Highways Agency
Temple Quay House
The Square, Temple Quay
Bristol.
BS1 6HA

# A Guide to RCC Remote Access

**March 2006**

Mott MacDonald
1 Atlantic Quay
Broomielaw
Glasgow
G2 8JB
UK
Tel: 44 (0)141 222 4500
Fax: 44 (0)141 221 8083

# A Guide to RCC Remote Access

## Issue and Revision Record

| Rev | Date | Originator | Checker | Approver | Description |
|---|---|---|---|---|---|
| A | 2005-03-04 | E.Patton | M. Lynch | A. MacKenzie | Initial draft issue for comment. |
| B | 2005-05-05 | E.Patton | M. Lynch | A. MacKenzie | Final issue |
| C | 2007-04-04 | E.Patton | M. Lynch | A.MacKenzie | Update to reflect technology changes and risk assessment |

## List of Contents

Page

## Summary

This document describes the facilities available for remote access to systems within the Regional Control Centres and provides guidance in their use.

This document has been published to assist Software Maintenance Contractors by providing guidance on possible technology choices, configurations and a description of the capabilities available for use within the remote access system.

# 1 Introduction

This document describes the facilities available for remote access to systems within the Regional Control Centres and provides guidance in their use.

This document has been published to assist Software Maintenance Contractors by providing guidance on possible technology choices, configurations and a description of the capabilities available for use within the remote access system.

Remote access facilities are aimed at improving the effectiveness of support by Software Maintenance Contractors. It is not intended that the facilities described within this document be used for the purposes of achieving remote operation of functions normally available within the RCC.

Correct operation of the remote access system will be in accordance with the appropriate Code of Connection [2.].

This document also provides guidance on the implementation of two different remote access approaches so that a solution is available for all of the major OS types present in the RCCs. The use of alternative approaches is not precluded by the provision of the guidance.

## 1.1 Document structure

This document is structured as follows:

Section 1 provides an introduction.

Section 2 provides contact details for remote access support.

Section 3 details the requirements for SMC remote access to RCC-based equipment

Section 4 discusses recommended remote access protocols and the reason for their selection.

Section 5 provides a technical description of the available remote access facilities

Section 6 provides an example showing how to use the remote access facilities

Section 7 provides guidance on the use of various remote access technologies

Appendix A details assigned Public IP addresses.

Appendix B: covers sub-system - access port assignments

Appendix C: provides references to some of the available resources describing remote access tools and technologies.

## 1.2 Document Conventions

In following sections the following text conventions are used.

**`This font is used to indicate a command that should be entered.`**

`This font is used to indicate the results of a command.`

## 1.3 References

[1.] RFC 1918 - Address Allocation for Private Internets

[2.] HATMS-NRTS Code of Connection MCH1514 dated April 2007 (Draft).

## 1.4 Omissions

No omissions have been identified.

## 1.5 Abbreviations and other terms

| | |
|---|---|
| ATM | Active Traffic Management |
| ATM | Asynchronous Transfer Mode |
| GUI | Graphical User Interface |
| ICA | Independent Computing Architecture |
| IP | Internet Protocol |
| OS | (Computer) Operating System |
| PKI | Public Key Infrastructure |
| RCC | Regional Control Centre |
| RDP | Remote Desktop Protocol |
| RFC | Request For Comments |
| SMC | Software Maintenance Contractor |
| SMTP | Simple Mail Transport Protocol |
| SSH | Secure SHell |
| SSHD | Secure SHell Daemon |
| SSL | Secure Sockets Layer (treated as synonymous with TLS) |
| TBD | To Be Determined |
| TCP | Transmission Control Protocol |
| TLS | Transport Layer Security |
| UDP | User Datagram Protocol |
| VPN | Virtual Private Network |

# 2 Support contact details

Support for remote access may be obtained from the Mott MacDonald Help Desk.

The contact details are:

Help Desk
Integrated Transport
Mott MacDonald
1 Atlantic Quay
Broomielaw.
Glasgow. G2 8JB.
United Kingdom.
Telephone: +44(0) 141 222 4666
Facsimile: +44 (0)141 222 4667
E-mail: helpdesk.itg@mottmac.com

The Help Desk operates 0830-1700 weekdays.

Allocation of access ports will be by arrangement with the remote access system operator. Please contact the Mott MacDonald to discuss port allocations.

# 3 Remote access requirements

## 3.1 Terminology

For the purposes of discussion, the following definitions are used:

| | |
|---|---|
| Client system | Computer located within the SMC's premises that will be used to effect remote access. |
| Remote host | System located within the RCC and subject to remote access |

## 3.2 Requirements for RCC remote access

In order to improve the efficiency and effectiveness of SMCs maintaining HATMS sub-systems, it was identified that the ability to access core HATMS sub-systems remotely would enable SMCs to provide a more efficient and timely administration of complex fault diagnosis and software updates.

The remote access system shall enable the following activities:

- Transmission of updated software to the remote host (prior to the on-site engineer effecting the application of the software update).
- Retrieval of system logs to assist fault diagnosis by factory located personnel.
- General system management and monitoring.
- More rapid out-of-hours access by maintenance personnel located at the SMC's base of operations.
- Generation and transmission of e-mail alerts and other support information by sub-systems.
- Routine transfer of small amounts of data from RCC-based systems to external client systems

The RCC remote access system must support remote access technologies that can be used by the disparate range of systems in use within the RCCs. Although the facilities provided will be general in nature, in particular the remote access system must be able to support remote access techniques normally used with the following classes of systems:

– MS Windows (XP, 2000,2003)
– Unix and variants (e.g. AIX, Linux)
– OpenVMS
– "Web-based" management systems over **https:**

The remote access system must be capable of monitoring usage to provide an audit trail and manage requirements for growth.

As a minimum, the remote access system must be capable of recording the following parameters:

- Initiation of a connection to or from a RCC or SMC
- Total bandwidth by time of day used by each RCC, the remote access system and HALOGEN on the HA WAN
- Total bandwidth by time of day used by the remote access system Internet connection.

If possible, the system should be capable of generating alerts whenever any of the measured quantities deviate significantly from their normal pattern.

The remote access system should generate a daily summary report of usage.

The summary report should be capable of being e-mailed to at least three external addresses.

The remote access system must be secure and meet the requirements of the Code of Connection [2.].

# 4 Remote access protocols and techniques

## 4.1 Requirements summary

The remote access system shall be capable of supporting most Network Address Translation compatible TCP or UDP-based protocols.

The choice of recommended protocols is driven by the need to ensure that information transferred between the SMC client systems and the remote host remains secure. In practice that means that protocols in use have to support suitable encryption, authentication and business needs (i.e. A secure protocol is of no use if it does not support the necessary remote access techniques).

There exists the additional constraint that the existing disparate operating systems within the RCCs have to be supported.

The configuration of any remote access server daemons (or equivalent) within RCCs should be standard to enable commonality across RCCs. The remote access point shall perform a Port Address Translation to ensure that only the standard configuration is required within the different RCCs for remote access server daemons (or equivalent).

## 4.2 SSH

Version 2 of the Secure SHell protocol was identified as a suitable mechanism for achieving remote access.

Version 2 of the Secure SHell protocol provides the following features:-

- Well respected, secure and supported protocol offering both client and server authentication and encryption facilities.
- Secure SHell daemon implementations available for both OpenVMS (Serco) and AIX (Peek) operating systems as used on COBS.
- A number of Secure SHell client implementations are available for a wide variety of operating systems including Linux and MS Windows 2000 and above.
- Support for SFTP allows secure file transfer.
- Much simpler to set up than other alternatives.
- Although SSH does not require a key management exercise, it does allow for key-based authentication to be used.
- SSH supports TCP tunnelling and so can be used to support GUI-based remote access for both X and MS Windows-based systems . For example, VNC can be tunnelled quite simply using SSH.
- Products such as NX can also be used for some systems that support SSH.

## 4.3 TLS/SSL

Techniques based on the use of Transport Layer Security (or Secure Sockets Layer) are appropriate for implementing remote access to RCC-based systems.

TLS provides the following features:-

- Mandatory server and optional client authentication
- Well-understood for the tunnelling of protocols such as http (https)
- Broad support over all of the necessary platforms (OpenVMS, AIX, Linux and MS Windows)
- When used with https, provides good client system support through web browsers

## 4.4 OpenVPN

OpenVPN is a TLS-based VPN solution that is appropriate for implementing remote access to RCC-based systems.

OpenVPN implements a OSI layer 2 or 3 secure network extension using the industry standard SSL/TLS protocol, supports flexible client authentication methods based on certificates, smart cards, and/or 2-factor authentication, and allows user or group-specific access control policies using firewall rules applied to the VPN virtual interface.

OpenVPN supports Linux, Windows 2000/XP and higher, OpenBSD, FreeBSD, NetBSD, Mac OS X, and Solaris.

OpenVPN is suitable for tunnelling other protocols such as RDP, ICA, ftp or VNC.

## 4.5 Other remote access protocols

The remote access system can directly support (as opposed to offering a tunnelling solution) other remote access technologies such as Symantec PCAnywhere - providing broad cross-platform support - (http://www.symantec.com/home_homeoffice/products/overview.jsp?pcid=pf&pvid=pca12).

# 5 Remote access facilities

## 5.1 Physical features

The point of entry to the Highways Agency Wide Area Network is an access point located at Coleshill Transmission Station.

The access point connects to the Internet via a business-class ADSL circuit providing 832kbps upload and up to 8Mbps download bandwidth. In practice, the achievable performance is somewhat less than that because of factors such as ISP network contention, line quality and protocol overhead.

The access point is connected via the HA Broadband network to the RCC Inter-RCC and sub-system LANs.

The access has a 1Mbps connection to the HA Broadband network.

## 5.2 Network features

A publicly visible IP address has been allocated for each of the major SMCs to use to access equipment within the RCCs. The IP address allocations are detailed in Appendix A. Each such IP address represents the access point for all of the equipment for which a SMC has responsibility and a need to remotely access. Access to different pieces of equipment is distinguished by allocating a TCP or UDP port to represent that particular piece of equipment (normally a sub-system). Where a single system is accessed by multiple SMCs, access for a given SMC will be via that SMC's allocated IP address.

The remote access point provides Network Address and Port Translation along with routing to ensure that the TCP or UDP connection is delivered to the correct sub-system. A key design requirement was that the configuration of any server daemons (or equivalent) within RCCs should be standard. So, for example, although different TCP ports might be used at the access point for COBS in different RCCs, the SSHD for both of the COBS will listen on the standard port (22). The remote access point in Coleshill will perform a Port Address Translation to ensure that only the standard configuration is required within the different RCCs.

In effect, the remote access system will support any client-server NAT-friendly TCP or UDP protocol that comes from a known fixed IP address or subnet. Server-side call-backs are supported where they use known fixed ports (or small port ranges). If a tunnelling solution such as OpenVPN is employed then it will be possible to use server-side call-backs on wide port ranges (for example, port mode ftp).

It is possible to create multiple connections to a remote host if that remote host supports multiple services. For example, if a remote host supported a web status page and also a SSH Server then it is possible to arrange that both can be accessed simultaneously.

Although normally left enabled, it is possible on request that one or any of the connections to a given remote host be disabled. For example, a web status page might be left enabled permanently but the corresponding SSH server connection only enabled whilst software updates were being transferred and then left disabled until next needed.

It is possible for the RCC system to originate network traffic and route that back to known destinations (which could be "anywhere"). For example, e-mail alerts or automatic transfer of a crash log from the RCC system to a SMC's location.

# 6 An example remote access configuration

The following example shows the configuration details for a set of example systems based in the NW RCC.

SMC SMC01 wishes to access four different systems in the North West RCC. The different systems use a variety of different protocols. Two of the systems are capable of generating SMTP e-mails and a third system needs to "phone home" with a status update using a https "**put**". One of the systems communicates via the Inter-RCC LAN rather than the sub-system LAN. The systems are supported from multiple locations.

The details of the four systems are:

| System | System IP address (RFC1918) | Services/protocol | System IP address - Public, routable. (DNS entry) | External Access TCP/UDP port | SMC Client system IP address[:port] |
|---|---|---|---|---|---|
| smc01_a | 10.52.1.150 | SSH on TCP port 22 | 213.36.35.120 (smc01.harcc.org.uk) | TCP 33520 | 138.104.152.20[1] |
| smc01_a | 10.52.1.150 | Outgoing e-mail to TCP port 25 | 213.36.35.120 (smc01.harcc.org.uk) | N/A | 138.104.152.20:25 |
| smc01_b | 10.52.1.151 | OpenVPN on UDP port 1194 | 213.36.35.120 (smc01.harcc.org.uk) | UDP 33521 | 138.104.152.20 |
| smc01_b | 10.52.1.151 | Outgoing e-mail to TCP port 25 | 213.36.35.120 (smc01.harcc.org.uk) | N/A | Anywhere:25 |
| smc01_c | 10.52.1.152 | https on port 443 | 213.36.35.120 (smc01.harcc.org.uk) | TCP 33522 | 138.104.152.20 |
| smc01_c | 10.52.1.152 | https on port 443 | 213.36.35.120 (smc01.harcc.org.uk) | TCP 33523 | 138.104.152.21 |
| smc01_d | 10.52.2.140 | PCAnywhere<br>5631 TCP<br>5632 UDP | 213.36.35.120 (smc01.harcc.org.uk) | TCP 33524<br>UDP 33525 | 138.104.152.20/29 |
| smc01_d | 10.52.2.140 | Outgoing https **put** to TCP port 443 | 213.36.35.120 (smc01.harcc.org.uk) | N/A | 138.104.152.23:443 |

As can be seen from the table:

- All systems controlled by a SMC are allocated a single public IP address no matter where the system is located.
- The first system maintained by a SMC is accessed via the Public IP address on port 33XXY where XX is the second dotted quad in the RCC IP address subnet and Y is a number starting from 0 for each connections to a SMC system within a given RCC.
- RCC-based systems listen on the standard port for a given service so that the configuration is identical between RCCs.
- The combination (Public IP address:access TCP/UDP port) is used to distinguish to a particular service on a particular system.
- More than one service may be offered by a single system.
- The remote access system supports both specific rules to allow access to known e-mail servers or a general rule to allow access to any e-mail server.
- Rules to support network traffic originated in the RCC may be to specific addresses or to unspecified addresses ("anywhere").
- There is a DNS entry associated with each SMC's public IP address. This should be used in preparing digital certificates to avoid hostname mismatch warnings.
- Protocols that require multiple simultaneous connections or mixed TCP/UDP can be supported

[1] Please note that 138.104.152.0/24 is an address range assigned to Mott MacDonald and is used for the purposes of the example.

- Access is from a known fixed IP address or small subnet.
- Accesses may come from multiple locations provided they are from known fixed IP addresses.

# 7 Remote access technology guidance

The purpose of this section is to provide an introduction to some of the issues that must be addressed in deploying common remote access systems. It is not intended to be a complete deployment guide and SMCs must satisfy themselves that the chosen technology and the manner of its deployment will allow the SMC to meet its requirements with respect to remote maintenance capability and also the requirements of the Code of Connection [2.].

## 7.1 General guidance

This section details guidance that is applicable to all remote access mechanisms.

### 7.1.1 Authentication and authorisation

Authorisations attached to an authenticated client system are a matter for the SMC and would normally be managed by the use of an appropriate remote host account to effect the remote access. The general principle that privileges should be the minimum necessary to perform a given role should be followed. For example, a non-privileged account could be used for routine logins to check logs whereas a privileged account would be necessary to perform a software update.

#### (i) Account and password

The minimum requirement is that access (that is, authentication and authorisation) should be controlled through the use of accounts and well-chosen passwords. The possibilities for passwords will vary with the operating system of the remote host but passwords should generally be a minimum of 8 characters long and contain a mixture of alphanumeric and non-alphanumeric characters.

**Where possible, direct login to an administrative account should be avoided.**

#### (ii) How public/private key-based authentication works[2]

The following discussion describes public/private key-based authentication in terms of the SSH implementation. However, the principles described apply to other systems of public/private key-based authentication.

Public/private key-based authentication uses the following sequence:

- The client system connects to the remote host's SSH server port.
- The SSH client system and remote host SSH server perform their handshaking. (Client system verifies the remote host's server host key, encryption keys and algorithms are negotiated, etc.)
- The SSH client system sends its (default) public key to the server and offers to authenticate with this key.
- The remote host SSH server checks the remote host's user's `authorized_keys` file, and determines if the client system public key is present. If it is, the remote host offers to accept this key from the client system by presenting a randomly chosen number encoded using the client system's public key.
- If the client system private key is protected by a passphrase (and it should be - only host keys should not have a passphrase), SSH client software will prompt the client system user to provide the passphrase in order to decrypt the client system private key.

[2] Simplifying slightly.

- The client system uses its private key to decode the number that was presented by the remote host's SSH server and returns the decoded number to the remote system in order to prove that the client system is in possession of the corresponding private key.
- If the number returned by the client system to the remote host matches then the remote host's SSH server logs the user in without asking for their password.
- If the client system cannot prove it has the private key, it may offer other keys instead.
- If the client system has no other keys it can offer, then the server offers to authenticate the user using standard password authentication.

Using key-based authentication improves security as there is no need to transmit an account password over the network. All that is required is the transmission of a randomly chosen and encrypted number. Barring errors in the cryptographic implementation of the software, it is currently computationally infeasible to deduce and then substitute the correct client system response before the client system can make the response itself. Once established, sessions (SSH, TLS etc.) are generally resistant to "man-in-the-middle" attacks.

The above discussion generally holds for SSL/TLS based techniques.

It is possible to configure web server software to require the presentation of a valid key from a client and this approach should be adopted whenever the web server interface provides more than simple status or logging functions.

SSL/TLS based VPNs such as OpenVPN require correctly configured server and client certificates to function and offer in addition, facilities such as PKI and certificate revocation.

## 7.2 SSH

SSH is a flexible protocol that supports a number of authentication and encryption mechanisms along with the ability to tunnel other TCP based protocols should it be required.

This section discusses a number of these options along with guidance on their use.

### 7.2.1 Installation of SSHD on the remote host

Installation of SSHD on the remote host to be controlled will be operating system specific. The appropriate instructions for the host should be followed.

### 7.2.2 SSHD Configuration

The configuration will normally reflect the following:

- All remote hosts will listen on port 22 (the default).

- Remote hosts should listen on all available network adapters. This may be done explicitly or by leaving the `ListenAddress` at `0.0.0.0`.

- Only version 2 of the SSH protocol should be enabled. Version 1 is now regarded as containing a number of significant vulnerabilities and should be disabled.

- PermitRootLogin should be set to a value consistent with the capabilities of the remote host's operating system. Where possible, direct login to a fully privileged account should be avoided.

- X11 forwarding should be disabled unless it is intended to that it should be used.

- Appropriate values for the various authentication mechanisms are discussed in the next section of this document.

### 7.2.3 Authentication & authorisation using SSH

This section will deal primarily with authentication of the client system against the remote host when using SSH.

#### (i) Account and password

SSH should be configured to prevent direct login to a root (or equivalent) account whenever possible.

#### (ii) Public/private key-based authentication

RSA and public key authentication should be enabled on the remote host. This can be done by setting the appropriate values in `sshd_config`

```
RSAAuthentication yes
PubkeyAuthentication yes
```

#### (iii) Host key configuration

For each remote host that the client system must control it will be necessary for the client system to have a local copy of that remote host's public key. The following discussion assumes that RSA keys have been used (the default). The procedure for using DSA keys is similar other than that file references will be to "dsa" rather than "rsa".

The remote host key is normally found in `/etc/ssh/ssh_host_rsa_key.pub`[3].

Each remote host key should be appended to the client system's `known_hosts` file. The `known_hosts` file may be established on a system-wide basis or may be in `.ssh/known_hosts` within $HOME or the equivalent default login directory.

It is important that the transfer of the remote host's public key is performed by a secure out of band method. Whilst it is possible to connect to a remote host and simply accept the presented key, this technique should be avoided as there is no certainty that the correct remote host has been reached.

#### (iv) Client system key configuration

The above allows the client system to know that it has connected to the correct remote host. It is now necessary to ensure that the client system is properly authorised on the remote host. This process is essentially a reverse of the previous process in that the client system's public key is transferred to the remote host in order to allow authentication to take place.

#### (v) Creating the client system digital signature

First it is necessary for the client system to create a suitable digital signature. This need only be done once as the same public key can be used for each remote system that must be accessed.

```
[root@ukglaswk10785 root]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

[3] In Cygwin installations this might be in `/etc/ssh_host_rsa_key.pub`. The location can be controlled by altering the entries in `sshd_config`.

```
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
44:f4:51:da:d9:bb:0d:00:cd:3a:16:b0:75:1c:ef:58 root@ukglaswk10785
[root@ukglaswk10785 root]# cat .ssh/id_rsa.pub
ssh-rsa AAAB3NzaC1yc2EAAAABIwAAAIEAw8f0Ylf+I7hx+qyoi4AqGH2GQd4
QMBykR2gjvtLne5z69yJMUp6N15fIYX0fcUMBbdtqxZN3356jbzw5N7j3ZGy8g
1hbIK1clNhdH+aamqJM6BPhY0mhuNxCwOkK4MGo+UohfnxVmeDCPaDIZbAOUB9
xY4RoIoNJlVOJOqNquW8= root@ukglaswk10785
```

The passphrase should be a string of 10-30 characters length. It should contain a variety of alphabetic, numeric and non-numeric characters. Simple English phrases containing only alphabetic characters should be avoided as they make poor passphrases.

The client system's private key should have appropriate permissions set on it, otherwise `ssh` will issue a warning and decline to use what it considers to be an insecure key. On Unix or Unix-like systems a file permission of `600` is appropriate. Note that for Cygwin installations running on MS Windows NT/2000/XP that it might be necessary to set "**`SET CYGWIN=ntea ?`**" in the batch file used to invoke Cygwin in order to allow the **`chmod`** utility to correctly set file permissions.

### (vi) Configuring the remote host with the client system digital signature

The client's file `id_rsa.pub` should be securely transferred using an out of band method to each remote host that will be accessed by the client system

The client system's public key should be appended to the file `.ssh/authorized_keys`[4] resident in the $HOME directory of the login account (or equivalent) on the remote host. Alternative locations for this file can be chosen by amending the appropriate entry in `sshd_config`

[4] Please note "authorized" with a "z".

## 7.3 OpenVPN

This document details the process in setting up an OpenVPN system (http://openvpn.net).

OpenVPN is a full-featured SSL VPN solution which can accommodate a wide range of configurations, including remote access, site-to-site VPNs, WiFi security, and enterprise-scale remote access solutions with load balancing, failover, and fine-grained access-controls.

OpenVPN implements OSI layer 2 or 3 secure network extension using the industry standard SSL/TLS protocol, supports flexible client authentication methods based on certificates, smart cards, and/or 2-factor authentication, and allows user or group-specific access control policies using firewall rules applied to the VPN virtual interface. OpenVPN is not a web application proxy and does not operate through a web browser.

OpenVPN runs on Linux, Windows 2000/XP and higher, OpenBSD, FreeBSD, NetBSD, Mac OS X, and Solaris.

Using OpenVPN, it is possible to:

- tunnel any IP subnetwork or virtual ethernet adapter over a single UDP or TCP port,
- configure a scalable, load-balanced VPN server farm using one or more machines which can handle thousands of dynamic connections from incoming VPN clients,
- use all of the encryption, authentication, and certification features of the OpenSSL library to protect your private network traffic as it transits the internet,
- use any cipher, key size, or HMAC digest (for datagram integrity checking) supported by the OpenSSL library,
- choose between static-key based **conventional encryption** or certificate-based **public key** encryption,
- use static, pre-shared keys or TLS-based dynamic key exchange,
- use real-time adaptive link compression and traffic-shaping to manage link bandwidth utilisation,
- tunnel networks whose public endpoints are dynamic such as DHCP or dial-in clients,
- tunnel networks through connection-oriented stateful firewalls without having to use explicit firewall rules,
- tunnel networks over NAT,
- create secure ethernet bridges using virtual **tap** devices, and
- control OpenVPN using a GUI on Windows or Mac OS X.

This section discusses a number of these options along with guidance on their use, including:

- Installation of OpenVPN on clients and servers
- Establishing a Public Key Infrastructure
    - Establishing a Certificate Authority
    - Creating the server certificate
    - Establishing the Diffle-Hellman parameters
    - Creating the client certificates
- Server configuration

- Starting OpenVPN as a service
- Client configuration
- Securing and starting the client

As alternative solutions are available for other operating systems, this document concentrates on the use of OpenVPN with Windows XP and 2003 systems. It should be noted that OpenVPN will operate on several platforms other than MS Windows.

### 7.3.1 Obtaining the OpenVPN software

The latest version of OpenVPN can normally be obtained from http://openvpn.net/download.html

### 7.3.2 OpenVPN Installation

The installation of OpenVPN is common for both servers and clients.

Configurations will differ between servers and clients.

Servers (the RCC-resident systems) will normally configure OpenVPN to run as a service.

Clients will normally configure separate instance of OpenVPN for each server connection. Although each of these instances could be installed as a service, it will usually be preferable to start and stop each connection as required.

OpenVPN for Windows uses a conventional installer. Note that the TUN driver installation might generate a warning that the driver has not passed Windows Logo compatibility testing.

### 7.3.3 Establishing a Public Key Infrastructure

#### (i) Overview

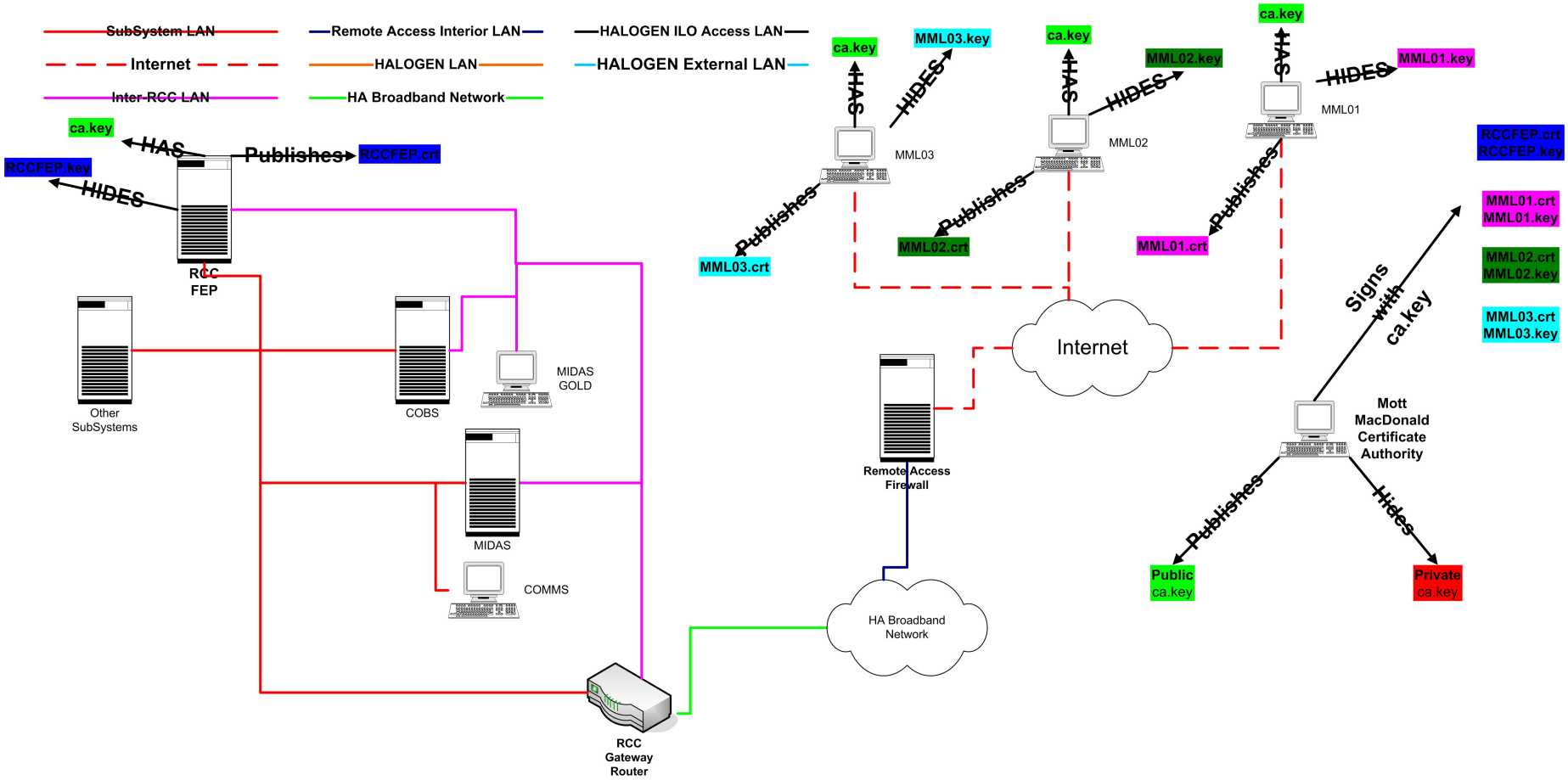
The first step in building an OpenVPN 2.0 configuration is to establish a PKI (public key infrastructure). The PKI consists of:

- a separate certificate (also known as a public key) and private key for each server and each client
- a master Certificate Authority (CA) certificate and key which is used to sign each server and client certificates.

OpenVPN supports bidirectional authentication based on certificates, meaning that the client must authenticate the server certificate and the server must authenticate the client certificate before mutual trust is established.

Both server and client will authenticate the other by first verifying that the presented certificate was signed by the master certificate authority (CA), and then by testing information in the now-authenticated certificate header, such as the certificate common name or certificate type (client or server).

The following figure shows an example that details which systems possess keys and certificates and how those items are processed on those systems - for example, public keys are *published* and private keys are *hidden*. The example is taken from HALOGEN and shows a PKI-based VPN from a Mott MacDonald (OpenVPN client) system to a RCC-based HALOGEN FEP (OpenVPN server).




# Public Key Infrastructure

### (ii) Establishing a Certificate Authority (CA)

A Certificate Authority (CA) signs the certificate/key of each user of the PKI. Other users are then able to use the CA's publicly available certificate to verify that a presented certificate is genuine. In effect, all users trust the CA's certificate - if a presented certificate can be validated through the CA's published certificate then that certificate is genuine.

As the CA's authority is used to validate the entire network of certificates, it is essential that the CA's master key is kept secure. If that key is compromised then it will be possible for others to generate certificates that appear to be valid.

# The master CA key must be kept secure.

As a CA defines a trust group, consideration must be given to the scope that applies to a particular master certificate. There is a tension between having too few master certificates which causes difficulties in managing changes and having too many master certificates which causes an administrative overhead. It is suggested that master certificates should be aligned along customer group or product lines.

### (iii) Creating the Master Certificate

OpenVPN on Windows XP/2003 provides a number of scripts that support the establishment of a PKI.

The following instructions assume that the PKI will be established from a PC running Windows XP. Similar scripts are available for OpenVPN on other operating systems.

Open up a Command Prompt window and `cd` to `\Program Files\OpenVPN\easy-rsa`.

Run the following batch file to copy configuration files into place (this will overwrite any pre-existing `vars.bat` and `openssl.cnf` files):

```
init-config
```

Now edit the `vars.bat` file and set the parameters shown below. It is important that none of these parameters are left blank. The table below shows the values chosen on the HALOGEN project by way of example. The values entered should be adapted to reflect a SMC's particular circumstances. The default key size is 1024. That may be changed to 2048 at some cost in OpenVPN performance for the benefit of greater security. *Unless the proposed application specifically requires it, it is recommended that the key size be left at 1024.*

| **Parameter** | **Value** |
|---|---|
| KEY_SIZE | 1024 |
| KEY_COUNTRY | UK |
| KEY_PROVINCE | SC |
| KEY_CITY | Glasgow |
| KEY_ORG | MottMacDonald |
| KEY_EMAIL | helpdesk.itg@mottmac.com |

Next, initialize the PKI by executing:

**vars**

**clean-all**

The OpenVPN default is to establish a master certificate that is valid for 10 years from the date of signature. Depending on the nature of the application or the length of the associated contract, it might be desirable to consider a shorter period for certificate validity. The length of the certificate validity can be altered by modifying the parameter **-days 3650** given in the **build-ca.bat** file to the desired value (e.g. for a certificate valid for 2 years, set the parameter to **-days 730**)

The final command in the establishment of the master CA is

**build-ca**

That will build the certificate authority (CA) certificate and key by invoking the interactive openssl command. This will request information needed to establish the master CA. Some information will already be present (as entered by editing the file **vars.bat**) and may be selected by pressing **enter**.

The display will be similar to that shown below. Additional information that is entered at this stage is shown in ***bold italics***:

**build-ca**

```
Generating a 1024 bit RSA private key
.............++++++
...........++++++
writing new private key to 'ca.key'
-----
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) [UK]:
State or Province Name (full name) [SC]:
Locality Name (eg, city) [Glasgow]:
Organization Name (eg, company) [MottMacDonald]:
Organizational Unit Name (eg, section) []:ITG
Common Name (eg, your name or your server's hostname) []:MottMacDonald-ITG-CA
Email Address [helpdesk.itg@mottmac.com]:
```

The above process creates 3 files:

| File | Purpose | Used by | Secret? |
|---|---|---|---|
| **XX.pem** | (where XX is a 2 digit number) This file is not used. It should be stored with the ca.key file. | No one | YES |
| **ca.crt** | This is the public certificate of the CA. It should be distributed to all servers and hosts that are part of the PKI certified by the CA. | All servers,<br><br>all clients | NO |
| **ca.key** | This is the CA's private key. It is essential that this is stored in a secure location as compromise of this key will lead to the compromise of the entire | CA machine only | YES |

| | | | |
|---|---|---|---|
| | PKI network certified by the CA. | | |

### (iv) Creating the Server Certificate

Run the build-key-server batch file:

```
build-key-server server
```

As in the previous steps, most parameters can be defaulted. When the Common Name is queried, enter "**server**". Two other queries require positive responses, "`Sign the certificate? [y/n]`" and "`1 out of 1 certificate requests certified, commit? [y/n]`".

Diffie-Hellman parameters must be generated for the OpenVPN server:

```
build-dh
Generating DH parameters, 1024 bit long safe prime, generator 2
This is going to take a long time
..................+.........................................
...................+............+.................+.........
.....................................
```

Depending on the environment variables, it might be necessary to edit the build-dh.bat file to point to the **keys** sub-directory of **easy-rsa** and (re-)name the **.pem** file. The table below assumes that the **.pem** file has been named **dh1024.pem**..

The above processes create 5 files:

| File | Purpose | Used by | Secret? |
|---|---|---|---|
| **XX.pem** | (where XX is a 2 digit number) This file is not used. It should be stored with the **ca.key** file. | No one | YES |
| **server.crt** | This is the public certificate of the server. | Server | NO |
| **server.csr** | Server Certificate Request file. This file is not used. It should be stored with the **ca.key** file. | No one | YES |
| **server.key** | This is the server's private key. It is essential that this is stored in a secure location on the server as compromise of this key will allow the server to be impersonated. | Server only | YES |
| **dh1024.pem** | Diffie-Hellman parameters. | Server only | NO |

The server key and certificate files should be renamed to reflect the server for which they were generated e.g. **server.crt -> SWCOBSComms-server.crt**

### (v) Creating the Client Certificates

Generating client certificates is very similar to the previous step. For each client that will connect to a server that is part of the PKI (NB this need only be done once per client),

```
build-key client
```

If you would like to password-protect your client keys, substitute the build-key-pass script.

"**client**" should be the name of an individual (without spaces or punctuation marks). The output files will then take the form **client.crt** (for example).

So, for example, to generate a client key for the individual Joe Bloggs, one should run:

```
build-key JoeBloggs
```

and the output will be:

```
JoeBloggs.crt
JoeBloggs.key
JoeBloggs.csr
```

For each client, make sure to type the appropriate Common Name when prompted, e.g. "JoeBloggs". Always use a unique common name for each client. So, if there were more than one Joe Bloggs requiring a client certificate then the CN (and key filenames) for the second Joe Bloggs might be "JoeBloggs01"

The script will also ask for an Organisational Unit - this should be appropriate for the particular application.

Securely record any passwords created.

The above processes create 4 files:

| File | Purpose | Used by | Secret? |
|---|---|---|---|
| **XX.pem** | (where XX is a 2 digit number) This file is not used. It should be stored with the **ca.key** file. | No one | YES |
| **client.crt** | This is the public certificate of the client. | Client only | NO |
| **client.csr** | Client Certificate Request file. This file is not used. It should be stored with the **ca.key** file. | No one | YES |
| **client.key** | This is the client's private key. It is essential that this is stored in a secure location on the client as compromise of this key will allow the client to be impersonated. | Client only | YES |

### 7.3.4 Server Configuration

#### (i) Server Configuration File Location

The server configuration file is normally in
`C:\Program Files\OpenVPN\config\`[5].

#### (ii) Preparing the OpenVPN Server Configuration File

The server configuration file may be created by copying and modifying the sample configuration file given in `C:\Program Files\OpenVPN\sample-config\`.

In preparing the configuration file, consideration has to be given to a number of items that will take non-default values. Suggested example values are shown in the next section.

#### (iii) Typical OpenVPN Server Settings

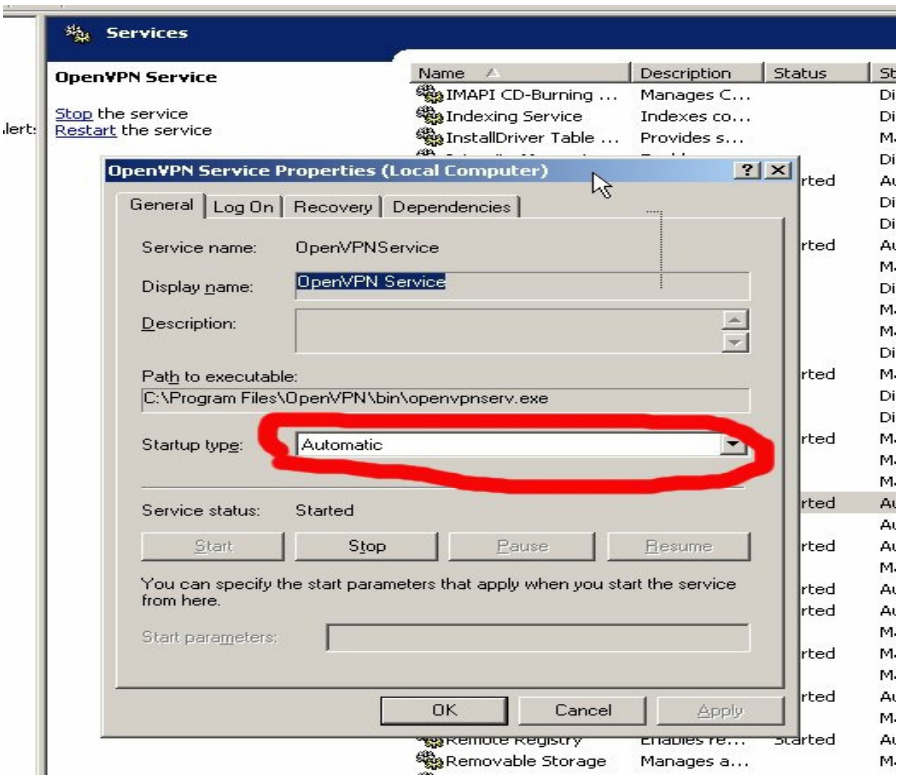
| Parameter | Value | Comment/Notes |
|---|---|---|
| local | 10.5x.1.y | This should be the local IP address of the RCC-based system that OpenVPN should listen on for incoming connections. |
| proto | udp | **udp** is the default value and will provide better performance. **tcp** should be chosen if a slightly higher degree of security is preferred (although it can cause problems when the network connection is poor). It is recommended that **udp** be chosen for most applications. |
| dev | tun | **tun** is the default value. It is strongly recommended that **tap** be avoided unless there is no other way to link systems and the subject protocol requires that the connecting systems be in the same LAN sub-net. The rest of this guide provides assumes that **tun** has been chosen |
| ca | "C:\\Program Files\\OpenVPN\\easy-rsa\\keys\\ca.crt" | This file will be the Certificate Authority certificate. Note that filenames on Windows systems **must** be quoted and that directory slashes must be escaped (by another slash). |
| cert | "C:\\Program Files\\OpenVPN\\easy-rsa\\keys*SomeSystemName*-server.crt" | This file will be the server certificate. Note that the file should be named to reflect the server system name. Note that filenames on Windows systems **must** be quoted and that directory slashes must be escaped (by another slash). |
| key | "C:\\Program Files\\OpenVPN\\easy-rsa\\keys*SomeSystemName*-server.key" | This file will be the server secret key. Note that the file should be named to reflect the server system name. Note that filenames on Windows systems **must** be quoted and that directory slashes must be escaped (by another slash). |
| dh | "C:\\Program Files\\OpenVPN\\easy-rsa\\keys\\dh1024.pem" | This file will be the Diffie-Hellman parameters for the system. Note that filenames on Windows systems **must** be quoted and that directory slashes must be escaped (by another slash). |
| server | An RFC1918 address range. | Sets OpenVPN in server mode and defines the sub-net to be used with the VPN. The server will be allocated the |

[5] If OpenVPN has been installed in a non-standard location then substitute the appropriate drive letter/directory path.

| | | |
|---|---|---|
| | 192.168.238.0<br>255.255.255.0 | first address in the block (in this case, 192.168.238.1). This subnet must be a unique subnet. Clients will be allocated blocks from the subnet (typically 255.255.255.248 for Windows clients) and so the subnet should be large enough to accommodate the maximum number of simultaneous connections anticipated. |
| ;client-to-client | Leave commented. | It should not normally be necessary for clients of the server to be able to see other clients and this directive should normally be left commented. |
| keepalive | 10 120 | This directive should be enabled. The defaults (shown) are normally sufficient. They should only be varied if there is a need for earlier detection of a network failure. |
| tls-auth | tls-auth "C:\\Program Files\\OpenVPN\\easy-rsa\\keys\\ta.key" 0 | This directive should be enabled.<br>Note that filenames on Windows systems **must** be quoted and that directory slashes must be escaped (by another slash). |
| ;cipher | BF-CBC | The parameter to this directive should match that chosen for clients. As the default cipher (Blowfish) is quite satisfactory, there should be no need to uncomment this unless a non-default cipher is required. |
| comp-lzo | Uncommented | This directive should be enabled. The corresponding setting should also be set in the client configurations. |
| max-clients | 10 | This directive should be enabled and the maximum number of clients reduced from 100 to a value that better reflects the likely number of connections. Unless there is a specific requirement to allow a large number of simultaneous connections, it is suggested that 10 is a more reasonable limit. |
| status | "F:\\OpenVPNLogs\\openvpn-status.log" | This should be a standard location for log files.<br>Note that filenames on Windows systems **must** be quoted and that directory slashes must be escaped (by another slash). |
| log | "F:\\OpenVPNLogs\\openvpn.log" | This should be a standard location for log files.<br>Note that filenames on Windows systems **must** be quoted and that directory slashes must be escaped (by another slash).<br>NB This setting will cause the OpenVPN log file to be overwritten on a restart of the service. |
| verb | 4 | Logfile verbosity. 4 will normally be sufficient unless there is a requirement to diagnose a problem with the OpenVPN connection. |
| mute | 20 | This directive should be enabled. It prevents repeating messages filling up the log by preventing more than 20 instances of a particular message from being entered into the log. |

### (iv) Starting the OpenVPN Server as a Service

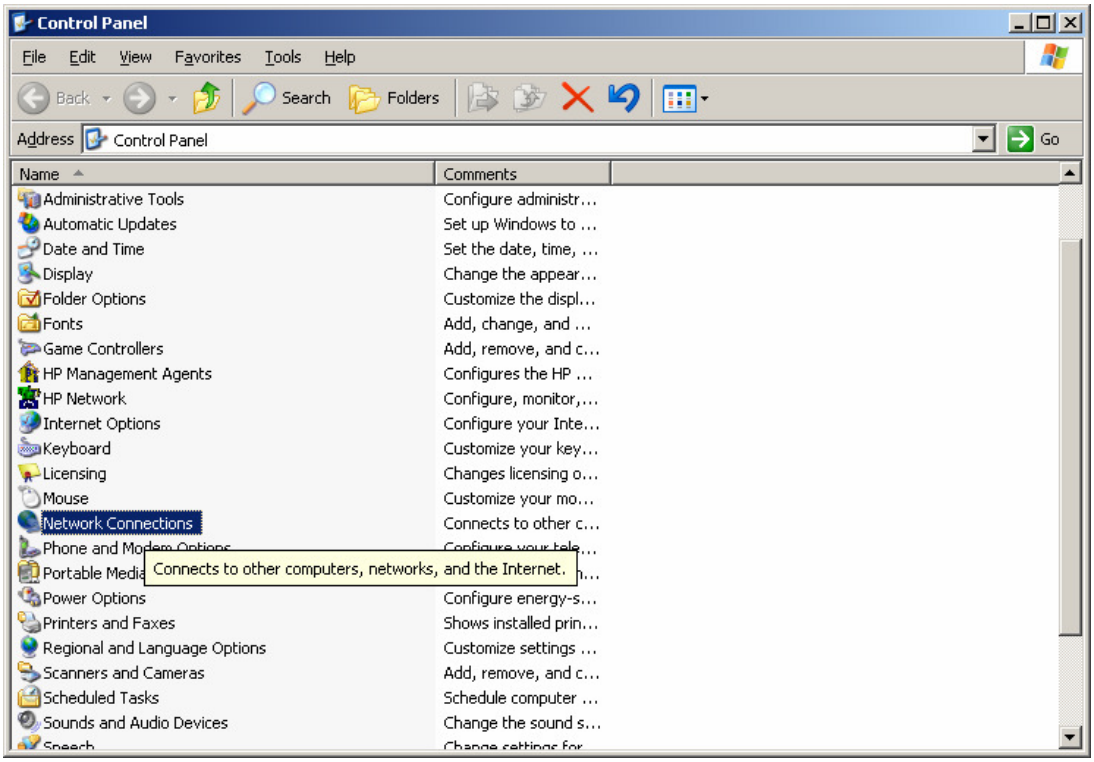
OpenVPN should be configured to run as a service. From the Services Control Panel, ensure that the **startup type** is set to **Automatic** (as shown below):



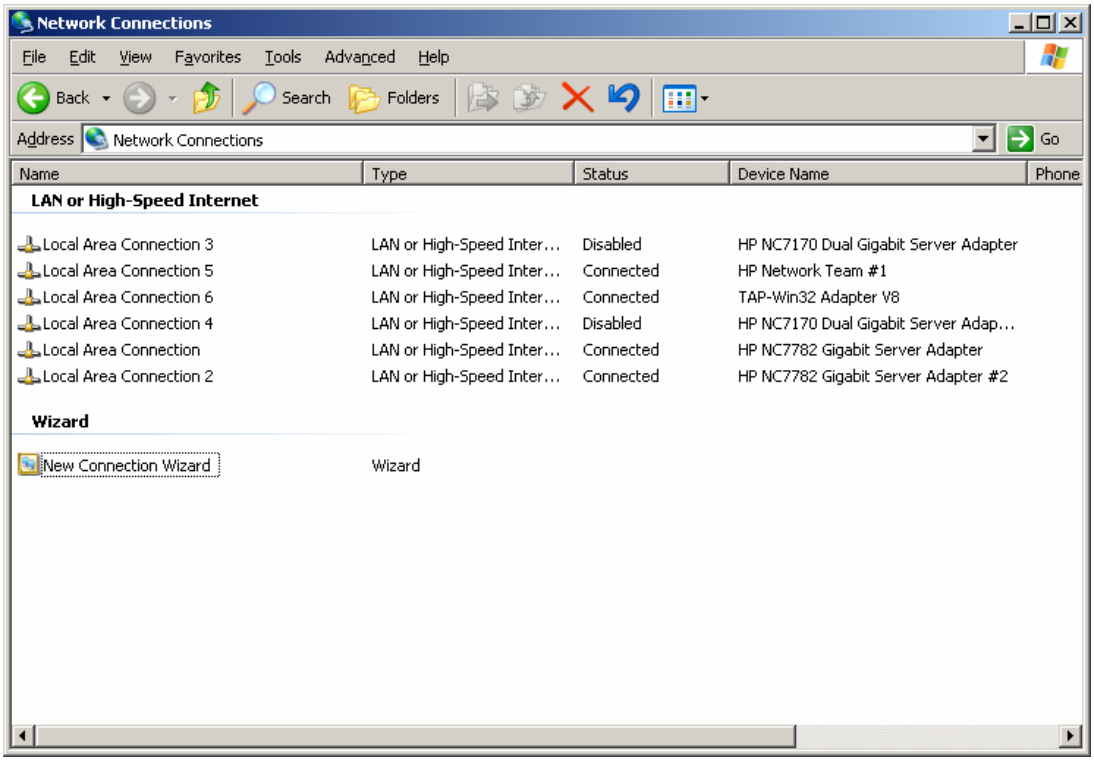

If OpenVPN is configured to run as a service then it will initiate OpenVPN listeners for each **xxxxx.ovpn** file in the **C:\Program Files\OpenVPN\config**. directory.

### (v) Network Adapter Order

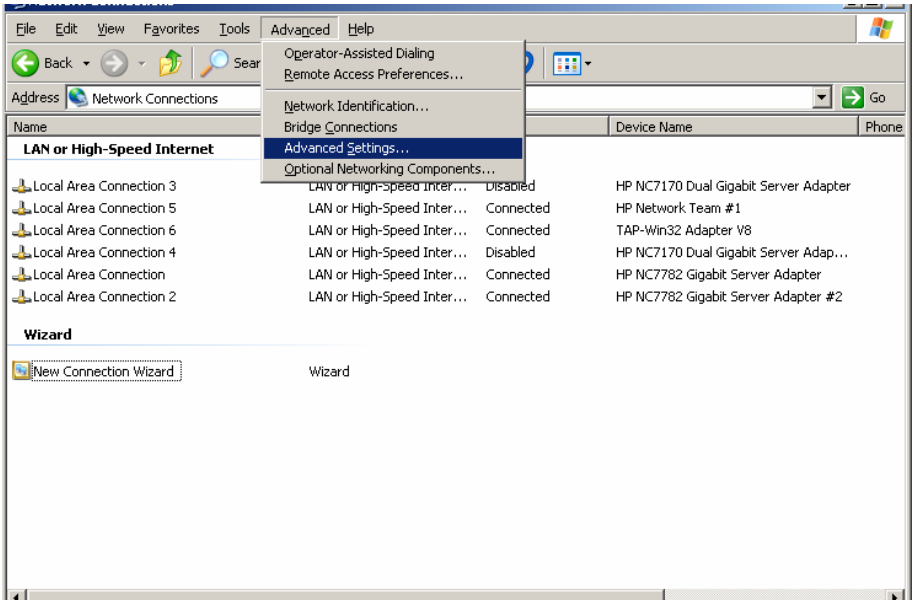
When the OpenVPN TAP-Win32 network adapter is added to the Windows configuration it can become the default adapter for certain functions. This can confuse some applications into using the OpenVPN tunnel rather than the local physical LAN. In order to avoid this problem it is necessary to move the TAP-Win32 adapter down in the list of adapters. This is done as follows:



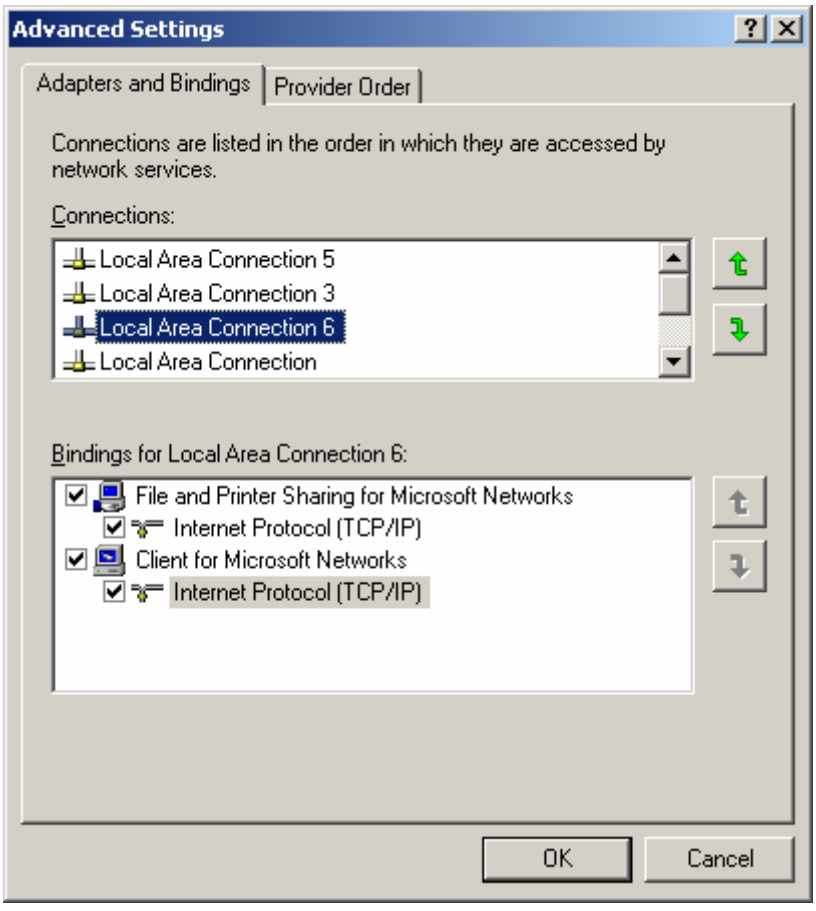

Open the **Control Panel** and double-click on **Network Connections**.



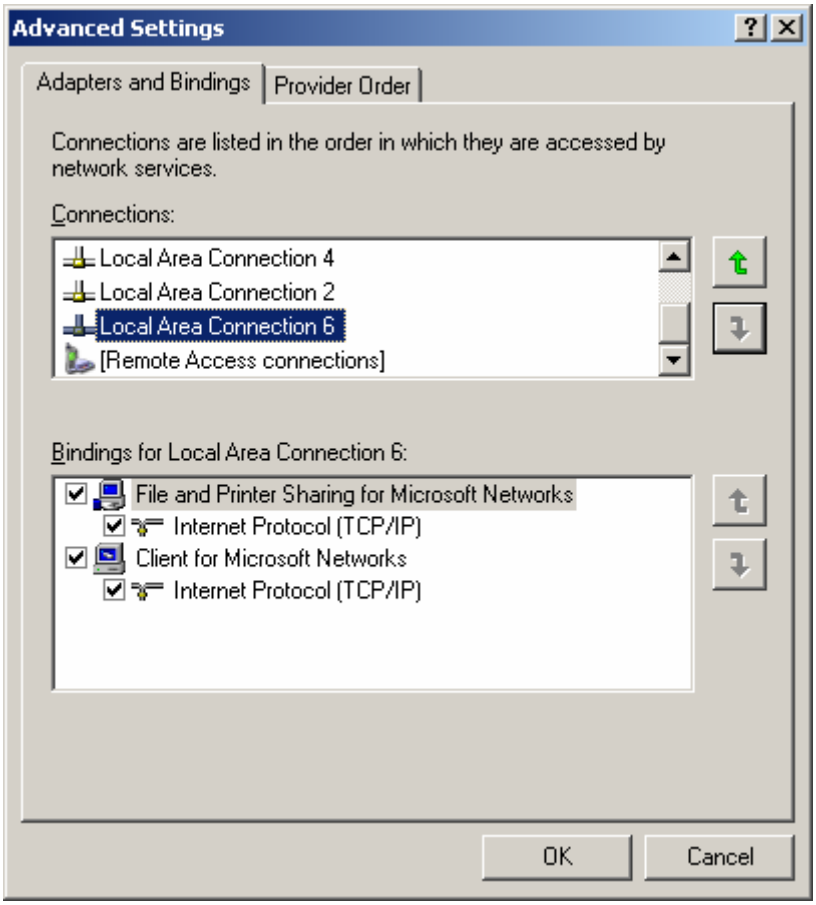

The Network Connections panel will look similar to the above. Make a note of which **Local Area Connection** is associated with the **TAP-Win32 Adapter V8** (in the case above, that is **Local Area Connection 6**).




Select **Advanced -> Advanced Settings**.




The Advanced Settings panel will appear similar to that shown above. Under **Connections**, select the **Local Area Connection** that is associated with the **TAP-Win32 Adapter V8**.




Use the green arrows to move the selected **Local Area Connection** down the list until it is the last connection above **[Remote Access connections]**. Select **OK**.

### 7.3.5 Client Configuration

#### (i) Client Configuration File Location

Each client configuration should be named to reflect the name of the system with which it will establish a VPN. Using the example above, the client configuration file to establish a VPN to the FEP in RCC 40 could be called "fep40-client.ovpn".

The client configuration file may be kept where ever is convenient as long as it correctly identifies the location of the relevant keys and certificates within that particular client environment.

#### (ii) Preparing the OpenVPN Client Configuration File

The client configuration file may be created by copying and modifying the sample configuration file given in C:\Program Files\OpenVPN\sample-config\.

In preparing the configuration file, consideration has to be given to a number of items that will take non-default values. These values should be recorded.

In general, client configurations should reflect the corresponding server settings, for example, if compression is enabled on server then it should also be enabled on the client.

### (iii) Typical OpenVPN Client Settings

The recommended settings for an OpenVPN client are shown below:

| Parameter | Value | Comment/Notes |
|---|---|---|
| client | | This directive **must** be set. |
| persist-key | | This directive should be set in order to assist in maintaining state across restarts. |
| persist-tun | | This directive should be set in order to assist in maintaining state across restarts. |
| proto | udp | **udp** is the default value and will provide better performance. **tcp** should be chosen if a slightly higher degree of security is preferred. It is recommended that **udp** be chosen for most applications. |
| dev | tun | **tun** is the default value. It is strongly recommended that **tap** be avoided unless there is no other way to link systems and the subject protocol requires that the connecting systems be in the same LAN sub-net. The rest of this guide provides assumes that **tun** has been chosen |
| resolv-retry | infinite | This directive should normally be set to "infinite" so that the VPN client will continue to attempt to restore itself should the connection be lost or not be available for any reason. |
| nobind | | This directive should normally be set unless there is some specific reason to force the client to use a particular port. |
| remote | 217.36.35.115 33550<br>This will be an address from those given in Appendix A. | This should be set to the externally visible IP address of the remote access point - not the RFC1918 IP address of the server. The port number for OpenVPN defaults to 1194 and the normal configuration will be to have the server listen on that port. However, if multiple systems are being access through a single access point then it is likely that there will be a Port Address Translation in effect and that the port that the client connects to in the first instance will be different from the default port (the PAT will be configured to transparently re-direct the client connection to the default port). |
| ns-cert-type | server | This is necessary to force the client to verify the server and prevent "man-in-the-middle" attacks. |
| ca | "<driveletter>:\\OpenVPN\\easy-rsa\\keys\\ca.crt" | This file will be the Certificate Authority certificate. Note that filenames on Windows systems **must** be quoted and that directory slashes must be escaped (by another slash). The location of this file should normally be on a removable device. |
| cert | "<driveletter>:\\OpenVPN\\easy-rsa\\keys*SomeSystemName*-server.crt" | This file will be the server certificate. Note that the file should be named to reflect the server system name. Note that filenames on Windows systems **must** be quoted and that directory slashes must be escaped (by another slash). The location of this file should normally be on a removable device. |
| key | "<driveletter>:\\OpenVPN\\easy-rsa\\keys*SomeSystemName*-server.key" | This file will be the server secret key. Note that the file should be named to reflect the server system name. Note that filenames on Windows systems **must** be quoted and that directory slashes must be escaped (by another slash). The location of this file should normally be on a |

| | | |
|---|---|---|
| | | removable device. |
| tls-auth | <driveletter>:\\OpenVPN\\easy-rsa\\keys\\ta.key" 1 | This directive should be enabled.<br>Note that filenames on Windows systems **must** be quoted and that directory slashes must be escaped (by another slash). The location of this file should normally be on a removable device. |
| ;cipher | BF-CBC | The parameter to this directive should match that chosen for clients. As the default cipher (Blowfish) is quite satisfactory, there should be no need to uncomment this unless a non-default cipher is required. |
| comp-lzo | Uncommented | This directive should be enabled. The corresponding setting should also be set in the server configuration. |
| verb | 4 | Logfile verbosity. 4 will normally be sufficient unless there is a requirement to diagnose a problem with the OpenVPN connection. |
| mute | 20 | This directive should be enabled. It prevents repeating messages filling up the log by preventing more than 20 instances of a particular message from being entered into the log. |

### (iv) Securing and starting the OpenVPN client

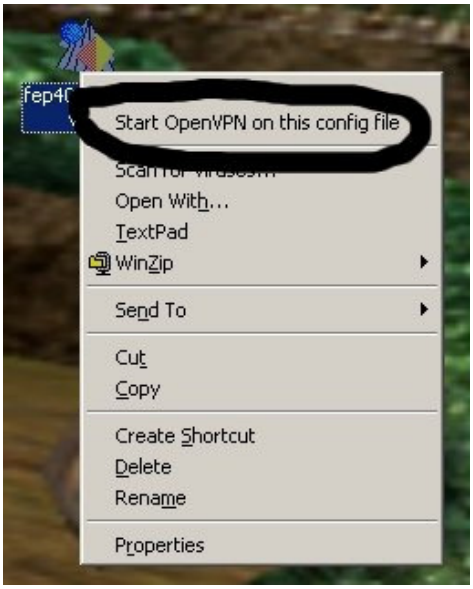
Key and certificate files should be kept in a secure location. For client's, it might be appropriate for key and certificate files to be kept on a USB memory stick or similar removable device. The relevant client configurations (there might have to be more than one depending on drive letter allocations for the removable device) should specify where the keys and certificates are to be found (i.e. in a directory located on the removable device).

It is recommended that the key and certificate files are placed in a similar directory structure to that use in a standard OpenVPN installation. So, for a user "Joe Bloggs" we would get:

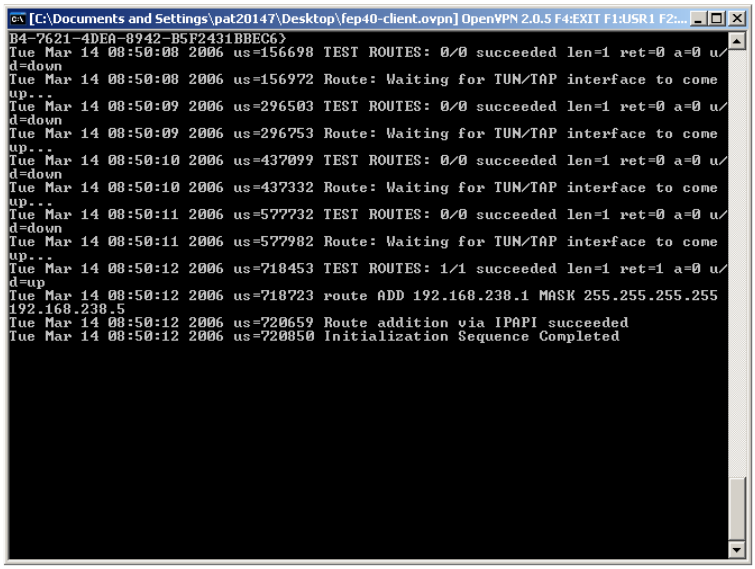
| File | Directory |
|---|---|
| ca.crt | <driveletter>:\OpenVPN\easy-rsa\keys\ca.crt |
| JoeBloggs.crt | <driveletter>:\OpenVPN\easy-rsa\keys\JoeBloggs.crt |
| JoeBloggs.key | <driveletter>:\OpenVPN\easy-rsa\keys\JoeBloggs.key |
| ta.key | <driveletter>:\OpenVPN\easy-rsa\keys\ta.key |

Where "<driveletter>" will be the drive letter normally allocated to the storage device. NB filepaths must be quoted and directory slashes escaped for Windows systems.

In MS Windows systems, the client connection for OpenVPN can be initiated by right-clicking on the relevant xxxx.ovpn file and selecting "Start OpenVPN on this config file"




A successful connection will result in a client-side display similar to that below:




Once the Initialisation Sequence Completed message has been displayed, the OpenVPN channel will have been established and it will be possible for the client to communicate with the server over the VPN.

### (v) Controlling the OpenVPN session

The OpenVPN client has some control over the session once it has been established:

Once the OpenVPN window has input focus, the following functions are available:

| Key | Function |
|---|---|
| F1 | This causes an OpenVPN restart similar to that generated by using the F3 option except that OpenVPN will not re-read the configuration file and possibly will not close and reopen the TUN/TAP device. It will re-read key files and attempt to preserve the local IP address/port. |
| F2 | Causes OpenVPN to generate session statistics for the current session |
| F3 | Cause OpenVPN to close all TUN/TAP and network connections, restart, re-read the configuration file (if any), and reopen TUN/TAP and network connections. |
| F4 | This will cause OpenVPN to perform a graceful exit and the VPN connection to the server will be terminated. |

## 7.3.6 Applications that can use OpenVPN

OpenVPN will permit the operation of most common network aware applications over the VPN tunnel. Examples of applications that can operate over OpenVPN are:

- remote management software such as
    - PC Anywhere
    - VNC (standard, Tight and Ultra variants)
- Database control software such as SQLAdvantage, RapidSQL
- Web browsers
- Most other applications that use TCP/IP or UDP/IP

In the case of VNC-type software, the VNC Server address will be the address of the server on the VPN.

Some software might need to be configured with an explicit listener on the server's VPN address.

## 7.4 Remote access user interface options

### 7.4.1 Terminal or shell style

SSH tools will provide this type of access by default. Tools, as a minimum, normally include a terminal utility **`ssh`** and file transfer utilities such as **`sftp`** or **`scp.`** GUI-based versions of these tools are readily available.

Terminal access will provide the most efficient use of bandwidth. Terminal access will normally be sufficient for OpenVMS or Unix based systems. It will also be appropriate for recovering log files etc. (via **`sftp`**) from MS Windows based systems.

### 7.4.2 X Windows

X Windows access can be achieved by tunnelling X via SSH. It will be necessary to enable X forwarding in `sshd_config.` Example instructions, detailing the required SSH configuration on both client system and remote host and using the VNC package are given later in this document.

It should be noted that remote access using X does consume more bandwidth than simple shell-style access. If X is to be used then the remotely managed display should be configured to use the minimum resolution and colour depth consistent with the normal operation of the remote host.

X Windows can also be presented (over SSH) by way of the NX system (http://www.nomachine.com/). It is the author's experience that NX is significantly more efficient than VNC for X Windows-based systems.

### 7.4.3 MS Windows GUI

Access to the MS Windows can be achieved using a VNC package tunnelled over SSH or OpenVPN.

It should be noted that remote access using VNC does consume more bandwidth than simple shell-style access. If VNC is to be used then the remotely managed display should be configured to use the minimum resolution and colour depth consistent with the normal operation of the remote host.

Several variants of VNC are available (e.g. VNC, Tight VNC, UltraVNC).

Commercial packages such as PCAnywhere provide similar functionality.

RDP may also be tunnelled over OpenVPN.

### 7.4.4 Secure web server implementing https for the display of status or log information.

If the remote system offers a web page (or pages) displaying status or log information then this should be implemented over **https** unless tunnelled over a VPN - in which case, plain http is acceptable. Note that the server certificate should be prepared using the external name/address (see Appendix A.) of the remote system unless tunnelled over a VPN. If the web server is capable only of display and has no control capabilities then basic authentication over https should be sufficient. Certificate-based mandatory client authentication may also be implemented.

### 7.4.5 Secure web server implementing https for the input of command or configuration information.

If the remote system offers a web page (or pages) that provides for the input of command or configuration information then this should be implemented over https. Note that the server certificate should be prepared using the external name/address (see Appendix A.) of the remote system.

As the web server is capable of control in addition to display or status reporting then basic authentication over https should be implemented but is not on its own sufficient. Certificate-based client authentication must also be implemented.

## 7.5 Remote access software updates

The implementation of remote access introduces an element of risk to the RCC network that was not there previously. In particular, as remote hosts could now be connected to hostile systems, it is necessary that security updates to server software are considered in prompt manner and applied as soon as reasonably possible (following appropriate testing to ensure that the update does not interfere with the normal operation of the remote host).

It is the responsibility of the SMC using the client system to ensure that the client system is appropriately configured and updated.

The above does not remove the need to follow the procedures that have been agreed between the SMC and the Highways Agency. SMCs must satisfy themselves that the chosen technology and the manner of its deployment and maintenance will allow the SMC to meet its requirements with respect to remote maintenance capability and also the requirements of any agreed Code of Connection [2.].

## 7.6 Client-side implementation requirements

Consideration was also given to the requirements that ought to apply to remote access and those systems used to effect the remote access.

These include:-

- Network connections to the client remote access system should be minimised. Whilst a standalone, dedicated system and network are to be preferred, in practice, facilities often have to be shared. For example, access might come from a controlled corporate network and often it will be necessary to use the already available corporate Internet connection.
- The client remote access system should be configured with up to date anti-virus software and protected by a suitable firewall.
- The client remote access system (and any associated router or firewall) should be configured such that no incoming connections are accepted unless from pre-arranged RCC addresses (for example, transfer of log files initiated by the RCC-based system).
- The operating system (and other software packages) used for the remote access system should normally be maintained at the latest patch status.
- The remote access system should be dedicated to that purpose and not normally used for other purpose such as casual web browsing.

For the case of access to a web server displaying pages capable only of providing status or log information and with no control capability, the above requirements may be relaxed. The requirement for the provision of a fixed IP address has not been removed.

It is the responsibility of the SMC to ensure that the client-side requirements are met and SMCs must satisfy themselves that the chosen technology and the manner of its deployment and maintenance will allow the SMC to meet its requirements with respect to remote maintenance capability and also the requirements of any agreed Code of Connection [2.].

## Appendix A: HARCC.ORG.UK External IP Addresses

| Primary DNS name | IP Address | Comment |
|---|---|---|
| | 217.36.35.112 | Network address |
| halrca.harcc.org.uk | 217.36.35.113 | Firewall |
| ipl.harcc.org.uk | 217.36.35.114 | Access point for IPL. |
| mottmac.harcc.org.uk | 217.36.35.115 | Access point for Mott MacDonald. |
| peek.harcc.org.uk | 217.36.35.116 | Access point for Peek. |
| serco.harcc.org.uk | 217.36.35.117 | Access point for Serco. |
| simsys.harcc.org.uk | 217.36.35.118 | Access point for Simulation Systems. |
| ultradatel.harcc.org.uk | 217.36.35.119 | Access point for Ultra-Datel. |
| ccl.harcc.org.uk | 217.36.35.120 | Access point for Cambridge Consultants Limited |
| wsp.harcc.org.uk | 217.36.35.121 | Access point for WSP. |
| ha.harcc.org.uk | 217.36.35.122 | Access point for HA users |
| smc04.harcc.org.uk | 217.36.35.123 | Unallocated access point. |
| smc05.harcc.org.uk | 217.36.35.124 | Unallocated access point. |
| smc06.harcc.org.uk | 217.36.35.125 | Unallocated access point. |
| | 217.36.35.126 | Router |
| | 217.36.35.127 | Broadcast address |

## Appendix B: Sub-system - port assignments

Sub-system - port assignments are available on request from the Mott MacDonald Help Desk (See section 2 Support contact details).

## Appendix C: Remote access resources

The following is a list of some resources that available to support remote access. It does not represent an endorsement of any particular resource. It is the responsibility of the SMC to satisfy itself that a selected resource is appropriate to its need. It should not be assumed that because a resource does not appear on this list that the resource is not suitable. It is not intended that this list should be considered exhaustive.

| MS Windows | | |
|---|---|---|
| Resource | Location | Notes |
| Cygwin | http://www.cygwin.com/ | The Cygwin tools are ports of the popular GNU development tools for Microsoft Windows. In particular SSH and X are supported.<br><br>A commercially supported version may be found here: http://www.redhat.com/software/cygwin/ |
| OpenSSH for Windows | http://sshwindows.sourceforge.net/ | OpenSSH for Windows is a free package that installs a minimal OpenSSH server and client utilities in the Cygwin package without needing the full Cygwin installation. It provides full SSH/SCP/SFTP support, SSH terminal support provides a familiar Windows Command prompt while retaining Unix/Cygwin-style paths for SCP and SFTP.<br><br>This package does not appear to being actively maintained and is probably best avoided. |
| PuTTY | http://www.chiark.greenend.org.uk/~sgtatham/putty/ | A common ssh client package. |
| SecureCRT | http://www.vandyke.com/products/securecrt/index.html | Commercial SSH client package |
| VShell™ Server for Windows and UNIX | http://www.vandyke.com/products/vshell/index.html | Commercial SSH server package |
| Secure KoalaTerm | http://www.foxitsoftware.com/mkt/skt_intro.php?key=ssh | Commercial SSH client package |
| WAC Server | http://www.foxitsoftware.com/wac/ssh_intro.htm | Commercial SSH server package |
| SSH Tectia Client | http://www.ssh.com/products/tectia/client/ | Commercial SSH client package |
| SSH Tectia Client and Server | http://www.ssh.com/products/client-server/ | Commercial SSH package |
| WinSCP | http://winscp.net/eng/index.php | WinSCP is an open source free SFTP client for Windows using SSH. Legacy SCP protocol is also supported. Its main function is safe copying of files between a local and a remote computer.<br><br>It features:<br><br>- Graphical user interface |

| | | |
|---|---|---|
| | | - Translated into several languages<br>- Integration with Windows (drag&drop, URL, shortcut icons)<br>- U3 support<br>- All common operations with files<br>- Support for SFTP and SCP protocols over SSH-1 and SSH-2<br>- Batch file scripting and command-line interface<br>- Directory synchronisation in several semi or fully automatic ways<br>- Integrated text editor<br>- Support for SSH password, keyboard-interactive, public key and Kerberos (GSS) authentication<br>- Integrates with Pageant (PuTTY authentication agent) for full support of public key authentication<br>- Windows Explorer-like and Norton Commander-like interfaces<br>- Optionally stores session information<br>- Optionally supports standalone operation using a configuration file in place of registry entries, suitable for operation from removable media |
| OpenVPN | http://www.openvpn.net | OpenVPN is a full-featured SSL VPN solution which can accommodate a wide range of configurations, including remote access, site-to-site VPNs, WiFi security, and enterprise-scale remote access solutions with load balancing, failover, and fine-grained access-controls.<br><br>OpenVPN implements OSI layer 2 or 3 secure network extension using the industry standard SSL/TLS protocol, supports flexible client authentication methods based on certificates, smart cards, and/or 2-factor authentication, and allows user or group-specific access control policies using firewall rules applied to the VPN virtual interface. OpenVPN is not a web application proxy and does not operate through a web browser.<br><br>NB Requires MS Windows 2000 or later. |
| RealVNC<br><br>TightVNC<br><br>UltraVNC | http://www.realvnc.com/download.html<br>(Windows, Linux, Solaris,HP-UX, OS X)<br>http://www.tightvnc.com/download.html<br>(Windows, Linux/Unix, Java viewer)<br>http://www.uvnc.com/index.html<br>(Windows, Java) | VNC (Virtual Network Computing) is available from a number of suppliers. It is remote controls software which allows one to view and fully interact with a remote computer desktop. There is no requirement for the server and client to be running the same operating system (e.g. it is possible to use VNC from a Windows XP system to |

| | | |
|---|---|---|
| Chicken of the VNC (sic) | http://sourceforge.net/projects/cotvnc/ (OS X) | manage a Linux system and vice versa).<br><br>The different VNC implementations differ in their cost, support for Enterprise features, file transfer capabilities and also in their ability to support downloadable Java-based client applets.<br><br>Most Linux distributions ship with VNC client and server software (although it might have to be separately installed). |
| PC Anywhere | http://www.symantec.com/home_homeoffice/products/overview.jsp?pcid=pf&pvid=pca12 | Enables simple, secure connection to remote devices<br><br>Simplifies working across multiple platforms Connects to Microsoft® Windows®, Linux®, and Mac OS® X based hosts from these (plus Pocket PC) remote systems<br><br>Ensures high security with encryption and password protection<br><br>Remotely accesses pcAnywhere hosts through firewalls and routers |