AAA Functionality for Handheld Systems

Ji Cao

Kongens Lyngby 2005

Abstract

This master thesis project presents a VPN solution with AAA functionality for handheld device. We analyze the requirement of the system especially the security and mobility, and then we come up with a solution composed of a set of software or protocol.

Preface

This thesis is submitted to fulfill the requirements of the Master of Science in Computer System Engineering. The project was done by Ji Cao during the period February 2005 to July 2005 at the department of Informatics and Mathematical Modelling (IMM), Technical University of Denmark (DTU). The work was supervised by Professor Robin Sharp.

I would like to thank my supervisor Robin Sharp for his assistance throughout the thesis. He assisted me on all the stages of my thesis and always gave me good idea and helpful instruction.

Special thanks to my parents and my girlfriend for their support, especially at my difficult time.

Lyngby, July 2005

Ji Cao

Contents

Chapter 1

Introduction

1.1 Background

1.1.1 Traveling employee

An employee that is on traveling often need to access the intranet of his/her company. Because the employee can be anywhere, like hotel, cafe bar, airport, the security of access to the company is not guaranteed by nature. Since it is such a common case, The security about visiting the intranet is a very hot topic nowadays.

1.1.2 e-Library user

Suppose there is such an e-library, it composes of several servers, each of which contains tons of resource that user is interested. The user who holds the PDA needs to get authenticated before being able to visit all those servers. And the library charges the user by hours that the user spends on accessing the library.

How to create such kind of authentication and billing system?

1.1.3 Summary

The common characteristic of these two cases is: A PDA or PC user with an unknown IP address, needs to visit a LAN. How to implement the AAA (authentication, authorization, accounting) functionality on that system. And how to make sure that the operation is with a high level of security.

1.2 Goal of this project

The aim of this project is to investigate how so-called AAA (Authentication, Authorization and Accounting) functionality is to be incorporated into a distributed system based on mobile PDAs. This functionality makes it possible to check that a PDA which attempts to communicate from anywhere via the Internet is correctly identified and receives suitable authorization to handle remotely stored, potentially confidential information.

1.3 Outline of this thesis

Chapter 2 introduces the requirement of the project and the analysis of the requirement.
Chapter 3 introduces the existed technology which might be suitable to use in this project.
Chapter 4 compares these technologies and chooses the most suitable one.
Chapter 5 presents the whole solution's architecture and components.
Chapter 6 and 7 present the installation and configuration of the solution, both in the server and client side.
Chapter 8 made a verification of this solution's functionality and security.
Chapter 9 is the security analysis of the system.
Chapter 10 is the summary of the whole thesis.

Chapter 2

Requirement

2.1 Environment introduction

To focus on the point, we choose a simple network model, which is close to the real network environment with some extension:

- Server side
  There is a LAN, which can access the Intenet though, the gateway server. The gateway has a global IP address (e.g. 130.225.76.9), and also a private IP address (e.g 192.168.0.2). The machine in the LAN only has one private IP address (e.g 192.168.0.3).
- Client side
  There is a PDA which has a global IP address, assigned by an Access Point + Route. The PDA can access the Internet (including the gateway server mentioned above), with the wireless service of the Access Point. The Access Point and the Route mentioned above, most likely will be those of the airport, hotel, Campus, because it will probably not always the same one, and mostly likely we have no control on them at all. We can just think of a computer or PDA with Internet connection.

  And the PDA's functionality is also very limited. PDA is basically a computer: it has the CPU, memory, input device and output device. But

it's destined to have only a few software installed. So it is not a good solution if it needs the PDA to install a lot of lib files or some kind of heavyweight software.

- Internet
  The Internet basically provides the connection between the client and server.

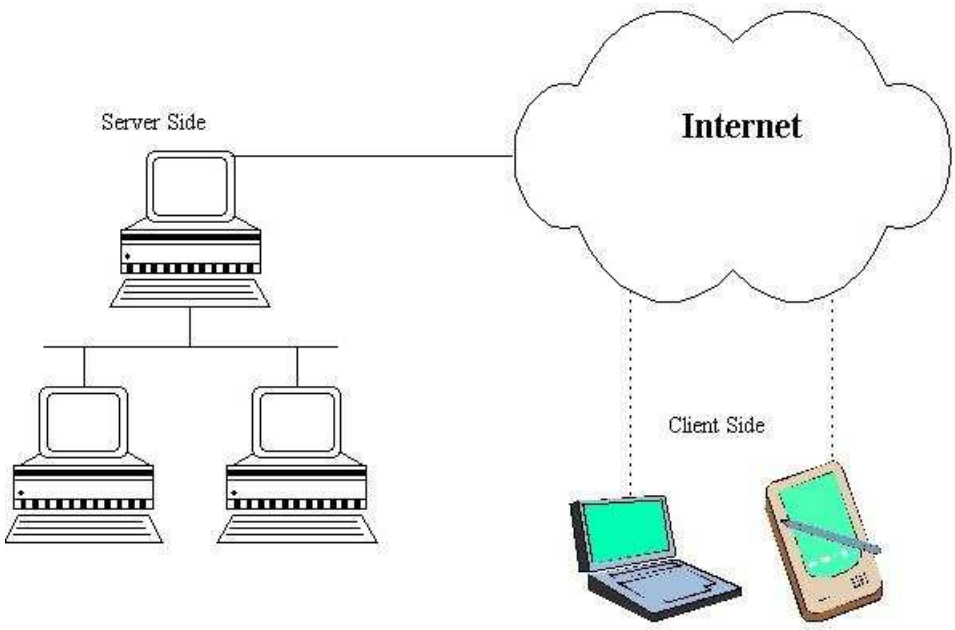

Figure 2.1: The basic network model.

2.2 Functional requirement

1. *Ability to Access to the LAN.* This is the most important requirement of the whole project. The PDA should be able to access the Internal Email Server, web server, and file storage server etc. which is located inside the LAN. The outside handheld device should have almost the same privilege as it is inside the LAN after dialing in.

2. *Authentication, Authorization and Accounting (AAA) enabled.* This is also an important requirement. The AAA service must be dependent from the NAS (network access server). That means it should allow a distributed system: one or more NAS server work with one or more AAA server.

3. *Secure.* It is one of the most important requirements of the whole solution. We always need to consider about this important issue when making choices. It basically includes:

   - The account and password must be safe.
   - The data communication exposed to the Internet needs to be kept confidential and not modified.

   The security level should be at least on the industry level.

4. *Support the handheld device* It must support the client which can be normal PC or PDA, etc. which have internet connection.

2.3 Non-functional requirement

1. *Easy.* It means the whole solution should try to avoid unnecessary complexity. It should be easy to install, configure and maintain. And we should take advantage of using existed protocol or software instead of designing a new one. Because designing a new one means non-standard, hard to understand, and a lot of work. We shall avoid doing so, unless it is really necessary. Easy also means the client side's configuration must be as simple as possible. Because there might be a lot of clients and the client user might make mistakes if it is not easy.

2. *Cheap.* According to the budget we have no plan to import any commercial software to this solution.

3. *Interoperable.* This is also a very important requirement for the following reason.

   - *Cooperate with other system.* Since there are a lot of existed systems, maybe they are old and slow; maybe they have a lot of bugs. But they exist and is currently running. So our system has to be compatible with them. And we also know our solution will become old, so being stick to the standard means to have more chance to be compatible with the other system.
   - *Replaceable.* If the user is not satisfied with the software used in this solution(they have their own favorite software for that functionality for whatever reason), it could be very easy if the software is interoperable with others.
   - *Easy to use.* If the system is interoperable, it must be following some rules. So as long as the user familiar with the similar system, it would be very easy for him/her to use this system.

2.4 Solution for the Requirement

Basically, those requirements can be met by the so-called "AAA+VPN". AAA server provides AAA service and the VPN server provide 'access to the internal LAN with security'. The difficult part is that there is a special requirement – supporting handheld device. We know the handheld device will normally access the Internet though a wireless Access Point. So its IP address is varying from time to time. This is so called "Road warrior". This must be taken into account when choosing the VPN solution, because some VPN doesn't have this feature.

Chapter 3

Existed Technology

After analysis of the requirement, we take a look at what technologies related are available in the industry, focusing on AAA, VPN and Cryptography.

3.1 AAA

AAA stands for Authentication, Authorization and Accounting. They provide protection of investments and businesses against malicious users, but also offer auditing and session information or support for billable services by allowing and tracking network access, gateway services, high bandwidth or low latency or jitter paths. In general, the three A's are defined in "Criteria for Evaluating AAA Protocols for Network Access" as:[4]

1. *Authentication.* The act of verifying the claimed identity of an entity (user or device)[4]
2. *Authorization.* The act of determining if a requester can be granted a right (e.g. network access, high bandwidth service, etc.) [4]
3. *Accounting.* The act of collecting information on resource usage for the purpose of trend analysis, auditing, billing or cost allocation. [4]

In a typical AAA serivce scenario, (figure 3.1), a client submits its identification to the NAS [1], NAS connects to the AAA server, then decide to let the user use the network or not.

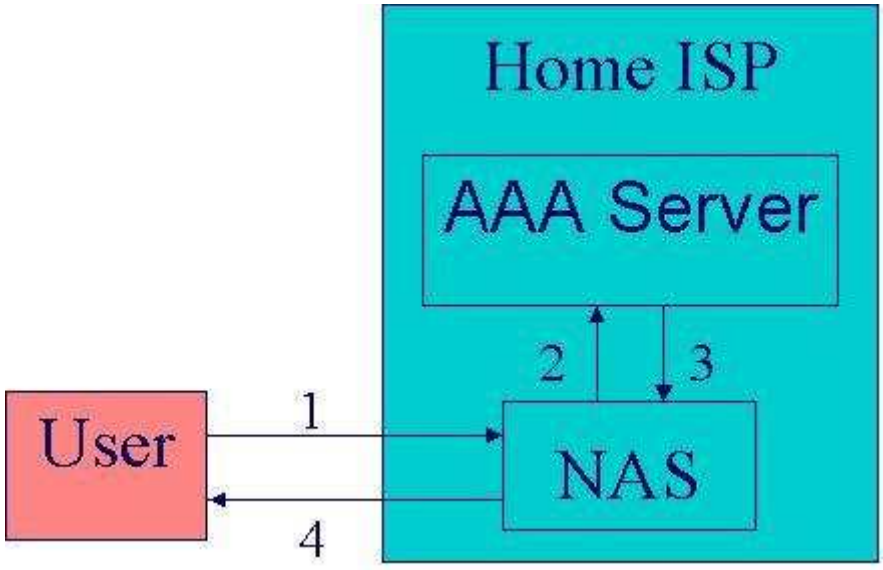

Figure 3.1: The AAA server.[4]

3.1.1 Radius

3.1.1.1 Introduction

The Remote Authentication Dial-In User Service (RADIUS) protocol was an access server authentication and accounting protocol. The RADIUS specification is defined by RFC 2865. The RADIUS accounting standard is defined by RFC 2866. RADIUS servers receive user connection requests, authenticate the user, and then return the configuration information necessary for the client to deliver service to the user. In addition a RADIUS server can act as a proxy client to other RADIUS servers or other kinds of authentication servers.

RADIUS is a client/server protocol. The RADIUS client is a usually a network access server (NAS) or authenticator and the RADIUS server (or authentica-

[1]Network Access Server, provides a network service to the dial-in user as a gateway

tion server) is usually a daemon process running on an appliance, a UNIX or Windows NT machine. The client passes user information to designated RADIUS servers and acts on the response that is returned. The RADIUS server can support a variety of methods to authenticate a user including PAP, CHAP, MS-CHAP, and MS-CHAP2

3.1.1.2 Feature

1. *Centralized user administration.* Radius server makes user management centralized, so the insert and removal,modification of user information can be operated on the radius server. The burden of user management is far from the NAS server, which is not good at doing that and has some more important work to do.

2. *Secure.* RADIUS consistently provides some level of protection against a sniffing, active attacker. Other remote authentication protocols provide intermittent protection, inadequate protection or non-existent protection. RADIUS's primary competition for remote authentication is TACACS+ and LDAP. LDAP natively provides no protection against sniffing or active attackers. TACACS+ is subtly flawed, as discussed by Solar Designer in his advisory. [1]

3. *Popular.* RADIUS is currently the de-facto standard for remote authentication. It is very prevalent in both new and legacy systems. It is get consistent support from hardware vendors.[1]

3.1.1.3 Known weakness

1. *Security Issues.* It is reported the RADIUS protocol has a set of vulnerabilities that are either caused by the protocol or caused by poor client implementation and exacerbated by the protocol. It is mostly about the Shared secret, User-Password attribute and Request Authenticator.[1]

2. *Limitation.* There are several general shortcomings of the RADIUS protocol that were addressed in the design of the Diameter base protocol. Like 1)Limited size of attribute data. 2)Limited number of concurrent pending messages. 3)Limited server failure detection. 4)Suffed from replay Attacks.

3.1.2 Diameter

3.1.2.1 Introduction

The Diameter base protocol is intended to provide an Authentication, Authorization and Accounting (AAA) framework for applications such as network access or IP mobility. Diameter is also intended to work in both local Authentication, Authorization & Accounting and roaming situations. This document specifies the message format, transport, error reporting, and accounting and security services to be used by all Diameter applications. The Diameter base application needs to be supported by all Diameter implementations.[2]

The Diameter model is a base protocol and a set of applications. The base protocol provides common functionality to the supported applications. The following figure depicts the Diameter architecture. The Diameter protocol consists of two main components, the Diameter Base Protocol and the CMS (Cryptographic Message Syntax) Security Module. The base protocol, as the name suggests, offers all basic functionality needed to provide full AAA services. The CMS Module, which had been a separate entity in earlier versions of the protocol but has been tightly implemented later on, adds the necessary safety features such as encryption and digital signatures.[5]

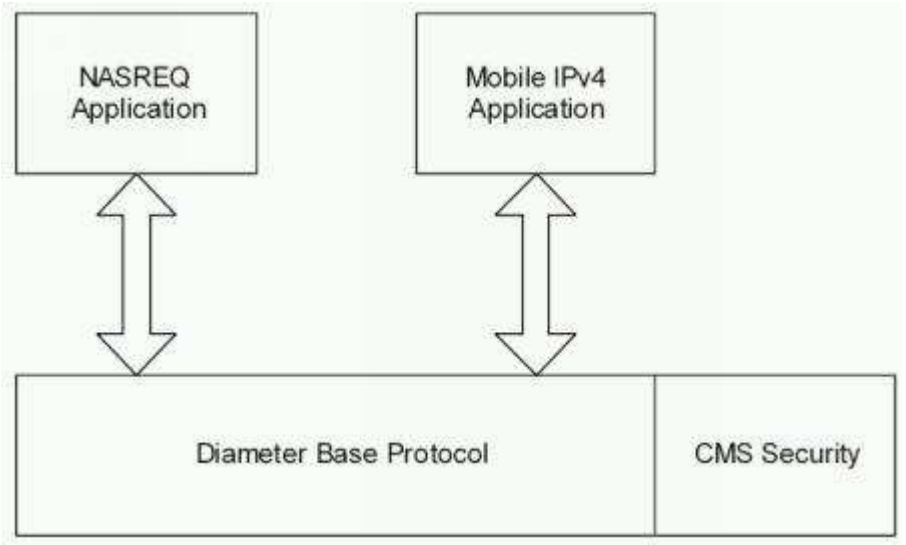

Figure 3.2: Diameter protocol architecture.[5]

3.1.2.2 Feature

Diameter comes as the replacement of Radius. Comparing to Radius, it has these feature:

1. *Better transport.*
   - Diameter runs over a reliable transport TCP or SCTP while Radius runs over UDP.
   - Lost packets will be retransmitted at each hop.
2. *Better Proxying.*
   - Hop-by-hop transport failure detection allows failover to occur at the appropriate place ¡[a] proxies can locally failover to an alternate next-hop peer.
   - The proxy automatically does retransmission of pending request messages following a failover.
3. *Better Security.*
   - Hop-by-hop security is provided using IPsec or TLS.
   - End-to-end security protects the integrity and/or confidentiality of sensitive AVPs through intermediate proxies. [5]
4. *More Information.*
   - able to ask for additional logon information beyond the basic authentication.
   - able to exchange user accounting information among different ISPs.

3.1.2.3 Know weakness

The RFC [3588] 'Diameter Base Protocol' which describe the specification of Dieameter's protocol is released on September 2003. So it is only less than 2 years old. Although more and more hardware vendors begin to support Diameter, however, Diameter is still quite a new thing. It hasn't been widely used for production. And that also means little experience about diameter and more maintenance cost to use Diameter.

3.1.3 TACACS

3.1.3.1 Description.

TACACS is an authentication scheme that can be used to validate users who are attempting to gain access to information servers, networks, and remote access servers. TACACS was originally developed by the U.S. Department of Defense and BBN Planet Corp. and then further developed by Cisco. There are three versions of the protocol: the original TACACS as just mentioned, XTACACS (Extended TACACS), and TACACS+. The first two versions are discussed in RFC 1492 (An Access Control Protocol, Sometimes Called TACACS, July 1993). TACACS+ is the latest version and should be used whenever TACACS is called for. TACACS is also discussed in RFC 2975 (Introduction to Accounting Management, October 2000). Note that TACACS, in general, is no longer being maintained.[6] TACACS uses UDP port and provides authentication, authorization but no accounting service.

3.1.3.2 Known weakness.

TACACS is now somewhat dated and is not used as frequently as it once was. A later version of TACACS was called XTACACS (Extended). These two versions have generally been replaced by TACACS+ and RADIUS in newer or updated networks. TACACS+ is a completely new protocol and is therefore not compatible with TACACS or XTACACS. [7]

3.2 VPN

A virtual private network is the creation of private links across public networks such as the Internet. The idea is to create what appears to be a dedicated private link on a shared network using encryption and tunneling techniques. Anybody can create a private connection by encrypting the contents of the traffic being sent across a network, but truly secure VPNs are better built with the cooperation of service providers that can create dedicated paths with guaranteed service levels across their networks.[8]

The advantage of VPN are:

- The networks which are physically located distantly can communicated

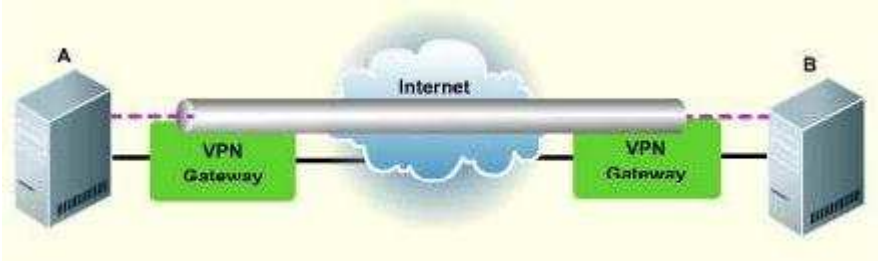

Figure 3.3: VPN.[37]

as if they are inside a private network. And the cost is much lower than traditional private network.

- VPN has more control on the network management than traditional private network.
- VPN make internal network accessible from any place where Internet connection is available. So home office is much easier and traveling people can access their internal network with his/her account. All these access are reasonably secure if the VPN has proper security settings.

There are several different technologies (e.g. PPTP, IPSEC) can be used to implement the VPN solution, they will be discussed in the following sections.

3.2.1 PPTP

3.2.1.1 Introduction

Point-to-Point Tunneling Protocol (PPTP) is a network protocol that enables the secure transfer of data from a remote client to a private enterprise server by creating a virtual private network (VPN) across TCP/IP-based data networks. PPTP supports on-demand, multi-protocol, virtual private networking over public networks such as the Internet.

PPTP is an extension of the remote access Point-to-Point protocol defined in the document by the Internet Engineering Task Force (IETF) titled "The Point-to-Point Protocol for the Transmission of Multi-Protocol Datagrams over Point-to-Point Links," referred to as RFC 1171. PPTP is a network protocol that encapsulates PPP packets into IP datagrams for transmission over the Internet

or other public TCP/IP-based networks. PPTP can also be used in private LAN-to-LAN networking.[9] PPTP uses the same types of authentication as PPP (PAP, SPAP, CHAP, MS-CHAP, EAP). PPTP establishes the tunnel but does not provide encryption. It is used in conjunction with the Microsoft Point-to-Point Encryption (MPPE) protocol to create a secure VPN. PPTP has relatively low overhead, making it faster than some other VPN methods.
pptp datagram:

IP Header	GRE Header	PPP Header	IP Header	TCP Header	Data

3.2.1.2 Advantage

Easy to use is a big and might be the biggest advantage of PPTP. The PPTP client is part of the Windows operating system (including Windows 98,Windows 2000,Windows ME,Windows XP and Windows 2000 server). PPTP server is part of Windows 2000 Server, Windows XP. So they come for free as long as you have the operating system. There are also free PPTP client and server software in Linux, Unix and Mac system. The settings are comparably easy.

3.2.1.3 Known weakness

In 1998, Bruce Schneier and Mudge released an "analysis of Microsoft PPTP" [30]. Some serious flaws are found in the following areas[31]:

1. *Password hashing.* Weak algorithms allow eavesdroppers to learn the user's password.
2. *Challenge/Reply Authentication Protocol.* A design flaw allows an attacker to masquerade as the server
3. *Encryption.* Implementation mistakes allow encrypted data to be recovered
4. *Encryption key.* Common passwords yield breakable keys, even for 128-bit encryption
5. *Control channel.* Unauthenticated messages let attackers crash PPTP servers

Then Microsoft released an upgrade to the pptp protocol. This upgrade is available for Windows 95, Windows 98, and Windows NT as DUN 1.3. Microsoft has made the following security upgrades to the protocol. [31]

1. The weaker LAN Manager hash is no longer sent along with the stronger Windows NT hash. This is to prevent automatic password crackers like L0phtcrack (http://www.l0pht.com/l0phtcrack) from first breaking the weaker LAN Manager hash and then using that information to break the stronger NT hash.

2. An authentication scheme for the server has been introduced. This is to prevent malicious servers from masquerading as legitimate servers.

3. The change password packets from MS-CHAPv1 have been replaced by a single change password packet in MS-CHAPv2. This is to prevent the active attack of spoofing MS-CHAP failure packets.

4. MPPE uses unique keys in each direction. This is to prevent the trivial cryptanalytic attack of XORing the text stream in each direction to remove the effects of the encryption.

These changes address most of the major security weaknesses of the orginal protocol. However, the fundamental weakness of the authentication and encryption protocol is that it is only as secure as the password chosen by the user. As computers get faster and distributed attacks against password files become more feasible, the list of bad passwords(dictionary words, words with random capitalization, words with the addition of numbers, words with numbers replacing letters, reversed words, acronyms, words with the addition of punctuation) becomes larger.[32] Because the revised pptp protocol is still vulnerable to offline password-guessing attacks from hacker tools such as L0phtcrack (http://www.atstake.com/research/lc3/)[31] Mudge and Bruce Schneier recommand to use IPSec for security reason, because IPsec use Encrypted Key Exchange, and key-exchange and its variants protocols do not allow passive dictionary attacks against the user's password.

3.2.2 L2TP

The Layer 2 Tunneling Protocol (L2TP) was developed in cooperation between Cisco and Microsoft, combining features of PPTP with those of Cisco's proprietary Layer 2 Forwarding (L2F) protocol. Like PPTP (and as its name implies), L2TP operates at the data link layer of the OSI networking model. L2TP VPNs are supported by many major firewall products, including ISA Server, Check-Point, Cisco PIX, and WatchGuard.[10]
l2tp datagram:

IP Header	UDP Header	L2TP Header	PPP Header	IP Header	TCP Header	Data

3.2.2.1 Advantage

L2TP has many advantage over PPTP

1. *Applicable.*L2TP can be used on non-IP networks such as ATM, frame relay and X.25.
2. *Security.*PPTP provide data confidentiality, while L2TP goes further and even provide:
   - *Data integrity.* The protection against modification of the data between the time it leaves the sender and the time it reaches the recipient.
   - *Authentication of origin.* The confirmation for that the user who claims to have sent the data really did so.
   - *Replay protection.* That keeps a hacker from being able to capture data that is sent, such as the sending of credentials, and then "replay" it to "trick" the server.

3.2.2.2 Disadvantage

L2TP has more data in the header than PPTP, which may translate to a bigger performance hit. It is also less mature and has less support than PPTP. Addtionally, it is not widely used. Using L2tp together with IPSec is much more common than using L2tp alone.

3.2.3 IPSec

Short for IP Security, a set of protocols developed by the IETF to support secure exchange of packets at the IP layer. IPsec has been deployed widely to implement Virtual Private Networks (VPNs).

IPsec supports two encryption modes: Transport and Tunnel. Transport mode encrypts only the data portion (payload) of each packet, but leaves the header untouched. The more secure Tunnel mode encrypts both the header and the payload. On the receiving side, an IPSec-compliant device decrypts each packet. [13] See figure 3.5.

IPSec is an official Internet standard. It is defined by RFCs 1825 through 1829. RFC 1825, 1826, and 1827 are replaced by RFCs 2401, 2402, and 2406

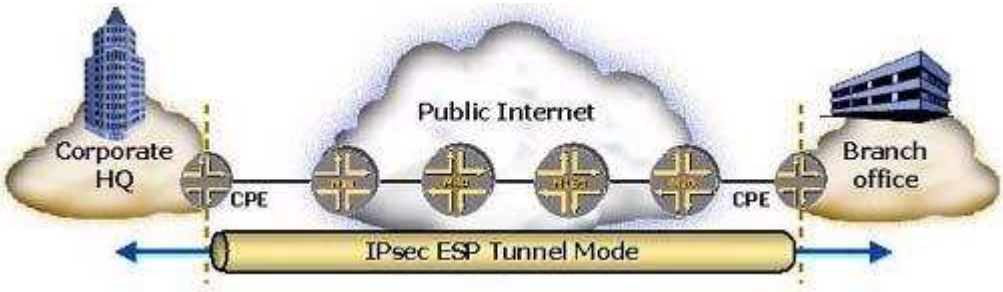

Figure 3.4: IPSec.[37]

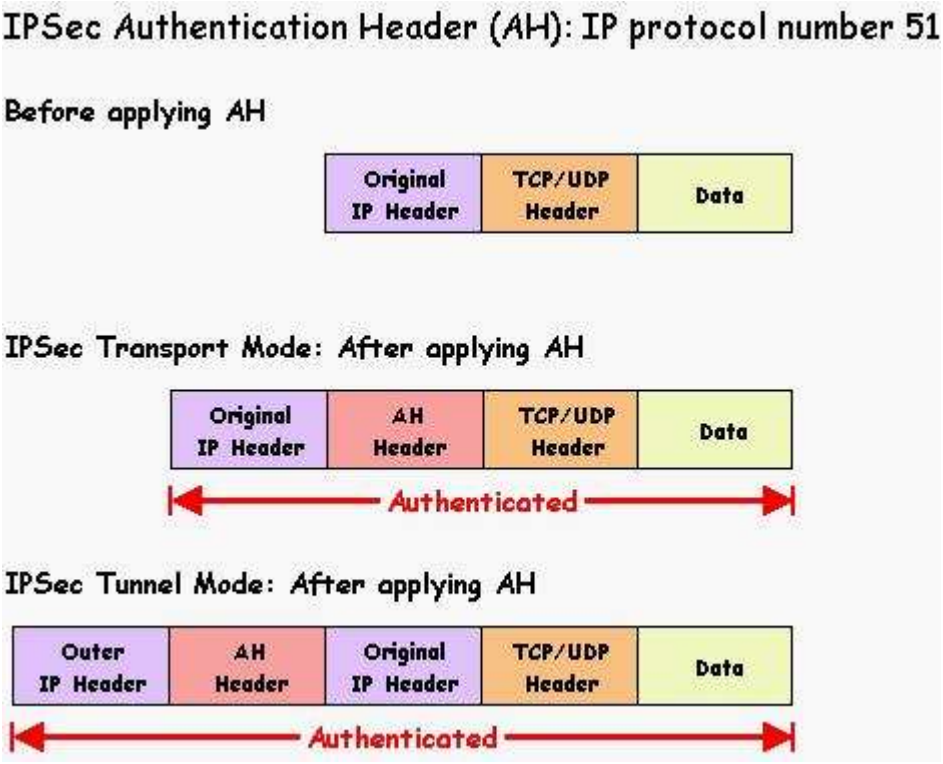

Figure 3.5: IPSec tunnel mode and transport mode.[36]

respectively. IPSEC services are implemented at the IP network layer. Therefore protocols Using IP or above are protected. IPSec supports VPN for "Host to Host" and "Gateway to gateway".

IPSEC contains one or more of the implementation: Authentication Headers (AH) - RFC 2402, Encapsulation Security Protocol (ESP) header - RFC 2406, Key Exchange (ISAKMP) - RFC 2408.

1. *Authentication Header (AH)*: provides authenticity guarantees for packets, by attaching strong cryptographic checksum to packets.[40]
2. *Encapsulating Security Payload (ESP)*: provides confidentiality guarantees for packets, by encrypting packets with encryption algorithms. ESP also provides optional authentication services for packets.[40]
3. *Internet Key Exchange (IKE)*: provide ways to securely negotiate shared keys.[40]

Since IPSEC is designed to be able to use various security protocols, it uses Security Associations (SA) to specify the protocols to be used. SA is a database record, which specify security parameters controlling security operations. They are referenced by the sending host and established by the receiving host. An index parameter called the Security Parameters Index (SPI) is used. SAs are in one direction only and a second SA must be established for the transmission to be bi-directional.

IPSec support algorithms for encryption are: 3DES (mandatory algorithm) DES, CAST-128, Blowfish, AES algorithm. For data authentication, it supports HMAC-MD5, HMAC-SHA1.

3.2.3.1 How it works

In the figure 3.6,IPSec's operation can be broken down into five main steps:

1. "Interesting traffic" initiates the IPSec process. Traffic is deemed interesting when the IPSec security policy configured in the IPSec peers starts the IKE process.
2. IKE phase 1. IKE authenticates IPSec peers and negotiates IKE SAs during this phase, setting up a secure channel for negotiating IPSec SAs in phase 2.

3. IKE phase 2. IKE negotiates IPSec SA parameters and sets up matching IPSec SAs in the peers.

4. Data transfer. Data is transferred between IPSec peers based on the IPSec parameters and keys stored in the SA database.

5. IPSec tunnel termination. IPSec SAs terminate through deletion or by timing out.[38]

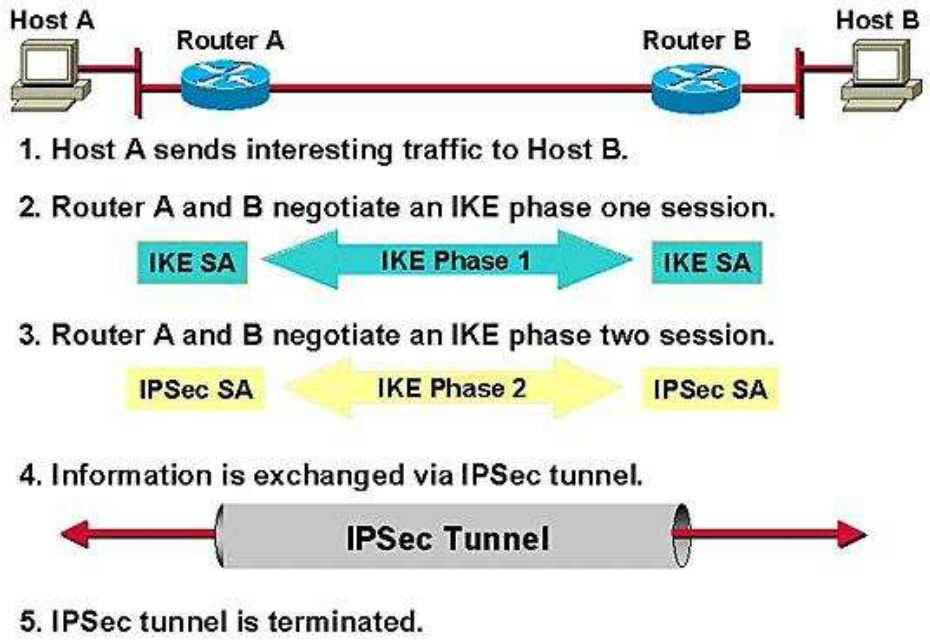

Figure 3.6: IPSec's operation's 5 steps [38]

3.2.3.2 Advantage

1. *Applicable.* Allows encryption in the network layer, which can be used by all applications

2. *Flexibility.* Because IPSec doesn's rely on the underlying network in any way, except to provide IP connectivity, IPSec VPNs can be established between any two points on a public IP network such as the Internet.

3. *Secure.* IPsec is considered more secure than any IP security protocol that has come before: Microsoft PPTP, L2TP, etc

3.2.3.3 Disadvantage

1. *No internal IP addresses.* So the support for Road Warrior is very limited

2. *Complex.* High functionality brings high complexity. IPSec is far complex than PPTP and L2tp.

3. *May not work with NAT.*[2]. NAT rewrites the packet's IP headers so the packet will get rejected when it reaches the other IPSEC node. However, there is solution called NAT Traversal, NAT-T for short. NAT Traversal is a method for encapsulating IPsec ESP packets into UDP packets for passing through routers or firewalls employing Network Address Translation (NAT). However, Some old IPSec implementation needs a patch to enable this feature, like Microsoft Windows 2000/XP.

Although IPSec is famous as its security, one security vulnerability is reported on 09 May 2005[40] at National Infrastructure Security Co-ordination Centre (http://www.niscc.gov.uk). According to that report, any configuration of IPsec that uses Encapsulating Security Payload (ESP) in tunnel mode with confidentiality only, or with integrity protection being provided by a higher layer protocol. Some configurations using AH to provide integrity protection are also vulnerable. If the vulnerability is exploited, it is possible for an active attacker to obtain the plaintext version of the IPsec-protected communications using only moderate effort. to rectify this issue, use any of the following methods :

1. Configure ESP to use both confidentiality and integrity protection. This is the recommended solution.

2. Use the AH protocol alongside ESP to provide integrity protection. However, this must be done carefully: for example, the configuration where AH in transport mode is applied end-to-end and tunnelled inside ESP is still vulnerable.

3. Remove the error reporting by restricting the generation of ICMP messages or by filtering these messages at a firewall or security gateway.

[2]Network Address Translation, an Internet standard that enables a local-area network (LAN) to use one set of IP addresses for internal traffic and a second set of addresses for external traffic

3.2.4 L2TP/IPSec

3.2.4.1 Introduction

L2TP/IPSec is an IETF Internet standard VPN networking and encryption protocol that assures confidentiality of data moving through the link. Unlike firewalls that depend on proprietary IPSec tunnel mode VPN connections that rely on pre-shared "keys" or passwords, secure Internet standards-based L2TP/IPSec connections require that each VPN router identify itself with a user name and password and a machine certificate. The machine certificates guarantee the VPN routers are who they claim to be, and not another VPN router that might be owned by an attacker who has misappropriated a pre-shared key or password.[11] The protocol L2TP over IPSEC is defined in RFC 3193

3.2.4.2 Advantage

1. *Easy in client side.* A l2tp/IPSec client is installed by default on Windows 2000XP, Pocket PC 2003 and Mac OS X v10.3+. On Win9x/Me/NT4, a free l2tp/IPSec client is available. There are also some third-party clients that can be installed.[12]

2. *Secure.* IPsec is generally considered a more secure VPN protocol than PPTP. All the data through the channel are encrypted. [12]

3. *Virtual IP addresses.* The remote client will get an IP address from the internal network once it has logged on. To other computers it will seem as if the remote client is on the internal network. [12]

4. *TCP/IP and IPX tunnelling.*With L2TP, a layer 2 tunnel is created, so in theory any layer 3 protocol can be tunnelled. In most cases, however, TCP/IP will be used within the VPN tunnel. IPX is reported to work as well. [12]

5. *Standard.* The specification is described in RFC 2661 and it is supported by multiple vendors.[12]

6. *NAT-Traversal.* Most of the IPsec clients support this. So even the client doesn't have a global IP address. it still can connect to the VPN server. [12]

3.2.4.3 Disadvantage

1. *Possibly difficult to install on the server.* L2TP/IPsec may be easier to use on the client, but the trade-off is that it is more difficult to install on the server.[12]

2. *Fewer features.* AES encryption, for instance, is currently not supported by any of the Microsoft VPN clients. Many commercial clients and Mac OS X do support AES, which is considerably faster than 3DES. Perfect Forward Secrecy is a security feature that can be enabled for IPsec connections but the Windows and Mac L2TP/IPsec clients do not support it.[12]

3. *Performance.* The payload traffic gets encapsulated a couple of times (IPsec, L2TP, PPP). This requires more bandwidth. It could also result in a problem with MTU size. , IPsec has an overhead of 56 bytes per packet. L2TP will add an extra 16 bytes per packet.[12]

3.2.5 SSL

3.2.5.1 Introduction

SSL is a VPN technology that has been growing in popularity is the Secure Sockets Layer (SSL) VPN. SSL VPN's pros and cons are all very clear. A big advantage of SSL VPNs is that no special VPN client software on the VPN clients is needed. That's because the SSL VPN uses the Web browser as the client application. Thus, SSL VPNs are known as "clientless" solutions. This also means the protocols that can be handled by an SSL VPN are more limited. However, this can also be a security advantage. With SSL VPNs, instead of giving VPN clients access to the whole network or subnet as with IPSec, it can be can restricted them to specific applications. If the applications exposed are not browser-based, however, custom programming might be necessary to create Java or Active-X plug-ins to make the application accessible through the browser. A disadvantage of this is that in order to use such plug-ins, the client's browser settings will have to be opened up to allow active content which means exposing the browser to malicious applets.

SSL VPNs operate at an even higher layer of the OSI model than IPSec VPNs: the session layer. This gives users the ability to control access more granularly. SSL VPNs use digital certificates for server authentication. Other methods can be used for client authentication, but certificates are preferred as the most secure one.

Even though there is no client software installed (other than the Web browser), SSL VPN gateways can still provide the advantages of "managed clients" by forcing the browser to run applets, for example, to verify that anti-virus software is in place before the VPN connection can be established.[10]

SSL provides Data confidentiality (RC4, DES, 3DES...) and data integrity and authentication (MD5, SHA-1) and optional peer authentication with public key cryptography.

3.3 Identification Authentication

Identification Authentication is the process of identifying an individual, usually based on a username and password. In security systems, authentication is distinct from authorization, which is the process of giving individuals access to system objects based on their identity. Authentication merely ensures that the individual is who he or she claims to be, but says nothing about the access rights of the individual. There are several ways to do the authentication. The simplest one is sending the username and password in plaintext.

3.3.1 PAP

Password Authentication Protocol, the most basic form of authentication, in which a user's name and password are transmitted over a network and compared to a table of name-password pairs. Typically, the passwords stored in the table are encrypted. The Basic Authentication feature built into the HTTP protocol uses PAP. The main weakness of PAP is that both the username and password are transmitted "in the clear "– that is, in an unencrypted form. Contrast with CHAP. [14]

3.3.2 CHAP

Challenge Handshake Authentication Protocol, a type of authentication in which the authentication agent (typically a network server) sends the client program a random value that is used only once and an ID value. Both the sender and peer share a predefined secret. The peer concatenates the random value (or nonce), the ID and the secret and calculates a one-way hash using MD5. The hash value is sent to the authenticator, which in turn builds that same string on its side,

calculates the MD5 sum itself and compares the result with the value received from the peer. If the values match, the peer is authenticated. By transmitting only the hash, the secret can't be reverse-engineered. The ID value is increased with each CHAP dialogue to protect against replay attacks. [15]

3.3.3 MS-CHAP

Microsoft's PPP CHAP dialect (MS-CHAP) extends the user authentication functionality provided on Windows networks to remote workstations. MS-CHAP is closely derived from the PPP Challenge Handshake Authentication Protocol (CHAP). Microsoft created MS-CHAP to authenticate remote Windows workstations, providing the functionality to which LAN-based users are accustomed while integrating the encryption and hashing algorithms used on Windows networks. Where possible, MS-CHAP is consistent with standard CHAP. Briefly, the differences between MS-CHAP and standard CHAP are:
The MS-CHAP Response packet is in a format designed for compatibility with Microsoft's Windows NT 3.5, 3.51 and 4.0, and Windows95 networking products. The MS-CHAP format does not require the authenticator to store a clear-text or reversibly encrypted password.
MS-CHAP provides authenticator-controlled authentication retry and password changing mechanisms.[16]

3.3.4 MS-CHAPv2

MS-CHAP v2 is a password-based, challenge-response, mutual authentication protocol that uses the industry-standard Message Digest 4 (MD4) and Data Encryption Standard (DES) algorithms to encrypt responses. The authenticating server challenges the access client and the access client challenges the authenticating server. If either challenge is not correctly answered, the connection is rejected. MS-CHAP v2 was originally designed by Microsoft as a PPP authentication protocol to provide better protection for dial-up and virtual private network (VPN) connections. With Windows XP SP1, Windows XP SP2, Windows Server 2003, and Windows 2000 SP4, MS-CHAP v2 is also an EAP type.

Although MS-CHAP v2 provides better protection than previous PPP-based challenge-response authentication protocols, it is still susceptible to an offline dictionary attack. A malicious user can capture a successful MS-CHAP v2 exchange and methodically guess passwords until the correct one is determined. Using the combination of PEAP with MS-CHAP v2, the MS-CHAP v2 exchange

is protected with the strong security of the TLS channel.[17]

3.3.5 Certificate

The most common use of a digital certificate is to verify that a user sending a message is who he or she claims to be, and to provide the receiver with the means to encode a reply. An individual wishing to send an encrypted message applies for a digital certificate from a Certificate Authority (CA). The CA issues an encrypted digital certificate containing the applicant's public key and a variety of other identification information. The CA makes its own public key readily available through print publicity or perhaps on the Internet.

The recipient of an encrypted message uses the CA's public key to decode the digital certificate attached to the message, verifies it as issued by the CA and then obtains the sender's public key and identification information held within the certificate. With this information, the recipient can send an encrypted reply.

The most widely used standard for digital certificates is X.509.[18] Generally speaking, if a certificate gets accepted on these conditions:

1. *Trusty.* The certificate's issuer CA or the issuer's up level CA is in the trusted CA list. Then the certificate is considered trustable.
2. *Valid.* The current date doesn't exceed the expire date of the certificate

3.4 Cryptography

Cryptography is the art of protecting information by transforming it (encrypting it) into an unreadable format, called cipher text. Only those who possess a secret key can decipher (or decrypt) the message into plain text. Encrypted messages can sometimes be broken by cryptanalysis, also called codebreaking, although modern cryptography techniques are virtually unbreakable. As the Internet and other forms of electronic communication become more prevalent, electronic security is becoming increasingly important. Cryptography is used to protect e-mail messages, credit card information, and corporate data. One of the most popular cryptography systems used on the Internet is Pretty Good Privacy because it's effective and free.[20]

There are three primary cryptographic techniques.

1. *Secret-key.* also call Symmetric Encryption. A single key is used to encrypt and decrypt information. This technique is called symmetric key encryption. Encrypted information may be stored on disk or transmitted over non-secure channels. Since there is only one key, some form of secure key exchange is necessary (in-person, courier, and so on). Typically, it is used for secrecy and integrity of data-single characters to blocks of data, messages and files. DES, Triple DES and AES are belongs to this type.

2. *Public-key.* also call Symmetric Encryption. Two keys are used in this scheme-one to encrypt and one to decrypt. Thus, the scheme is asymmetric. Every person has a set of keys and one is held private while the other is made publicly available. To send a private message to someone, you encrypt it with the recipient's public key. The recipient then decrypts it with his or her private key. This eliminates the problems of exchanging keys in advance of using the encryption. But public-key system is slow, typically, 10,100 times slower than Secret-key system. It is widely used in key exchange, certificate and authentication. The most famous example is RSA Encryption.

3. *Hash functions.* A hash function is an algorithm that produces a unique "fingerprint" of a message that can prove that it has not been altered since its creation. The output of the algorithm is called a message digest. A recipient that runs the same algorithm on the message should arrive at the same digest; otherwise, the message is suspect. It can be used for checking integrity and authentication. Most widely used hash functions are MD4/MD5 and SHA. [21]

The first two are used to encrypt text, graphics, and other information in a form that can be recovered by someone who has an appropriate key. The third, used in authentication and integrity schemes, scrambles input without any intention to recover it.

3.4.1 DES

Short for Data Encryption Standard, a popular symmetric-key encryption method developed in 1975 and standardized by ANSI in 1981 as ANSI X.3.92. DES uses a 56-bit key and uses the block cipher method, which breaks text into 64-bit blocks and then encrypts them.[19]

The Data Encryption Standard (DES) specifies a FIPS approved cryptographic algorithm as required by FIPS 140-1. This publication provides a complete

description of a mathematical algorithm for encrypting (enciphering) and decrypting (deciphering) binary coded information. Encrypting data converts it to an unintelligible form called cipher. Decrypting cipher converts the data back to its original form called plaintext.The algorithm described in this standard specifies both enciphering and deciphering operations which are based on a binary number called a key.

3.4.2 Triple DES

Also referred to as 3DES, a mode of the DES encryption algorithm that encrypts data three times. Three 64-bit keys are used, instead of one, for an overall key length of 192 bits (the first encryption is encrypted with second key, and the resulting cipher text is again encrypted with a third key). [22]

3.4.3 AES

Short for Advanced Encryption Standard, a symmetric 128-bit block data encryption technique developed by Belgian cryptographers Joan Daemen and Vincent Rijmen. The U.S government adopted the algorithm as its encryption technique in October 2000, replacing the DES encryption it used. AES works at multiple network layers simultaneously. The National Institute of Standards and Technology (NIST) of the U.S. Department of Commerce selected the algorithm, called Rijndael (pronounced Rhine Dahl or Rain Doll), out of a group of five algorithms under consideration, including one called MARS from a large research team at IBM. [23]
AES is considered safer than DES and 3DES.

3.4.4 RSA

RSA is an public-key encryption technology developed by RSA Data Security, Inc. The acronym stands for Rivest, Shamir, and Adelman, the inventors of the technique. The RSA algorithm is based on the fact that there is no efficient way to factor very large numbers. Deducing an RSA key, therefore, requires an extraordinary amount of computer processing power and time. The RSA algorithm has become the de facto standard for industrial-strength encryption, especially for data sent over the Internet. It is built into many software products, including Netscape Navigator and Microsoft Internet Explorer. The technology

	DES	AES
Data	1976	1999
Block size	64 bits	128 bits
Key length	56 bits	128,192,256 bits
Encryption primitives	Substitution,permutation	Substitution, shift, bit mixing
Cryptographic primitives	Confusion,diffusion	Confusion,diffusion
Design	Open	Open
Design rationale	Closed	Open
Selection process	Secret	Secret
Source	IBM,enhanced by NSA	Independent Dutch cryptographers

Table 3.1: Comparison of DES and AES[35]

is so powerful that the U.S. government has restricted exporting it to foreign countries. [24]

3.4.5 MD4/MD5

The algorithm takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given prespecified target message digest. The MD5 algorithm is intended for digital signature applications, where a large file must be "compressed" in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as RSA.

The MD5 algorithm is an extension of the MD4 message-digest algorithm. MD5 is slightly slower than MD4, but is more "conservative" in design. MD5 was designed because it was felt that MD4 was perhaps being adopted for use more quickly than justified by the existing critical review; because MD4 was designed to be exceptionally fast, it is "at the edge" in terms of risking successful cryptanalytic attack. MD5 backs off a bit, giving up a little in speed for a much greater likelihood of ultimate security. It incorporates some suggestions made by various reviewers, and contains additional optimizations.[25] The MD5 algorithm is a block-chained hashing algorithm. The first block is hashed with an initial seed, resulting in a hash. The hash is summed with the seed, and that result becomes the seed for the next block. When the last block is computed, it's "next-seed' value becomes the hash for the entire stream. Thus, the seed for block depends on both the hash and the seed of its preceding block. As a result, blocks cannot be hashed in parallel.[26]

Recently, MD5 is not considered very safe. And Vlastimil Klima's paper "Finding MD5 Collisions ¨C a Toy For a Notebook"[41] demonstrats a technique for finding MD5 collisions quickly: eight hours on 1.6 GHz computer.

3.4.6 SHA

SHA1, also known as SHA160, and the Secure Hash Algorithm 160, is a hash algorithm which was developed by the National Institute of Standards.

It is commonly used on the Internet to verify the integrity of software archives, as a unique identifier, and for digital signatures. The SHA-1 is called secure because it is computationally infeasible to find a message which corresponds to a given message digest, or to find two different messages which produce the same message digest.Any change to a message in transit will, with very high probability, result in a different message digest, and the signature will fail to verify.[27] SHA is considered more secure than MD4/MD5. However, Xiaoyun Wang's paper "Finding Collisions in the Full SHA-1"[42] showes the vulnerability of SHA-1.

Chapter 4

Technical choice

To meet those requirement, there are lots of different possible solution, we need to choose the most suitable one, which has high security, full functionality and easy to use.

4.1 AAA Technology

4.1.1 Choice of Protocol

The candidate protocol are TACACS, Radius and Diameter.

TACACS Generally speaking, TACACS is a protocol out of date and is no longer being maintained. It was used for authentication and authorization, but its age is gone. Its successor, XTACACS and TACACS+, however, are not widely used anyway.

Diameter is designed as a replacement of Radius and it is considered more powerful than Radius. It has a lot of features that Radius doesn't have. And

it is also more secure than Radius, because it requires the message for authentication, authorization and accounting is encrypted.

However, Diameter is far complex than Radius and it is still a quite new protocol. It needs more time to get accepted widely in the market. Currently, there are not much experience and support for Diameter. Furthermore, there are very few software(including Server, Client, add-on and plug-in) supporting Diameter. It is fine using Diameter in this solution, but thinking of interoperability, maintenance and support, that is very risky. Few software or hardware support, few document, little experience.

Radius In the AAA area, Radius is very mature comparing to TACACS and Diameter. It is very widely used in the industry. A lot of hardware vendors support it. And there are also a lot of free and commercial software written by different programming language. However, it has its own security flaws and limitation[1], it will be discussed later. So we decide to choose it as the AAA protocol.

4.1.2 How it works

4.1.2.1 protocol summary

A Radius packet contains following data:

1. Code.

Code	Type of Radius packet
1	Access-Request
2	Access-Accept
3	Access-Reject
4	Accounting-Request
5	Accounting-Response
11	Access-Challenge
12	Status-Server (experimental)
13	Status-Client (experimental)
255	Reserved

2. Identifier. The identifier is a one-octet value that allows the RADIUS client to match a RADIUS response with the correct outstanding request.

[1] That's one reason of the birth of the Diameter

3. Length. That is the length of the packet, including the Code, Identifier, Length, Authenticator and Attribute fields.

4. Authenticator. Response Authenticator = MD5(Code + ID + Length + RequestAuth + Attributes + Secret) where + denotes concatenation.

5. Attributes. The attributes section is where an arbitrary number of attribute fields are stored. The only pertinent attributes for this discussion are the User-Name and User-Password attributes.

4.1.2.2 Authentication process

The Authentication process can be divided into three basic steps:

1. *Client creates the packet.* The client creates an Access-Request RADIUS packet, including at least the User-Name and User-Password attributes.

   The Access-Request packet's identifier field is generated by the client. The generation process for the identifier field is not specified by the RADIUS protocol specification, but it is usually implemented as a simple counter that is incremented for each request.

   The Access-Request packet contains a 16 octet Request Authenticator in the authenticator field. This Request authenticator is a randomly chosen 16 octet string.

   Radius packet is a completely unprotected UDP packet, except the User-Password attribute is encrypted by MD5 Hashing and XORed with the shared secret string.

2. *Server checks it.* The server receives the RADIUS Access-Request packet and verifies that the server possesses a shared secret for the client. If it doesn't, the packet will be dropped.

   The server also possesses the shared secret, it can go through a slightly modified version of the client's protection process on the User-Password attribute and obtain the unprotected password. It then uses its authentication database to validate the username and password. If the password is valid, the server creates an Access-Accept packet to send back to the client. If the password is invalid, the server creates an Access-Reject packet to send back to the client.

3. *Client gets the answer.* When the client receives a response packet, it attempts to match its identifier field and also the Response Authenticator, if one of them doesn't match, then the packet will be dropped. If the client received a verified Access-Accept packet, the username and password

are considered to be correct, and the user is authenticated. If the client received a verified Access-Reject message, the username and password are considered to be incorrect, and the user is not authenticated.

4.1.3 Vulnerabilities.

According to Joshua Hill's "An Analysis of the RADIUS Authentication Protocol"[1], RADIUS protocol has a set of vulnerabilities that are either caused by the protocol or caused by poor client implementation and exacerbated by the protocol. [1]

- Response Authenticator Based Shared Secret Attack
- User-Password Attribute Cipher Design Comments
- User-Password Attribute Based Shared Secret Attack
- User-Password Based Password Attack
- Passive User-Password Compromise Through Repeated Request Authenticators
- Active User-Password Compromise through Repeated Request Authenticators
- Replay of Server Responses through Repeated Request Authenticators
- DOS Arising from the Prediction of the Request Authenticator

The detailed description of these attacks is in the Appendix cited from Joshua Hill's paper "An Analysis of the RADIUS Authentication Protocol" 2001.

Because the whole packet is not protected except the User-Password field, the attacker can get some hint of the shared secret or password by monitoring the traffic between the Radius server and client for enough time. And the MD5 is not good enough also lease some hint information about the sensitive data. There are 3 possible solutions:

1. put the NAS(Radius client) and the Radius server into a private LAN to prevent any kind of sniffing or monitoring.
2. Use IPSec to protect the data traffic between the Radius server and client.

3. Use Diameter instead of Radius, it comes as the replacement of Radius and it has more features and is more secure.

In this solution, it will be guaranteed that the traffic between the Radius server and client is secure.

4.1.4 Choice for the implementation

We need an implementation of Radius, non-commercial, open source preferred. And there is a pretty good one–'Freeradius'. Freeradius is one of the best open source radius implementation. In its own website (http://www.freeradius.org), it said:

> *"The RADIUS server has more features and is more flexible than any other free software RADIUS server, and many commercial servers. Most commercial servers are distributed as a "base" system ($), and an "enhanced" version ($$) with more features. FreeRADIUS has all the features of a commercial "enhanced" server, without the associated cost. FreeRADIUS features more than 50 vendor-specific dictionary files. It ships with support for LDAP, MySQL, PostgreSQL, Oracle databases. It supports EAP, with EAP-MD5, EAP-SIM, EAP-TLS, EAP-TTLS, EAP-PEAP, and Cisco LEAP sub-types. It supports proxying, with fail-over and load balancing. It has reached a stable 1.0 release, with incremental improvements being added and tested daily. In short, it is a powerful, fast, and complex RADIUS server which is compatible with the latest network protocols and practices, and is well suited for deployment in any size network.*[29]

And actually, in non-commerial area, Freeradius is the most widely used radius server. It is powerful, and open sourced. The only so called drawback is that it doesn't have a good GUI, like a typical Linux/Unix software. FreeRADIUS is available for a wide range of platforms, including Linux, FreeBSD, OpenBSD, OSF/Unix, and Solaris

4.2 VPN Technology

Comparing to AAA server, there are more possible VPN solution can be used. Some of them need special hardware support; they are not infeasible for this project. Finally, we have following candidate VPN protocols: PPTP, L2TP, IPSec, L2TP/IPSec, SSL. Each of them have many implementations.

4.2.1 Choice of Protocol

PPTP is considered not a very secure protocol. It has been discussed above. On the other hand,it is easy to use: It is by default installed in Windows operating system and also Pocket PC 2002/2003. So if the client is Microsoft's system, no need to install anything. It is meant to be lightweight solution, easy but not very secure.

L2TP is considered more secure than PPTP. However, it is usually used with IPSec, namely L2TP over IPSec.

IPSec is usually used as gate-to-gate or host-to-host solution. the former, needs installation of IPSec on both gateways, in this project, the gateway of the handheld is unknown and not under our control; And the latter, it can only visit only one machine, which doesn't meet the requirement either.

L2TP/IPSec is the combination of l2tp and IPSec. It sounds very suitable for this project.

1. *Secure.* IPSec is considered a secure protocol. L2TP is running on the channel IPSec established. All the data traffic in that channel is encrypted with 3DES.

2. *Less restriction on Client.* It supports client with an unfixed IP address to access the LAN. And it support NAT-T, so even the client doesn't have a global IP address, it doesn't matter. So as long as the client PC/PDA has an L2tp/IPSec client, it will be able to access the LAN. Furthermore, the client will even get an IP address from the sever, so the client seems to be in the LAN in the eyes of other machines.

3. *Client software.* The Windows 2000,XP, Pocket PC2003 has l2tp/Ipsec client default installed. There are also free client for MAC and Linux/Unix. And all these software are free.

4. *Ignorance of application.* The IPSec is running on the IP layer, so the application protocol over that layer will be protected without being noticed. That means most web application like Web browser, ftp client, and Email client. And any other application using TCP or UDP will just work.

Figure 4.1 shows networks set up of the l2tp/ipsec's client and server. Figure

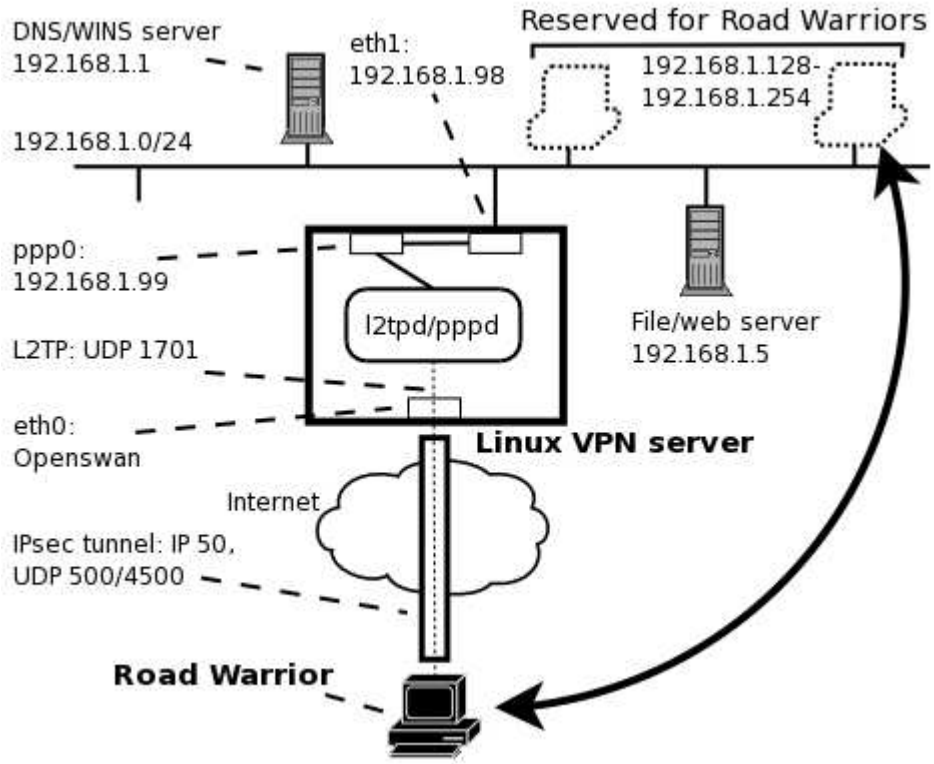

Figure 4.1: l2tp/IPSec.[12]

4.2 and figure 4.3 show the L2TP/IPSec's datagram

SSL is famous as a "clientless" VPN solution. Actually, it still needs a client–web browser, only because almost all the clients' computers have a web browser.

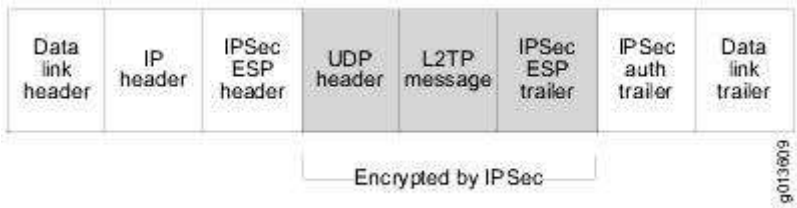

Figure 4.2: L2TP/IPSec-encapsulated control frame.[39]

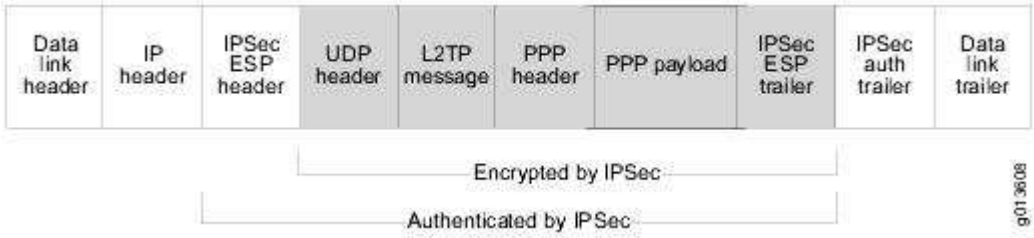

Figure 4.3: L2TP/IPSec-encapsulated data packet.[39]

But its limitation is also very clear: it only supports the service that a web browser can provide. If client also want secure ftp service or secure smtp service, it will need a ftps client and Secure POP3 or Secure IMAP client.

Usually, ActiveX is used to extend the functionality of a Web browser, and it is well supported in Microsoft Internet Explorer. This is a general way that SSL also use to provide service more than web. And actually the Pocket Internet Explorer[2] in Pocket PC 2003 support ActiveX. So at the first glance, it seems also a good solution. But with further understanding of Pocket Internet Explorer, it is not good enough to be a solution.

1. *Not support ActiveX online installation.* Internet explorer support install ActiveX online, but Pocket Internet Explorer doesn't, it needs to get the ActiveX by some ways first and use a "regsvrce.exe" program to register it.[3]

2. *Pocket Internet Explorer not good enough.* The Pocket PC's functionality is very limited. So it is not able to take the responsibility of being an VPN client.

[2]Default installed Web browser of Pocket PC 2003
[3]Doesn't it say "clientless"?

3. *Poor Extendibility.*To support secure ftp or secure pop3, Some ActiveX components need to be developed, and it the user need an extra service. Further development is needed and the new ActiveX needed to be installed to the PDA again.

In conclusion, L2TP/IPSec is a very suitable solution for this project, it meets all the requirement of this project. While PPTP can also do the job, but there is some doubt about its security. l2tp is seldom used alone. IPSec can't give the access of the LAN to the 'Road warrior'. SSL vpn is only convenient for the web application.

4.2.2 Choice for the implementation

After deciding to use the L2TP/IPSec protocol, there is still a lot of implementations to choose. Comparing to choosing protocol, the actually implementation becomes not that important. What it should do is explicitly defined in the protocol specification, the difference is usually user friendliness, effectiveness and documentation support.

4.2.2.1 IPSEC

The most popular IPSec implementation is FreeS/WAN, OpenSwan and Strongswan. Besides that there are also a lot of commercial IPSec implementation, but they are out of discussion. FreeS/Wan is the firs implementation in Linux. However, it is not in active development. Openswan and Strongswan are the successors of FreeS/Wan. All these three are open source implementation.

Basicly, FreeS/Wan is not a good choice, it is old and needs several patches to fill its security hole, to support NAT-T, certificate or road warrior. Between Openswan and Strongswan, it is difficult to say which is better, and actually, they are quite similar, from the configuration file to the documentation. Since there are more people using Openswan than Strongswan, it might be easy to use Openswan. And actually, the latest version of Openswan is very suitable to the project, it supports "road warrior", certificate authentication, NAT-T , Linux Kernel and so on.

4.2.2.2 L2TP

Several open source implementation available:

1. *l2tpd.* Most widely used one, it was the first L2tp server available and released under GPL license.
2. *rp-l2tp.* One L2tp implementation. But it has the drawback that it cannot assign dynamic internal (virtual) IP addresses by itself license.
3. *OpenL2TP.* Latest implementation by Katalix. Supports kernel-mode which means that it should be faster but it also requires recompiling the kernel.
4. *l2tp.* A kernel-mode implementation. No activity since January 2002.

Among these, l2tpd is much more widely used than the others, maybe because:

- *Easy to install.* L2tpd runs in user mode so there is no kernel recompilation needed. Recompiling the kernel is often a lot of trouble.
- *Easy to configure.*L2tpd has only one configuration file called l2tpd.conf which is reasonably intuitive to configure.
- *Support IP address pool.* L2tpd has built-in support for IP pools which means it can assign internal IP addresses from a pool that l2tpd maintains.

There is no big difference between those implementations. So the choice is actually not that important.

4.3 Authentication Methods

The choice here is different from the choice for VPN or AAA service. The choice of authentication methods is more like "Which authentication method should the system support?", and the choice are not exclusive. Basically, as described in previous chapter, there are 5 ways of authentication. PAP, CHAP, MSCHAP, MSCHAP-V2, Certificate. There are three parties involved in the authentication: The VPN client, VPN server and the Radius server. The VPN server works as a mailman, it does the transportation for the other two parties. So the authentication's actual client and server is the VPN client and the Radius server.

1. *PAP.* It is the most primitive way of authentication, sending the password in plaintext. Unless there is already a secure channel for data traffic, never use this way to get authentication. Because as long as there is a sniffer on the user's network, the password is almost completely exposed to it. However it can do one thing that CHAP cannot do: save the password in encrypted form in the authentication server side.

2. *CHAP.* It is a more advanced way of authentication than PAP. Roughly speaking, it uses the password as a parameter for the MD5 hash function to generate a hash string, which will be transported to the server side. The server side use the real password to generate the hash string, if these two hash strings match, the client gets authentication. The drawback is that the password in the server side must keep in a plain text form. Anyway, it is much better that PAP, the password can be protected by a Database system or other way.

3. *MSCHAP.* Microsoft created MS-CHAP to authenticate remote Windows workstations, providing the functionality to which LAN-based users are accustomed while integrating the encryption and hashing algorithms used on Windows networks.

4. *MSCHAPv2.*Improved version of MSCHAP, more secure than MSCHAP.

The Radius protocol support PAP and CHAP. And Freeradius support not only PAP and CHAP, but also MSCHAP and MSCHAPv2 by an extention module with vendor specific attribute.
l2tp/IPSec server supports all these authentication methods.
Windows 2000 and XP's client supports all of them.
Pocket PC 2003's l2tp/IPSec client supports MSCHAP and MSCHAPv2.
So, in order to support the Pocket PC 2003's client, either MSCHAP or MSCHAPv2 must be enabled, PAP and CHAP are optional.

Chapter 5

Solution

Based on the analysis in the above, FreeRadius +Openswan +l2tpd + Pocket Pc l2tp/IPSec client will be the main application of this solution. And of course, it still needs some other program to let them work together correctly.

5.1 Network setup

Figure 5.1 shows the concerned part of network for the solution. It can be divided into 4 parts.

1. *VPN server.* The VPN server is a Linux server installed with Openswan and l2tpd.
2. *AAA Server.* FreeRadius is installed and this server can be located in any place that is safe and reachable by the Linux server.
3. *PC inside the LAN.* Those are the machines that the PDA interested in.
4. *PDA.* The PDA have the Internet access with the Access Point. It has the l2tp/IPSec client default installed.

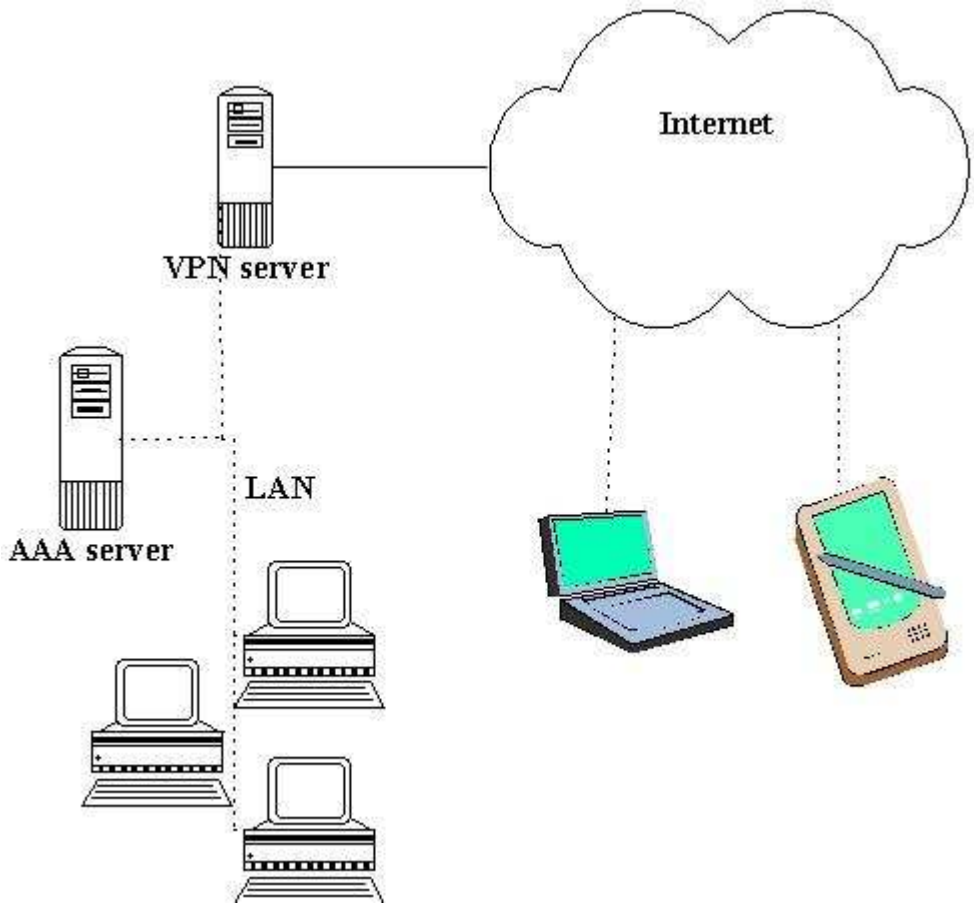

Figure 5.1: The network model in the solution.

The VPN server and Radius server are the most important part of the project. VPN provides client the ability to access the PC in the LAN safely, and the Radius server checks whether the client should be given the ability.

5.2 Firewall or Router

There might a route or firewall in front of the LAN, protecting the whole LAN. Then it should allow some IPsec and l2tp packets coming in and going out.

1. *IPSec.* IPSec use UDP port 500 and IP protocol 50(ESP) and IP protocol 51(AH).
2. *L2tp.* L2tp use L2tp UDP 1701.

And if the VPN server has firewall on, it should allow those packets passing by as well.

5.3 VPN

To install the l2tp/IPSec VPN, the following software are needed.[1]

1. *ppp.*PPP is needed to establish the point-to-point connection. PPP (the Point to Point Protocol) is a mechanism for creating and running IP (the Internet Protocol) and other network protocols over a serial link - be that a direct serial connection (using a null-modem cable), over a telnet established link, or a link made using modems and telephone lines (and of course using digital lines such as ISDN).[28] The latest ppp version is 2.4.3.
2. *L2tp.* L2tp server.
3. *Openssl.* The Openssl is used to create the Certificate Authority and issue the certificate. The certificate is needed to identify the VPN client. It is not mandatory, but using certificate increases the security of the whole system.
4. *Openswan.* IPSec server. It creates a secure tunnel for the client to access the LAN. It has two ways for IPSec authentication.[2]
   - *Pre-shared key(PSK).* The pre-shared key is actually a secret string shared between the IPSec server and the client. And the PSK should be distributed in a safe way, (never over the hostile network, like Internet). However, the PSK poorly support the "dynamic IP address": all the "dynamic IP" users pre-shared key must be the same, this can cause some potential problems, for example, if the PSK needs to be changed by some reason, all the users needs to be noticed.
   - *Certificate* Only when the IPSec server found accept client's certificate. (See section 3.3.5 on page 25 for the condition of accepting)

[1] Other software like gcc,rpm should be installed as well if they are used.
[2] IPSec has its own authentication ,which has nothing to do with the Radius AAA service.

5. *ppp-radius plugin.* The plugin enable the l2tp to use Radius AAA service. As the name imply, it is actually the ppp that uses this plugin to "speak" Radius protocol and communicate with the Radius server.

5.4 Radius server

The Radius server is relatively easy than VPN. Only Freeradius is necessary. Freeradius can just use the account in file, which is the simplest way of accounts information storage. But in real production server, it usually use database system like MySql or other authentication/authrozation system, like Unix system user or LDAP system. This kind of system needs to be installed in the radius server or migrate into the radius server.

5.5 Awareness of the vulnerability

1. VPN. The security of the VPN is counting on the security of IPSec, which is considered as a secure protocol. About the vulnerability mentioned in NISCC Vulnerability Advisory IPSEC - 004033[40], as long as we configure the Openswan correctly(using AH and ESP) together, the vulnerability doesn't matter.

2. Radius. Some vulnerabilities of the Radius is inside its design, so there is no way to fix it without change the protocol. However, it can be a secure solution if the Radius's packets are running in a secure channel. This can be achieved by either putting Radius server in a private network or establishing a secure channel with IPSec. The former way seems more common and will be used in this solution.

3. PDA. As long as the PDA is cummunicating to the VPN server using IPSec, the security of the data traffic is guaranteed by IPSec. The security of the PDA such as wireless connection is beyond the scope of this system.

Chapter 6

Server Installation and configuration

We'll demosrate the solution implementation in the practical network.

6.1 Basic

This server has SuSE 9.2(kernel version 2.6) installed. And since we don't have extra machine to install the IPSec and FreeRadius separately. We have installed them at this Linux server. This inconsistence doesn't matter, and we come to this issue later. The following software will be installed on the server. These are all the programs with the latest version.
Note:

1. *Different operating system.* We use SuSE 9.2, kernel version 2.6. But in principle, all the Linux distribution with kernel 2.4+ can be used as the server. Some old distribution might need some patches to make it work with Openswan. Refer its document to work it out.
2. *Different software version.* In the real world, the server might already have some software installed, like ppp. Then that corresponding step can

be skipped. But if the program is old than our chosen version, and it doesn't work properly, use the latest version.

3. *Binary version.* If you choose to use binary version (e.g. RPM)of these software instead of compiling the source code. It is fine. But for the ppp, it is recommended to compile the source code instead of install the RPM, Because the RPM version of PPP doesn't include the ppp plugin for radius.

6.2 ppp

The Point-to-Point Protocol (PPP) originally emerged as an encapsulation protocol for transporting IP traffic over point-to-point links. It is used by the l2tp to create the ppp channel between the server and the client.

Version 2.4.3 is preferred. If you install the binary version of ppp, you might also need to install the ppp-radius plugin rpm. But ppp source code includes both.

6.2.1 Installation

Download ppp-2.4.3.tar.gz source code from http://ppp.samba.org/

```
tar xzvf ppp-2.4.3.tar.gz
cd ppp-2.4.3
./configure
make
make install
```

6.2.2 Configuration

/etc/ppp/options.l2tp

name MyVPN	
ipcp-accept-local	
ipcp-accept-remote	
noccp	
auth	
crtscts	
idle 1800	
mtu 1410	
mru 1410	
nodefaultroute	
debug	
lock	
+pap	accept the authentication way of pap
+chap	accept chap
+mschap	accept mschap
+mschap-v2	accept of mschap version2
proxyarp	
connect-delay 5000	
plugin /usr/local/lib/pppd/2.4.3/radius.so	use the radius plugin.
logfile /var/log/l2tpd.log	

6.3 l2tpd

l2tpd is the server for l2tp protocol.

6.3.1 Installation

Download l2tpd-0.69-10jdl.i586.rpm from
http://www.jacco2.dds.nl/networking/RPMS/SuSE9.1/l2tpd-0.69-10jdl.i586.rpm

```
rpm -ivh l2tpd-0.69-10jdl.i586.rpm
```

6.3.2 Configuration

[global]	
;listen-addr = 192.168.1.98	
;auth file=/etc/l2tpd/l2tp-secrets	
;[lns 1default]	
ip range = 192.168.0.240-192.168.0.250	the range for the client
local ip = 192.168.0.2	the local IP address
require authentication = yes	
name = MyVPN	
ppp debug = yes	
pppoptfile=/etc/ppp/options.l2tpd	specify the ppp configuration files
length bit = yes	

6.4 OpenSSL

The installation of OpenSSL is optional. It provides some facilities to issue the certificate and change the format of the certificate. The certificate is used for IPSec authentication. Because other way of IPSec authentication PSK[1] is not very flexible ,certificate is needed. If the user has his/her own certificate, and can provide the required certificate format, it is not neccessary to install OpenSSL.

6.4.1 installation of OpenSSL

Download openssl-0.9.7b-74.i586.rpm from
ftp://fr.rpmfind.net/linux/SuSE-Linux/i386/9.0/suse/i586/openssl-0.9.7b-74.i586.rpm

```
rpm -ivh openssl-0.9.7b-74.i586.rpm
```

6.4.2 Setup of Certificate Authority

radius:~# /usr/lib/ssl/misc/CA.sh -newca
CA certificate filename (or enter to create)
(enter)

[1] Pre-Shared Key

Making CA certificate ...
Using configuration from /usr/lib/ssl/openssl.cnf
Generating a 1024 bit RSA private key
.....+++
.....+++
writing new private key to './demoCA/private/./cakey.pem'
Enter PEM pass phrase:(enter password)
(This is the password we will need to create any other certificates.)
Verifying password - Enter PEM pass phrase:(repeat password)
—
You are about to be asked to enter information that will be incorporated into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN . There are quite a few fields but you can leave some blank For some fields there will be a default value,If you enter '.', the field will be left blank.
—
Country Name (2 letter code) [AU]:DK(enter) Enter your country code here
State or Province Name (full name) [Some-State]:State(enter) Enter your state/province here
Locality Name (eg, city) []:City(enter) Enter your city here
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ExampleCo(enter)
Enter your company name here (or leave blank)
Organizational Unit Name (eg, section) []:(enter)
Common Name (eg, YOUR name) []:CA(enter) The name of your Certificate Authority
Email Address []:ca@example.com(enter) E-Mail Address

6.4.3 Generate a certificate request

radius:~/sslca# openssl ca -gencrl -out crl.pem
Using configuration from /usr/lib/ssl/openssl.cnf
Generating a 1024 bit RSA private key
.....+++
.....+++
writing new private key to 'newreq.pem'
Enter PEM pass phrase:(enter password) Password to encrypt the new cert's private key with - you'll need this!
Verifying password - Enter PEM pass phrase:(repeat password)
—
You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN. There are quite a few fields but you can leave some blank. For some fields there will be a default value, if you enter '.', the field will be left blank.

—

Country Name (2 letter code) [AU]:US(enter)
State or Province Name (full name) [Some-State]:State(enter)
Locality Name (eg, city) [ ]:City(enter)
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ExampleCo(enter)
Organizational Unit Name (eg, section) [ ]:(enter)
Common Name (eg, YOUR name) [ ]:host.example.com(enter)This can be a hostname, a real name, an e-mail address, or whatever
Email Address [ ]:user@example.com(enter) (optional)
Please enter the following 'extra' attributes to be sent with your certificate request
A challenge password []:(enter)
An optional company name []:(enter)
Request (and private key) is in newreq.pem

Until now, we have the certificate request, it needs to be signed by the Certificate Authority, the authority can be the CA in this server or more more formal Certificate authority.

6.4.4 Sign the certificate

If the use choose to sign the certificate by the CA in this server, following steps can be followed:
radius:~/sslca# /usr/lib/ssl/misc/CA.sh -sign
Using configuration from /usr/lib/ssl/openssl.cnf
Enter PEM pass phrase:(password you entered when creating the ca)
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName :PRINTABLE:'US'
stateOrProvinceName :PRINTABLE:'State'
localityName :PRINTABLE:'City'
organizationName :PRINTABLE:'ExampleCo'
commonName :PRINTABLE:'host.example.com'
emailAddress :IA5STRING:'user@example.com'
Certificate is to be certified until Feb 13 16:28:40 2012 GMT (3650 days)
Sign the certificate? [y/n]:y(enter)
1 out of 1 certificate requests certified, commit? [y/n]y(enter)

```
Write out database with 1 new entries
Data Base Updated
(certificate snipped)
Signed certificate is in newcert.pem
```

Until now, we get the certificate(newreq.pem) and the key (newcert.pem), rename them to openswan.pem and openswan.key.

6.5 Openswan

Openswan provides the IPSec service.

6.5.1 Installation

Download openswan-2.3.0-1suse9x.i586.rpm from
http://www.openswan.org/download/binaries/suse/9/
i386/openswan-2.3.0-1suse9x.i586.rpm

```
rpm -ivh openswan-2.3.0-1suse9x.i586.rpm
```

6.5.2 Configuration

6.5.2.1 Add the certificate

Copy the certificate, the corresponding key and some CA files into IPSec's directory.

```
$ cp /var/sslca/openswan.key /etc/ipsec.d/private
$ cp /var/sslca/openswan.pem /etc/ipsec.d/certs
$ cp /var/sslca/demoCA/cacert.pem /etc/ipsec.d/cacerts
$ cp /var/sslca/crl.pem /etc/ipsec.d/crls/crl.pem
```

6.5.2.2 Configuration file

/etc/ipsec.conf is the main configuration file for IPSec.
/etc/ipsec.conf - Openswan IPsec configuration file
RCSID $Id: ipsec.conf.in,v 1.132004/03/2404:14:39kenExp$
This file: /usr/share/doc/openswan/ipsec.conf-sample
#
Manual: ipsec.conf.5
version 2.0 # conforms to second version of ipsec.conf specification
basic configuration
config setup
# Debug-logging controls: "none" for (almost) none, "all" for lots.
interfaces=%defaultroute
#klipsdebug=all # enable this to see the debug information of klips
plutodebug=all
nat_traversal=yes #support nat traversal
virtual_private =%4:10.0.0.0/8,%4:172.16.0.0/12,%4:192.168.0.0/16
#define the private address # Add connections here
conn %default
keyingtries=1
compress=yes
disablearrivalcheck=no
authby=rsasig
leftrsasigkey=%cert
rightrsasigkey=%cert
conn roadwarror-all
leftsubnet=0.0.0.0/0
also=roadwarrior
conn roadwarrior-net
leftsubnet=130.225.76.0/24
also=roadwarrior
conn rodwarrior
left=%defaultroute
leftcert=openswan.pem
right=%any
rightsubnet=vhost:%no,%priv
auto=add
pfs=yes
conn roadwarrior-l2tp #for road warrior with l2tp
type=transport
left=%defaultroute
leftcert=openswan.pem
leftprotoport=17/1701

```
right=%any
rightprotoport=17/1701
pfs=no
auto=add
conn roadwarrior-l2tp-oldwin #for road warrior with l2tp
left=%defaultroute
leftcert=radius.pem
leftprotoport=17/0
right=%any
rightprotoport=17/1701
rightsubnet=vhost:%no,%priv
pfs=no
auto=add
conn block
auto=ignore
conn private
auto=ignore
conn private-or-clear
auto=ignore
conn clear-or-private
auto=ignore
conn clear
auto=ignore
conn packetdefault
auto=ignore
#include /etc/ipsec.d/L2TP.conf
#Disable Opportunistic Encryption
#include /etc/ipsec.d/examples/no_oe.conf
```

6.6 Freeradius

6.6.1 Installation

download freeradius-1.0.2.tar.gz from
ftp://ftp.freeradius.org/pub/radius/freeradius-1.0.2.tar.gz

```
tar xzvf freeradius-1.0.2.tar.gz
cd freeradius-1.0.2
./configure
make
make install
```

6.6.2 Configuration

The main configuration file of Freeradius is /etc/raddb/radius.conf There are serveral different ways to get the user authentication, authorization and accounting information. And it is totally up to the user to decide. The easiest way to use freeradius is using file as data storage, we use introduce that here.

6.6.2.1 Specify the method

In the /etc/raddb/radius.conf, make sure it contains this block

```
authorize {
preprocess
chap
mschap
suffix
eap
files# Read the 'users' file for authorization
}
```

6.6.2.2 provide the user data

In the /etc/raddb/user,we create we user Jonh and test

```
John Auth-Type := Local, User-Password == "hello"
Reply-Message = "Hello, %u"
test Auth-Type :=MS-CHAP , User-Password == "chap"
Reply-Message = "Hello, %u"
```

6.6.2.3 Acconting files

In the /etc/raddb/radius.conf, just use the default setting.

```
prefix = /usr/local
exec_prefix = ${prefix}
sysconfdir = ${prefix}/etc
localstatedir = ${prefix}/var
sbindir = ${exec_prefix}/sbin
logdir = ${localstatedir}/log/radius
raddbdir = ${sysconfdir}/raddb
radacctdir = ${logdir}/radacct
# Location of config and logfiles.
confdir = ${raddbdir}
run_dir = ${localstatedir}/run/radiusd
# Write a detailed log of all accounting records received.
detail {
detailfile = $radacctdir/%Client-IP-Address/detail-%Y%m%d
detailperm = 0600
}
```

6.7 ppp-radius plugin

6.7.1 Installation

The plugin includes two parts:

1. *ppp-radius plugin.* It is included in the ppp-2.4.3 source package. And it is already installed if the ppp is installed by compiling the source code. But if the ppp-2.4.3 is installed with just binary code, eg. RPM file, then the ppp-radius plugin is excluded.

2. *radiusclient.*Download radiusclient-0.3.2-142.i586.rpm from ftp.suse.com/pub/suse/i386/9.2/suse/i586/radiusclient-0.3.2-142.i586.rpm

```
rpm -ivh radiusclient-0.3.2-142.i586.rpm
```

6.7.2 Configuration

In the /etc/radiusclient/radiusclient.conf, make sure it contains

```
authserver localhost
# authserver indicates the location of the radius server.
```

And in the /etc/radiusclient/dictionary, make sure it contains

ATTRIBUTE	UserName	1	string
ATTRIBUTE	Password	2	string
ATTRIBUTE	CHAP-Password	3	string
ATTRIBUTE	NAS-IP-Address	4	ipaddr
ATTRIBUTE	NAS-Port-Id	5	integer
ATTRIBUTE	Service-Type	6	integer
ATTRIBUTE	Framed-Protocol	7	integer
ATTRIBUTE	Framed-IP-Address	8	ipaddr
ATTRIBUTE	Framed-IP-Netmask	9	ipaddr
ATTRIBUTE	Framed-Routing	10	integer
ATTRIBUTE	Filter-Id	11	string
ATTRIBUTE	Framed-MTU	12	integer
ATTRIBUTE	Framed-Compression	13	integer
ATTRIBUTE	Login-IP-Host	14	ipaddr
ATTRIBUTE	Login-Service	15	integer
ATTRIBUTE	Login-TCP-Port	16	integer
ATTRIBUTE	Reply-Message	18	string
ATTRIBUTE	Callback-Number	19	string
ATTRIBUTE	Callback-Id	20	string
ATTRIBUTE	Framed-Route	22	string
ATTRIBUTE	Framed-IPX-Network	23	ipaddr
ATTRIBUTE	State	24	string
ATTRIBUTE	Class	25	string
ATTRIBUTE	Session-Timeout	27	integer
ATTRIBUTE	Idle-Timeout	28	integer
ATTRIBUTE	Termination-Action	29	integer
ATTRIBUTE	Called-Station-Id	30	string
ATTRIBUTE	Calling-Station-Id	31	string
ATTRIBUTE	NAS-Identifier	32	string
ATTRIBUTE	Acct-Status-Type	40	integer
ATTRIBUTE	Acct-Delay-Time	41	integer
ATTRIBUTE	Acct-Input-Octets	42	integer
ATTRIBUTE	Acct-Output-Octets	43	integer
ATTRIBUTE	Acct-Session-Id	44	string
ATTRIBUTE	Acct-Authentic	45	integer
ATTRIBUTE	Acct-Session-Time	46	integer
ATTRIBUTE	Acct-Input-Packets	47	integer
ATTRIBUTE	Acct-Output-Packets	48	integer
ATTRIBUTE	Acct-Terminate-Cause	49	integer
ATTRIBUTE	Chap-Challenge	60	string
ATTRIBUTE	NAS-Port-Type	61	integer
ATTRIBUTE	Port-Limit	62	integer
ATTRIBUTE	Connect-Info	77	string

6.8 How to run

In the above programs, some needs to be start manually (Openswan, L2tp, freeradius), some are running inside another program (ppp,ppp-radius plugin, radiusclient), and some are just assistant programs (openssl)

6.8.1 Start the VPN server

Start the IPsec: /etc/init.d/ipsec start
Start the l2tpd: /etc/init.d/l2tpd start
And the commands /etc/init.d/ipsec stop or /etc/init.d/l2tpd stop can stop the server.

6.8.2 Start the Radius server

Start the FreeRadius by "radiusd -X"

Chapter 7

Client

7.1 Basic

The PDA is HP iPAQ Pocket PC h5500 series. The operating system inside is 'Pocket PC 2003'. It can connect to the Internet with its embedded wireless card.

7.2 Certificate importing

Since the PDA's IP address might vary from time to time and IPsec needs to bind 'pre-shared key' with a fix IP address, so it should use certificate instead of pre-shared key to identify itself. Because. And unfortunately Pocket PC 2003 only support direct certificate issued from a Windows 2003 server. However, we have an unofficial tool 'crtimprt' to install a certificate issued from a Linux/Unix server. It can be done in following steps.

7.2.1 Get a certificate

The PDA holder needs a certificate from a Certificate Authority, if the user doesn't have his/her own certificate, then he/she should ask the administrator of the Certificate Authority to generate one for him/her.

7.2.2 Extracting PEM form PKCS#12

With the certificate from the CA(let's call it client.p12)

Extract the user certificate contained within the PKCS#12 file:
openssl pkcs12 -in user.pfx -nokeys -clcerts -out usercrt.pem

Extract the CA certificate(s) contained within the PKCS#12 file:
openssl pkcs12 -in user.pfx -nokeys -cacerts -out cacrt.pem # Extract the private key contained within the PKCS#12 file.
(Warning: the resulting file userkey.pem is not encrypted! # Be careful with it!). openssl pkcs12 -in user.pfx -nocerts -nodes -out userkey.pem

7.2.3 Extracting form PEM

Download pvk program from http://www.jacco2.dds.nl/networking/SRPMS/pvk-0.12-3jdl.src.rpm

openssl crl2pkcs7 -certfile usercrt.pem -certfile cacrt.pem -nocrl -outform PEM -out usercrt.p7b
pvk -in userkey.pem -topvk -nocrypt -out userkey.pvk

Finally, we got the required certificate format pvk and p7b

7.2.4 Transfer files to PDA

Download crtimprt program from http://www.jacco2.dds.nl/networking/crtimprt.zip install the Microsoft ActiveSync on a Windows machine to transfer the files from PC to PDA, make sure the synchronizing folder includes crtimprt.exe, crtim-

prt.cfg, usercrt.p7b, and userkey.pvk. Save them at '/My Documents' at the PDA device.

7.2.5 Run the crtimprt

Make sure the crtimprt.exe, crtimprt.cfg, usercrt.p7b, and userkey.pvk are all at the '/My Documents' of PDA device then run the crtimprt program. if succeed, it should say "Cert Has Been Added Successfully".

7.3 l2tp/IPSec client setup

The setup is comparable easy and straightforward, just 4 steps:(see figure 7.1 and 7.2)

(a) Settings for the connection

(b) Make a new connection

Figure 7.1: Pocket PC 2003 VPN client setup

(a) Enter the username and password

(b) Dial the VPN connection

Figure 7.2: Pocket PC 2003 VPN client setup

Chapter 8

Verification

After all the installation and configuration, we need to do the verification. Maybe it has already been partly done in the process of installation and configuration. But here, we are doing the "put-all-together" verification.

8.1 Verification goal

The project's goal is enabling the handheld device to access the LAN remotely with high level security and AAA service. So the verification can be divided into 3 small parts, Which is separately focusing on : VPN(l2tp/Openswan), AAA(FreeRadius) and Security verification.

- Can the client access the LAN remotely?
  We need to show the client can really reach the computer inside the LAN.

- Are the authentication, authorization and accounting done though the Radius server?
  We need to show the access of the LAN requires a correct user account and the access's time gets recorded for accounting.

- Are the whole process secure at a reasonable level?
  Make sure that necessary security features are enabled and the sensitive data is protected.

8.2 Does VPN work?

8.2.1 VPN Environment

- A simple LAN, composed of 2 machines: A SuSE Linux server with Openswan installed. It has a global IP address: 130.225.76.9 and a private IP address 192.168.0.2/255.255.255.0.It also has http service running on 80 TCP port. A PC in LAN with the IP address 192.168.0.3/255.255.255.0.

- A PDA of HP iPAQ h5500 with Pocket PC 2003, connected to the Internet with wireless network card. Its IP address is unpredictable.

- A PC of windows 2000,connected to the Internet with network card with unpredictable IP. This PC sometimes is used because of the limited function in the Pocket PC 2003 and the LAN we have is extremely simple. [1] So we sometimes need to use the PC to do the verification.

8.2.2 VPN verification

We use the l2tp/IPSec client on the Windows 2000 to connect to the Linux server. We can see that it gets an IP address, 192.168.0.240. (See Figure 8.1)And it can get a ping answer from 192.168.0.2 and 192.168.0.3. This means the l2tp/IPSec client can reach the computer inside the LAN. (See Figure 8.2)

We also need to try the PDA, after establishing the VPN connection to the global IP address of the VPN server. We use the Pocket Internet Browser to visit the 192.168.0.2, And we open the web page successfully. That means we can access the LAN thought the VPN.

[1] This is because of the limited resource for the project.

Figure 8.1: The client got IP address 192.168.0.240

8.3 Does Radius work?

8.3.1 Radius Environment

The involved software here is the radius client (VPN server) and the radius server. Because of we don't have an extra machine, we install the radius client (VPN server) and the radius server at the same machine. In real production situation, they are usually located in different machines. But since we are testing the functionality of the radius, this inconsistency makes no difference.

8.3.2 Radius verification

The file containing the user account information is stored in the Radius directory and used by the radius server. If the user's name and password don't correspond to that in the file, the VPN connection request gets refused. In the console of

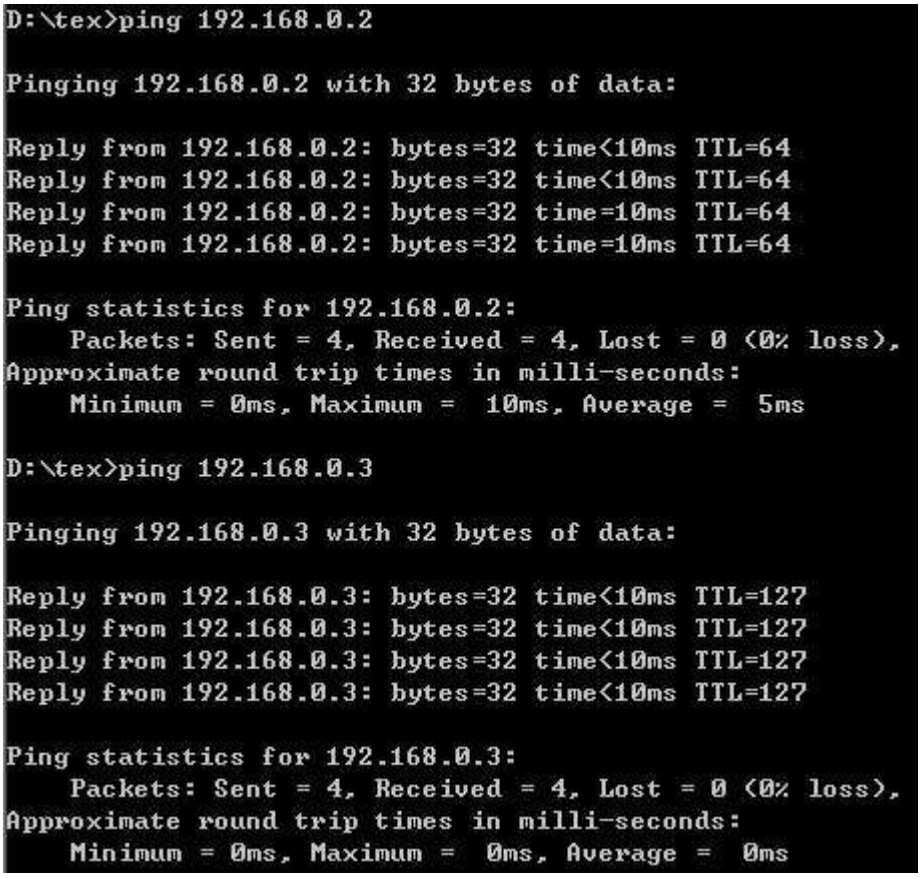

Figure 8.2: The client got ping answer from 192.168.0.2 and 192.168.0.3

the 'radiusd' program, we can see the information about the acceptance and the refusal. That means the radius's authentication and authorization work. About the accouting service, we can found the records of each connection in the directory /usr/local/var/log/radius/radacct/. In this directory, we can found a subdirectory, named '127.0.0.1'. There are serveral files in this subdirectory, log what had happened. In those files, we can clearly see when, who from where get authenticated and also the time for ending. So radius's authentication, authorization and accouting are working there.

Figure 8.3: The pocket PC can visit the machine with its private IP address

8.4 Is it secure?

8.4.1 Theoretical analysis

Since this project uses l2tp/IPSec and Radius for a solution. It is necessary to emphasis its security again.

8.4.1.1 IPSec

The security of l2tp/IPSec is depending on the IPSec. IPsec provides IP datagrams with confidentiality, authenticity and data integrity with two protocols:

Authentication Header (AH) and Encapsulating Security Payload (ESP).

1. *Authentication header (AH)* This header, when added to an IP datagram, ensures the integrity and authenticity of the data, including the invariant fields in the outer IP header. It does not provide confidentiality protection. AH uses a keyed-hash function rather than digital signatures, because digital signature technology is too slow and would greatly reduce network throughput.
2. *Encapsulating security payload (ESP)* This header, when added to an IP datagram, protects the confidentiality, integrity, and authenticity of the data. If ESP is used to validate data integrity, it does not include the invariant fields in the IP header. [34]

IPsec is used widely by nearly all security vendors. It is the primary security protocol used in VPNs. IPSec is generally considered secure. A Cryptographic Evaluation of IPsec can be found at http://www.schneier.com/paper-ipsec.pdf

8.4.1.2 Radius

Radius protocol has its own vulnerability, it has been described in Section 4.1.3. It is basicly the flaw in the protocol design. The possible solution is protecting the data traffic of the radius protocol. That can be done by puting radius server inside the LAN or using IPSec to create a secure channel for the data.

8.4.2 Pratical check

In the client side, we can see the connection is IPSec encrypted with 3DES. In the console of the VPN, we use the tcpdump to monitor the network data traffic between VPN server and VPN client. (See figure 8.4) The result is that at first there are some packets coming and going though udp port 500, and afterwards, all the rest are IP Protocol 50 packets (ESP packets). ISAKMP (Internet Security Association and Key Management Protocol) is a protocol for establishing Security Associations (SA) and cryptographic keys in an Internet environment. It is part of IPSec, used for key exchange. ESP(Encapsulating security payload) protects the confidentiality, integrity, and authenticity of the data. Since all the real data are protected by this kind of packet, all the rest packets are all ESP packet.
So they are running on IPSec protocol, we shall believe in its security is guaranteed at the industrial level.

```
radius:~ # tcpdump  -n not port 22 and host 82.211.196.88
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
09:05:39.624258 IP 82.211.196.88.500 > 130.225.76.9.500: isakmp: phase 1 I ident
09:05:39.630582 IP 130.225.76.9.500 > 82.211.196.88.500: isakmp: phase 1 R ident
09:05:39.693500 IP 82.211.196.88.500 > 130.225.76.9.500: isakmp: phase 1 I ident
09:05:39.709179 IP 130.225.76.9.500 > 82.211.196.88.500: isakmp: phase 1 R ident
09:05:39.743039 IP 82.211.196.88.500 > 130.225.76.9.500: isakmp: phase 1 I ident
[E]
09:05:39.791614 IP 130.225.76.9.500 > 82.211.196.88.500: isakmp: phase 1 R ident
[E]
09:05:39.798695 IP 82.211.196.88.500 > 130.225.76.9.500: isakmp: phase 2/others
I oakley-quick[E]
09:05:39.837583 IP 130.225.76.9.500 > 82.211.196.88.500: isakmp: phase 2/others
R oakley-quick[E]
09:05:39.840364 IP 82.211.196.88.500 > 130.225.76.9.500: isakmp: phase 2/others
I oakley-quick[E]
09:05:39.849122 IP 82.211.196.88 > 130.225.76.9: ESP(spi=0x17980e3f,seq=0x1)
09:05:39.849122 IP truncated-ip - 20 bytes missing! 82.211.196.88.13442 > 69.0.0
.124.0: UDP, length: 31753
09:05:39.859755 IP 130.225.76.9 > 82.211.196.88: ESP(spi=0x3edbea7a,seq=0x2e)
09:05:40.844888 IP 82.211.196.88 > 130.225.76.9: ESP(spi=0x17980e3f,seq=0x2)
09:05:40.844888 IP truncated-ip - 20 bytes missing! 82.211.196.88.13446 > 69.0.0
.124.0: UDP, length: 31753
```

Figure 8.4: The data traffic between VPN server and client.

8.5 Support for PDA

Since there are tons of different types PDA, and some of them only have very limited functionality, so it is impossible to support a few PDA type. The PDA that the system supports needs to have a l2tp/IPSec client. Pocket PC 2003 has that default installed, PDA with other Operating system like Pocket PC 2002 or Palm needs to have a third-party L2tp/IPSec client to use the system. We have used the Pocket PC 2003 to verify the VPN functionality and it works.

8.6 Non-functional requirement

8.6.1 Easy

"Keep it simple and stupid" is one concern when making the technical decisions in this project.

1. *Client* The client user is not with good knowledge about computer, so

it is important to keep the whole settings easy. In this solution, the PDA's VPN client is defaulted installed and the settings are clear and self-explained except importing the certificate. However this can be solved conditionally.

- Using Microsoft's Certificate Services. This is the easiest way to import a certificate to a Pocket PC 2003, and Microsoft recommended it. However, this requires a Windows 2000/2003 Server's Certificate Services, and it is rather expensive.
- Using pre-shared key. This is an alternative way of doing IPSec's own authentication. The cost is that all the VPN clients have to use the share preshared key which might cause security problem especially when the number of users is large.

2. *Server.* The server side's setting are much more complex than the client, as expected. But all the software used in the project are essential and necessary, they can't hardly be simplified without losing functionality or lowering the security. However, all the installation, configuring and maintenance is supposed to be done by the network administrator, the knowledge required for the server settings is absolutely within his/her area.

8.6.2 Cheap

All the software used in this solution are free[2] or part of the operating system[3]. So it is a very economical solution.

8.6.3 Interoperable

The reason to keep it interoperable is being independent of a specific software vendor or a specific implementation. And it also make the system can cooperate with other system more easily. The easiest way to keep it interoperable is using or following the standards.

This requirement is met quite well in this solution. The protocols we used in this solution are standard protocols. Radius is defined by Network Working Group RFC 2865 ; l2tp over IPSec is defined by Network Working Group RFC 3193. So the component of the software is replaceable. These protocols are

[2] all the software in the server side
[3] Pocket PC 2003's VPN client

very widely used, and there are a lot of implementation in hardware, linux/unix or Windows system. If the user has his/her own favorite or a part of function should be provided by another system. Then it can be done with a simple replacement.

Chapter 9

Risk Analysis

9.1 The general attack

Generally speaking, there are several possible ways that security attack might happen to this system:

1. *Attack the Operating system.* Because of the security hole or misconfiguration, the attacker might have a chance to hack the computer, according to the seriousness of the security hole, the attack might be able to crash the computer or take over its control.

2. *Attack the Firewall.* Because of misconfiguration of the firewall or some security flaws of the firewall, the attacker might be able to reach the inside computer and then attack those computers.

3. *Attack the network protocol.* There are some known flaws in some network protocols which might cause security problems. For example, the radius protocol, has some security flaws which is described above. If the attacker armed with the suitable tools like some software and powerful computer, he/she has the chance to crash the radius server, steal the shared secret, or steal an account.

4. *Attack the application.* Sometimes the network protocol is fine, but the implementation has security holes. for example, the IIS [1] without patch has a lot of security holes, that is not the HTTP protocol's fault, but the IIS's. So even the protocol is well designed, bad implementation can still cause problems. According to the seriousness of the security hole, the attacker might be able to crash the service, or read some serect files, get a user account, or even get the full control of the server.

5. *"Man in the middle" Attack.* The "man in the middle' is always pretending to be party A when communicating to party B ,and pretending to be party B when communicating to party A. It make them belielve they are talking to the machine they want to talk. It is also one way of exploiting the network protocols. It is so widely used because a lot of network protocols suffering from this kind of Attack.

6. *Sniffing network traffic.* It's a very easy attack. If the user is not aware of the security, the password might be transfered in plain text or with weak encrpting algorithm (eg. ftp or email account's password). In that case, the attacker can easily get the user account. And if the network traffic is unencrypted or encrypted with a weak algorithm. The attacker can know the content of the network traffic and get some confidential information.

Usually, the attacker will use more than one mathod to attack. However, most of this kind of attacks need a very high skill, are very time consuming and have a rather low possibility to succeed.
We will make an analysis to the possible attack to our system.

9.2 Attack the Operating system

9.2.1 Means

Taking advantage of some security hole or using Trojan, worm, virus.

9.2.2 Difficulty

Depending on the target's security level. If the target gets security patches very often and have firewall on and be aware of these Trojan, worms, it would be

[1] Microsoft Internet Information Server

rather difficult to succeed. Otherwise, the attacker might find a hole that the target machine has be aware of and succeed.

9.2.3 Result if attack succeeds

Depending on the seriousness of the security hole and the power of the Trojan, the attacker might get part or full control of the target computer. Theoretically, the attacker might be able to monitor the computer's input, steal the user account, read or write files, almost do anything he/she wants to do with the computer.

9.2.4 Resist

However, this kind of attack is beyond the discussion of this VPN + Radius system, this system cannot prevent that happening and might be seriously affected by this kind of intrusion. However, the owner of the server and client has the responsibility to resist this intrusion. That includes keeping the operating system updated frequently, not running strange program. This can't solve this problem completely, but definitely highly increase the security.

9.3 Attack the Firewall

9.3.1 Means

Taking advantage of some security flaws or misconfiguration of the firewall and then successfully get the control of one machine inside or something similar.

9.3.2 Difficulty

It is actually rather difficult to succeed unless the firewall has some serious misconfiguration.

9.3.3 Result if attack succeeds

But if that happens, that means the all the LAN machines are exposed. The attack can reach the machine inside. If the radius server is also reachable, then the attack might have a chance to obtain a valid account. And if the attacker found the radius server's security hole, he/she might even have a chance intruding into the machine, which means all the accounts' information will be exposed.

9.3.4 Resist

This is basically also a network and firewall security problem, which is out of the scope of our system.

9.4 Attack the IPSec server.

Usually the vpn server is protected by the firewall, so only the ports for vpn protocol is open to the user. If the these open ports were security hole because of the l2tp/IPsec design,[2] then theoretically the attacker might have a chance to crash the VPN server or even control the VPN server.

9.5 Attack the IPSec protocol or OpenSwan.

9.5.1 Means

Find the vulnerability of the IPSec protocol or OpenSwan and put it into practise. The attack can try to decrypt the message sent between IPSec server and client, it can also try to play "man in the middle", or try to find the flaw of IKE and ESP protocol.

[2]It is very unlikely to be true, at least there is not report about this issue

9.5.2 Difficulty

In the paper "A Cryptographic Evaluation of IPsec"[44], IPSec is proven to be a secure protocol. Until now, there is no efficient way to break the IPSec protocol, which means trying to break the IKE or ESP protocol or trying to decrypt the message will always fail. And there are no security flaw reported about latest version of OpenSwan. So if the IPSec is properly used, it is very difficult to succeed in this way of attacking.

9.5.3 Result if attack succeed

If the attack succeed by any means, the security of the channel is broken, the information in the channel is no longer secure.

9.5.4 Resist

The system is quite good at resisiting this kind of attack, as long as it is properly configured, which is not difficult, it is considered safe.

9.6 Attack the Radius protocol or Freeradius.

9.6.1 Means

The radius protocol has flaws in its design, it has been described above. The attacker can sniff the traffic between the radius server and client. and then take advantage of those flaws and try to steal the pre-shared secret or an user account.

9.6.2 Difficulty

The attacker should understand the radius's security flaw very well and have some powerful machines available. Even that, the attacking will still take a very long time, and might fail if he/she is not lucky enough, Because the Radius's security flaw is not so serious that the attacker can break it very soon.

9.6.3 Result if attack succeeds

The attacker might be able to steal an account without being discovered. The attacker might also be able to crash the radius server.

9.6.4 Resist

The radius server is not fragile, but it is not wise to put it into a public place that everybody can reach. So the easiest way of resisting is protecting it from being reached from outside or unauthorized user. And our system require does that.

9.7 "Man in the middle" Attack

This attack takes advantage of the weakness in the network protocol that the two parites can't prove that it is really who it claims to be. In our system, the IPSec's identification authentication is provided with the certificate, so it doesn't have that weakness. Of course, then the "Man in the middle" can't spoof the IPSec server and the client. And the Radius has a pre-share secret, and it is also kept from the outside, so it can hardly suffer from this attack.

9.8 Sniffing network traffic

This is a very basic attack, only very unprotected protocol like ftp or pop3 will suffer from this and directly leak its account information. Most of the time, sniffing is an assistant for other attacking. In our system, the traffic is protected by IPSec channel, and the second protection is that password is not sent directly[3] through the network. So the sniffer has no way to get the account information at all.

[3]Using CHAP, MSCHAP or MSCHAP2

Chapter 10

Summary

10.1 Project review

The project's goal is letting the PDA access the LAN with AAA (Authentication, Authorization, Accounting) service, and it also makes sure the data traffic between the PDA and the LAN is secure.
With the analysis of the requirement, we know the whole system can be divided into 3 parts: VPN server, Radius server and the PDA client. And these three parts need to be connected by some protocols.
After analyzing the protocol for VPN and AAA service, we decided to use l2tp/IPsec as the VPN protocol, and use Radius as the AAA protocol. l2tp/IPSec establishes the secure channel between VPN server and PDA and Radius is a mature AAA protocol running between the VPN server and the Radius server. Then we chose the implementation for these two protocol and demosrate the installation and configuration.
Finally, we made the verification to make sure that those requirements are really met.

10.2 Candidate solution for this project

Since there are so many technologies existed, l2tp/IPSec is not the only solution. It is the one that we think suit the requirement best. Here is the list of candidate solutions:

1. *PPTP+Radius.* If lower security requirement is acceptable, PPTP+Radius can be used. It is much easier to setup and use.
2. *l2tp/IPSec+Diameter.* If more feature is needed and the expense is acceptable, IPSec+ Diameter can be used. It is much complex but is more powerful.
3. *IPSec+Radius.* If it requires secure access to only one machine, then IPSec+Radius can be used. It provides Machine-to-Machine security.
4. *SSL+Radius.* If the service provided for the PDA is mostly based on Web, then SSL+Radius can be used. It doesn't require PDA to install a new software. And at the same time, the security is also guaranteed.

10.3 Future work

If more time and resource permitted, there are some aspects can be improved.

1. PDA client. Currently, there is no free l2tp/IPSec client for the PDA, except the Pocket PC 2003's VPN client. It would be very helpful to develop a generic l2tp/IPSec client which can run on Pocket PC series and Palm OS[1] as well.
2. Performance test. It would be nice to take some benchmark tests like how many VPN client can connect to the same VPN server without downgrading the connection speed. How much the network is slowered down because of using IPSec to protect the network traffic.
3. Interoperability test. Theoretically, the system's software is replaceable, because the protocols are standards. However, it would be more convincing if we can take some tests like:
    - *Using another Radius server,* e.g. a hardware device or a server with Radius service installed.

[1] Palm OS is one of the most popular OS for PDA

- *Using another l2tp/IPSec VPN server* e.g. Freeswan or other commercial VPN software.
- *Using another l2tp/IPSec VPN client* Some other l2tp/IPsec VPN client running on PDA or PC.

4. Use Diameter to provide AAA service. Diameter is much more complex and powerful than Radius. The actual steps are finding out or developing a diameter plugin for ppp and then install Diameter server software on a server.

Appendix A

Vulnerability of Radius protocol

The following text is cited from Joshua Hill's paper "An Analysis of the RADIUS Authentication Protocol" 2001,http://www.untruth.org/~josh/security/radius/radius-auth.html

A.1 Response Authenticator Based Shared Secret Attack

The Response Authenticator is essentially an ad hoc MD5 based keyed hash. This primitive facilitates an attack on the shared secret. If an attacker observes a valid Access-Request packet and the associated Access-Accept or Access-Reject packet, they can launch an off-line exhaustive attack on the shared secret. The attacker can pre-compute the MD5 state for (Code+ID+Length+RequestAuth+Attributes) and then resume the hash once for each shared secret guess. The ability to pre-compute the leading sections of this keyed hash primitive reduces the computational requirements for a successful attack.

A.2 User-Password Attribute Cipher Design Comments

The User-Password protection scheme is a stream-cipher, where an MD5 hash is used as an ad hoc pseudorandom number generator (PRNG). The first 16 octets of the stream cipher display the same properties as a synchronous stream cipher. After the first 16 octets, the stream cipher state integrates the previous ciphertext, and becomes more accurately described as a self-synchronizing stream cipher.

The security of the cipher rests on the strength of MD5 for this type of use and the selection of the shared secret. It is unclear what the requirements for this cipher are, so it is unclear if the MD5 function is appropriate for this use. MD5 is not designed to be a stream cipher primitive, it is designed to be a cryptographic hash. This sort of misuse of cryptographic primitives often leads to subtly flawed systems.

A.3 User-Password Attribute Based Shared Secret Attack

Because of the selection of a stream cipher for protection of the User-Password attribute, an attacker can gain information about the Shared Secret if they can observe network traffic and attempt an authentication. The attacker attempts to authenticate to the client with a known password. The attacker then captures the resulting Access-Request packet and XORs the protected portion of the User-Password attribute with the password they provided to the client. This results in the value of the MD5(Shared Secret + Request Authenticator) operation. The Request Authenticator is known (it is in the client's Access-Request packet), so the attacker can launch an off-line exhaustive attack on the shared secret. Note, though, that the attacker cannot pre-compute the MD5 state of the hash for the Request Authenticator, because the Request Authenticator is hashed second.

A.4 User-Password Based Password Attack

The use of a stream cipher to protect the User-Password attribute results in a vulnerability that allows an attacker to circumvent any authentication rate limits imposed by the client. The attacker first attempts to authenticate to the client using a valid username and a known (and likely incorrect) user password. The attacker then captures the resulting Access-Request packet and determines the result of the MD5(Shared Secret + Request Authenticator) operation (in the same way as in the previous attack). The attacker can then replay modified Access-Request packets, using the same Request Authenticator and MD5(Shared Secret + Request Authenticator) value, changing the password (and the associated User-Password attribute) for each replay. If the server does not impose user based rate limits, this will allow the attacker to efficiently perform an exhaustive search for the correct user password.

Note that the attacker can only use this method to attack passwords that are 16 characters or less, as the User-Password protection mechanism uses a chaining method that includes previous ciphertext in the state after the first 16 octets of output.

Any sort of strong data authentication in the Access-Request packet would make this attack impossible.

A.5 Request Authenticator Based Attacks

The security of RADIUS depends on the generation of the Request Authenticator field. The Request Authenticator must be both unique and non-predictable in order for the RADIUS implementation to be secure. The RADIUS protocol specification does not emphasize the importance of the Request Authenticator generation, so there are a large number of implementations that use poor PRNGs to generate the Request Authenticator. If the client uses a PRNG that repeats values (or has a short cycle), the protocol ceases to provide the intended level of protection.

The last two of these attacks require the attacker to cause the client to produce a particular identifier value. This is generally not particularly difficult, as identifiers were never meant as a security feature. The actual method of identifier

generation is not specified by the protocol specification, but the most common method of generating the identifier is to increment a one octet counter for each request, and include the counter value as the identifier. Because the identifier generation is normally deterministic, it often doesn't increase the work factor very much at all. An attacker can insert a series of extra requests to the client, forcing the desired identifier to reoccur much more rapidly than it would normally. Even if the identifier were not generated in a readily attackable way, it would still only increase the work factor by 256 times.

A.5.1 Passive User-Password Compromise Through Repeated Request Authenticators

If the attacker can sniff the traffic between the RADIUS client and the RADIUS server, they can passively produce a dictionary of Request Authenticators, and the associated (protected) User-Password attributes. If the attacker observes a repeated Request Authenticator, they can remove any influence of the Shared Secret from the first 16 octets of the passwords by XORing the first 16 octets of the protected passwords together. This yields the first 16 octets of the two (now unprotected) user passwords XORed together.

The impact of this attack varies according to how good the user passwords are. If the users all chose random passwords of the same length, the attacker can gain nothing because no information about either password can be extracted. Unfortunately, this is a somewhat unlikely occurrence. In reality, users choose passwords of varying lengths (generally less than 16 characters) and of varying quality.

The easiest problem for the attacker to exploit is the case where the two passwords are of different lengths. Ideally for the attacker, the passwords are both less than 16 characters long and are significantly different lengths. In this situation, one of the passwords has more padding than the other, so the non-overlapping characters of the longer password are XORed with '0' (the characters do not change). This results in the non-overlapping characters of the longer password being exposed to the attacker with no analysis.

More complex attacks are available if the attacker makes the assumption that the users chose low-entropy passwords. In this situation, the attacker can perform an intelligent dictionary attack guided by statistical analysis of the overlapping

region. This dictionary attack can be further refined by noting the length of the two passwords and the trailing portion of the longer password, and then only trying passwords with this length and ending.

Even passwords longer than 16 characters are at risk from this attack, because the attacker still gains information about the first 16 characters of the password. This provides a firm basis for later attack, if nothing else.

A.5.2 Active User-Password Compromise through Repeated Request Authenticators

The attacker can attempt to authenticate many times using known passwords and intercept the generated Access-Request packets, extracting the Request Authenticator and User-Password attributes. The Attacker can then XOR the known password with the User-Password attribute and be left with the MD5(Shared Secret + Request Authenticator) value. The attacker generates a dictionary of Request Authenticator values and associated MD5(Shared Secret + Request Authenticator) values.

When the attacker sees a valid Access-Request packet that has a Request Authenticator value that is in the attacker's dictionary, the attacker can recover the first 16 octets from the protected region of the User-Password field by looking up the associated MD5(Shared Secret + Request Authenticator) value from the dictionary and XORing it with the intercepted protected portion of the User-Password attribute.

A.5.3 Replay of Server Responses through Repeated Request Authenticators

The attacker can build a dictionary of Request Authenticators, identifiers and associated server responses. When the attacker then sees a request that uses a Request Authenticator (and associated identifier) that is in the dictionary, the attacker can masquerade as the server and replay the previously observed server response.

Further, if the attacker can attempt to authenticate, causing the client to produce an Access-Request packet with the same Request Authenticator and identifier as a previously observed successful authentication, the attacker can replay the valid looking Access-Accept server response and successfully authenticate to the client without knowing a valid password.

A.5.4 DOS Arising from the Prediction of the Request Authenticator

If the attacker can predict future values of the Request Authenticator, the attacker can pose as the client and create a dictionary of future Request Authenticator values (with either the expected identifier, or with every possible identifier) and associated (presumably Access-Reject) server responses. The attacker can then masquerade as the server and respond to the client's (possibly valid) requests with valid looking Access-Reject packets, creating a denial of service.

A.6 Shared Secret Hygiene

The RADIUS standard specifically permits use of the same Shared Secret by many clients. This is a very bad idea, as it provides attackers with more data to work from and allows any flawed client to compromise several machines. All RADIUS clients that possesses the same shared secret can be viewed as a single RADIUS client for the purpose of all these attacks, because no RADIUS protection is applied to the client or server address.

Most client and server implementations only allow shared secrets to be input as ASCII strings. There are only 94 different ASCII characters that can be entered from a standard US style keyboard (out of the 256 possible). Many implementations also restrict the total length of the shared secret to 16 characters or less. Both of these restrictions artificially reduce the size of the keyspace that an attacker must search in order to guess the shared secret.

Bibliography

[1] Joshua Hill. An Analysis of the RADIUS Authentication Protocol. http://www.untruth.org/~josh/security/radius/radius-auth.html 2001

[2] [RFC 3588] Diameter Base Protocol 2003

[3] Network Working Group. Criteria for Evaluating AAA Protocols for Network Access . http://www.scit.wlv.ac.uk/rfc/rfc29xx/RFC2989.html 2000

[4] Christian Schulze. Diameter http://www.ibr.cs.tu-bs.de/lehre/ws0203/skm/articles/schulze_diameter.pdf 2003

[5] Interlink Networks, Inc. Introduction to Diameter http://www.interlinknetworks.com/images/resource/Introduction_to_Diameter.pdf 2002

[6] TACACS (Terminal Access Controller Access Control System) http://www.linktionary.com/t/tacacs.html 2005

[7] webopedia http://www.webopedia.com/TERM/T/TACACS.html .2005

[8] http://www.linktionary.com/v/vpn.html 2005

[9] Microsoft Corporation Understanding Point-to-Point Tunneling Protocol (PPTP) http://msdn.microsoft.com/library/default.asp?url=/archive/en-us/dnarwebtool/html/understanding_pptp.asp 1997

[10] http://www.windowsecurity.com/pages/article_p.asp?id=1334 2004

[11] http://www.isaserver.org/pages/article_p.asp?id=1189 2004

[12] Jacco de Leeuw. Using a Linux L2TP/IPsec VPN server http://www.jacco2.dds.nl/networking/freeswan-l2tp.html 2005

[13] http://www.webopedia.com/TERM/I/IPsec.html 2005

[14] http://www.webopedia.com/TERM/P/PAP.html 2005

[15] http://www.webopedia.com/TERM/C/CHAP.html 2005

[16] http://www.faqs.org/rfcs/rfc2433.html 1998

[17] http://www.microsoft.com/technet/community/columns/cableguy/cg0702.mspx 2002

[18] http://www.webopedia.com/TERM/d/digital_certificate.html 2005

[19] http://www.webopedia.com/TERM/D/DES.html 2005

[20] http://www.webopedia.com/TERM/c/cryptography.html 2005

[21] http://www.linktionary.com/c/cryptography.html 2005

[22] http://www.webopedia.com/TERM/T/Triple_DES.html 2005

[23] http://www.webopedia.com/TERM/A/AES.html 2005

[24] http://www.webopedia.com/TERM/R/RSA.html 2005

[25] http://www.faqs.org/rfcs/rfc1321.html 1992

[26] http://www.faqs.org/rfcs/rfc1810.html 1995

[27] http://www.faqs.org/rfcs/rfc3174.html 2001

[28] Joshua Drake. Linux PPP HOWTO http://www.tldp.org/HOWTO/PPP-HOWTO/ 2000

[29] http://www.freeradius.org 2005

[30] Analysis of Microsoft PPTP. http://www.schneier.com/paper-pptp.html 1998

[31] Analysis of Microsoft PPTP Version 2. http://www.schneier.com/pptp.html 1999

[32] 2http://www.schneier.com/paper-pptpv2.html 1999

[33] NATTraversal. http://wiki.openswan.org/index.php/NATTraversal 2005

[34] White Paper IPSec. http://www.cisco.com/warp/public/cc/so/neso/sqso/eqso/ipsec_wp.htm 2000

[35] Charles P.Pfleeger, Shari Lawrence Pfleeger. Security in Computing Third Edition .2003

[36] Stefaan Pouseele How to pass IPSec traffic through ISA Server. www.isaserver.org/articles/ IPSec_Passthrough.html 2003

[37] An Introduction to Virtual Private Networks. http://users.cs.dal.ca/˜qiufen/pdfs/4171.pdf 2005

[38] Andrew Mason. IPSec Overview Part Four. http://www.ciscopress.com/articles/article.asp?p=25474&seqNum=7&rl=1 2002

[39] Juniper Networks L2TP/IPSec Tunnels. http://www.juniper.net/techpubs/software/erx/junose53/swconfig-routing-vol1/html/ l2tp-over-ipsec-config4.html#1028082 2005

[40] NISCC Vulnerability Advisory IPSEC - 004033. http://www.niscc.gov.uk/niscc/docs/al-20050509-00386.html?lang=en 2005

[41] Vlastimil Klima. Finding MD5 Collisions ¨C a Toy For a Notebook. http://cryptography.hyperlink.cz/md5/MD5_collisions.pdf 2005

[42] Xiaoyun Wang. Finding Collisions in the Full SHA-1 http://www.infosec.sdu.edu.cn/paper/sha1-crypto-auth-new-2-yao.pdf 2005

[43] Configuring an ipsec tunnel with openswan and l2tpd http://www.natecarlson.com/linux/ipsec-l2tp.php

[44] Niels Ferguson and Bruce Schneier. A Cryptographic Evaluation of IPsec http://www.schneier.com/paper-ipsec.pdf 2005