Wireless VPN: IPSec vs SSL/TLS

Åsa Pehrsson – asap@kth.se

1 Abstract

This report is written as part of the assignment given in the course 2G1330 Wireless and Mobile Network Architectures. The aim of the assignment is by practical experience gain deeper knowledge within a chosen topic. The topic is an end-to-end Virtual Private Network (VPN) deployment scenario over the GPRS network and a Wireless Internet Service Provider (WISP) 802.11 based network. Two VPN solutions have been compared, IPSec and SSL/TLS based software.

2 Introduction

A virtual private network is a way to use a public communication infrastructure to provide remote sites or individual users with secure access to their organization's private network. In order to extend that concept to the Internet, solutions such as IPSec and SSL/TLS have evolved.

The secure connection can consist of two types of end points, either an individual computer or a LAN with a security gateway. Traditionally the LAN-to-LAN connection, where a security gateway at each end point with known IP addresses serves as the interface between the secure connection and the private LAN, was the most used. Today, when telecommuting and mobile device such as a laptop or PDA are common, another case must also be considered: clients using individual computers with dynamic IP addresses connected to the VPN gateway. This type of client is also referred to as a road warrior.

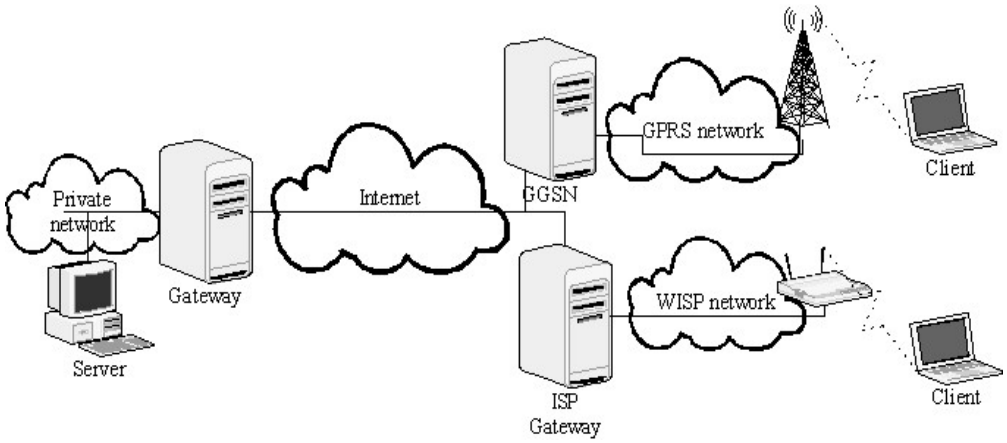

Figure 1. Wireless VPN

The scenario we will study is a road warrior that uses wireless access, either by using a Wireless LAN (WLAN) provided by a Wireless Internet Service Provider (WISP) or General Packet Radio Service (GPRS). In this scenario, the user does not move around when using the Internet access thus no handovers between base stations or between networks are considered.

The rest of this document describe an experiment. We begin with an overview of the GPRS and WISP/WLAN architectures. Next, we go through the security tools IPSec and SSL/TLS. Finally we present the findings from deploying the security tools in a GPRS and WISP/WLAN environment, and finish with some conclusions.

3 GPRS

General Packet Radio Service (GPRS) is a standard for wireless communications. It is a packet-switching technology for GSM networks, where air interface resources (time slots) are shared among users (including circuit-switched voice). GPRS runs at speeds from 9.05 to 171.2 kilobits per second. The bandwidth may be asymmetrical. Charging is based on usage rather than on the duration of the connection. Figure 1 shows this architecture.

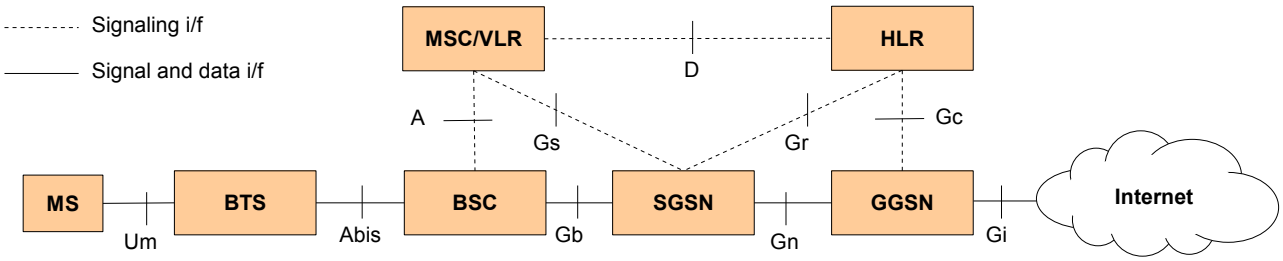

Figure 2. GPRS Architechture

The GPRS network introduces two new nodes to the GSM infrastructure: Serving GPRS Support Node (SGSN) and Gateway GPRS Support Node (GGSN). The connection between different GSN nodes and other components of the core network is called the GPRS backbone. The backbone is a regular IP network.

The SGSN routes incoming and outgoing IP packets addressed to or from a GPRS subscriber via the radio network. Each SGSN is responsible for:

- ciphering (encryption and decryption) and authentication
- session management and communication set-up to the mobile subscriber
- roaming and handover within and between mobile networks
- logical link management to the mobile subscriber
- connection to other nodes (HLR, MSC, BSC,GGSN).

The GGSN serves as the interface to external IP packet networks. The GGSN routes the IP addresses of subscribers served by the GPRS network, exchanging routing information with the external network. A GGSN is responsible for:

- access to external ISP functions such as routers and RADIUS servers.
- setting up communication and managing GPRS sessions.
- associating subscribers with the appropriate SGSN.
- collecting charging data based on use of the external data network and use of GPRS network resources.

The SGSN and GGSN can either be co-located or placed far from each other and connected via the backbone. Because the backbone can be shared with other operators and with others, a tunneling protocol called the GPRS Tunneling Protocol (GTP) is used.

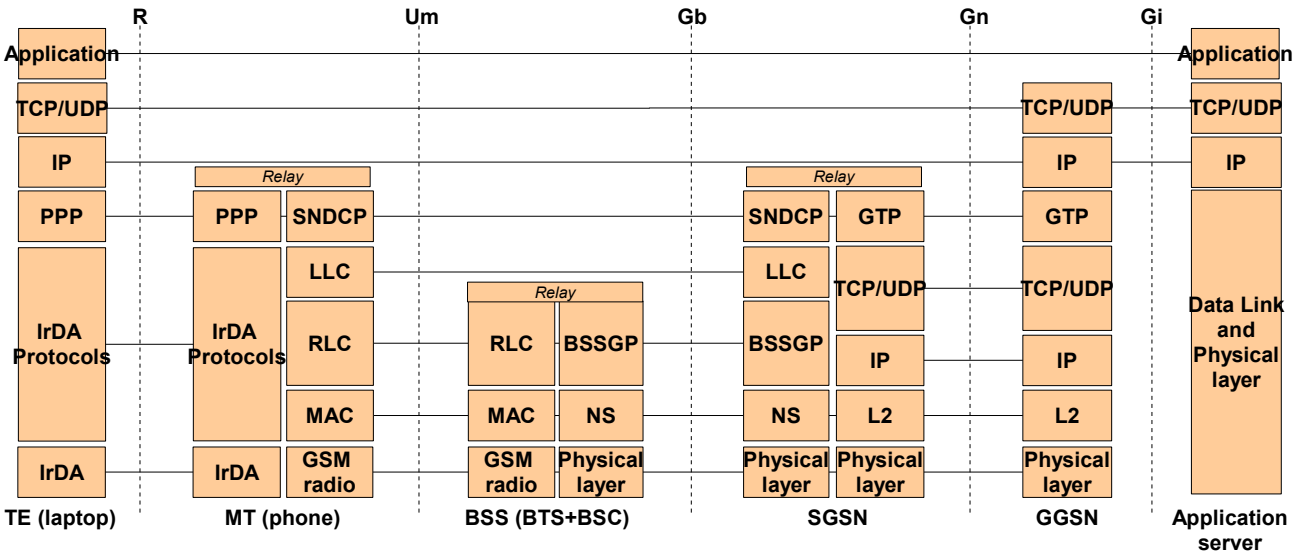

Figure 3.GPRS Data Plane Protocol Stack

Um is located between MS and the GPRS fixed network part. The Um radio interface gives MS access to the GPRS network.

Gb interface carries both payload and signaling between the BSS and the SGSN. Flow control is based on Frame Relay.

The Gn interface connects the SGSN and the GGSN. GPRS tunneling protocol (GTP) tunnels the data and signaling between the SGSN and the GGSN. TCP carries packet data in the GPRS backbone network for protocols that need reliable data link for, e.g., X.25 protocol.

Gi resides between the GGSN and an external network. The external data network is connected to GPRS network via the Gi interface.

The GPRS Tunneling Protocol (GTP) encapsulates higher layer Protocol Data Units (PDUs) and tunnels user data and signaling information between the GPRS support nodes. It also provides flow control between the GSNs. It provides logical connectivity, but it does not ensures security.

The TCP and UDP protocols are situated below the GTP protocol. The TCP provides reliable transmission of the GTP PDU whereas the UDP is used for unreliable GTP PDU transmission.

The Subnetwork Dependent Convergence Protocol (SNDCP) handles the PDU transmission between the SGSN and the MS. It maps network layer characteristics into characteristics of the underlaying logical link. The SNDCP performs also multiplexing of multiple layer 3 messages into a single logical link connection. In addition, ciphering, segmentation and compression are performed by SNDCP.

The Logical Link Layer (LLC) provides a highly reliable and secure logical link between the MS and the SGSN. The reliability of packet transmission is provided by retransmission of the packet in acknowledged mode. In unacknowledged mode retransmission is not used.

LLC ensures security from MT to SGSN, then the connection is unprotected. We have VPN solutions, where the connection is terminated in the operator network, and then connected to the company over a VPN. The solution is not end-to-end, and of course the operator needs to be trusted.

4 WLAN and WISP

Wireless Local Area Networks (WLAN) is a network that uses high-frequency radio waves rather than wires to communicate between nodes. A Wireless Internet Service Provider (WISP) is an operator that offers WLAN access to the Internet.

IEEE 802.11 is the standard for the family of wireless network protocols. It deals with the Medium Access Control (MAC) layer and the Physical Layer (PHY). The table below shows a comparison of the different 802.11 standards.

IEEE 802.11a	Using OFDM (Orthogonal Frequency Division Multiplexing) achieves up to 54 Mbps. Operates in 5 GHz band
IEEE 802.11b	One of the most used protocols in this family. Operates at the 2.4GHz unlicensed frequency band. 1, 2, 5.5 and 11 Mbps
IEEE 802.11g	Enable data transmission speeds of up to 54 Mbps, with backwards compatibility to 802.11b infrastructure. Operates in 2.4GHz band
IEEE 802.11h	Designed to adapt 802.11a to the European HiperLAN/2 requirements; operates in 5 GHz band

Table 1 Comparison of 802.11 standards

There are two modes of operation in 802.11 networks, ad-hoc and infrastructure. The ad-hoc mode is used when stations communicate directly with each other. Infrastructure mode is used when building networks. In this mode stations communicate via an associated access point.

The IEEE 802.11 architecture consists of several components and concepts that is used when building a infrastructure network. Figure 4 shows the architecture for a fictive WISP as an example of an 802.11 infrastructure network with Internet access.

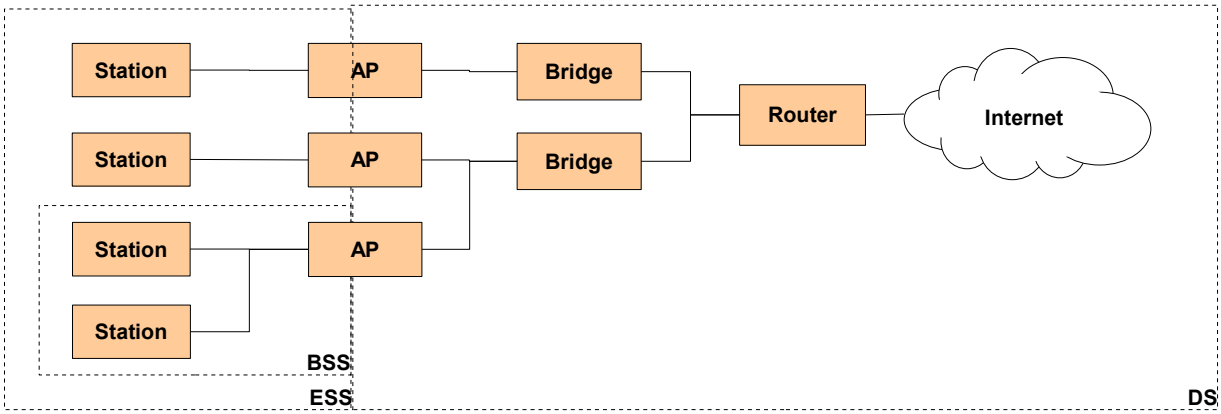

Figure 4. WLAN Architechture for a fictive WISP

The access point (AP) is an entity that provides access to the distribution services via the wireless interface for associated stations. Basic service set (BSS) is a set of stations controlled by a single coordination function. Coordination function is the logical function that determines when a station operating within a basic service set (BSS) is permitted to transmit and may be able to receive protocol data units via the wireless medium. An extended service set (ESS) is one or more interconnected basic service sets (BSSs) The distribution system (DS) is used to interconnect one or several BSSs and a LAN

In wireless networks, authentication and association needs to be considered. Before a station may communicate with an AP, it must first authenticate itself and must then send an association request. There are two modes of authentication: open system and shared key. Open system is a null authentication algorithm in which anyone may associate. Shared key authentication, on the other hand, is achieved by using the Wired Equivalent Privacy (WEP) algorithm, which requires prior agreement of a shared secret WEP key. After a successful authentication request and association response, the client is in the associated state and may start sending data.

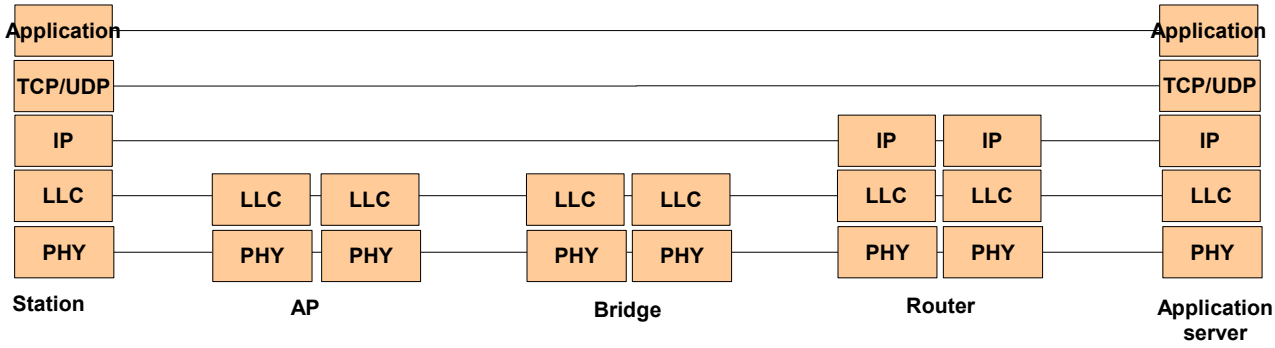

Figure 5. WISP protocol stack example

In the protocol stack example illustrated above one bridge and one router is pictured. In a real deployment, there will probably be several. Without a VPN, the entire channel from the Station to the Application server is unprotected.

5 SSL/TLS

The Secure Sockets Layer (SSL) is a protocol designed by Netscape Communications to enables secure data transfer between two devices over a public network. SSL protects applications running over TCP, and is mostly utilized to protect HTTP transactions. It can be used for other applications like IMAP, POP etc. SSL has been replaced by Transport Layer Security (TLS).

SSL provides authentication, confidentiality, and data integrity through the use of encryption. The protocol has two primary functions: privacy between client and server, and to authenticate the server to the client. SSL can accommodate a variety of algorithms for key agreement (RSA, DH, etc) encryption (RC4, 3DES etc) and hashing (MD5, SHA, etc). The specification explicit lists combinations of these algorithms, called cipher suites.

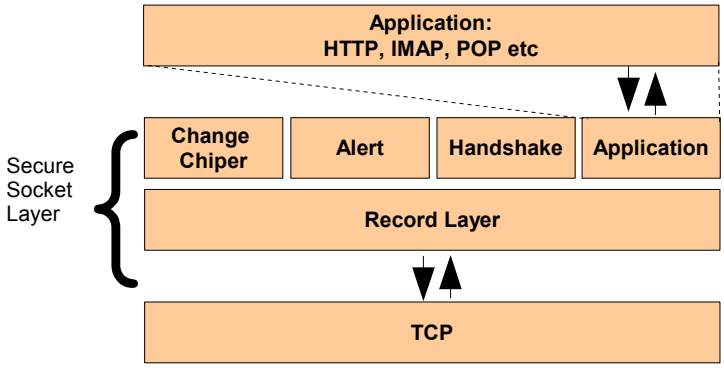

Figure 6. SSL Architechture

As shown in the figure above SSL is a layered protocol. These layers are:

- Change Cipher Spec protocol - indicate that the chosen keys will now be used
- Alert protocol - used for signaling errors and session closure
- Handshake protocol - performs authentication and exchange keys
- Application Data protocol - transmits and receives application data
- Record layer – provides encryption and authentications services.

SSL consists of two phases: handshake and data transfer. During the handshake phase, the client and server use a public-key encryption algorithm to determine secret-key parameters. During the data transfer phase, both sides use the secret key to encrypt and decrypt successive data transmissions.

One approach when providing SSL based VPN is to presented the user with a centralized Web interface in which all available servers/services hosted by the VPN server are listed as hyperlinks. Another approach is to use a virtual point-to-point network interface, in that way all traffic between the client and the server will be protected.

Khanvilkar and Khokhar [1] have evaluated open source VPN software. They have identified two SSL based product that support the basic security properties (confidentiality, data integrity, authentication/non repudiation and anti replay protection). These products were OpenVPN [2] and tinc [3].

For wireless communication, V. Gupta and S. Gupta have showed in their study [4] that SSL is a practical solution for ensuring end-to-end security of wireless connections.

6 IPSec

IP Security (IPSec) is a set of open standards developed by the IETF and documented in Internet general Request-For-Comments (RFC 2401) and related RFCs. It provides for encryption and authentication at the network layer to protect IP packets between IPSec compliant devices.

IPSec allows the sender (or a security gateway acting on his behalf) to authenticate or encrypt each IP packet or apply both operations to the packet. Separating the application of packet authentication and encryption has led to two different methods of using IPSec, called modes. In transport mode, only the transport-layer segment of an IP packet is authenticated or encrypted. The other approach, encrypting the entire IP packet, is called tunnel mode.

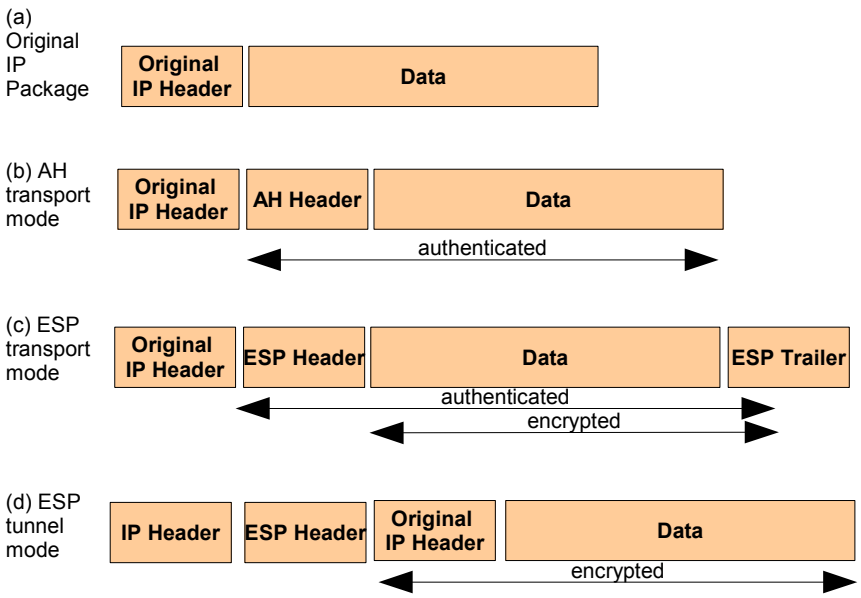

Figure 7. IPSec modes

Encapsulating Security Payload (ESP) is the IPSec protocol that provides confidentiality, data integrity, and data source authentication of IP packets, and also provides protection against replay attacks. It inserts a header after an IP header and before the data to be protected and appends an ESP trailer.

Authentication Header (AH) provides data integrity, data source authentication, and protection against replay attacks. It does not provide confidentiality.

IPSec is built around a number of standardized cryptographic technologies to provide confidentiality, data integrity, and authentication. For example, IPSec uses:

- Diffie-Hellman key exchanges to deliver secret keys between peers on a public net
- public-key cryptography for signing Diffie-Hellman exchanges, to guarantee the identities of the two parties and avoid man-in-the-middle attacks
- data encryption standard (DES) and other bulk encryption algorithms for encrypting data
- keyed hash algorithms (HMAC, MD5, SHA) for authenticating packets
- digital certificates for validating public keys

The Security Association (SA) is a contract between two parties. The SA determine the IPSec protocols used for securing the packets, the transforms, the keys, and the duration for which the keys are valid to name a few. The SPI is a very important element in the SA. It used to uniquely identify an SA at the receiver.

In an environment where IPSec is deployed, the SAs are created through an Internet standard key management protocol such as IKE. With the automatic key exchange (IKE) it is possible create and manage the session keys automatically by the system. The mostly used authentication methods are the preshared key (also known as shared secret) and x.509 certificates.

The use of NAT in conjunction with IPSec could cause incompatibilities issues, since IPSec either hides the private address through encryption (and thus let them escape translation), or it will detect integrity violations as a consequence of NAT manipulating the IP address. However, there are NAT routers that are IPSec aware.

Xenakis, Gazis, and Merakos pinpointed in [5] that the GGSN in the GPRS network may apply NAT.

7 Public Key Infrastructure

Public Key Infrastructure (PKI) is a policy for establishing a secure method for exchanging information within an organization. It includes the cryptographic methods, the use of digital certificates and certification authorities (CA).

X.509 is an ITU-T standard for public key infrastructure. X.509 certificates are based on public/private key pairs. Each certificate contains a public key, along with other information (identifying and not), such as the owner's common name and the certificate's expiration date. The owner keeps her private key in a separate file.

The certificates are signed by a CA, and contain information as to which authority signed it. This digitally proves that the certificate is authentic and that the information contained within it is accurate. The CA's authenticity is verified by its certificate, which is generally available to the public.

8 Experiment

8.1 Strategy

The goal of this study was to compare IPSec vs SSL/TLS based end-to-end VPNs when using GPRS or WLAN as carrier. Factors such as installation and configuration complexity and performance has been taken into account.

8.2 Local network

The local network configuration is used as a base for initial installation and configuration of IPSec and OpenVPN. It is also used when comparing installation and configuration complexity. A very simple performance test is also done.

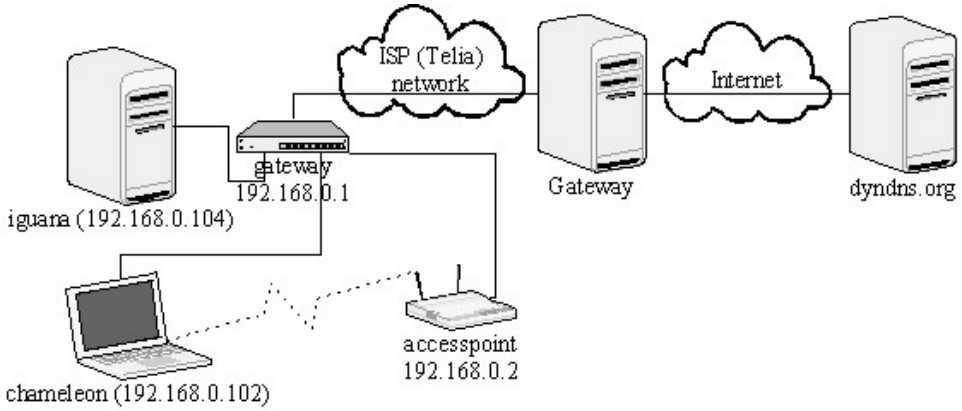

Figure 8. Local network testbench

The local network has the following configuration:

- access point: Cisco Aironet 340.
- chameleon: Laptop with Microsoft's Windows XP SP2. 512M memory
- iguana: a desktop with Linux, Fedora Core 3, Linux 2.6.11-1.14. 256M memory
- gateway: D-Link DI-704.

To get external access to the server iguana, we use a dynamic DNS (DDNS). It allows us to alias a dynamic IP address to a static host name. The provider DynDNS has a free DDNS service. We choose the name chikchak.homeip.net, and then installed an update client at the Linux host.

8.2.1 IPSec transport mode with x.509 certificate and a road warrior client

The current release of the Linux kernel (2.6) has built-in IPSec support. Hence we installed Fedora Core 3, and follow the setup guidelines for IPSec provided at http://www.ipsec-howto.org/.

IPSec support is part of Microsoft Window's XP Professional and there is two ways to set up IPSec policies. Either with a command line utility 'ipseccmd.exe', or with IP Security Policies snap-in from the Microsoft Management Console. We used IP Security Policies snap-in from the Microsoft Management Console – it is rather messy but all the needed configuration options are available.

The process to get IPSec up and running was very cumbersome and time consuming.

Server configuration

We will use two software packages at the Linux machine:

- Setkey, a tool to manipulate and dump the kernel Security Policy Database (SPD) and Security Association Database (SAD).

- Racoon, a Internet Key Exchange (IKE) daemon for automatically keying IPsec connections.
- Openssl, an open source implementation of the SSL and TLS protocols that can be used to create a local Certificate Authority (CA)

We start by creating a local Certificate Authority - meaning that we can create, sign, and distribute certificates. While others may not recognize us as a CA, at least we will recognize ourself as one.

We begin with creating the certificate authority.

```
[root@localhost certs]# /usr/share/ssl/misc/CA -newca
CA certificate filename (or enter to create) [enter]
Making CA certificate ...
Generating a 1024 bit RSA private key
.....++++++
.......++++++
writing new private key to './demoCA/private/./cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
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) [GB]:SE
State or Province Name (full name) [Berkshire]:n/a
Locality Name (eg, city) [Newbury]:Stockholm
Organization Name (eg, company) [My Company Ltd]:KTH
Organizational Unit Name (eg, section) []:SEDS
Common Name (eg, your name or your server's hostname) []:Asa Pehrsson
Email Address []:asap@kth.se
```

By default, the created certificate authority is only valid for one year. Even if this will be enough for the purpose of this experiment, we change the lifetime manually to ten years:

```
[root@localhost certs]# cd demoCA/
[root@localhost demoCA]# openssl x509 -in cacert.pem -days 3650 -out cacert.pem -signkey
./private/cakey.pem
Getting Private key
Enter pass phrase for ./private/cakey.pem:
```

The certificate authority is now ready. Next step is to create a certificate signing request:

```
[root@localhost demoCA]# /usr/share/ssl/misc/CA -newreq
Generating a 1024 bit RSA private key
.....................++++++
..............++++++
writing new private key to 'newreq.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
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) [GB]:SE
State or Province Name (full name) [Berkshire]:n/a
Locality Name (eg, city) [Newbury]:Stockholm
Organization Name (eg, company) [My Company Ltd]:KTH
Organizational Unit Name (eg, section) []:2G1330
Common Name (eg, your name or your server's hostname) []:Asa Pehrsson
Email Address []:asap@kth.se

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Request (and private key) is in newreq.pem
[root@localhost demoCA]#
```

The file `newreq.pem` contains the certificate signing request and the encrypted private key. This file will be

used as a private key for our server (iguana). Once the request is created, we can sign it using the certificate authority.

```
[root@localhost certs]# /usr/share/ssl/misc/CA -sign
Using configuration from /usr/share/ssl/openssl.cnf
Enter pass phrase for ./demoCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: May  8 12:56:27 2005 GMT
            Not After : May  8 12:56:27 2006 GMT
        Subject:
            countryName               = SE
            stateOrProvinceName       = n/a
            localityName              = Stockholm
            organizationName          = KTH
            organizationalUnitName    = 2G1330
            commonName                = Asa Pehrsson
            emailAddress              = asap@kth.se
        X509v3 extensions:
            X509v3 Basic Constraints:
            CA:FALSE
            Netscape Comment:
            OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
            BD:A0:63:03:24:1E:81:DD:8D:27:3F:D4:36:FC:8F:4E:6E:E8:AB:55
            X509v3 Authority Key Identifier:
            keyid:31:8E:86:99:AF:92:AD:8E:24:EA:A4:9B:43:34:10:75:4A:5F:B7:06
            DirName:/C=SE/ST=n/a/L=Stockholm/O=KTH/OU=SEDS/CN=\xC3Asa
Pehrsson/emailAddress=asap@kth.se
            serial:00

Certificate is to be certified until May  8 12:56:27 2006 GMT (365 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=SE, ST=n/a, L=Stockholm, O=KTH, OU=SEDS, CN=\xC3Asa
Pehrsson/emailAddress=asap@kth.se
        Validity
            Not Before: May  8 12:56:27 2005 GMT
            Not After : May  8 12:56:27 2006 GMT
        Subject: C=SE, ST=n/a, L=Stockholm, O=KTH, OU=2G1330, CN=Asa
Pehrsson/emailAddress=asap@kth.se
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:d6:6c:ae:99:c0:35:10:54:3a:67:26:ec:c1:72:
[...]
                    ee:9d:f3:16:a9:53:e3:95:ff:20:aa:2e:50:6f:b7:
                    a7:54:d7:fd:9b:59:de:8a:e3
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
            CA:FALSE
            Netscape Comment:
            OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
            BD:A0:63:03:24:1E:81:DD:8D:27:3F:D4:36:FC:8F:4E:6E:E8:AB:55
            X509v3 Authority Key Identifier:
            keyid:31:8E:86:99:AF:92:AD:8E:24:EA:A4:9B:43:34:10:75:4A:5F:B7:06
            DirName:/C=SE/ST=n/a/L=Stockholm/O=KTH/OU=SEDS/CN=\xC3Asa
Pehrsson/emailAddress=asap@kth.se
            serial:00

    Signature Algorithm: md5WithRSAEncryption
        ae:46:bf:16:88:e9:60:6d:3a:f1:50:d2:26:18:e1:6a:8a:52:
[...]
        3e:4c:bf:d1:91:ad:4a:35:a3:7d:43:07:b9:c9:72:8b:da:a5:
        dc:10
-----BEGIN CERTIFICATE-----
MIIDhzCCAvCgAwIBAgIBATANBgkqhkiG9w0BAQQFADCBgDELMAkGA1UEBhMCU0Ux
[...]
```

```
acVeuM19hVodToHsu7errug3TcxP2To+TL/Rka1KNaN9Qwe5yXKL2qXcEA==
-----END CERTIFICATE-----
Signed certificate is in newcert.pem
[root@localhost certs]#
```

We need to tell racoon about our CA certificate (cacert.pem). This file is copied to the certificates directory, then the following command is entered:

```
[root@localhost ]# ln -s cacert.pem `openssl x509 -noout -hash -in cacert.pem`.0
```

Road warriors are clients using unknown dynamic IP addresses. In combination with racoon this poses two problems: The IP address is not known and cannot be specified in the racoon configuration file. No security policy can be created for racoon to act on, since the destination IP address is not known. Therefore racoon must create the security policy and the security association when the connection is initiated.

Below is the configuration file racoon.conf:

```
path include "/etc/racoon";
path certificate "/root/ipsec/racoon/certs";

remote anonymous {
       exchange_mode main;
       generate_policy on;
       passive on;
       lifetime time 24 hours;
       verify_cert on;
       verify_identifier on;
       peers_identifier asn1dn;
       my_identifier asn1dn;
       certificate_type x509 "newcert.pem" "privkey.pem";

        proposal {
                encryption_algorithm 3des;
                hash_algorithm md5;
                authentication_method rsasig;
                dh_group 2;
        }
}
sainfo anonymous
{
       pfs_group 2;
       lifetime time 24 hour ;
       encryption_algorithm 3des, blowfish 448, rijndael ;
       authentication_algorithm hmac_sha1, hmac_md5 ;
       compression_algorithm deflate ;
}
```

The option `generate_policy` instructs racoon to create the appropriate policy when a new connection is initiated. The option `passive` tells racoon to remain passive and wait for new connection to be started from the outside

The most important is the definition of `anonymous` in the remote and sainfo line. This instructs racoon to accept a connection from anywhere.

Client Configuration

A certificate request is created and signed using the certificate authority as described in the previous chapter. Then it is convert to a p12 format:

```
[root@localhost certs]#  openssl pkcs12 -export -in privkey.pem -inkey privkey.pem -certfile
cacert.pem -out export.p12
```

We configure IPSec and handle certificates with Microsoft Management Console. This tool is started by “Click Start, click Run, type MMC, and then click OK”.

First, we import the previously generated certificate. Select 'Åtgärd' -> 'Alla aktiviteter' -> 'Importera'. A wizard is opened for importing certificates. Select "Automatically select the certificate store".

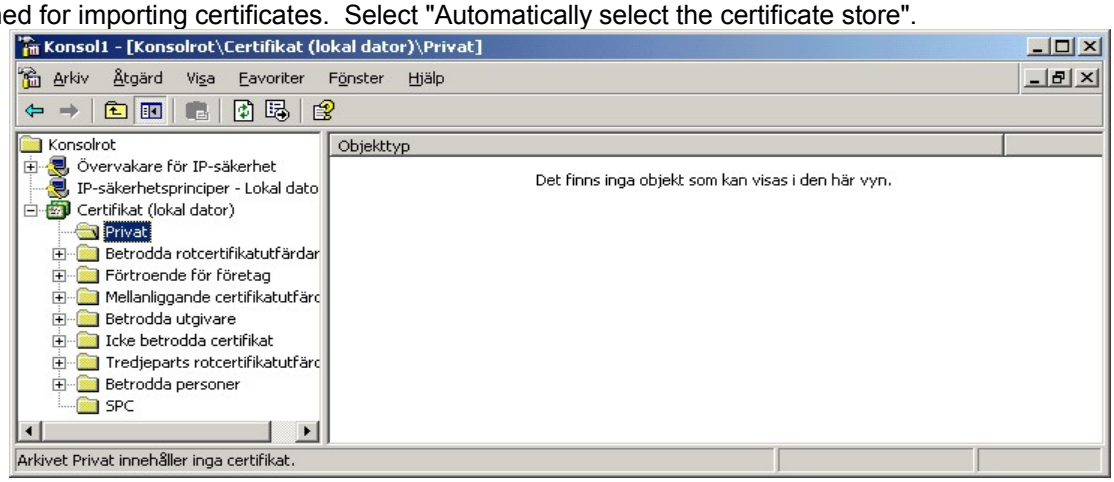

Figure 9. Microsoft Management Console

Next step is to configure Ipsec, IP Security Policies snap-in from the Microsoft Management Console.

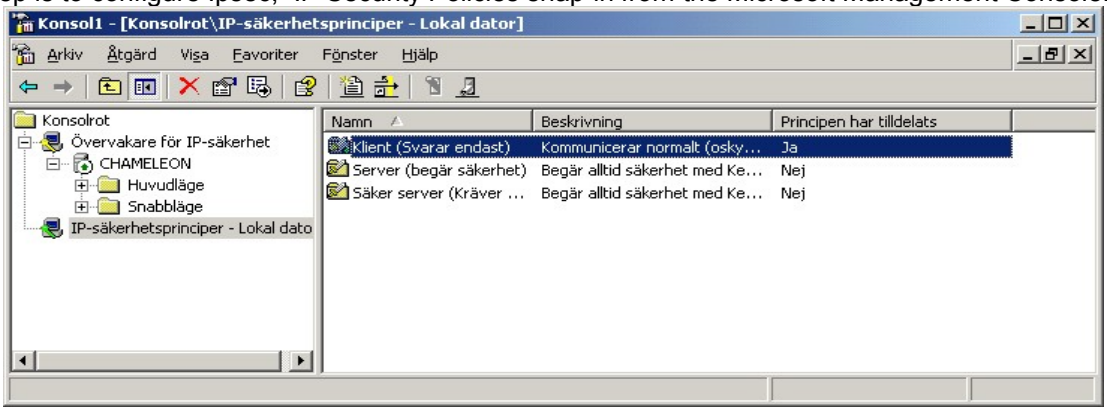

Figure 10. IP Security Policies snap-in

The following screen shots show the settings for 'Klient princip'.

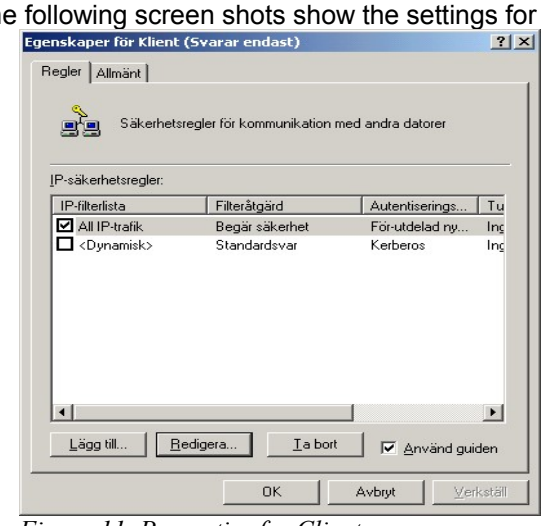

Figure 11. Properties for Client

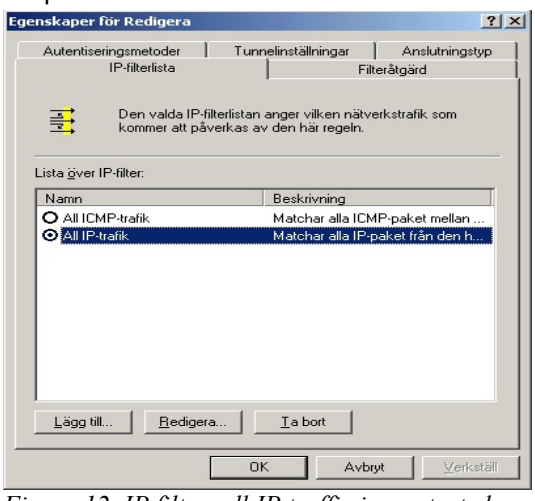

Figure 12. IP filter, all IP traffic is protected

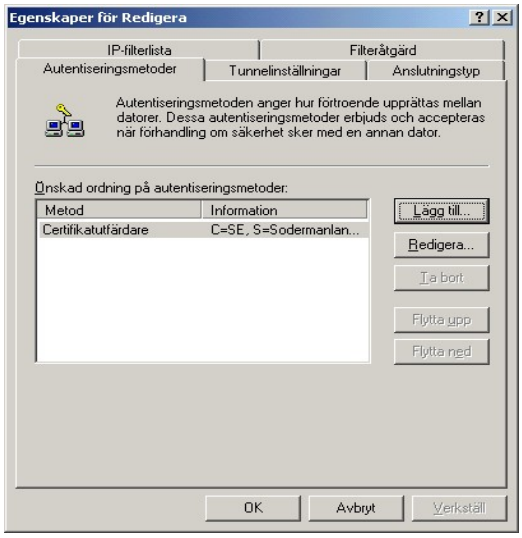

Figure 13. Authentication with a certificate

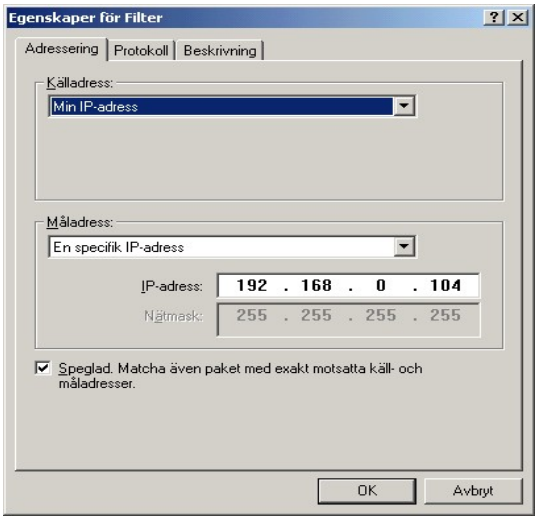

Figure 14. The IPSec policy is applied to 192.168.0.104.

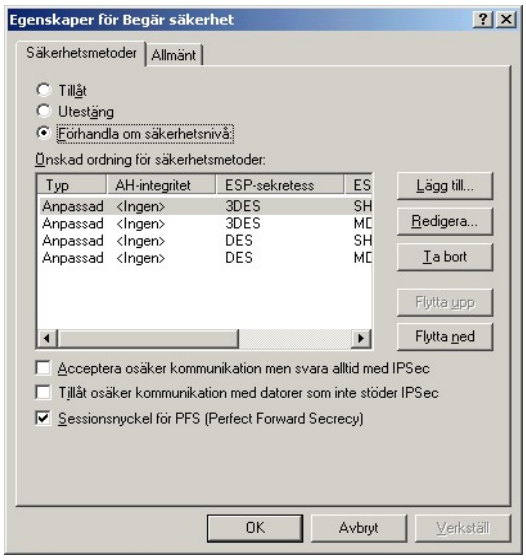

Figure 15. Rules used for negotiating ISAMK

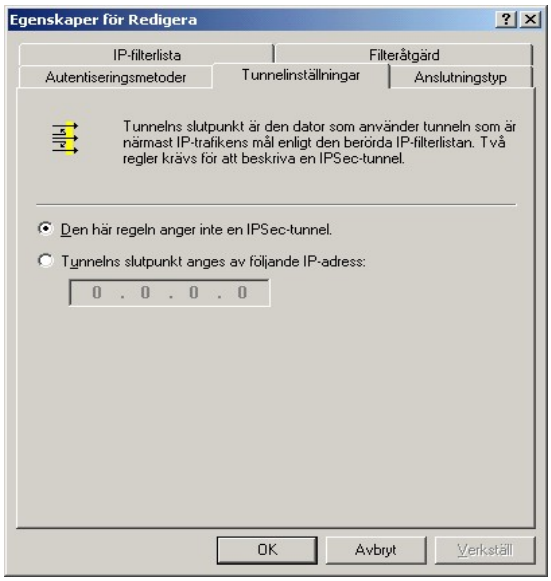

Figure 16. Transport mode is used.

Test

Since IGMP uses port 500 for both client and server, we realized that it was not possible to address the server as 'chikchak.homeip.net'. We will test the secure connection simply by 'pinging' iguana with the local network address. The result was successful, see below:

```
>ping 192.168.0.104
Skickar signaler till 192.168.0.104 med 32 byte data:
IP-säkerhet förhandlas.
IP-säkerhet förhandlas.
Svar från 192.168.0.104: byte=32 tid < 1 ms TTL=64
Svar från 192.168.0.104: byte=32 tid < 1 ms TTL=64
Ping-statistik för 192.168.0.104:
    Paket: Skickade = 4, mottagna = 2, Förlorade = 2 (50 %),
Ungefärligt överföringstid i millisekunder:
    Lägsta = 0 ms, Högsta = 0 ms, Medel = 0 ms
```

Below is the capture from ethereal (running at 192.168.0.104):

```
No.Time        Source                Destination           Protocol Info
 1 0.000000    AcerTech_8f:be:a3     Broadcast             ARP      Who has 192.168.0.104?  Tell
192.168.0.102
 2 0.000172    Elitegro_a1:2c:88     AcerTech_8f:be:a3     ARP      192.168.0.104 is at
00:0a:e6:a1:2c:88
 3 0.000041    192.168.0.102         192.168.0.104         ISAKMP   Identity Protection (Main Mode)
 4 0.376183    192.168.0.104         192.168.0.102         ISAKMP   Identity Protection (Main Mode)
 5 0.495730    192.168.0.102         192.168.0.104         ISAKMP   Identity Protection (Main Mode)
 6 0.528527    192.168.0.104         192.168.0.102         ISAKMP   Identity Protection (Main Mode)
```

```
 7 0.555187    192.168.0.102         192.168.0.104         ISAKMP   Identity Protection (Main Mode)
 8 0.628755    192.168.0.104         192.168.0.102         ISAKMP   Identity Protection (Main Mode)
 9 0.641630    192.168.0.104         192.168.0.102         ISAKMP   Informational
10 0.703166    192.168.0.102         192.168.0.104         ISAKMP   Quick Mode
11 0.979995    192.168.0.104         192.168.0.102         ISAKMP   Quick Mode
12 1.006990    192.168.0.102         192.168.0.104         ESP      ESP (SPI=0x05c7a049)
13 1.007571    192.168.0.102         192.168.0.104         ISAKMP   Quick Mode
14 1.962714    192.168.0.102         192.168.0.104         ESP      ESP (SPI=0x05c7a049)
15 1.962990    192.168.0.104         192.168.0.102         ESP      ESP (SPI=0x279a4ab0)
16 2.964317    192.168.0.102         192.168.0.104         ESP      ESP (SPI=0x05c7a049)
17 2.964526    192.168.0.104         192.168.0.102         ESP      ESP (SPI=0x279a4ab0)
```

8.2.2 OpenVPN with x.509 certificate and a roadwarrior client

OpenVPN is an open source SSL/TLS-based VPN solution. It can tunnel both Ethernet connections using a "tap" interface (a virtual Ethernet network device) or IP. UDP or TCP port can be used for the connections. No special kernel modules or modifications are necessary, OpenVPN runs purely in user space. All encryption operations are handled by OpenSSL. There is two ways of authenticating the remote connection: a private shared key, or public key encryption with x.509 certificates. We will use x.509 certificates.

The installation and configuration of OpenVPN went smoothly.

Server Configuration

We installed OpenVPN from a binary RPM package. The RPM distribution is dependent on openssl, lzo and pam. Lzo was missing, but easily installed.

The next step was to set up your own Certificate Authority (CA) and to generate certificates and keys for an OpenVPN server and one client. The CA and certificates that was generated earlier was used.

Finally the configuration file was updated with the name of the certificate and key files. The entire configuration file, server.conf is below:

```
# Which TCP/UDP port should OpenVPN listen on?
port 1194

# TCP or UDP server?
;proto tcp
proto udp

# "dev tun" will create a routed IP tunnel, "dev tap" will create an ethernet tunnel.
dev tun

# SSL/TLS root certificate (ca), certificate (cert), and private key (key)
ca /etc/openvpn/easy-rsa/keys/ca.crt
cert /etc/openvpn/easy-rsa/keys/server.crt
key /etc/openvpn/easy-rsa/keys/server.key

# Diffie hellman parameters.
dh /etc/openvpn/easy-rsa/keys/dh1024.pem

# Configure server mode and supply a VPN subnet for OpenVPN to draw client addresses from.
server 10.8.0.0 255.255.255.

# Maintain a record of client <-> virtual IP address associations in this file.
ifconfig-pool-persist ipp.txt

# The keepalive directive causes ping-like messages to be sent back and forth over the link
keepalive 10 120

# Enable compression on the VPN link.
comp-lzo

# The persist options will try to avoid accessing certain resources on restart
persist-key
persist-tun
```

Finally the server was started and the IP configuration examined (parts deliberately omitted):

```
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0A:E6:A1:2C:88
          inet addr:192.168.0.104  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20a:e6ff:fea1:2c88/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
[...]
```

```
tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.8.0.1  P-t-P:10.8.0.2  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
[...]

[root@localhost ~]#
```

Client Configuration

OpenVPN for Windows was installed from a self-installing .exe file. After the installation, the client configuration file was modified:

```
# Specify that we are a client
client

#
dev tun

# Connecting to a TCP or UDP server
proto udp

# The hostname/IP and port of the server.
remote 195.198.40.204 1194
;remote 192.168.0.104 1194

# Keep trying indefinitely to resolve the host name of the OpenVPN server.
resolv-retry infinite

# Most clients don't need to bind to a specific local port number.
nobind

# Try to preserve some state across restarts.
persist-key
persist-tun

# SSL/TLS parms.
ca ca.crt
cert client.crt
key client.key

# Enable compression on the VPN link.
comp-lzo
```

Finally the client was started and the ip configuration examined:

```
C:\DEV\2G1330>ipconfig

IP-konfiguration för Windows
[...]

Ethernet-kort Anslutning till lokalt nätverk:
        Anslutningsspecifika DNS-suffix . :
        Beskrivning . . . . . . . . . . . : Intel(R) PRO/100 VE Network Connect on
        Fysisk adress . . . . . . . . . . : 00-00-E2-8F-BE-A3
        DHCP aktiverat  . . . . . . . . . : Ja
        Autokonfiguration aktiverat . . . : Ja
        IP-adress . . . . . . . . . . . . : 192.168.0.102
        Nätmask . . . . . . . . . . . . . : 255.255.255.0
        Standard-gateway  . . . . . . . . : 192.168.0.1
        DHCP-server . . . . . . . . . . . : 192.168.0.1
        DNS-servrar . . . . . . . . . . . : 195.67.199.18
                                            195.67.199.19
[...]

Ethernet-kort Anslutning till lokalt nätverk 3:
        Anslutningsspecifika DNS-suffix . :
        Beskrivning . . . . . . . . . . . : TAP-Win32 Adapter V8
        Fysisk adress . . . . . . . . . . : 00-FF-92-01-F8-2F
        DHCP aktiverat  . . . . . . . . . : Ja
        Autokonfiguration aktiverat . . . : Ja
        IP-adress . . . . . . . . . . . . : 10.8.0.6
        Nätmask . . . . . . . . . . . . . : 255.255.255.252
        Standard-gateway  . . . . . . . . :
        DHCP-server . . . . . . . . . . . : 10.8.0.5
[...]

C:\DEV\2G1330>
```

Gateway Configuration

iguana (192.168.0.104) was set as the virtual host for port 1194.

Test

We test the secure connection simply by 'ping'. The result was successful. An interesting feature is that it is possible to use ethereal to listen to the traffic before it is encrypted - very useful for debugging. Below is the ethereal capture from the tap interface:

No.	Time	Source	Destination	Protocol	Info
1	0.000000	10.8.0.6	10.8.0.1	ICMP	Echo (ping) request
2	0.029012	10.8.0.1	10.8.0.6	ICMP	Echo (ping) reply
3	1.000028	10.8.0.6	10.8.0.1	ICMP	Echo (ping) request
4	1.047126	10.8.0.1	10.8.0.6	ICMP	Echo (ping) reply
5	2.000144	10.8.0.6	10.8.0.1	ICMP	Echo (ping) request
6	2.025385	10.8.0.1	10.8.0.6	ICMP	Echo (ping) reply
7	3.005559	10.8.0.6	10.8.0.1	ICMP	Echo (ping) request
8	3.025794	10.8.0.1	10.8.0.6	ICMP	Echo (ping) reply

Here is the corresponding capture from the Ethernet interface:

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.0.102	195.198.40.204	UDP	Source port: 2355 Destination port: 1194
2	0.020812	195.198.40.204	192.168.0.102	UDP	Source port: 1194 Destination port: 2355
3	1.000175	192.168.0.102	195.198.40.204	UDP	Source port: 2355 Destination port: 1194
4	1.019228	195.198.40.204	192.168.0.102	UDP	Source port: 1194 Destination port: 2355
5	2.000382	192.168.0.102	195.198.40.204	UDP	Source port: 2355 Destination port: 1194
6	2.020073	195.198.40.204	192.168.0.102	UDP	Source port: 1194 Destination port: 2355
7	3.002482	192.168.0.102	195.198.40.204	UDP	Source port: 2355 Destination port: 1194
8	3.022320	195.198.40.204	192.168.0.102	UDP	Source port: 1194 Destination port: 2355

Both traces are from the client machine.

8.2.3 Metrics

IPSec requires approximately 1.8 seconds and 4300 bytes for the initial ISAKMP messages.

The OpenVPN client requires approximately 1.4 seconds and 12300 bytes when it is started.

To get an feeling on the increased bandwidth need when using a VPN, a simple test was performed. Three files of size 100k, 500k, and 1M byte(s) was transmitted using FTP. Each file was transmitted several times, and the average number of bytes sent on the wire was counted. OpenVPN require 10% and IPSec approx. 3% more bandwidth compared with an unprotected connection.

8.3 GPRS

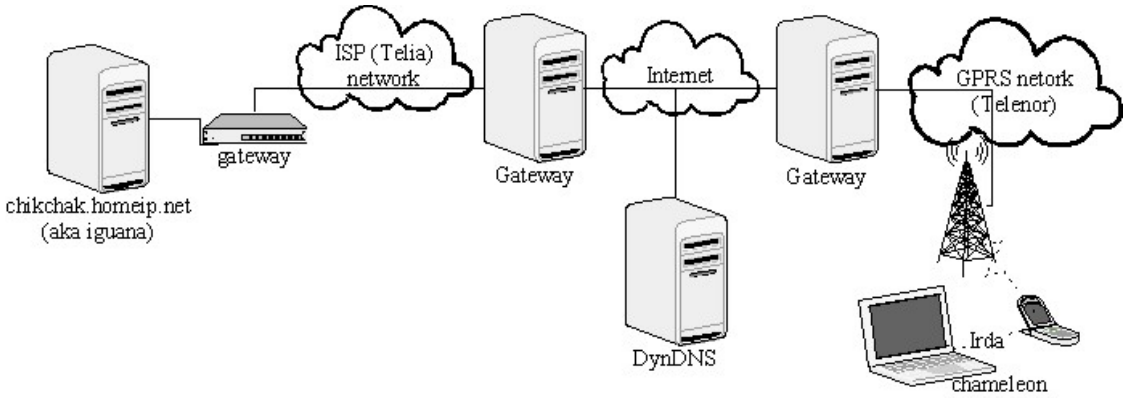

Figure 17. GPRS testbench

In addition to the devices used in the home network configuration, we used a SonyEricsson T300 phone. GPRS provider: Telenor.

8.3.1 Client Configuration

In the test bench, the laptop is connected to the phone via IR. To make this work, the phone is used as a

modem. The problem is that the modem that gets installed if placing the phone at the IR transceiver is a standard IR modem. The solution was to download modem scripts for the phone from SonyEricssons web site.

The IR was enabled at the phone, and the phone was placed in front of the computer. The “Found new hardware” wizard started, and modem driver was installed. Finally Sony Ericsson Dial-Up suite version 1.5.5 software was used to set up the GPRS dial-up connection.

8.3.2 Phone Configuration

The settings for the phone was requested from the service provider's homepage, the configuration was sent to the mobile phone via SMS.

8.3.3 First connection to Internet

After the modem has been installed and the phone configured, we can make the first test to connect to Internet. The connection went well, see below:

```
>ipconfig /all

[...]

PPP-kort djuice gprs:
        Anslutningsspecifika DNS-suffix . :
        Beskrivning . . . . . . . . . . . : WAN (PPP/SLIP) Interface
        Fysisk adress . . . . . . . . . . : 00-53-45-00-00-00
        DHCP aktiverat  . . . . . . . . . : Nej
        IP-adress . . . . . . . . . . . . : 212.17.144.22
        Nätmask . . . . . . . . . . . . . : 255.255.255.255
        Standard-gateway  . . . . . . . . : 212.17.144.22
        DNS-servrar . . . . . . . . . . . : 212.17.131.3
                                            192.237.125.2
[...]
```

Next step is to check connectivity to our server:

```
C:\Documents and Settings\Åsa Pehrsson>tracert chikchak.homeip.net

Spårar väg till chikchak.homeip.net [195.198.40.204]
över högst 30 hopp:

  1   944 ms   872 ms   876 ms  10.0.0.1
  2   859 ms   871 ms   890 ms  10.1.4.1
  3   853 ms   872 ms   867 ms  10.1.4.125
  4   852 ms   881 ms   872 ms  gw-internett.mobil.telenor.no [212.17.134.33]
  5   873 ms   848 ms   881 ms  c7206-1.mobil.telenor.no [212.17.134.36]
  6   855 ms  1139 ms   921 ms  nb02b01-s0-0-0.nb.telenor.net [217.70.230.17]
  7  1413 ms   959 ms   950 ms  nb02b11-pos0-0.nb.telenor.net [217.70.228.17]
  8   891 ms  1102 ms  1414 ms  nb01b11-ge3-1.nb.telenor.net [217.70.228.13]
  9  1221 ms  1409 ms  1405 ms  nb01b12-ge5-3-0.nb.telenor.net [217.70.228.21]
 10   929 ms   992 ms  1075 ms  217.70.229.34
 11   892 ms   941 ms  1432 ms  ge-2-3-0.se-snams001-pe-1.tu.telenor.net [212.105.101.41]
 12  1110 ms   874 ms  1208 ms  hy-peer1-link.se.telia.net [212.105.100.30]
 13   870 ms   871 ms   872 ms  fre-c1-link.se.telia.net [81.228.72.42]
 14  1372 ms   876 ms  1910 ms  fre-c2-link.se.telia.net [81.228.72.217]
 15   902 ms   862 ms   872 ms  hy-d5-link.se.telia.net [81.228.72.115]
 16   854 ms   871 ms   872 ms  h204n2fls12o893.telia.com [195.198.40.204]
 17   881 ms   862 ms   881 ms  h204n2fls12o893.telia.com [195.198.40.204]
```

8.3.4 Configuration: OpenVPN over GPRS

```
C:\DEV\2G1330>ftp 10.8.0.1
Connected to 10.8.0.1.
220 (vsFTPd 2.0.1)
User (10.8.0.1:(none)): asap
331 Please specify the password.
Password:
230 Login successful.
ftp> get t10k.bin
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for t10k.bin (10000 bytes).
226 File send OK.
ftp: 10000 bytes received in 3.43Seconds 2.92Kbytes/sec.
ftp> bye
221 Goodbye.
```

```
C:\DEV\2G1330>
```

Ethereal trace below:

```
No. Time        Source                Destination           Protocol Info
 1 0.000000    212.17.144.22    195.198.40.204    UDP    Source port: 1633  Destination port: 1194
 2 0.851224    195.198.40.204   212.17.144.22     UDP    Source port: 1194  Destination port: 1633
 3 1.682419    212.17.144.22    195.198.40.204    UDP    Source port: 1633  Destination port: 1194
 4 3.074420    195.198.40.204   212.17.144.22     UDP    Source port: 1194  Destination port: 1633
 5 3.635227    195.198.40.204   212.17.144.22     UDP    Source port: 1194  Destination port: 1633
 [...]
29 9.884212    195.198.40.204   212.17.144.22     UDP    Source port: 1194  Destination port: 1633
30 9.884212    212.17.144.22    195.198.40.204    UDP    Source port: 1633  Destination port: 1194
31 9.884212    212.17.144.22    195.198.40.204    UDP    Source port: 1633  Destination port: 1194
32 11.837020   195.198.40.204   212.17.144.22     UDP    Source port: 1194  Destination port: 1633
```

Total number of captured bytes was 14272.

8.3.5 IPSec over GPRS

We tested the connectivity to our server by nslookup:

```
C:\>ping chikchak.homeip.net

Skickar signaler till 195.198.40.204 med 32 byte data:

IP-säkerhet förhandlas.
IP-säkerhet förhandlas.
IP-säkerhet förhandlas.
IP-säkerhet förhandlas.

Skickar signaler till chikchak.homeip.net [195.198.40.204] med 32 byte data:
    Paket: Skickade = 4, mottagna = 0, Förlorade = 4 (100 %)
```

Ethereal capture at chikchak:

```
No.Time        Source                Destination           Protocol Info
 1 0.000000    212.17.144.56         195.198.40.204        ISAKMP   Identity Protection (Main Mode)
 2 0.921324    212.17.144.56         195.198.40.204        ISAKMP   Identity Protection (Main Mode)
 3 1.812606    195.198.40.204        212.17.144.56         ISAKMP   Identity Protection (Main Mode)
 4 1.832635    195.198.40.204        212.17.144.56         ISAKMP   Identity Protection (Main Mode)
 5 1.902736    212.17.144.56         195.198.40.204        ISAKMP   Identity Protection (Main Mode)
 6 1.902736    212.17.144.56         195.198.40.204        ISAKMP   Identity Protection (Main Mode)
...
23 9.303377    212.17.144.56         195.198.40.204        ISAKMP   Quick Mode
24 10.204673   195.198.40.204        212.17.144.56         ISAKMP   Quick Mode
25 10.204673   212.17.144.56         195.198.40.204        ISAKMP   Quick Mode
...
```

The connection failed, probably due to the IPSec-NAT interoperability issue.

8.4 WISP/WLAN

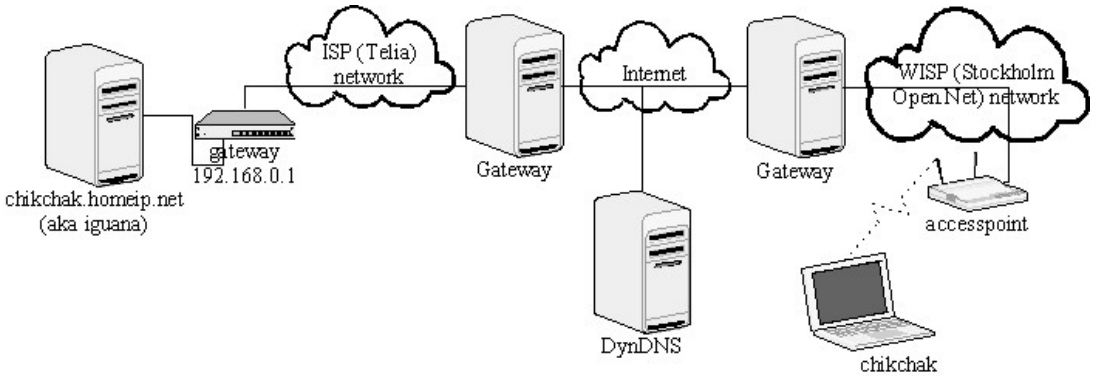

Figure 18. WISP/WLAN test bench

WISP network: stockholmopen.net/KTH

8.4.1 Client Configuration

Windows XP has built in support for WLAN connections. Therefore, not much needed to be done, except logging on to the WISP network. This was easily done from a web-page.

Then we checked connectivity to our server:

```
C:\Documents and Settings\Åsa Pehrsson>tracert chikchak.homeip.net

Spårar väg till chikchak.homeip.net [195.198.40.204] över högst 30 hopp:

  1     3 ms     4 ms     3 ms  it-gw.wan.it.kth.se [130.237.250.3]
  2     2 ms     3 ms     2 ms  130.237.203.2
  3     3 ms     5 ms     4 ms  cn4-kf4-p2p.gw.kth.se [130.237.211.205]
  4     4 ms     2 ms     5 ms  cn5-cn4-p2p.gw.kth.se [130.237.211.201]
  5     3 ms     3 ms     3 ms  kth1-cn5-p2p.gw.kth.se [130.237.211.41]
  6     3 ms     3 ms     3 ms  stockholm3-SRP2.sunet.se [130.242.85.65]
  7     6 ms     3 ms     3 ms  fre-peer1-pos0-3.se.telia.net [195.67.220.149]
  8     5 ms     4 ms     4 ms  fre-c1-link.se.telia.net [81.228.72.32]
  9     6 ms     6 ms     3 ms  fre-c2-link.se.telia.net [81.228.72.217]
 10   156 ms    10 ms     8 ms  hy-d5-link.se.telia.net [81.228.72.115]
 11     4 ms     3 ms     4 ms  h204n2fls12o893.telia.com [195.198.40.204]
```

8.4.2 OpenVPN over 802.11

The test went smoothly.

8.4.3 IPSec over 802.11

As for IPSec over GPRS, this test did not work. The ethereal trace looked pretty much the same.

9 Conclusion

In this paper, an end-to-end VPN scenario deployment over (1) the GPRS mobile network and (2) a WLAN network has been presented and analyzed. The VPN deployment is based on the IPSec protocol suite and OpenVPN, a software based on SSL/TLS.

IPSec is time consuming and rather complicated to install and configure. The IPSec solution do not work in the configuration used for this test, probably because of NAT interoperability.

OpenVPN is easy to install and configure. It increases the bandwidth consumption with approximately 10%. If there are firewalls involved, ports needs to be open, but otherwise, both ends of the tunnel can be forwarded through NAT routers without any problems.

10 References

[1] S. Khanvilkar and A. Khokhar,“Virtual Private Networks: An Overview with Performance Evaluation” October 2004. IEEE Communications Magazine, pp. 146-154.

[2] OpenVPN.[homepage on the Internet] Available from: http://openvpn.org [cited May 2005]

[3] tinc.[homepage on the Internet] Available from: http://www.tinc-vpn.org/ [cited May 2005]

[4] V. Gupta and S. Gupta, “Securing the Wireless Internet”. December 2001. IEEE Communications Magazine, pp. 68-74.

[5] C. Xenakis, E. Gazis and L. Merakos, “Secure VPN Deployment in GPRS Mobile Networks”. 2002. Department of Informatics & Telecommunications, University of Athens