bootstrap information within IPsec negotiation.

It is important to mention that the IKE phase 1 and phase 2 exchanges remain completely unchanged in their role and function. Only after the completion of phase 1 is there a possibility to start with phase 1-1/2 where certain configuration parameters are pushed towards the remote peer. Phase 1-1/2 by the protocol definition, must be protected by the IKE phase 1 Security Association negotiated crypto suite and is as such secure as any phase 2 SA negotiations. In Table 1, we could also see that there are a number of attributes reserved for future use and also a significant number of attributes that are reserved for other private purpose. A reason for that is that the IKE mode config is not a stand-alone protocol. Rather it is an enabler for exchanges that could also be used in other protocols and for other purposes, such as, the user authentication that is described in the coming chapter.

5.2 Extended Authentication within IKE

The IKE as described in [12], except in the case of using SmartCards as already previously mentioned, does not provide any ways to leverage legacy user authentication methods that are widely deployed today like RADIUS, TACAS+ or OTP. The purpose of another extension of the IKE, named **IKE Xauth** (**Extended Authentication within IKE**) [4] is not to replace or enhance the existing device authentication mechanisms, but rather to allow them to be used together with the IPsec device authentication mechanisms.

The Xauth protocol combined with the IKE mode config protocol enables support for a user authentication mechanism like the two-factor authentication, challenge/response and other remote access unidirectional authentication methods.

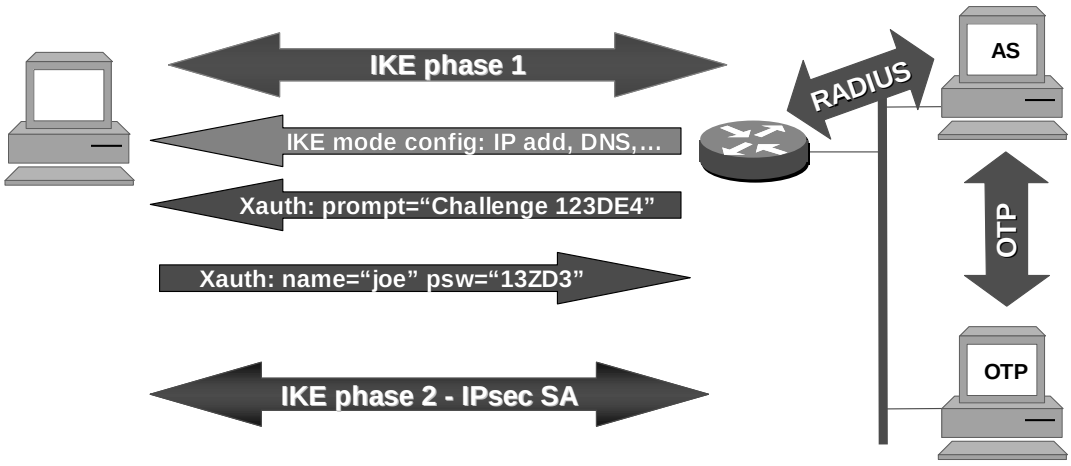


Figure 12. IKE Extended authentication mode

Xauth is designed in such a way that extended authentication may be accomplished using any mode of operation for phase 1 (Main Mode or Aggressive Mode) as well as any authentication method supported by IKE (pre-shared keys, public key cryptography or digital signatures). IKE and IPsec SAs phases, together with mode config and extended authentication intermediate phases are illustrated in Figure 12. The message exchange negotiation, for example, the RADIUS-CHAP Xauth method is illustrated in Figure 13.

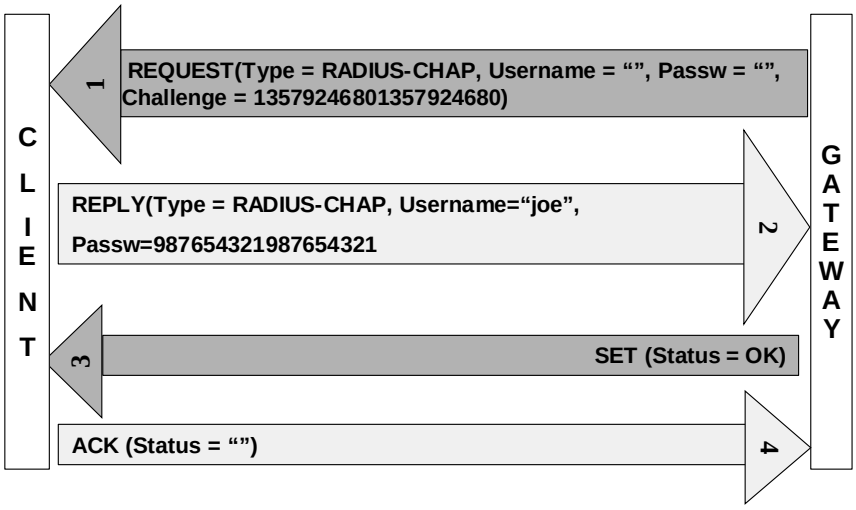


Figure 13. Xauth negotiation for RADIUS-CHAP protocol

A list of the IKE mode config attributes, which could be used with extended authentication with IKE, is given in Table 2.

Attribute	Value	Type
XAUTH-TYPE	16520	Basic
XAUTH-USER-NAME	16521	Variable ASCII
XAUTH-USER-PASSWORD	16522	Variable ASCII
XAUTH-PASSCODE	16523	Variable ASCII
XAUTH-MESSAGE	16524	Variable ASCII
XAUTH-CHALLENGE	16525	Variable ASCII
XAUTH-DOMAIN	16526	Variable ASCII
XAUTH-STATUS	16527	Basic
XAUTH-NEXT-PIN	16528	Variable
XAUTH-ANSWER	16529	Variable ASCII

Table 2. IKE Extend Authentication Attributes

The Xauth authentication protocol does not affect the nature of the IKE phase 1 authentication mechanism in any way. Both IPsec peers still must mutually authenticate each other in the IKE phase 1 exchange either via the authentication methods defined for IKE or using some other authentication method within the ISAKMP framework. It is important to mention that the Xauth exchange for remote user authentication starts only after a successful phase 1 device authentication. If the remote VPN user is successfully authenticated with any of the authentication methods supported by Xauth, IKE will continue with the negotiation of

phase 2 SAs. If however the user fails the Xauth authentication, both IKE phase 1 and Xauth phase will fail and there won't be any phase 2 negotiation.

Xauth methods, by protocol design, provide unidirectional user authentication only, meaning only one-sided, the side where the remote user resides, is authenticated using both IKE authentication methods and Extended Authentication. The other side, the VPN gateway is authenticated only with IKE device authentication methods. Types of the user authentication methods that are supported by Xauth are listed in Table 3.

Value	Authentication Required
0	Generic
1	RADIUS-CHAP
2	OTP
3	S/Key
4-32767	Reserved for future use
32768-65535	Reserved for private use

Table 3. Xauth authentication types

The IKE mode config, together with Xauth extended authentication mode of IKE, gives us the possibility to achieve vital requirements for large-scale remote access VPN deployment. With an IKE mode config, we can dynamically pass necessary configuration parameters to a remote IPsec device, while with an extended authentication, we can also authenticate the user behind the remote VPN device with legacy user authentication methods. This is the main reason why there are several implementations already existing, based on both protocols, although they are not yet accepted nor finalized within the IETF working group. The IETF directions in changing IKE protocol will be explained in detail in the chapter of IPsec future directions, however, at this point of time, there is also an alternative proposal within the IETF working group that addresses remote access VPN requirements.

5.3 Pre-IKE Credential Provisioning Protocol

While previous user authentication mechanism defines a new user authentication mode for IKE, an alternative approach defined by the IPSRA (IP Security Remote Access) working group of the IETF proposes a separate mechanism for obtaining user credentials with the **Pre-IKE Credential Provisioning Protocol** or **PIC** [31]. This is the approach to offload the user authentication task into a separate server, called an AS (Authentication Server), which

upon user authentication will provide the client machine with credentials that allow for standard IKE authentication. Such a process consists of two phases where the user client machine contacts first the AS in order to receive IKE-acceptable credentials (either the public key certificates or a strong shared key), and in the second phase, connects to a regular IKE/IPsec VPN gateway and uses these credentials within a regular IKE phase 1 establishment as illustrated in Figure 14.

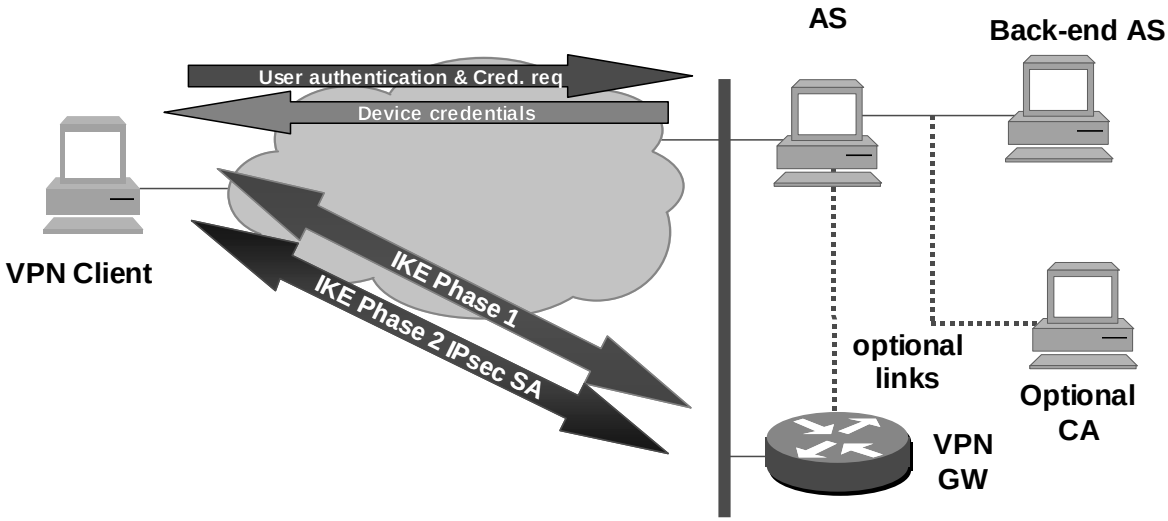


Figure 14. PIC Model

User authentication with username and password exchange via any of the EAP methods (MD5, CHAP, OTP...) in case of PIC is protected with an “IKE like” negotiation protocol as illustrated in Figure 15. Once the user is authenticated, the client machine obtains credentials from the AS that can later be used to authenticate the client in a standard IKE exchange with an IPsec-enabled security gateway. The later stage does not require user intervention. The proposed server-authenticated key exchange uses an ISAKMP-based protocol, similar to a simplified IKE exchange, and an arbitrary user authentication, which is supported via the use of the EAP protocol. The PIC method accomplishes user authentication by using an exchange, which supports legacy authentication mechanisms, and then provides the user with a private/public key pair and a certificate that are used for subsequent authentication operations with the VPN gateway. PIC protocol may be terminated by the target VPN gateway, or by a separate authentication server.

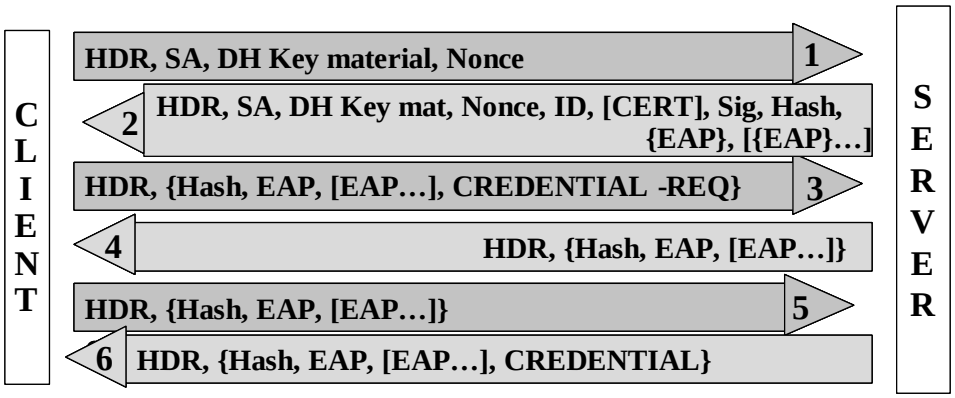


Figure 15. PIC negotiation

The PIC approach however, requires a larger number of round-trips before an IPsec association is established between a remote access VPN user and the VPN gateway as it involves the interaction with the AS first, in addition to the normal interaction with a VPN gateway. Using a separate AS for user authentication, however, has its advantages and disadvantages.

The **advantages** of the PIC mechanism are that it provides a method for integrating legacy user authentication with existing IPsec deployments without the need for modifying the underlying IPsec implementations on the VPN gateway. It is also not exposed to any denial of a service attack targeted at the AS. The migration process from legacy user name password authentication methods to advanced certificate based user authentication systems might be easier with the simple elimination of AS user authentication phase. By using short-lived certificates, PIC could also be used to achieve a single sign-on mechanism by accessing multiple resources with only one credential.

The **disadvantage** of the PIC is that it requires implementation of a separate protocol on the AS side and on the VPN client side for obtaining the user credential. Although the proposed protocol is ISAKMP based, the end result is that it adds-on to the complexity of a complete user authentication mechanism similar to the previous proposals. In spite of its advantages, due to the fact that it was developed relatively late compared to other protocol proposals for remote VPN user authentication, and also due to new developments of the IKE protocol, PIC does not have any practical acceptance yet.

5.4 Layer 2 Tunneling protocol IPsec specific extensions - L2TP/IPsec

The L2TP tunneling protocol has an advantage over the native IPsec tunneling in being capable of transporting multicast IP traffic. Being a layer two tunneling protocol, it is also capable of tunneling other layer three protocols supported by PPP, like IPX (Internetwork Packet Exchange) or AppleTalk. Based on the fact that both protocols, independently, are accepted standards, these are the additional reasons why the native L2TP/IPsec protocol combination problems, which were already discussed in the chapter of L2TP protected with IPsec, have resulted in the further development of this protocol combination within the IETF Layer Two Tunneling Protocol Extensions working group [3]. The outcome of that development is the RFC 3193 [27]. It defines requirements of a security protocol for L2TP tunnel protection and interoperability guidelines when both protocols are used together. These requirements and guidelines are listed in details in following chapters.

5.4.1 L2TP/IPsec interoperability guidelines

- When either of the peers terminates the L2TP tunnel, any phase 1 and phase 2 SA that still exist as a result of the L2TP tunnel between the peers should be also terminated. When IKE receives a phase 1 or phase 2 delete message, it should notify L2TP that this has occurred so that the L2TP tunnel state and any associated filters can also be safely removed.

- Per-packet security checks for an L2TP tunnel should assure that each packet that arrived from a tunnel was decrypted and authenticated by IPsec. Since IPsec already verifies that the packet arrived in the correct SA, L2TP can be assured that the packet was indeed sent by a trusted peer and that it did not arrive in the clear.

- As per IKE, when using pre-shared key in phase 1 negotiations, a key must be present for each peer where secure communication is required. When using a Main Mode (which provides identity protection), this key must correspond to the IP address for the peer. When using an Aggressive Mode (which does not provide identity protection), the pre-shared key must map to one of the valid identity types defined by the IPsec framework. One may wish to consider the implications for the scalability of using pre-shared keys as

the authentication method for phase 1 and whenever possible, and for a scalable deployment to consider using digital signatures.

- During the IKE phase 2 negotiations, the peers agree on which traffic is to be protected by the IPsec protocols. In Quick Mode, the traffic, which the peers agree to protect, is defined with address space, protocol, and port information. The IPsec protocol is typically agnostic about the variations of the application running on top of it, however the L2TP protocol allows the port number to float during the protocol negotiations. The L2TP specification states that implementations may use a dynamically assigned UDP source port. Another difficulty is that the current L2TP specification allows the responder to use a new IP address in its response as well. This can cause problems within the current IKE framework, so when securing the L2TP with an IPsec, the following cases must be considered:

Initiator Port	Responder Address	Responder Port
1701	Fixed	1701
1701	Fixed	Dynamic
1701	Dynamic	1701
1701	Dynamic	Dynamic
Dynamic	Fixed	1701
Dynamic	Fixed	Dynamic
Dynamic	Dynamic	1701
Dynamic	Dynamic	Dynamic

Table 4. L2TP address and port choices

To support the general case as defined in the last line of Table 4, mechanisms must be designed into the L2TP and the IPsec that allows the L2TP to dynamically inject filters into the IPsec filter database. When the initiator uses dynamic ports, L2TP must inject the filters into the IPsec filter database, once its source port number is known. If the initiator uses a fixed port of 1701, these filters may be statically defined. The any-port definition in the initiators inbound filter is needed to handle the potential port change, which may occur as a result of the responder changing its port number.

5.4.3 L2TP protection guidelines

5.4.3.1 Authentication mechanisms

In addition to the IKE authentication, L2TP implementations utilize PPP authentication methods. While the PPP provides initial authentication, it does not provide per-packet authentication, integrity or replay protection. This implies that the identity verified in the initial PPP authentication is not subsequently verified on reception of each packet.

- With IPsec, when the identity asserted in IKE is authenticated, the resulting derived keys are used to provide per-packet authentication, integrity and replay protection. As a result, the identity verified in the IKE conversation is subsequently verified upon reception of each packet. If we assume that the identity claimed in PPP is a user identity, while the identity claimed within the IKE is a machine identity, only the machine identity is verified on a per-packet basis and there is no way to verify that only the user authenticated within PPP is using the tunnel. In fact, IPsec implementations that only support machine authentication typically have no way of enforcing traffic segregation. As a result, where a machine authentication is used, once an L2TP/IPsec tunnel is opened, any user on a multi-user machine will typically be able to send traffic down the tunnel. In order to provide a segregation of traffic between users when a user authentication is used, the VPN client functionality must ensure that only traffic from that particular user is sent down the L2TP tunnel.

- When a digital signature authentication is chosen within IKE, the LNS should be able to trust several certificate authorities in order to allow tunnel client end-points to connect to it using their own certificate credential from their chosen CA (Certificate Authority) server.

When the L2TP is protected with an IPsec, both PPP and IPsec security services are available. If the VPN management is under control of the corporation, the L2TP will be used in the voluntary mode. In case of outsourced VPN solution, L2TP can be used in either voluntary, compulsory or in both modes. Depending on the mode of the L2TP tunnel, each side of the tunnel can negotiate different services.

5.4.4 L2TP/IPsec compulsory tunnel

In the case of a **compulsory tunnel**, the client sends PPP frames to the LAC, and will typically not be aware that the frames are being tunneled, nor that any security services are in place between the LAC and LNS. By obtaining the properties of the Security Association set up between the LNS and the LAC, the LNS can obtain information about security services in place between itself and the LAC. Thus, in the compulsory tunneling case, the client and the LNS have unequal knowledge of the security services in place between them. Since the LNS is capable of knowing whether confidentiality, authentication, integrity and replay protection are in place between itself and the LAC, it can use this knowledge in order to modify its behavior during PPP ECP negotiation. Since the client has no knowledge of the security services in place between the LAC and the LNS, and since it may not trust the LAC or the wire between itself and the LAC, the client will typically want to ensure sufficient security through the use of end-to-end IPsec or PPP encryption/compression between itself and the LNS.

The client will typically not trust the LAC and will negotiate confidentiality and compression services on its own. As a result, the LAC may only wish to negotiate the IPsec ESP with null encryption with the LNS, and the LNS will request replay protection. This will ensure that confidentiality and compression services will not be duplicated over the path between the LAC and the LNS. This results in better scalability for the LAC, since the client and the LNS will handle the encryption.

5.4.5 L2TP/IPsec voluntary tunnel

In the case of a **voluntary tunnel**, the client will send L2TP packets to the NAS, which will route them to the LNS. Over a dialup link, these L2TP packets will be encapsulated in IP and PPP. Assuming that it is possible for the client to retrieve the properties of the Security Association between itself and the LNS, the client will have knowledge of any security services negotiated between itself and the LNS. It will also have knowledge of PPP encryption and compression services negotiated between itself and the NAS. From the LNS point of view, it will note a PPP frame encapsulated in L2TP, which is itself encapsulated in an IP packet. If LNS retrieves the properties of the Security Association set up between itself

and the client, it can be informed of the security services in place between them. Thus in the voluntary tunneling case, the client and the LNS have symmetric knowledge of the security services in place between them.

Since the LNS is capable of knowing whether confidentiality, authentication, integrity check or replay protection is in place between the client and itself, it is able to use this knowledge to modify its PPP ECP and CCP (Compression Control Protocol) negotiation. If IPsec confidentiality is in place, the LNS can behave as though a "Require Encryption" directive had been fulfilled, not mandating the use of PPP encryption or compression. Typically LNS will not insist that PPP encryption or compression be turned off, leaving this decision instead to the client. Since the client has knowledge of the security services in place between itself and the LNS, it can act as though a "Require Encryption" directive had been fulfilled if IPsec ESP was already in place between itself and the LNS. Thus, it can request that PPP encryption and compression not be negotiated. If IP compression services cannot be negotiated, it will typically be desirable to turn off the PPP compression if no stateless method is available, due to the undesirable effects of a stateful PPP compression.

In the voluntary tunneling case, the client and LNS will typically be able to avoid the use of PPP encryption and compression, choosing to negotiate IPsec confidentiality, authentication, and integrity protection services instead, as well as IP compression, if available. This may result in duplicate encryption if the client is communicating with an IPsec-capable end-station. In order to avoid duplicate encryption/compression, the client may negotiate two Security Associations with the LNS, an ESP with null encryption, and another with confidentiality and compression. Packets going to an IPsec-capable end-station would run over the ESP with null encryption security association, and packets to a non-IPsec capable end-station would run over the other security association. Unfortunately, most of IPsec implementations today cannot support this without allowing L2TP packets on the same tunnel to be originated from multiple UDP ports.

To protect the client against eavesdropping on the wire between itself and the NAS, the PPP client may wish to put confidentiality services in place for non-tunneled packets traveling between itself and the NAS. As a result, the client may wish to negotiate PPP encryption and compression with the NAS. As in a compulsory tunneling, this will result in a duplicate encryption and possibly compression unless the PPP compression or encryption

can be turned off on a per-packet basis. Those are just some of the issues that still need to be resolved in L2TP/IPsec protocol combination further development.

5.5 Other proposals

There were several other protocol proposals [17] of how to solve remote access VPN requirement for user authentication. They have all in some way or another, contributed and influenced previously described protocols but are not further developed and are mentioned here just for historical reference and sorted alphabetically with no order of importance.

- The **GetCert** method [17] was a precursor to PIC, having provided the first example of the underlying model: as a result of a non-IPsec user authentication exchange, the user was required to obtain a certificate, which was than used to authenticate a subsequent IKE session. The primary difference between GetCert and PIC is in the transport. While PIC runs over a new ISAKMP exchange, GetCert is completely independent of IPsec, and runs over a TLS (Transport Layer Security) connection.

- The "**Hybrid"** authentication mechanism [11] attempted to address some of the shortcomings of Xauth, most notably the need to support global pre-shared keys when remote access client certificates are not available. The hybrid mechanism modified the Xauth mechanism by requiring the VPN gateway to authenticate itself using public key techniques, and deferring user authentication until after the phase 1 IKE SA is in place. That is, the hybrid requires the VPN gateway to authenticate itself to the VPN client, but not vice-versa and as such it is only a one-sided authentication.

- The **ULA** (**User-level Authentication mechanism for IPsec**) protocol [11] approach consists of forming an authenticated phase 1 SA in the same manner as Xauth, followed by the creation of a phase 2 SA which sole purpose is to protect the authentication exchange. Following a successful authentication, the phase 2 SA is either replaced, or the selectors are modified to permit access to appropriate resources. This method improves somewhat on a Xauth protocol by providing the ability to offload the user authentication to an outboard server reachable through the tunnel.

- The **CRACK** (**Challenge Response Authentication for Cryptographic Keys**) protocol [11,32] has proposed integration of the user authentication process into the key exchange negotiation by defining a new IKE phase 1 exchange type. It proposed to authenticate the VPN gateway by using public key techniques and authenticated the user by using an identity and one or more password phrases.

6 Other IPsec extensions

6.1 IPsec and Network Address Translation

While the **NAT** (**Network Address Translation**) was originally developed to address the problem of IPv4 protocol running out of the address space, it has also been used for different purposes. Home users and small office networks use NAT as an alternative to buying registered addresses and large corporations network implement NAT alone or with a firewall to protect their internal resources. This represents a problem, not just in IPsec remote access VPN scenarios, but whenever an IPsec based traffic needs to crossover to a NAT device. A particularly difficult case is if the NAT is used in many-to-one mode where it maps several private addresses to one single routable, public address which is also known as **PAT** (**Port Address Translation**) or **NAPT** (**Network Address and Port Translation**).

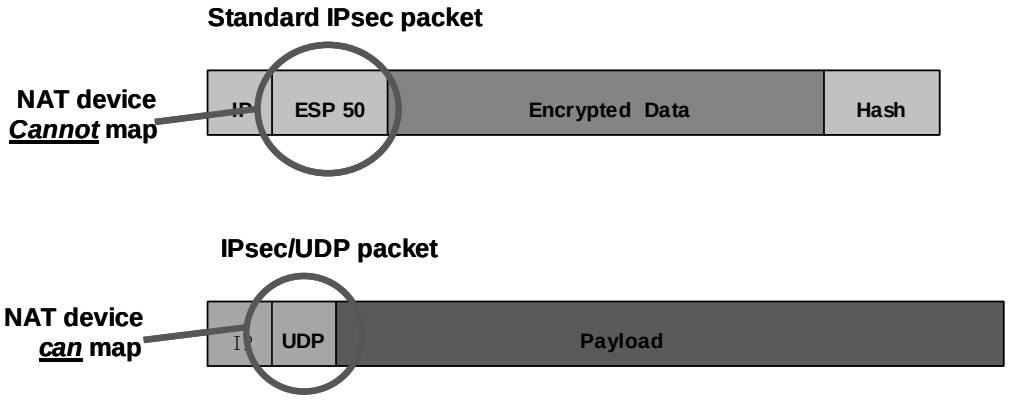


Figure 16. IPsec and NAT

Here are some of the reasons for NAT and IPsec incompatibilities [1]:

- As the IPsec AH header incorporates the IP source and destination addresses in the keyed message integrity check, the NAT or reverse NAT devices making changes to address fields will invalidate the message integrity check.
- TCP, UDP or SCTP checksums have a dependency on the IP source and destination addresses through the inclusion of the "pseudo-header" in the calculation. As a result, where checksums are calculated and checked on receipt, they will be invalidated by the passage through a NAT device. IPsec ESP will only pass without being impacted through a NAT if the TCP, UDP or SCTP protocols are not involved (as in IPsec tunnel mode or IPsec/GRE), or checksums are not calculated (as is possible with IPv4 UDP).
- If the IKE is used in Main Mode or Quick Mode, modification of the IP source or destination addresses by the NAT device will result in a mismatch between the identifiers and the addresses in the IP header.
- When multiple hosts behind the NAPT initiate IKE SAs to the same responder this can result in unpredictable behavior during the IKE session key re-generation, unless the floated IKE source port is used as the destination port for the re key.
- When multiple hosts behind the NAT attempt to bring up IPsec SAs to the same destination simultaneously, it is possible that the NAT will send the incoming IPsec packets to the wrong destination due to fact that IPsec SAs appear to be equivalent, since they exist between the same endpoints and can be used to pass the same traffic.
- Protocols that utilize embedded IP addresses (like IRC[7], LDAP[8], H.323, SIP[9] and many games) will not work since the packet payload is integrity protected so any IP addresses enclosed within IPsec packets will not be translatable by the NAT device.
- Some NAPT devices are not able to pass non-UDP/TCP traffic and hence will discard the ESP AH traffic.
- NAPTs will keep the UDP mapping in the absence of traffic only in a certain period of time. Thus, even where IKE packets can be correctly translated, the translation state may be removed prematurely.

[7] Internet Relay Chat
[8] Lightweight Directory Protocol
[9] Session Initiation Protocol

- The handling of fragmented outgoing IP packets in the case where outgoing packets are already fragmented is difficult for most NAPT devices as only the first fragment of the packet will typically contain a complete IP/UDP/TCP header and it might arrive out of order.

All the listed problems for IPsec one comes cross today in the networks frequently present with NAT or NAPT devices, have resulted in the development of another extension of the IPsec framework where the IPsec protocol is not used directly on top of the IP but has been moved one layer higher to run optionally on top of the UDP and achieves a **NAT transparency** [7] as illustrated in Figure 17.

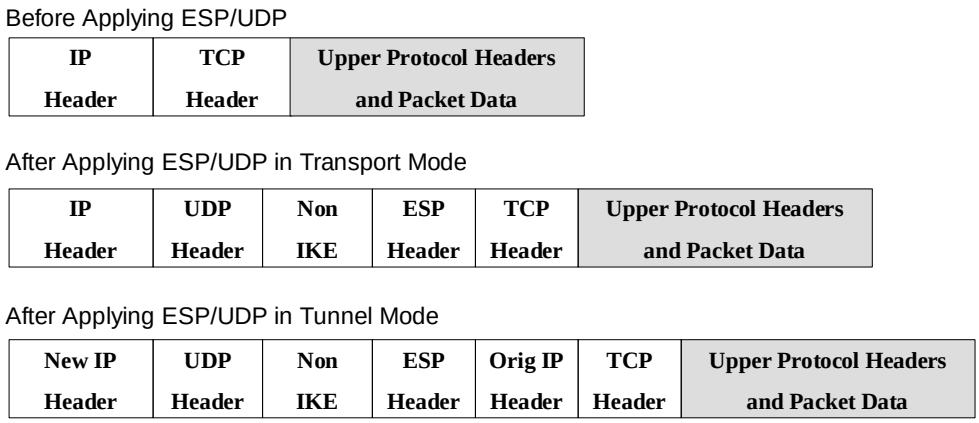


Figure 17. IPsec NAT Transparency

The goal of an IPsec-NAT compatibility solution is to expand the range of usable IPsec functionality beyond the limited NAT-compatible IPsec solution. The limited set of circumstances when it is possible for native IPsec packets to traverse NAT or NAPT successfully is as follows:

- If only the IPsec ESP header in tunnel mode is used.
- If there is no source address validation of a remote IPsec peer.
- If “any to any" SPD entries are used for IPsec tunnels.
- If only a single client is behind a NAT.
- If there is no packet fragmentation.
- If VPN sessions maintain ongoing traffic flow during their lifetime, so that UDP port mappings are not removed due to inactivity.

The last workable IPsec through NAT circumstance discloses another non-yet standardized solution within the current IPsec framework. It is the way to detect whether the IPsec remote peer is still active or not and, based on that information, keep up or clear any existing relationship among the IPsec peers.

6.2 Dead Peer Detection mechanism

When two IKE/IPsec peers communicate, the situation may arise where connectivity between them is unexpectedly lost. This could happen because of the routing connectivity problems, one host rebooting or any other problems. In such cases, there is often no way for the IKE and IPSec to identify the loss of peer connectivity. As a consequence, the SAs can remain active on one or both peers until their lifetime timers naturally expire, resulting in a situation where the packets are tunneled to a "black hole". It is often desirable to recognize “black holes” as soon as possible so that either of the peers can failover to a different, redundant peer as illustrated in Figure 18.

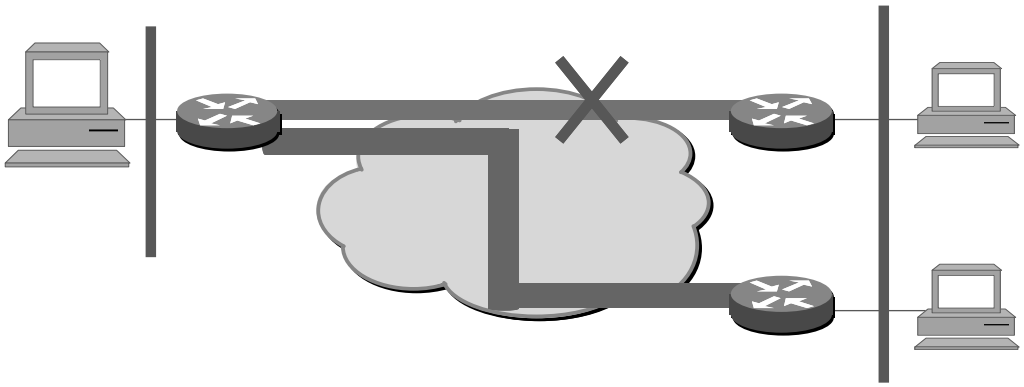


Figure 18. Failover scenario with Dead Peer Detection keepalive mechanism

The problem of detecting a dead IKE peer has been addressed in the new IPsec extension proposals that require sending hello messages to prove the liveliness of the peer. The schemes, which use unidirectional mechanisms, are commonly referred as "heartbeat”, while the bi-directional schemes are also known as "keepalive" mechanisms.

A mechanism, which combines both the unidirectional and bi-directional methods is referred as a **DPD** (**Dead Peer Detection)** protocol [15]. The DPD protocol addresses the shortcomings of the IKE keepalive and heartbeat schemes by introducing a more reasonable logic governing message exchange. Essentially, keepalives and heartbeats mandate the exchange of hello messages at regular intervals, while with the DPD, each peer DPD state is largely independent of the others. A peer is free to request proof of liveliness when it needs it - not at mandated intervals. This asynchronous property of DPD exchanges allows fewer messages to be sent, and achieves greater scalability. The DPD protocol is necessary in the IPsec extension for the site-to-site VPNs where it is mostly used for redundancy and failover purposes. However, it is also important in remote access VPN scenarios where the VPN gateway needs to maintain or clear a large number of relationships with potentially disconnected remote access VPN peers.

7 Usage of Remote Access VPN IPsec extensions on a router

A router[10] is in its most generic definition, a layer 3 forwarding device of an OSI network layer packets with multiple interfaces. Low-end routers today typically provide broadband high-performance connection to the Internet. However, business applications require not just a high-speed Internet access but also the security of VPN connections that perform a high level of authentication and encryption of the data between two particular endpoints. Establishing a VPN connection between the two routers can be complicated, and it usually requires a tedious coordination between the network administrator and an end user to correctly configure VPN parameters of the two routers.

7.1 Easy Virtual Private Network concept

To allow a simple and efficient VPN connectivity from low-end router devices, such as 800, 900 or 1700 family of routers, Cisco Systems Inc. decided to extend the IOS (Internetworking Operating System) router software with easier way of configuring Virtual

[10] This is only a generic definition of the router, while the modern router today acts mostly as a hybrid system.

Private Network functionality with a name **Easy VPN** [5]. The Cisco Easy VPN functionality in the IOS software simplifies most of the IPsec policy configuration by implementing IPsec extensions such as the IKE mode config and Xauth, which allows a majority of the VPN parameters to be defined at a VPN gateway acting as an IPsec server. After the IPsec server has been configured, a VPN connection can be created with minimal configuration on an IPsec client using the Easy VPN functionality. When the IPsec client initiates the VPN tunnel connection, the IPSec server pushes the IPSec parameters to the IPSec client and creates the corresponding VPN tunnel connection.

7.2 Router specific issues regarding IPsec extensions

By deploying VPNs for teleworkers and small branch offices, ease of deployment is critical, especially when skilled technical resources are not available for VPN configuration on remote site routers. Both the Easy VPN Remote and the Easy VPN Server features offer flexibility, scalability, and ease of use for site-to-site and remote-access VPNs.

The **Easy VPN Remote** feature allows low-end router routers to act as remote VPN clients. As such, a router can receive predefined security policies from the headquarter VPN head-end gateway (**Easy VPN Server**), thus minimizing the VPN configuration required at the remote location. This is not just a technically scalable solution but is also a cost effective solution ideal for remote offices with no technical support and particularly for large-scale deployments where it is impractical to individually configure multiple remote devices.

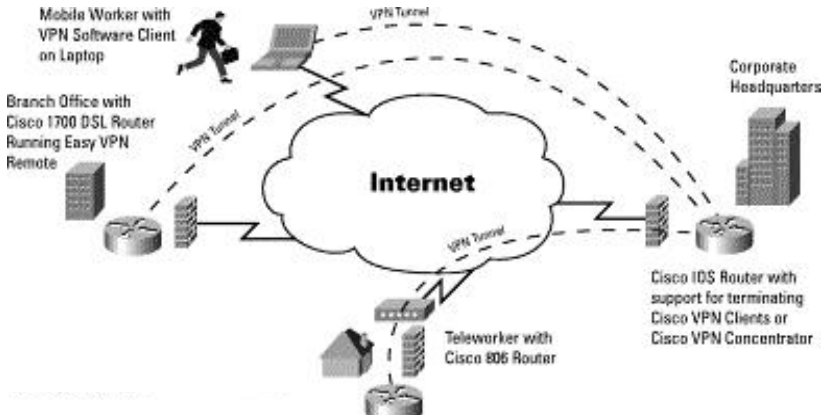


Figure 19. Cisco IOS Easy VPN concept

The **Easy VPN Server** feature allows a router to act as a VPN head-end device in site-to-site or remote-access VPNs, where the remote office routers are using the Easy VPN Remote feature. Using this feature, security policies defined at the head-end can be pushed to the remote office routers. In addition, an Easy-VPN-Server-enabled router can terminate VPN tunnels initiated by mobile remote workers running VPN client software on PCs (Personal Computers) as illustrated in Figure 19.

A router acting as a VPN termination point, compared to VPN software client running on individual PCs, has several benefits which includes the following:

- The centrally stored configurations allow dynamic configuration of end-user policy, require less manual configuration by end-users and field technicians, reducing errors or further end user support.
- The local VPN configuration is independent of the remote peer's IP address, allowing the VPN provider to change equipment and network configurations as needed, with little or no reconfiguration of the end-user equipment.
- Provides for centralized security policy management.
- Enables large-scale deployments with rapid user provisioning.
- Removes the need for end-users to purchase and configure external VPN devices.
- Removes the need for end-users to install and configure VPN client on their PCs.
- Offloads the creation and maintenance of the VPN connections from the PC to the router.
- Reduces interoperability problems between the different PC-based software VPN clients, external hardware-based VPN solutions, and other VPN applications.

As a mediator connectivity device to several PCs locally attached to its LAN segment, a router typically needs to provide additional functionalities besides VPN. These may be a dynamic address or services assignment, address or port translations (NAT/PAT), control of split tunneling or group and individual user authentication. Services and functions that a VPN router will provide depend on whether it is acting as a client or as a server side of the VPN connection, whether there is a single or multiple PCs attached to it and whether the ISP

(Internet Service Provider) provides single or multiple addresses assigned to a particular Internet connection.

7.2.1 Easy VPN Remote specific features

When being placed at a remote side of the VPN connection and using the Easy VPN Remote software feature set, a router can provide automatic management of the following:

- Negotiating tunnel parameters - such as addresses, algorithms or SA lifetime.
- Establishing tunnels according to the parameters.
- Automatically creating the NAT/PAT translation.
- Authenticating users - making sure users are who they say they are, by way of usernames, group names and passwords.
- Managing security keys for encryption and decryption.
- Authenticating, encrypting, and decrypting data through the tunnel.

Using IKE extended authentication (Xauth) does authentication of the individual user while IKE mode config is used for configuring the additional parameters on the client.

7.2.1.1 Network Address Translation considerations

With regard to the NAT, the Easy VPN Remote feature supports two modes of operation: Client mode and Network Extension mode. Both modes are illustrated in Figures 20 and 21.

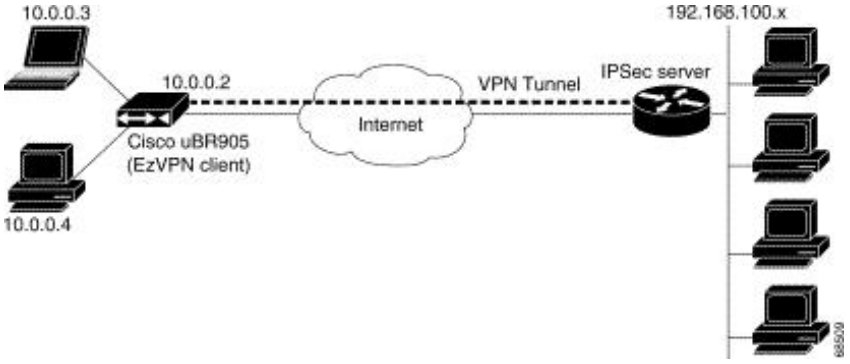


Figure 20. Easy VPN Remote - Client mode

In the **Client mode**, a remote VPN router acts as a single VPN client to a VPN gateway. This is achieved by using PAT, so that all PCs at the other side of the VPN tunnel use a private network address space that is translated to only single IP addresses in the destination network address space. In the Client mode, the Easy VPN Remote feature automatically configures the PAT translation and access lists that are needed to implement the VPN tunnel. These configurations are automatically created when the IPSec VPN connection is initiated. When the tunnel is torn down, the PAT and access list configurations are automatically deleted.

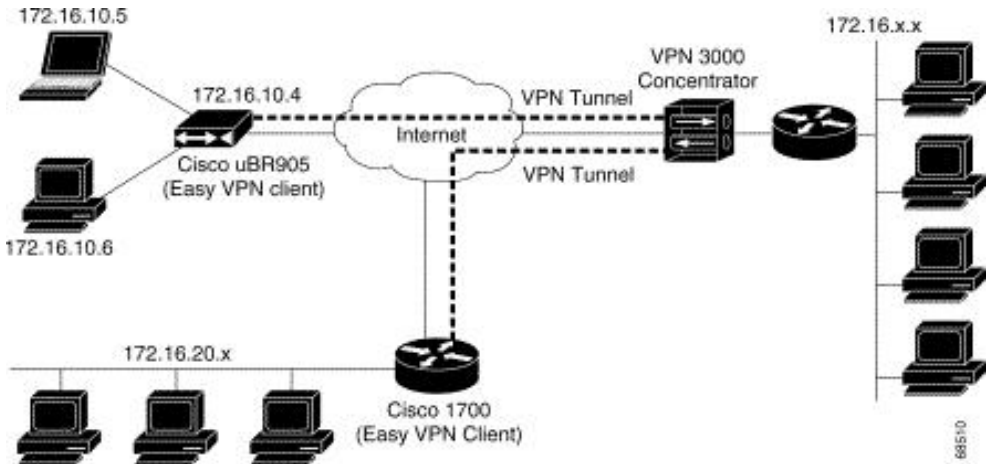


Figure 21. Easy VPN Remote - Network Extension mode

The **Network Extension mode** specifies that the PCs at the client end of the VPN tunnel should be given IP addresses that are fully routable and reachable by the destination network over the tunneled network. PAT is not used, which allows the client PCs to have direct access to the PCs at the destination network and vice versa.

Both modes of operation optionally support split tunneling, which allows a secure access to the corporate resources through the VPN tunnel while also allowing the Internet access through an ISP connection. A comparison of both modes of the Easy VPN Remote is given in Table 5.

	Client mode	Network Extension mode
VPN tunnel establishment	The VPN tunnel could be manually or automatically established.	By default, the VPN tunnel is automatically established but could be also manually controlled.
Mode config: IP address	Required. This is the PAT address for all outbound VPN traffic.	Not required. Local hosts keep routable addresses.
Mode config: DNS, WINS, and domain name	When configured as a DHCP server, the router will use the mode config pushed parameters in the DHCP response packet to local PCs.	When configured as a DHCP server, the router will use the mode config pushed parameters in the DHCP response packet to local PCs.
Mode config: split-tunnel	If not specified, all traffic will go through the VPN tunnel. If specified, packets matching the policy will route to the VPN tunnel otherwise will go through the router rules for clear traffic[11].	If not specified, all traffic will go through the VPN tunnel. If specified, packets matching the policy will route to the VPN tunnel otherwise will go through the router rules for clear traffic.

Table 5. Client mode versus Network Extension mode

In either of the two modes, there might be a need for using the IPsec NAT transparency functionality, explained in detail in previous chapters, but although this is currently not a supported feature, it is certainly under consideration for future development.

7.2.1.2 Dynamic update of DHCP parameters

The router in a branch office is, most of the time, also doing a dynamic assignment of IP addresses to locally attached PCs. Through the DHCP (Dynamic Host Configuration Protocol) protocol, attached PCs are obtaining not just IP address but also information about the default router, DNS or WINS servers as well as the default domain name. These parameters could, of course, be configured statically in the configuration file of each remote branch office router. However, in the case of a router with an Easy VPN Remote functionality, it is much more scalable to update and change DHCP parameters dynamically from the central site location. Each remote branch office router could obtain and dynamically import necessary DHCP parameters from the Easy VPN Server it connects to and update its

[11] The router typically has an additional filtering access list to allow or block certain clear traffic on the interface to the Internet.

DHCP server parameters upon the VPN tunnel establishment via the IKE mode config mechanism. With that, DHCP parameters need to be maintained and updated or changed only on the central VPN gateway. They will be pushed dynamically to all connecting remote VPN branch routers, which will then serve proper updated DHCP information to all remote PCs. An example of such a mechanism is shown in Appendix A.

7.2.2 Easy VPN Server specific features

An Easy-VPN-Server-enabled Cisco IOS router such as Cisco 7100, Cisco 7200, Cisco 3600, or a Cisco VPN Concentrator series is typically used as a head-end VPN gateway to accept connection from branch offices, remote offices, and teleworkers. The head-end VPN gateway must have pre-configured security policies to determine which VPN parameters will be used to communicate with remote devices. When the head-end security policies have been defined, branch offices can deploy Easy-VPN-Remote-enabled routers. Once VPN connections are established, the head-end security policies are pushed to the remote devices with minimal configuration. In regard to the dynamic negotiation of all IPsec related parameters, VPN head-end gateway can also, during the VPN tunnel establishment, have complete dynamic control of split tunneling policy, address and services present at the central site as well as the redistribution of connected remote network addresses towards the rest of the central site network.

7.2.2.1 Split tunneling

Remote VPN clients can support split tunneling, which is the ability for a VPN client to have intranet (though a VPN tunnel) and Internet accesses at the same time. If split tunneling is not configured, the client will direct all traffic through the VPN tunnel, even traffic destined for the Internet. As split tunneling poses security risk to a corporate network, it is up to the corporate security policy to define and enforce the split tunneling policy. Using attributes in the IKE mode config configure on the head-end VPN gateway, corporate can **control the split tunneling policy centrally**. During the VPN tunnel establishment, the VPN gateway

can push mode config attributes to allow or disallow usage of a split tunneling on remote VPN device and thus enforce split tunneling policy on the remote VPN end nodes.

7.2.2.2 Address and DHCP parameters assignments

In the case when the remote VPN end node is a single PC or a router running the Easy VPN Remote in the Client mode, it is necessary for a VPN head-end gateway to do a dynamic IP address assignment to a remote end of the VPN tunnel. First, the VPN gateway needs to have the capability to have a static poolls of addresses or a way to dynamically prevision IP addresses from a DHCP server. Next, it needs the capability to push that IP address to a remote end. This is also done also via a IKE mode config. For a large-scale deployment it is not enough to configure just the IP address for the PAT purpose, but also the information about the default router, DNS or WINS servers as well as the default domain name. These parameters should be either configured statically on a VPN head-end getaway or obtained from local DHCP server and then provisioned to a remote VPN end node via a IKE mode config.

7.2.2.3 Reverse Route Injection and Hot Standby Router Protocol

The central VPN head-end gateway is on one side terminating the VPN tunnels and encrypting or decrypting the traffic. On the other side, it is also responsible for the distribution of that traffic to internal parts of the network in addition to routing the packets targeted for remote VPN sites down the VPN tunnel. If there is just a single VPN head-end gateway in the network, this could be easily achieved via configuring static routes for all remotely available networks. However, this task represents a problem in the case of multiple VPN head-end gateways in a redundant scenario design as illustrated in Figure 22.

In the below or similar designs, it is necessary to track which VPN head-end gateway has the active IPSec connection with a remote VPN peer to ensure that tunnels are not duplicated across gateways.

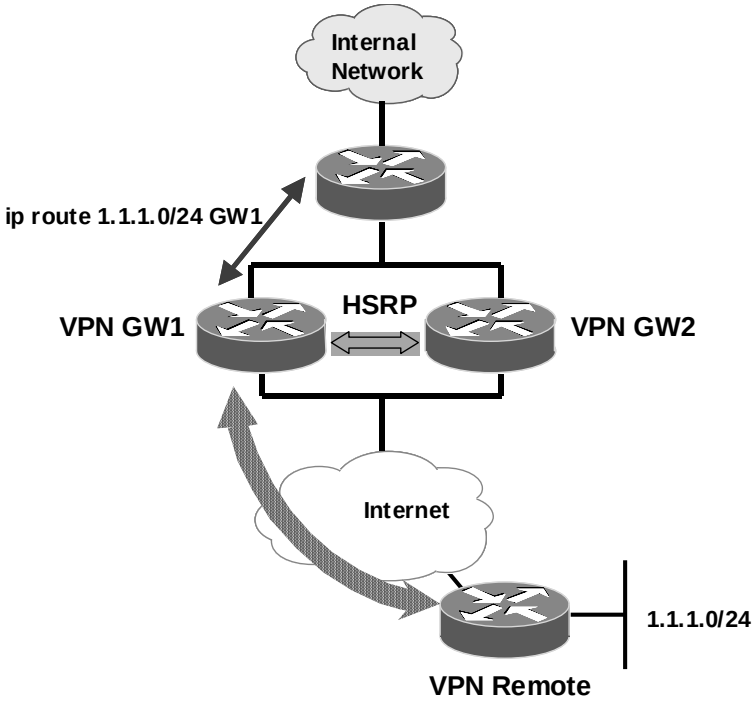


Figure 22. Reverse Route Injection and Host Standby Router Protocol

RRI (**Reverse Route Injection**) is a VPN head-end gateway feature that resolves the problem by injecting a static route in the routing table of the VPN getaway that has an active VPN tunnel. The primary benefit of RRI is that it enables the dynamic routing of VPN traffic to a specific VPN head-end device in environments with multiple, redundant VPN head-end devices. It is based on which device currently holds the VPN session for a specific peer. Advertising this route via dynamics routing protocol then ensures that return VPN traffic associated with the specific session will be routed back through the VPN gateway device that has the active VPN session.

HSRP (**Hot Standby Router Protocol**) is another function responsible for the resiliency of routers. While HSRP is the proprietary protocol of Cisco Systems, an equivalent functionality exists in a VRRP (Virtual Router Redundancy Protocol). HSRP creates a hot standby router group with a lead router that services all packets sent to the hot standby address. Other routers in the group monitor the lead router, so that if it fails, one of these standby routers inherits the lead position and the hot standby address.

HSRP complements the **RRI** feature in maintaining a network resiliency. Using HSRP, two or more routers can work in concert to present the single virtual router with a virtual IP address. The hosts on the internal network recognize the virtual router and IP address as the only router and IP address. The set of routers that comprises the virtual router is known as an HSRP group, or a standby group. A single router elected from the group is responsible for forwarding the packets that the hosts send to the virtual router. This router is known as the active router. Another router is elected as the standby router. In the event that the active router fails, the standby router assumes the packet forwarding duties of the active router. Although an arbitrary number of routers may run HSRP, only the active router forwards the packets sent to the virtual router. If the active router fails, the standby router takes over as the active router. If the standby router fails or becomes the active router, another router is elected as the standby router. RRI then informs peers of the active router, ensuring that peers use the active tunnel that HSRP has established.

While **HSRP** and **RRI** can be used in conjunction with each other for maximum network resiliency, they can also be **used separately**. RRI is also a good solution for topologies that require encrypted traffic to be diverted to a VPN router and all other traffic to a different router. In these scenarios, RRI again eliminates the need to manually define static routes on devices. Routing convergence can affect the success of a failover based on the routing protocol used to advertise routes (link state versus periodic update). It is recommended that a link state routing protocol such as OSPF (Open Shortest Path First) be used to help speed convergence time by ensuring that routing updates are sent as soon as a change in routing state is detected.

8 IPsec future directions

The IPsec standard versatility has already been recognized as a good mechanism for securing IP protocol based communication. Based on the present standard framework, numerous interoperable implementations[12] already exist among a significant number of vendors. The current IPsec framework also has the capability of easily adding new encryption or hashing

[12] The most comprehensive list of interoperability tests is hosted by the Virtual Private Network Consortium at www.vpnc.org/features-chart.html

algorithms. Proofs of that are the Internet draft proposals within the IETF IP Security working group for adding an AES or SHA-256. There are also additional AH and ESP header developments to accommodate high-speed networking. However, as we have seen so far, the IPsec framework, although complex and flexible, has not addressed all related problems when securing the IP protocol. We have already exposed problems when the IPsec packets need to traverse NAT devices, the detection of dead peers, as well as missing dynamic configuration or individual user authentication mechanisms.

8.1 New key exchange mechanism proposals

On one hand, IPsec framework flexibility is an advantage and was meant to block the development of nonstandard or proprietary extensions. On the other hand, it has the obvious drawback of increasing the complexity of the protocol [10,28,29]. The complexity, which may easily lead to faulty and non-secure implementations, is in particular present in the key exchange mechanism, IKE. This was one of the main reasons why the IETF decided to take the approach of developing new key management mechanisms for IPsec that will address new requirements and also reduce complexity by removing identified unnecessary components.

8.1.1 Son of IKE requirements

One of the approaches taken in the development of a new key exchange mechanism to succeed IKE (also known as "**son of IKE**" or **SOI**) [16], is to first define what are the requirements for the new key exchange mechanism and the new foreseen areas and where and how it may be used. Each of the potential domains of usage has its own requirements, which have been identified as:

- General - type of the VPN network site-to-site, remote access, full meshed…
- Dynamic addressing - static or dynamic, public or private.
- NAT support - depending on addressing type - may be required or not.
- QoS support - required traffic classification or not.
- Policy - static or dynamically configured.
- Security characteristics - authentication and identity protection.

The potential domains of usage for SOI that have been identified so far are listed below. However, in real life scenarios, there would certainly be situations that will require hybrid solutions including multiple domain requirements at once.

- Virtual Private Network Site-to-Site Tunnels
- Secure Remote Access
- End-to-End Security
- IP Storage
- PPVPN[13]/MPLS
- Other Areas (Mobile IP, Wireless, Delay-sensitive Applications…)

It is important to mention that the future key exchange protocol has, through SOI requirements document [24], recognized that the IPsec usage scope is currently mostly deployed in VPNs. However, it is also affecting other protocols based on IP transport and as such could be impacted by the other areas of usage as well. The details of the site-to-site VPN and the remote access requirements have already been described in previous chapters, the other domains are currently outside the scope of this document.

8.1.2 IKEv2 - Internet Key Exchange version 2

The IKEv2 (Internet Key Exchange version 2) protocol [13,14] is as the other proposals still in the development. The main goals of the current IKEv2 proposal, which is mostly based on the original IKE version 1 (IKEv1), are to simplify existing IKEv1 protocol without making any gratuitous changes and fix ambiguities or bugs in protocol definition. It also proposes to add flexibility deemed necessary and to reduce complexity of phase 1 exchange. The goals that authors of the proposal have put for IKEv2 are:

- To simplify IKE by eliminating the Aggressive Mode option and reduce the authentication algorithms, making phase 1 a single exchange based on public signature keys or pre-shared key.

[13] Provider Provisioned VPN in relation with IPsec VPN to MPLS VPN mapping

- To decrease IKE latency by making the initial exchange to be 2 round trips (4 messages), and allowing the ability to piggyback setup of a child IPsec SA.
- To replace the cryptographic algorithms for protecting the IKE messages themselves with one based closely on ESP to simplify implementation and security analysis.
- To increase robustness by allowing the Responder, if under attack, to require return of a cookie before the Responder commits any state to the exchange.
- To specify Traffic Selectors in their own payload type rather then overloading ID payloads, and facilitate the Traffic Selectors that may be specified;
- To avoid unnecessary exponential explosion of space in attribute negotiation, by allowing choices when multiple algorithms of one type (say, encryption) to work with any of a number of acceptable algorithms of another type (say, integrity protection).
- To specify required behavior under certain error conditions or when data that is not understood is received, in order to make it easier for future revisions in a way that does not break backwards compatibility.
- To simplify and clarify how a shared state is maintained in the presence of network failures and Denial of Service attacks.
- To maintain existing syntax and magic numbers to the extent that implementations of IKEv1 can be enhanced to support IKEv2 with minimum effort.

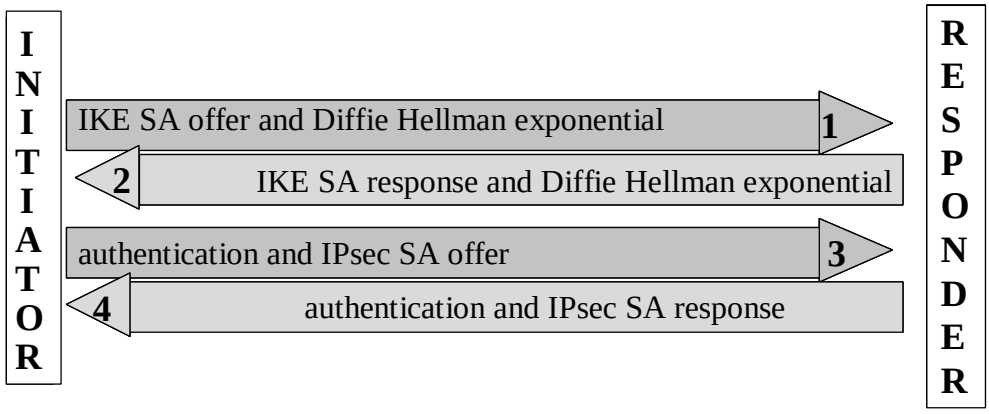


Figure 23. IKEv2 Main Mode exchange

The IKE v2 based on listed goals simplifies phase 1 and phase 2 exchanges by replacing the 8 messages (of Maine Mode and Quick Mode) with a single exchange of up to a minimum 4 messages as illustrated in Figure 23 simplified messages flow based on public signature keys or still debatable pre-shared keys.

The single exchange provides identity hiding, yet works in 2 round trips. Latency of setup of an IPsec SA is further reduced from IKEv1 by allowing the setup of an SA for ESP, AH or IP compression (IPcomp) to be piggybacked on the initial IKE exchange. The IKEv2 still exercises the option of having two phases and the capability of negotiating multiple child IPsec SAs. With keeping separate phase 2, IKEv2 avoids multiplexing of several conversations over the same SA and allows different security or quality of service policies over separate SAs. The IKEv2 protocol is still flexible enough to allow extensions and it can use another port number besides UDP 500, so that the new protocol would work through NAT. It is also presented in a single self-contained document, in contrast to IKEv1, which was described in three different RFC documents.

8.1.3 JFK - Just Fast Keying

Another proposal for a new key management protocol instead of the IKEv1 has a goal to start simple and hence its name, **JFK** (**Just Fast Keying**) [2]. The main issue, that JFK is trying to solve, is a complexity of IKEv1. The simplicity property is motivated by several factors, the primary one being efficiency. Motivation of efficiency is also colored by experience with the IKEv1 where even if the protocol is defined correctly, due to its complexity, it does not necessarily mean that it is also always implemented correctly. If its definition is too complex, implementers might get it wrong and by doing so hinder both security and interoperability. The JFK achieves simplicity by deliberately removing some functionality, which is present in the current key management protocol. In particular, JFK is lacking the following:

- Any form of authentication other than digital signatures.
- The JFK also completely eliminates negotiation, in favor of options issued by the Responder. The JFK standpoint is that the Responder is providing a service, so it is entitled to set its own requirements for that service. Any cryptographic primitive mentioned by the Responder is acceptable and the Initiator is free to choose any it wishes,

which thus eliminates complex rules for selecting the ”best” choice from two different sets.

- A re-keying mechanism is not existent in JFK. When a negotiated SA expires (or shortly before it does), the JFK protocol needs to run again. The JFK argument is that running the key management protocol is not a big performance consideration if the protocol is simple and fast enough.
- JFK does not have the notion of two different phases. It sees the practical benefits of a quick mode as limited and does not agree that frequent re-keying is necessary. The fundamental idea is that if the underlying block cipher is not cryptographically strong enough, the proper solution is to replace it with a stronger cipher. For example, if a 3DES is inadequate for protection of very high-speed transmissions, using AES instead of 3DES solves that problem without complicating the key exchange protocol.

The JFK proposal is also to also limit the set of algorithms and algorithm combinations for ESP and AH to only subsets as given in Table 6.

Header	Algorithm set
ESP	ESP-AES-CBC with HMAC-SHA1 ESP-3DES-CBC with HMAC-MD5 ESP-3DES-CBC with HMAC-SHA1 ESP-NULL with HMAC-MD5 ESP-NULL with HMAC-SHA1 ESP_BYPASS
AH	AH with HMAC-MD5 AH with HMAC-SHA1 AH_BYPASS

Table 6. JFK subset of ESP and AH algorithms

As we have seen, a JFK proposal does not support pre-shared keys or any other extension mechanisms for possible user authentication like the token-based authentication. It leaves other authentication mechanisms to be carried out by other new external protocols.

8.1.5 SIGMA - Signature Mode of Authentication

The third of the initial three proposals for new key exchange mechanism is a variant of the current digital signature mode of authentication of IKEv1. The proposed protocol, named

SIGMA (**Signature Mode of Authentication**) [23], describes a simplified variant of the signature modes of IKE as well as a new simple pre-shared key mode. The SIGMA protocol in essence suggests using an authenticated Diffie-Hellmann exchange with the MAC (Message Authentication Code) of a peer identity within a signature and tries to achieve several seemingly conflicting goals: simplification of the protocol, enhanced functionality, and performance improvement. The packet exchange of SIGMA with the digital signatures is illustrated in Figure 24.

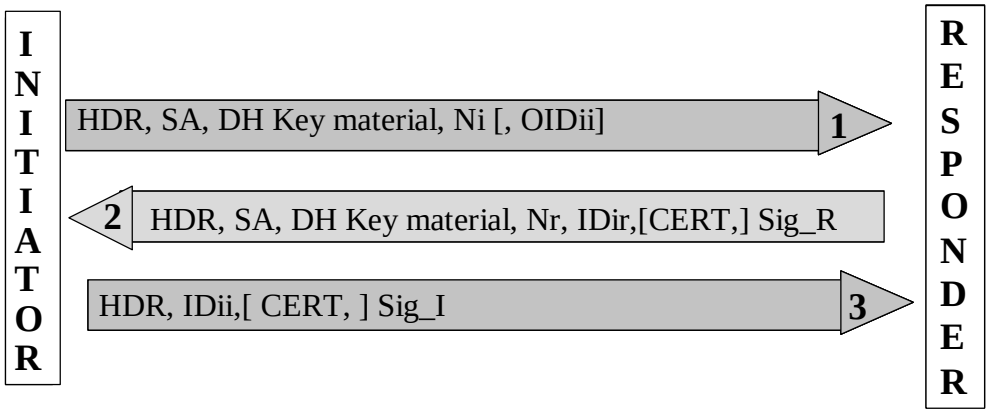


Figure 24. SIGMA exchange with digital signatures

The main characteristics of the SIGMA proposal are listed as follows:

- Main mode and aggressive mode are unified into a single protocol.
- Under normal conditions the SIGMA protocol requires just three messages to complete the establishment of a shared key.
- At the cost of two additional messages the protocol also provides defense against DoS attacks. A responder may choose to apply these mechanisms only in situations where an actual DoS attack on the system is suspected.
- The protocol provides identity protection for both sides as initiator’s and responder's identities are protected with full perfect forward secrecy. In the case of the initiator this is targeted against active attackers and in the case of the responder, only against passive eavesdroppers.
- The protocol enjoys a further privacy property: a party does not sign the peer's identity. Therefore, a regular run of the protocol does not leave a "proof of communication" that can later be used to prove that an exchange between specific peers took place.

- The protocol allows for an optional disclosure of an "outer identity" of the initiator that may serve the responder for making policy decisions at the start of the protocol. This simple mechanism replaces one of the main functionalities of aggressive mode without necessarily disclosing the initiator's full identity.
- In the present description, the SIGMA protocol is used to provide the equivalent of a secured phase 1 SA in IKE; the derivation of keying material for the specific IPsec transforms relies on a Quick Mode execution. This reliance on Quick Mode and its associated overhead should be avoided by piggybacking on SIGMA. The transform negotiation currently done in Quick Mode, which can allow having a working IPsec SA in just three exchange messages.

The major contribution of SIGMA proposal was to secure peer identities and with the re-use of the specification basis of an existing IKEv1, to also allow for the re-use of an existing code that already implements current key exchange protocol. Any user authentication mechanism is left out to external future protocol definitions.

Although the SIGMA proposal is currently out of a narrow choice for a future key exchange mechanism, it has inspired elements in both of the other two proposals which may end up in the eventual combined new key exchange mechanism.

9 Conclusion

The IPsec protocol came a long way to where it is today. An idea of IP security, as "the most important thing missing from the Internet", started from a "birds of a feeder" session at the 23rd IETF meeting in March 1992. By 1995 it had already multiple interoperable tests of the first draft-based implementations, which have resulted in the framework set of RFCs at the end of 1998. Amongst the other protocol proposals for securing the Internet traffic, the IPsec still wins as the least intrusive for the large suite of the existing TCP/IP based applications. Meant to replace leased lines with secure tunnel connection over the Internet, the widest usage of IPsec today is through so called Virtual Private Networks or VPNs. VPNs are enabling private traffic to safely traverse over any shared IP based network infrastructures or the Internet. Through the explosion of the Internet global presence, lowered costs and the increased speed of Internet access, the IPsec based VPNs also gained their popularity and widespread usage. However, this has also brought to light missing elements of the framework developed back in 1998.

One of the first missing parts of the protocol was the lack of a way to leverage existing user authentication infrastructure from the dial technologies, namely RADIUS, TACAS+ or OTP based services, for remote mobile users who wanted to use IPsec based VPN connectivity back to their corporation. The solution to that problem has been under way for a couple of years along the IPsec developments. The L2TP protocol, that by definition could leverage existing dial infrastructure has, however a complete lack of security elements. The combination of the two protocols seemed a natural follow-up, but has unfortunately resulted in a not-so-happy marriage that still required some fixes. At the same time, due to a late development and lack of widespread existence of an L2TP client on most popular desktop operating systems, several other parallel development efforts have resulted in first draft extensions of the IPsec framework to allow user authentication. None of the extensions has been yet accepted as a standard, yet two proposals gained advantage due to their practical usage: ISAKMP extended authentication mode - Xauth and IKE mode config. The reason is that they have addressed the missing elements of a user authentication and a remote peer configuration in IPsec framework by not changing the phases of IKE negotiation but rather inserting an optional middle exchange.

The IPsec was developed as an open set of standards and as such has got a fairly flexible architecture capable of easy adaptation to new developments and security algorithms. Looking it from today's perspective, the IPsec framework has also become a victim of its open development process and flexibility to negotiate too many options. On one hand, flexibility has advantages and was meant to block the development of nonstandard or proprietary extensions. On the other hand, the obvious drawback was the increased complexity of the framework. The complexity, which may easily lead to faulty and non-secure implementations, is in particular present in the key exchange mechanism - IKE. This was one of the main reasons why the IETF decided to take the approach of developing a new key management mechanisms for IPsec that would at the same time maintain flexibility, reduce complexity and remove currently identified unnecessary components or protocol negotiation combinations from it. Several proposals and approaches to resolve the complexity of the existing key exchange mechanism IKEv1 have been reduced to two basic approaches. The first one is to develop a new simpler protocol which will do just fast keying - JFK, while the second one is to reduce the complexity of the existing protocol by cleaning up and removing ambiguity and redundancies from existing ones and to create IKEv2. Unfortunately, none of the approaches yields a complete and ideal solution, so the resulting new key management protocol will most probably have to take the elements of both approaches and combine it into a new protocol.

The further development of the IPsec framework extensions for the purpose of remote access VPNs has already got some new draft proposals, like the IPsec NAT transparency, moving forward. Conversely, the user authentication extensions have been stalled and are waiting for the new key management protocol development results. In the meantime, IKE mode config and Xauth implementations, like the one in the Cisco VPN software client and IOS based Easy VPN are, although not ideal solution, still gaining ground for remote peer configuration and the user authentication based on legacy authentication mechanism. The configuration part is, as already recognized by the authors of the draft, best used for a bootstrapping mechanism and will most probably find its place in the future protocol developments. The protocol couple of combined L2TP with IPsec has also got a stronger standardized shape through the RFC 3193. Together with the maturity of the L2TP client on the desktop operating systems and the support for multicast traffic, it might get an advantage for the remote access VPNs in a foreseeable future.

10 Bibliography

[1] Aboba, B., IPsec-NAT Compatibility Requirements, IETF Internet Draft, <draft-ietf-ipsec-nat-reqts-01.txt>, March 2002.

[2] Aiello, W., Bellovin, S.M., Blaze, M., Canetti, R., Ionnidis, J., Keromytis, A.D., Reingold, O., Just Fast Keying (JFK) , IETF Internet Draft, <draft-ietf-ipsec-jfk-03.txt>, April 2002.

[3] Ball A. D., Standards Updates: L2TP Evolving, Cisco Systems Inc., Packet magazine, Vol. 14, No.1; p87, January 2002.

[4] Beaulieu S., Pereira R., Extended Authentication within IKE (XAUTH), IETF Internet Draft, <draft-beaulieu-ike-xauth-02.txt>, October 2001.

[5] Brown R., VPNs Made Easy, Cisco Systems Inc., Packet magazine, Vol. 14, No.2; p73-75, April 2002.

[6] Carlton R. Davis, IPSec: Securing VPNs, McGraw-Hill 2001.

[7] Dixon, W., Swander, B., Kivinen, T., Stenberg, M., Volpe, V., DiBurro, L., UDP Encapsulation of IPsec Packets, IETF Internet Draft, < draft-ietf-ipsec-udp-encaps-01.txt >, October 2001.

[8] Doraswamy N., Harkins D., IPSec The New Security Standard for the Internet, Intranets and Virtual Private Networks, Prentice Hall PTR 1999.

[9] Dukes D., Pereira R., The ISAKMP Configuration Method, IETF Internet Draft, <draft-dukes-ike-mode-cfg-02.txt>, September 2001.

[10] Ferguson, N., Schneier, B., A Cryptographic Evaluation of IPSec, <www.counterpane.com/ipsec.html>, Apr. 1999.

[11] Frankel, S., Demystifying the IPsec Puzzle, Artech House Inc., 2001.

[12] Harkins D., Carrel D., The Internet Key Exchange (IKE), RFC 2409, November 1998.

[13] Harkins, D., Kaufman, C., Kent, S., Kivinen, T., Perlman, R., Proposal for the IKEv2 Protocol , IETF Internet Draft, <draft-ietf-ipsec-ikev2-02.txt>, April 2002.

[14] Harkins, D., Kaufman, C., Kent, S., Kivinen, T., Perlman, R., Design Rationale for IKEv2, IETF Internet Draft, <draft-ietf-ipsec-ikev2-rationale-00.txt>, February 2002.

[15] Huang G., Beaulieu, S., Rochefort, D., A Traffic-Based Method of Detecting Dead IKE Peers, IETF Internet Draft, <draft-ietf-ipsec-dpd-00.txt>, July 2001.

[16] Hoffman, P., Features of Proposed Successors to IKE, IETF Internet Draft, <draft-ietf-ipsec-soi-features-00.txt>, April 2002.

[17] Kelly S., Comparing Proposed Solutions for IPsec Remote Access Legacy User Authentication, IETF Internet Draft, <draft-kelly-ipsra-eval-00.txt>, July 2001.

[18] Kelly S., Ramamoorthi S., Requirements for IPsec Remote Access Scenarios, IETF Internet Draft, <draft-ietf-ipsra-reqmts-05.txt>, March 2002.

[19] Kent, S., Atkinson R., Security Architecture for the Internet Protocol, RFC 2401, November 1998.

[20] Kent, S., Atkinson R., IP Authentication Header, RFC 2402, November 1998.

[21] Kent, S., Atkinson R., IP Encapsulating Security Payload (ESP), RFC 2406, November 1998.

[22] Kosiur D., Building and Managing Virtual Private Networks, Willey Computer Publishing 1998.

[23] Krawczyk, H., The IKE-SIGMA Protocol, IETF Internet Draft, <draft-krawczyk-ipsec-ike-sigma-00.txt>, November 2001.

[24] Madson C., Son-of-IKE Requirements, IETF Internet Draft, <draft-ietf-ipsec-sonofike-rqts-00.txt>, March 2002.

[25] Maughan D., Schertler M., Schneider M., Turner J., Internet Security Association and Key Management Protocol (ISAKMP), RFC 2408, November 1998.

[26] Meredith, G., Tunnel Vision: How to Build Remote-Access Virtual Private Networks, Cisco Systems Inc., Packet magazine, Vol. 12, No.4; p68-73, October 2000.

[27] Patel, B., Aboba, B., Zorn, G., Booth, S., Securing L2TP using IPsec, IETF RFC 3193, November 2001.

[28] Perlman, R., Kaufman, C., Key Exchange in IPSec: Analysis of IKE, IEEE Internet Computing Vol. 4, No.6; p50-56, November-December 2000.

[29] Perlman, R., Kaufman, C., Analysis of the IPsec key exchange Standard, WET-ICE Security Conference, MIT, <sec.femto.org/wetice-2001/papers/radia-paper.pdf>, 2001.

[30] Piper, D., The Internet IP Security Domain of Interpretation of ISAKMP, RFC 2407, November 1998.

[31] Sheffer, Y., Krawczyk, H., Aboba, B., PIC, A Pre-IKE Credential Provisioning Protocol, IETF Internet Draft, <draft-ietf-ipsra-pic-05.txt>, February 2002.

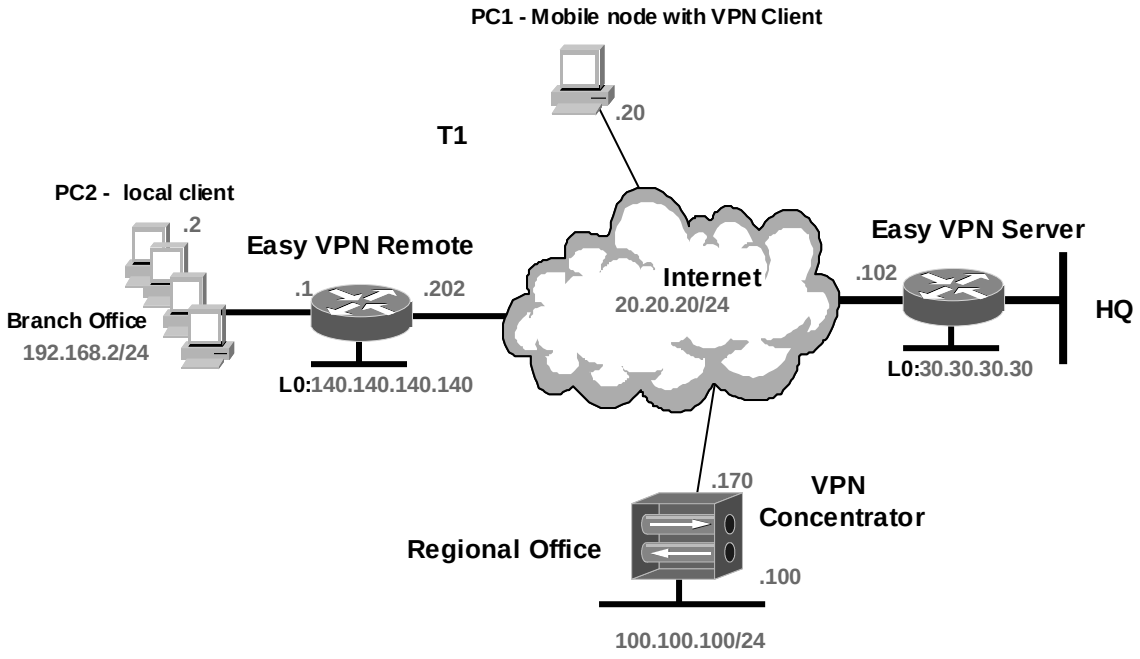
[32] Tiller, S.J., A Technical Guide to IPSec Virtual Private Networks, Auerbach Publications, 2001.

[33] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., and B. Palter, Layer Two Tunneling Protocol L2TP, RFC 2661, August 1999.

Appendix A - Remote Access IPsec extensions demonstration setup

A.1 Demonstration test bed description

Network diagram:



Description:

The demonstration test bed setup consists of a branch office, a regional office, a central office and the remote mobile client - all connected to each other via a simulated Internet connection. PC1 represents remote mobile user running Microsoft Windows operating system and a Cisco VPN client. Branch office and headquarter VPN gateways are Cisco routers running Easy VPN Remote and Easy VPN Server IOS software, while the regional office has VPN Concentrator as a VPN getaway. Two VPN connection scenarios that are demonstrated are PC1 via using the VPN client software connecting to a branch office VPN gateway and PC2 (without VPN stack) connecting via branch office router running Easy VPN Remote to a headquarter VPN gateway in two modes: Client mode and Network Extension mode. All VPN connections are described together with pre and post parameters as well as debug and screen logging outputs.

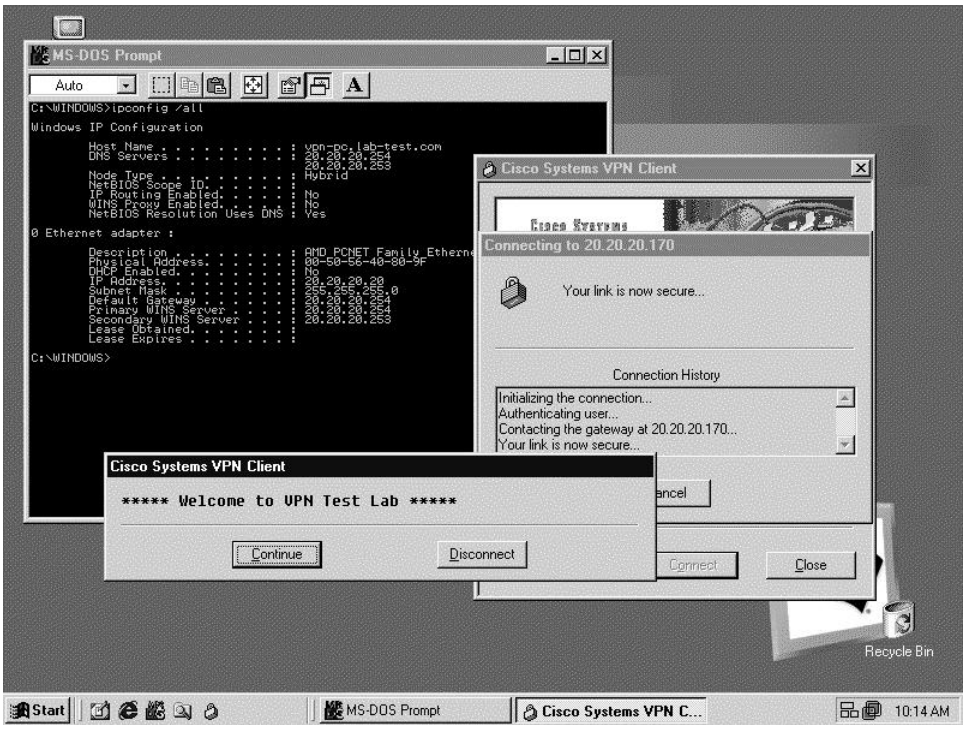
A.2 Demonstration setup with VPN software client

The PC1 mobile node with VPN software client running on Microsoft Windows operating system is pre-configured with the following parameters:

IKE device identity (group name): vpn_test_group
IKE pre-shred key (group password): cisco
VPN GW IP address: 20.20.20.170

C:\ >ipconfig /all
Windows IP Configuration
Host Name . . . . . : **vpn-pc.lab-test.com**
DNS Servers . . . . . : **20.20.20.254, 20.20.20.253**
Description . . . . . : AMD PCNET Family Ethernet Adapter
Physical Address. . . . . : 00-50-56-40-80-9F
IP Address. . . . . : 20.20.20.20
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 20.20.20.254
Primary WINS Server . . . . : **20.20.20.254**
Secondary WINS Server . . . : **20.20.20.253**

Client PC1 screen immediately after connection shows in an MS-DOS window still locally assigned parameters for WINS, DNS and domain name. In the VPN client window we can see the IKE mode config pushed banner “Welcome to VPN Test Lab”.



From connect session log on VPN client we can see connection establishment of IKE phase I in Aggressive Mode (AG) using pre-shared keys.

Attempting to establish a connection with 20.20.20.170.
SENDING >>> ISAKMP OAK AG (SA, KE, NON, ID, VID, VID, VID) to 20.20.20.170
Received ISAKMP packet: peer = 20.20.20.170
RECEIVING <<< ISAKMP OAK AG (SA, KE, NON, ID, HASH, VID, VID, VID, VID)
from 20.20.20.170

Both peers support DPD protocol:

Vendor ID payload = Peer is a Cisco-Unity compliant peer
Vendor ID payload = Peer supports DPD

Here follows the IKE mode config with an IP address, mask, primary and secondary DNS and WINS server addresses parameters push:

RECEIVING <<< ISAKMP OAK TRANS *(HASH, ATTR) from 20.20.20.170
MODE_CFG_REPLY: Attribute = INTERNAL_IPV4_ADDRESS: , value = **100.100.100.101**
MODE_CFG_REPLY: Attribute = INTERNAL_IPV4_NETMASK , value = **255.255.255.0**
MODE_CFG_REPLY: Attribute = INTERNAL_IPV4_DNS(1): , value = **111.111.111.111**
MODE_CFG_REPLY: Attribute = INTERNAL_IPV4_DNS(2): , value = **111.111.111.112**
MODE_CFG_REPLY: Attribute = INTERNAL_IPV4_NBNS(1) (a.k.a. WINS) : , value = **222.222.222.222**
MODE_CFG_REPLY: Attribute = INTERNAL_IPV4_NBNS(2) (a.k.a. WINS): , value = **222.222.222.223**

On the side of a domain name there are additional vendor specific attribute parameters pushed to client like banner, option to locally save password on VPN client and application version:

MODE_CFG_REPLY: Attribute = MODECFG_UNITY_BANNER, value =
******* Welcome to VPN Test Lab *******
MODE_CFG_REPLY: Attribute = MODECFG_UNITY_SAVEPWD: , value = 0x00000000
MODE_CFG_REPLY: Attribute = MODECFG_UNITY_DEFDOMAIN: , value = **vpn-test.com**
MODE_CFG_REPLY: Attribute = MODECFG_UNITY_PFS: , value = 0x00000000
MODE_CFG_REPLY: Attribute = APPLICATION_VERSION,
value = **Cisco Systems, Inc./VPN 3000 Concentrator**
Version 3.5.2.Rel built by vmurphy on Feb 14 2002 14:39:43

Follow up output shows the IKE Quick mode (QM) or phase II key derivation and IPsec SAs establishment with keys (ESP encryption and ESP authentication in each direction) and their lifetime:

Received a key request from Driver for IP address 20.20.20.170, GW IP = 20.20.20.170
SENDING >>> ISAKMP OAK QM *(HASH, SA, NON, ID, ID) to 20.20.20.170
Received a key request from Driver for IP address 10.10.10.255, GW IP = 20.20.20.170
SENDING >>> ISAKMP OAK QM *(HASH, SA, NON, ID, ID) to 20.20.20.170
Received ISAKMP packet: peer = 20.20.20.170

RECEIVING << ISAKMP OAK INFO *(HASH, NOTIFY:STATUS_RESP_LIFETIME) from 20.20.20.170

RESPONDER-LIFETIME notify has value of 86400 seconds

This SA has already been alive for 7 seconds, setting expiry to 86393 seconds from now

Received ISAKMP packet: peer = 20.20.20.170

RECEIVING <<< ISAKMP OAK QM *(HASH, SA, NON, ID, ID, NOTIFY:STATUS_RESP_LIFETIME) from 20.20.20.170

RESPONDER-LIFETIME notify has value of 28800 seconds

SENDING >>> ISAKMP OAK QM *(HASH) to 20.20.20.170

Loading IPsec SA (Message ID = 0xF8E1A29B OUTBOUND SPI = 0x43CD1F41 INBOUND SPI = 0x46FD2FE1)

Loaded OUTBOUND ESP SPI: 0x43CD1F41

Loaded INBOUND ESP SPI: 0x46FD2FE1

Received ISAKMP packet: peer = 20.20.20.170

RECEIVING <<< ISAKMP OAK QM *(HASH, SA, NON, ID, ID, NOTIFY:STATUS_RESP_LIFETIME) from 20.20.20.170

RESPONDER-LIFETIME notify has value of 28800 seconds

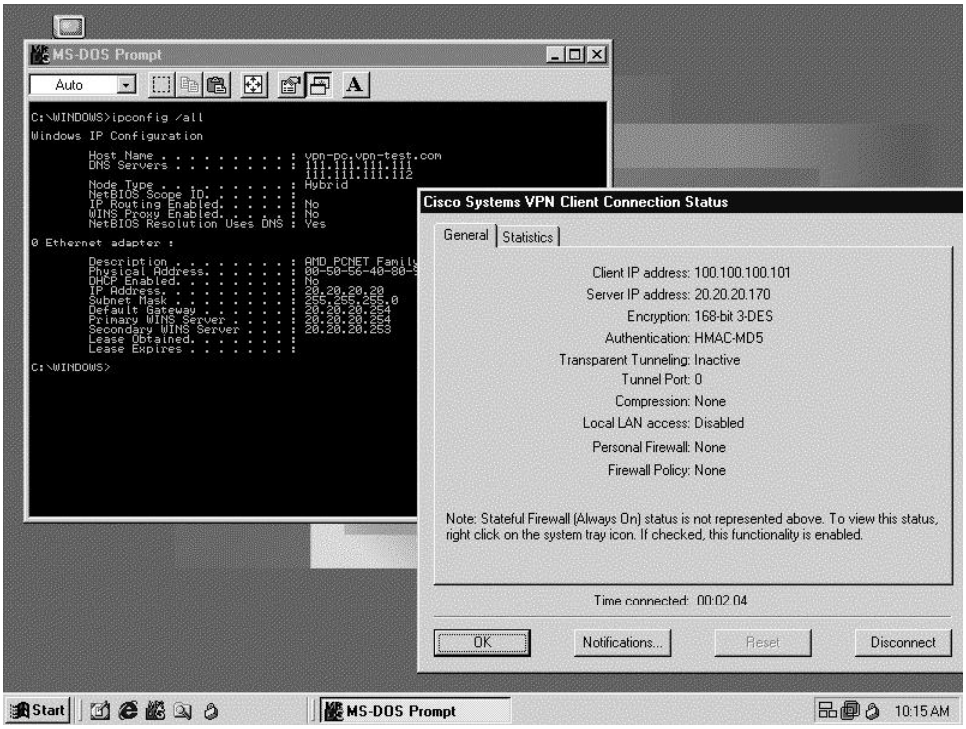
SENDING >>> ISAKMP OAK QM *(HASH) to 20.20.20.170

Loading IPsec SA (Message ID = 0xF14E37FD OUTBOUND SPI = 0x43003D52 INBOUND SPI = 0x344A08C8)

Loaded OUTBOUND ESP SPI: 0x43003D52

Loaded INBOUND ESP SPI: 0x344A08C8

On the client PC1 screen after a VPN connection establishment we can see in MS-DOS Prompt window, new IKE mode config assigned parameters for domain name, DNS and WINS servers, while the IP address shown represents the real tunnel end-point address. On the VPN client screen we can see the new inner IP address (100.100.100.101) assigned via IKE mode config as well as ESP-3DES encryption and ESP-HMAC-MD5 as tunnel authentication algorithms:



After the established VPN connection, we can see the new IKE mode config assigned parameters for a new domain name DNS and WINS servers also on the client PC1 screen via an issuing command "*ipconfig /all"* in the MS-DOS Prompt window:

```
C:\>ipconfig /all
Windows IP Configuration
        Host Name . . . . . . . . . . : vpn-pc.vpn-test.com
        DNS Servers . . . . . . . . . : 111.111.111.111, 111.111.111.112
        IP Routing Enabled. . . . . : No
        WINS Proxy Enabled. . . . . : No
        NetBIOS Resolution Uses DNS : Yes
        Description . . . . . . . . . : AMD PCNET Family Ethernet Adapter
        Physical Address. . . . . . : 00-50-56-40-80-9F
        DHCP Enabled. . . . . . . . . : No
        IP Address. . . . . . . . . . : 20.20.20.20
        Subnet Mask . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . : 20.20.20.254
        Primary WINS Server . . . . : 222.222.222.222
        Secondary WINS Server . . . : 222.222.222.223
```

A.3 Demonstration setup with IOS based VPN router

A.3.1 Easy VPN Remote router configuration

VPN-remote> show version

Cisco Internetwork Operating System Software
IOS (tm) C806 Software (C806-K9OSY6-M), Experimental Version 12.2(20020508:041726) [albra-BL4A_ezvpn 101]
Copyright (c) 1986-2002 by cisco Systems, Inc.
Compiled Wed 08-May-02 00:51 by albra
Image text-base: 0x80013170, data-base: 0x8082D210
ROM: System Bootstrap, Version 12.2(1r)XE2, RELEASE SOFTWARE (fc1)
VPN-remote uptime is 5 hours, 21 minutes
System returned to ROM by reload
System image file is "flash:c806_BL4A"
CISCO C806 (MPC855T) processor (revision 0x202) with 18432K/2048K bytes of memory.
Processor board ID JAD05260E53 (3723046521), with hardware revision 0000
CPU rev number 5
Bridging software.
2 Ethernet/IEEE 802.3 interface(s)
128K bytes of non-volatile configuration memory.
8192K bytes of processor board System flash (Read/Write)
2048K bytes of processor board Web flash (Read/Write)

VPN-remote> show running-config

```
Current configuration : 1282 bytes
!
version 12.2
!
hostname VPN-remote
!
enable secret 5 $1$3DpP$uR47iaey3s8r8jFUdOFFy.
!
ip subnet-zero
no ip domain-lookup
ip dhcp excluded-address 192.168.2.1
!
ip dhcp pool CLIENT
   import all
   network 192.168.2.0 255.255.255.0
   default-router 192.168.2.1
!
crypto ipsec client ezvpn default
```

```
 connect manual
 group ezvpn_client_group key cisco
 local-address Loopback0
 mode client
 peer 20.20.20.102
!
interface Loopback0
 ip address 140.140.140.1 255.255.255.0
 crypto ipsec client ezvpn default inside
!
interface Ethernet0
 ip address 192.168.2.1 255.255.255.0
 crypto ipsec client ezvpn default inside
 hold-queue 100 out
!
interface Ethernet1
 ip address 20.20.20.202 255.255.255.0
 crypto ipsec client ezvpn default
!
ip classless
ip route 0.0.0.0 0.0.0.0 Ethernet1
ip http server
!
banner motd ^C**********************************
*           Easy VPN Remote
**********************************^C
!
line con 0
 exec-timeout 0 0
 stopbits 1
line vty 0 4
 exec-timeout 120 0
 password cisco
 login local
 length 0
!
scheduler max-task-time 5000
end
```

A.3.2 Easy VPN Server router configuration

VPN-server> show version

Cisco Internetwork Operating System Software
IOS (tm) C806 Software (C806-K9OSY6-M), Version 12.2(8)T1, RELEASE SOFTWARE (fc2)
TAC Support: http://www.cisco.com/tac
Copyright (c) 1986-2002 by cisco Systems, Inc.
Compiled Sun 31-Mar-02 09:40 by ccai
Image text-base: 0x80013170, data-base: 0x8081D8CC
ROM: System Bootstrap, Version 12.2(1r)XE2, RELEASE SOFTWARE (fc1)
VPN-server uptime is 21 hours, 55 minutes
System returned to ROM by reload
System image file is "flash:c806-k9osy6-mz.122-8.T1.bin"
CISCO C806 (MPC855T) processor (revision 0x202) with 18432K/2048K bytes of memory.
Processor board ID JAD05250HTU (1172799700), with hardware revision 0000
CPU rev number 5
Bridging software.
2 Ethernet/IEEE 802.3 interface(s)
128K bytes of non-volatile configuration memory.
8192K bytes of processor board System flash (Read/Write)
2048K bytes of processor board Web flash (Read/Write)
Configuration register is 0x2102

VPN-server> show running-config

```
version 12.2
no service pad
no service timestamps debug uptime
no service timestamps log uptime
no service password-encryption
!
hostname VPN-server
!
aaa new-model
!
aaa authentication login test_list local
aaa authorization network ezvpn_client_group local
aaa session-id common
enable secret 5 $1$3DpP$uR47iaey3s8r8jFUdOFFy.
!
username test password 0 test
ip subnet-zero
no ip domain-lookup
!
crypto isakmp policy 1
 encr 3des
```

```
 authentication pre-share
 group 2
crypto isakmp client configuration address-pool local vpn_pool
crypto isakmp xauth timeout 60
!
crypto isakmp client configuration group ezvpn_client_group
 key cisco
 dns 111.111.111.111 111.111.111.112
 wins 222.222.222.222 222.222.222.223
 domain vpn-test.com
 pool vpn_pool
 acl 150
!
crypto ipsec transform-set transform_ezvpn esp-3des esp-sha-hmac
!
crypto dynamic-map ezvpn_map 1
 set transform-set transform_ezvpn
 reverse-route
!
crypto map ezvpn_map client authentication list test_list
crypto map ezvpn_map isakmp authorization list ezvpn_client_group
crypto map ezvpn_map client configuration address respond
crypto map ezvpn_map 1 ipsec-isakmp dynamic ezvpn_map
!
interface Loopback0
 ip address 30.30.30.30 255.255.255.255
!
interface Ethernet0
 ip address 192.168.1.1 255.255.255.0
 shutdown
 hold-queue 100 out
!
interface Ethernet1
 ip address 20.20.20.102 255.255.255.0
 crypto map ezvpn_map
!
ip local pool vpn_pool 192.168.1.2 192.168.1.10
ip classless
ip route 0.0.0.0 0.0.0.0 Ethernet1
ip http server
!
access-list 150 permit ip 30.30.0.0 0.0.255.255 any
banner motd ^C**********************************
*                  Easy VPN Server
**********************************^C
!
```

```
line con 0
 exec-timeout 0 0
 stopbits 1
line vty 0 4
 exec-timeout 0 0
 password cisco
!
end
```

A.3.3 Sample operational and debugging outputs

PC2 behind router before VPN connection:

C:\>ipconfig /all
Windows 2000 IP Configuration
Host Name . . . . . : secu-win2k
Primary DNS Suffix . . . . . :
Ethernet adapter LAB 3rd Interface:
Description . . . . . : 3Com EtherLink XL 10/100 PCI (3C905C-TX) #2
Physical Address. . . . . : 00-04-76-DE-C2-05
IP Address. . . . . : **192.168.2.2**
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.2.1
DHCP Server . . . . . : 192.168.2.1
DNS Servers . . . . . :
Lease Obtained. . . . . : Thursday, May 09, 2002 5:51:15 PM
Lease Expires . . . . . : Friday, May 10, 2002 5:51:15 PM

Easy VPN Remote connection operation in Client mode

VPN-remote> show ip dhcp import
Address Pool Name: CLIENT

VPN-remote> show crypto ipsec client ezvpn
Tunnel name : default
Inside interface list: Ethernet0, Loopback0,
Outside interface: Ethernet1
Current State: CONNECT_REQUIRED
Last Event: TUNNEL_INTERFACE_UP

VPN-remote> show ip nat statistics
VPN-remote> show ip nat translations

Both commands give an empty output at the beginning, as there is no dynamic NAT configuration yet.

VPN-remote> debug crypto ipsec client ezvpn
VPN-remote> crypto ipsec client ezvpn connect default

EZVPN(default): Current State: CONNECT_REQUIRED
EZVPN(default): Event: CONNECT
EZVPN(default): ezvpn_connect_request
EZVPN(default): Event: XAUTH_REQUEST
EZVPN(default): ezvpn_xauth_request
EZVPN(default): ezvpn_parse_xauth_msg

EZVPN: Attributes sent in xauth request message:
XAUTH_TYPE_V2(default): 0
XAUTH_MESSAGE_V2(default) <Username: >
XAUTH_USER_NAME_V2(default):
XAUTH_USER_PASSWORD_V2(default):
EZVPN(default): New State: XAUTH_REQ
EZVPN(default): Pending XAuth Request, Please enter the following command:
EZVPN: crypto ipsec client ezvpn xauth

VPN-remote> crypto ipsec client ezvpn xauth default

Username: : test
EZVPN(default): Current State: XAUTH_REQ
EZVPN(default): Event: XAUTH_PROMPTING
EZVPN(default): New State: XAUTH_PROMPT
Password: : test
EZVPN(default): Current State: XAUTH_PROMPT
EZVPN(default): Event: XAUTH_REQ_INFO_READY
EZVPN(default): ezvpn_xauth_reply
XAUTH_TYPE_V2(default): 0
XAUTH_USER_NAME_V2(default): test
XAUTH_USER_PASSWORD_V2(default): <omitted>
…
EZVPN(default): Event: MODE_CONFIG_REPLY
EZVPN(default): ezvpn_mode_config
EZVPN(default): ezvpn_parse_mode_config_msg
EZVPN: Attributes sent in message:
Address: **192.168.1.9**
DNS Primary: **111.111.111.111**
DNS Secondary: **111.111.111.112**
NBMS/WINS Primary: **222.222.222.222**
NBMS/WINS Secondary: **222.222.222.223**
Split Tunnel List: 1
Address : 30.30.0.0
Mask : 255.255.0.0
Protocol : 0x0
Source Port: 0
Dest Port : 0
Default Domain: vpn-test.com
EZVPN(default): Event: SOCKET_UP
EZVPN(default): New State: IPSEC_ACTIVE

VPN-remote> show crypto ipsec client ezvpn

Tunnel name : default
Inside interface list: Ethernet0, Loopback0,
Outside interface: Ethernet1
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
Address: **192.168.1.9**
Mask: **255.255.255.255**
DNS Primary: **111.111.111.111**
DNS Secondary: **111.111.111.112**
NBMS/WINS Primary: **222.222.222.222**
NBMS/WINS Secondary: **222.222.222.223**
Default Domain: **vpn-test.com**
Split Tunnel List: 1
Address : **30.30.0.0**
Mask : **255.255.0.0**
Protocol : 0x0
Source Port: 0
Dest Port : 0

VPN-remote> show ip dhcp import

Address Pool Name: CLIENT
Domain Name Server(s): **111.111.111.111 111.111.111.112**
NetBIOS Name Server(s): **222.222.222.222 222.222.222.223**
Domain Name Option: **vpn-test.com**

PC2 after VPN tunnel establishment:

C:\>ipconfig /all
Windows 2000 IP Configuration
Host Name . . . . . : secu-win2k
DNS Suffix Search List. . . . . : **vpn-test.com**
Ethernet adapter LAB 3rd Interface:
Connection-specific DNS Suffix . : **vpn-test.com**
IP Address. . . . . : 192.168.2.2
Subnet Mask . . . . . : **255.255.255.0**
Default Gateway . . . . . : **192.168.2.1**
DHCP Server . . . . . : **192.168.2.1**
DNS Servers . . . . . : **111.111.111.111, 111.111.111.112**
Primary WINS Server . . . . . : **222.222.222.222**
Secondary WINS Server . . . . . : **222.222.222.223**
Lease Obtained. . . . . : Thursday, May 09, 2002 9:11:38 PM
Lease Expires . . . . . : Friday, May 10, 2002 9:11:38 PM

VPN-remote> show ip nat statistics

Total active translations: 1 (0 static, 1 dynamic; 1 extended)
Outside interfaces:
  Ethernet1
Inside interfaces:
  Ethernet0, Loopback0
Hits: 37 Misses: 2
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 2] access-list 197 interface Loopback0 refcount 0
[Id: 1] access-list 198 pool default refcount 1
 pool default: netmask 255.255.255.0
        start **192.168.1.9** end **192.168.1.9**
        type generic, total addresses 1, allocated 1 (100%), misses 0

VPN-remote> show ip nat translations

Pro	Inside global	Inside local	Outside local	Outside global
tcp	**192.168.1.9**:1261	**192.168.2.2**:1261	30.30.30.30:23	30.30.30.30:23

VPN tunnel has been established and there are already 20 packets encrypted and 16 decrypted:

VPN-remote> show crypto engine connections active

ID	Interface	IP-Address		State	Algorithm	Encrypt	Decrypt
2	Ethernet1	20.20.20.202	set	HMAC_SHA+3DES_56_C		0	0
2000	Ethernet1	20.20.20.202	set	HMAC_SHA+3DES_56_C		0	**16**
2001	Ethernet1	20.20.20.202	set	HMAC_SHA+3DES_56_C		**20**	0

Output on Easy VPN Server:

debug crypto isakmp

IKE session start - UDP packet on port 500:

```
ISAKMP (0:0): received packet from 140.140.140.1 (N) NEW SA
ISAKMP: local port 500, remote port 500
…
```

Recognized support for DPD and Xauth:

```
ISAKMP (0:2): vendor ID is DPD
ISAKMP (0:2): processing vendor id payload
ISAKMP (0:2): vendor ID seems Unity/DPD but bad major
ISAKMP (0:2): vendor ID is XAUTH
ISAKMP (0:2): processing vendor id payload
ISAKMP (0:2): claimed IOS but failed authentication
ISAKMP (0:2): processing vendor id payload
ISAKMP (0:2): vendor ID is Unity
```

Sending proposal SAs for all combinations of encryption and authentication algorithms (DES/3DES, SHA/MD5 - abbreviated here for clarity) until they match:

```
ISAKMP (0:2): Checking ISAKMP transform 1 against priority 1 policy
ISAKMP:      encryption DES-CBC
ISAKMP:      hash SHA
ISAKMP:      default group 2
ISAKMP:      auth XAUTHInitPreShared
ISAKMP:      life type in seconds
ISAKMP:      life duration (VPI) of  0x0 0x20 0xC4 0x9B
ISAKMP (0:2): Encryption algorithm offered does not match policy!
ISAKMP (0:2): atts are not acceptable. Next payload is 3
…
ISAKMP:      encryption DES-CBC
ISAKMP:      hash MD5
ISAKMP (0:2): atts are not acceptable. Next payload is 3
…
ISAKMP:      encryption DES-CBC
ISAKMP:      hash SHA
ISAKMP (0:2): atts are not acceptable. Next payload is 3
…
ISAKMP:      encryption DES-CBC
ISAKMP:      hash MD5
ISAKMP (0:2): atts are not acceptable. Next payload is 3
…
```

```
ISAKMP (0:2): Checking ISAKMP transform 5 against priority 1 policy
ISAKMP:      encryption 3DES-CBC
ISAKMP:      hash SHA
ISAKMP:      default group 2
ISAKMP:      auth XAUTHInitPreShared
ISAKMP:      life type in seconds
ISAKMP:      life duration (VPI) of  0x0 0x20 0xC4 0x9B
ISAKMP (0:2): atts are acceptable. Next payload is 3
ISAKMP (0:2): processing KE payload. message ID = 0
ISAKMP (0:2): processing NONCE payload. message ID = 0
…
```

Recognizing that the peer wants to do the IKE phase I in Aggressive Mode with a pre-shared key and Xauth user authentication after phase I:

```
ISAKMP (0:2): Input = IKE_MESG_FROM_PEER, IKE_AM_EXCH
ISAKMP (0:2): SA is doing pre-shared key authentication plus XAUTH using id type ID_IPV4_ADDR
ISAKMP (2): ID payload
        next-payload : 10
        type         : 1
        protocol     : 17
        port         : 500
        length       : 8
ISAKMP (2): Total payload length: 12
ISAKMP (0:2): sending packet to 140.140.140.1 (R) AG_INIT_EXCH
ISAKMP (0:2): Input = IKE_MESG_FROM_AAA, PRESHARED_KEY_REPLY
ISAKMP (0:2): received packet from 140.140.140.1 (R) AG_INIT_EXCH
ISAKMP (0:2): processing HASH payload. message ID = 0
ISAKMP (0:2): SA has been authenticated with 140.140.140.1
ISAKMP (0:2): sending packet to 140.140.140.1 (R) QM_IDLE
ISAKMP (0:2): purging node 199511092
ISAKMP: Sending phase 1 responder lifetime 86400
```

Last IKE message of phase I, Aggressive Mode authentication with pre-shared keys is completed:

```
ISAKMP (0:2): Input = IKE_MESG_FROM_PEER, IKE_AM_EXCH
Old State = IKE_R_AM2  New State = IKE_P1_COMPLETE
```

Beginning of phase 1-1/2 doing request for Xauth and IKE mode config with address assignments:

```
ISAKMP (0:2): Need XAUTH
ISAKMP (0:2): Input = IKE_MESG_INTERNAL, IKE_PHASE1_COMPLETE
ISAKMP (0:2): received packet from 140.140.140.1 (R) CONF_XAUTH
```

```
ISAKMP/xauth: request attribute XAUTH_TYPE_V2
ISAKMP/xauth: request attribute XAUTH_MESSAGE_V2
ISAKMP/xauth: request attribute XAUTH_USER_NAME_V2
ISAKMP/xauth: request attribute XAUTH_USER_PASSWORD_V2
ISAKMP (0:2): initiating peer config to 140.140.140.1. ID = -1994158728
ISAKMP (0:2): sending packet to 140.140.140.1 (R) CONF_XAUTH
ISAKMP (0:2): Input = IKE_MESG_FROM_AAA, IKE_AAA_START_LOGIN
…
ISAKMP (0:2): received packet from 140.140.140.1 (R) CONF_XAUTH
ISAKMP (0:2): processing transaction payload from 140.140.140.1. message ID = -1994158728
ISAKMP: Config payload REPLY
```

Received Xauth response for username and password attributes and finishing Xauth negotiation:

```
ISAKMP/xauth: reply attribute XAUTH_TYPE_V2 unexpected
ISAKMP/xauth: reply attribute XAUTH_USER_NAME_V2
ISAKMP/xauth: reply attribute XAUTH_USER_PASSWORD_V2
…
ISAKMP (0:2): received packet from 140.140.140.1 (R) CONF_XAUTH
ISAKMP (0:2): processing transaction payload from 140.140.140.1. message ID = -1475077831
ISAKMP: Config payload ACK
ISAKMP (0:2):        XAUTH ACK Processed
ISAKMP (0:2): Input = IKE_MESG_FROM_PEER, IKE_CFG_ACK
```

IKE Xauth authentication has been successfully passed and is followed by the IKE mode config negotiation:

```
ISAKMP: Config payload REQUEST
ISAKMP (0:2): checking request:
ISAKMP:    IP4_ADDRESS
ISAKMP:    IP4_NETMASK
ISAKMP:    IP4_DNS
ISAKMP:    IP4_DNS
ISAKMP:    IP4_NBNS
ISAKMP:    IP4_NBNS
ISAKMP:    SPLIT_INCLUDE
ISAKMP:    DEFAULT_DOMAIN
…
```

Sending pre-configured parameters via IKE mode config: IP address (192.168.1.9) allocated from the local pool as well as mask, DNS, WINS (NBNS) servers and split tunneling information configured in the access list 150:

```
ISAKMP (0:2): allocating address 192.168.1.9
ISAKMP: Sending private address: 192.168.1.9
ISAKMP: Unknown Attr: IP4_NETMASK (0x2)
```

ISAKMP: Sending IP4_DNS server address: **111.111.111.111**
ISAKMP: Sending IP4_DNS server address: **111.111.111.112**
ISAKMP: Sending IP4_NBNS server address: **222.222.222.222**
ISAKMP: Sending IP4_NBNS server address: **222.222.222.223**
ISAKMP: **Sending split include name 150 network 30.30.0.0 mask 255.255.0.0 protocol 0, src port 0, dst port 0**
ISAKMP: Sending DEFAULT_DOMAIN default domain name: **vpn-test.com**

IKE mode config (phase 1-1/2) is finished and IKE Quick mode may begin to negotiate IPsec SA's:

ISAKMP (0:2): received packet from 140.140.140.1 (R) QM_IDLE
ISAKMP (0:2): **Checking IPSec proposal 1**
ISAKMP: transform 1, ESP_3DES
ISAKMP: attributes in transform:
ISAKMP: encaps is 1
ISAKMP: SA life type in seconds
ISAKMP: SA life duration (VPI) of 0x0 0x20 0xC4 0x9B
ISAKMP: SA life type in kilobytes
ISAKMP: SA life duration (VPI) of 0x0 0x46 0x50 0x0
ISAKMP: authenticator is HMAC-SHA
ISAKMP (0:2): atts are acceptable.
ISAKMP (0:2): processing NONCE payload. message ID = -2045312456
ISAKMP (0:2): asking for 1 spis from ipsec
ISAKMP (0:2): Node -2045312456, Input = IKE_MESG_FROM_PEER, IKE_QM_EXCH
…

Negotiated IPsec SA's has been established:

ISAKMP (0:2): received packet from 140.140.140.1 (R) QM_IDLE
ISAKMP (0:2): **Creating IPSec SAs**
inbound SA from 140.140.140.1 to 20.20.20.102
(proxy 192.168.1.9 to 30.30.0.0)
has spi 0xD1B4E7BF and conn_id 2000 and flags 4
lifetime of 2147483 seconds
lifetime of 4608000 kilobytes
outbound SA from 20.20.20.102 to 140.140.140.1 (proxy 30.30.0.0 to 192.168.1.9)
has spi 1834232203 and conn_id 2001 and flags C
lifetime of 2147483 seconds
lifetime of 4608000 kilobytes

ISAKMP (0:2): Node -2045312456, Input = IKE_MESG_FROM_PEER, IKE_QM_EXCH
Old State = IKE_QM_R_QM2 New State = **IKE_QM_PHASE2_COMPLETE**

Quick Mode (QM) IKE phase II has been completed and is ready to be used to encrypt the traffic.

The routing table of the VPN gateway has got an additional static route to a dynamically assigned IP address of the VPN client (192.168.1.9) and will proxy all packets forwarded to this address:

VPN-server> show ip route

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

     20.0.0.0/24 is subnetted, 1 subnets
C       20.20.20.0 is directly connected, Ethernet1
     192.168.1.0/32 is subnetted, 1 subnets
S       192.168.1.9 [1/0] via 0.0.0.0, Ethernet1
     30.0.0.0/32 is subnetted, 1 subnets
C       30.30.30.30 is directly connected, Loopback0
S*   0.0.0.0/0 is directly connected, Ethernet1
```

VPN-server> show crypto engine connections active

ID	Interface	IP-Address	State	Algorithm	Encrypt	Decrypt
2	Ethernet1	20.20.20.102	set	HMAC_SHA+3DES_56_C	0	0
2000	Ethernet1	20.20.20.102	set	HMAC_SHA+3DES_56_C	0	**20**
2001	Ethernet1	20.20.20.102	set	HMAC_SHA+3DES_56_C	**16**	0

VPN tunnel has been established and there are 16 packets encrypted and 20 decrypted, which is just opposite as we have seen on the VPN-remote side.

Easy VPN Remote in Network Extension mode

Only modification in configuration of Easy VPN Remote is in mode command:

VPN-remote>

```
crypto ipsec client ezvpn default
 connect manual
 group ezvpn_client_group key cisco
 local-address Ethernet0
 mode network-extension
 peer 20.20.20.102
```

After VPN tunnel establishment from PC2:

C:\>ping 30.30.30.30

```
Pinging 30.30.30.30 with 32 bytes of data:

Reply from 30.30.30.30: bytes=32 time=20ms TTL=254
Reply from 30.30.30.30: bytes=32 time=10ms TTL=254
Reply from 30.30.30.30: bytes=32 time=20ms TTL=254
Reply from 30.30.30.30: bytes=32 time=10ms TTL=254

Ping statistics for 30.30.30.30:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 10ms, Maximum =  20ms, Average =  15ms
```

VPN-remote> show ip nat statistics

```
Total active translations: 1 (0 static, 1 dynamic; 1 extended)
Outside interfaces:
  Ethernet1
Inside interfaces:
  Ethernet0, Loopback0
Hits: 4  Misses: 2
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 198 interface Loopback0 refcount 1
```

VPN-remote> show ip nat translations

There is no output due to no NAT applied on traffic through VPN tunnel.

VPN-remote> show ip access-lists

```
Extended IP access list 198
    deny ip 192.168.2.0 0.0.0.255 30.30.0.0 0.0.255.255 (87 matches)
    permit ip 192.168.2.0 0.0.0.255 any
    deny ip 140.140.140.0 0.0.0.255 30.30.0.0 0.0.255.255
    permit ip 140.140.140.0 0.0.0.255 any (2 matches)
```

Access list shows that all traffic toward the network 30.30.0.0 should be not NATed, (and go through VPN tunnel), while traffic destined to all other networks (keyword any) is split tunnel to Internet and NATed.

VPN tunnel establishment in network extension mode:

Routing table output on Easy VPN Server before VPN tunnel establishment shows no knowledge of the remote network in the headquarter VPN gateway routing table:

VPN-server> show ip route

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
Gateway of last resort is 0.0.0.0 to network 0.0.0.0

     20.0.0.0/24 is subnetted, 1 subnets
C       20.20.20.0 is directly connected, Ethernet1
     30.0.0.0/32 is subnetted, 1 subnets
C       30.30.30.30 is directly connected, Loopback0
S*   0.0.0.0/0 is directly connected, Ethernet1
```

Routing table output on Easy VPN Server after VPN tunnel establishment in Network Extension mode shows remote branch office network (192.168.2.0) locally attached to the interface where the VPN tunnel is coming from (Ethernet 1):

VPN-server> show ip route

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
Gateway of last resort is 0.0.0.0 to network 0.0.0.0

     20.0.0.0/24 is subnetted, 1 subnets
C       20.20.20.0 is directly connected, Ethernet1
S    192.168.2.0/24 [1/0] via 0.0.0.0, Ethernet1
     30.0.0.0/32 is subnetted, 1 subnets
C       30.30.30.30 is directly connected, Loopback0
S*   0.0.0.0/0 is directly connected, Ethernet1
```

Connectivity test to PC2 behind the Easy VPN Remote shows that there is no NAT applied on tunneled traffic in Network Extension mode:

VPN-server> ping 192.168.2.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/12 ms

Appendix B - List of Acronyms

Acronym/Term	Definition
ACL	Access Control List
AES	Advanced Encryption Standard
AH	Authentication Header
AS	Authentication Server
ATM	Asynchronous Transfer Mode
CA	Certification Authority
CAST	Charlie Adams and Stafford Tavares - CAST algorithm
CCP	Compression Control Protocol
CHAP	Challenge Handshake Authentication Protocol
CRACK	Challenge Response Authentication for Cryptographic Keys
CRC	Cyclic Redundancy Check
DES	Data Encryption Standard
3DES	Triple Data Encryption Standard
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Service
DoS	Denial Of Service
DPD	Dead Peer Detection
DSS	Digital Signature Standard
EAP	Extensible Authentication Protocol
ECP	Encryption Control Protocol
ESP	Encapsulating Security Payload
IKE	Internet Key Exchange
GRE	Generic Router Encapsulation
HDR	Header
HMAC	Hashed Message Authentication Code
HSRP	Hot Standby Router Protocol
IETF	Internet Engineering Task Force
IDEA	International Data Encryption Algorithm
IKE	Internet Key Exchange
IOS	Internetwork Operating System
IP	Internet Protocol
IPsec	Internet Protocol Security Protocol
IPPCP	IP Payload Compression Protocol
IPcomp	IP Compresion

Acronym/Term	Definition
IPsec	IP Security Protocol
IPSRA	IP Security Remote Access
IPX	Internetwork Packet Exchange
IRC	Internet Relay Chat
ISAKMP	Internet Security Association Key Management Protocol
ISP	Internet Service Provider
JFK	Just Fast Keying
L2TP	Layer 2 Tunneling Protocol
L2F	Layer 2 Forwarding
LAC	L2TP Access Concentrator
LDAP	Lightweight Directory Access Protocol
LNS	L2TP Network Server
MAC	Message Authentication Code
MD5	Message Digest 5
Mode Config	IKE Mode Configuration
MPLS	Multi Protocol Label Switching
NAS	Network Access Server
NAT	Network Address Translation
NAPT	Network and Port Address Translation
NTP	Network Time Protocol
OSI	Open System Interconnection
OSPF	Open Shortest Path First
OTP	One Time Password
PAP	Password Authentication Protocol
PAT	Port Address Translation
PC	Personal Computer
PIC	Pre-IKE Credentials
POP	Point of Presence
PPP	Point to Point Protocol
PPTP	Point to Point Tunneling Protocol
PPVPN	Provider Provisioned VPN
PSTN	Public Switch Telephony Network
PVC	Permanent Virtual Circuit
QoS	Quality of Service
RADIUS	Remote Authentication Dial-In User Service
RFC	Request For Comments
RRI	Reverse Route Injection
RSA	Rivest, Shamir, Adelman Protocol

Acronym/Term	Definition
SA	Security Association
SADB	Security Association Database
SCTP	Streaming Control Transport Protocol
SHA	Secure Hashing Algorithm
SIP	Session Initiation Protocol
SIGMA	Signature Mode of Authentication
SOI	Son of IKE
SPD	Security Association Policy Database
SPI	Security Parameter Index
SVC	Switched Virtual Circuit
TACACS+	Terminal Access Controller Access Control System +
TCP	Transport Control Protocol
TCP/IP	Transport Control Protocol/Internet Protocol
TLS	Transport Layer Security
TTL	Time To Live
UDP	User Datagram Protocol
ULA	User-level Authentication
VPN	Virtual Private Network
VRRP	Virtual Router Redundancy Protocol
Xauth	IKE Extended Authentication
WAN	Wide Area Network
WINS	Windows Internet Naming Service